

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТАВРІЙСЬКИЙ ДЕРЖАВНИЙ АГРОТЕХНОЛОГІЧНИЙ
УНІВЕРСИТЕТ**

Кафедра «Комп'ютерні науки»

ЛИТВИН Ю.О., СІЦИЛЦІН Ю.О., СТРОКАНЬ О.В.

КРОС-ПЛАТФОРМНЕ ПРОГРАМУВАННЯ

Лабораторний практикум для виконання лабораторних робіт

Мелітополь, 2019

УДК 004.4'23 (072)

К-83

Автори: Ю.О.Литвин, Ю.О. Сіциліцин, О.В.Строкань

Рекомендовано до друку рішенням методичної комісії факультету інженерії та комп'ютерних технологій Таврійського державного агротехнологічного університету

Рецензенти:

С.М.Прийма – д.пед.н., професор кафедри комп'ютерних наук Таврійського державного агротехнологічного університету

В.С. Єремєєв - д.т.н., професор кафедри інформатики і кібернетики, Мелітопольського державного педагогічного університету ім. Богдана Хмельницького.

К-83 Крос-платформне програмування: лабораторний практикум з виконання лабораторних робіт / укл. Ю.О.Литвин, Ю.О.Сіциліцин, О.В.Строкань – Мелітополь: Вид-во ТДАТУ, 2019 – 220 с.

У навчальному посібнику зібраний, систематизований та викладений теоретичний і методичний матеріал, який охоплює практичні питання дисципліни «Крос-платформне програмування». У навчальному посібнику розглядаються питання архітектури та стандарти компонентних моделей, комунікаційних засобів і розподілених обчислень; стратегії інтеграції програмних компонентів; основні платформи проміжного рівня та компонентні моделі; формальні та візуальні методи конструювання компонентів.

© Ю.О.Литвин, Ю.О.Сіциліцин, О.В. Строкань

ЗМІСТ

ЗАГАЛЬНІ ОРГАНІЗАЦІЙНО-МЕТОДИЧНІ	6
ПЕРЕДУМОВИ	6
ЛАБОРАТОРНА РОБОТА 1	7
Тема: ЛІНІЙНІ ПРОГРАМИ В JAVA.....	7
1.1 Практична частина	7
1.2 Самостійна частина.....	12
1.3 Питання для перевірки знань	15
ЛАБОРАТОРНА РОБОТА 2	16
Тема: РОЗГАЛУЖЕНІ ПРОГРАМИ В JAVA	16
2.1 Практична частина	16
2.2 Самостійна частина.....	17
2.3 Питання для перевірки знань	18
ЛАБОРАТОРНА РОБОТА 3	20
Тема: КЛАСИ І ОБ'ЄКТИ	20
3.1 Практична частина	20
3.2 Самостійна частина.....	26
3.3 Питання для перевірки знань	35
ЛАБОРАТОРНА РОБОТА 4	36
Тема: УСПАДКУВАННЯ В JAVA.....	36
4.1 Практична частина	36
4.2 Самостійна частина.....	43
4.3 Питання для перевірки знань	61
ЛАБОРАТОРНА РОБОТА 5	62
Тема: ПОЛІМОРФІЗМ.....	62
5.1 Практична частина	62
5.2 Самостійна частина.....	64
5.3 Питання для перевірки знань	82
Тема: ІНТЕРФЕЙСИ.....	83

6.1 Практична частина	83
6.2 Самостійна частина.....	87
6.3 Питання для перевірки знань	88
ЛАБОРАТОРНА РОБОТА 7	90
Тема: МАСИВИ.....	90
7.1 Практична частина	90
7.2 Самостійна частина.....	93
7.3 Питання для перевірки знань	103
ЛАБОРАТОРНА РОБОТА 8	105
Тема: РЯДКИ	105
8.1 Практична частина	105
8.2 Самостійна частина.....	119
8.3 Питання для перевірки знань	123
ЛАБОРАТОРНА РОБОТА 9	125
Тема: РОБОТА З КЛАСАМИ-КОЛЕКЦІЯМИ JAVA	125
9.1 Практична частина	125
9.2 Самостійна частина.....	136
9.3 Питання для перевірки знань	150
ЛАБОРАТОРНА РОБОТА 10	153
Тема: ПАКЕТ JAVA.IO, ФАЙЛИ.....	153
10.1 Практична частина	153
10.2 Самостійна частина.....	154
10.3 Питання для перевірки знань	167
ЛАБОРАТОРНА РОБОТА 11	169
Тема: ПАКЕТИ JAVA.AWT ТА JAVA.SWING.....	169
11.1 Практична частина	169
11.2 Самостійна частина.....	177
11.3 Питання для перевірки знань	183
ЛАБОРАТОРНА РОБОТА 12	184

Тема: ПАКЕТИ JAVA.AWT ТА JAVA.SWING	184
12.1 Практична частина	184
12.2 Самостійна частина.....	187
12.3 Питання для перевірки знань	187
ЛАБОРАТОРНА РОБОТА 13	188
Тема: ПАКЕТ JAVA.AWT. АНІМАЦІЯ.....	188
13.1 Практична частина	188
13.2 Самостійна частина.....	189
12.3 Питання для перевірки знань	190
ЛАБОРАТОРНА РОБОТА 14.....	191
Тема: РОБОТА З БАЗАМИ ДАНИХ. JDBC.....	191
14.1 Практична частина	191
14.2 Самостійна частина.....	209
14.3 Питання для перевірки знань	215
СПИСОК ЛІТЕРАТУРИ.....	217

ЗАГАЛЬНІ ОРГАНІЗАЦІЙНО-МЕТОДИЧНІ ПЕРЕДУМОВИ

Мета курсу - забезпечити отримання студентами теоретичних знань і практичних навичок компонентного програмування.

Завдання курсу - формування в студентів вміння розробляти крос-платформні програмні засоби.

В результаті вивчення дисципліни студент повинен знати: архітектуру та стандарти компонентних моделей, комунікаційних засобів і розподілених обчислень; стратегії інтеграції програмних компонентів; основні платформи проміжного рівня та компонентні моделі; формальні та візуальні методи конструювання компонентів.

В результаті вивчення дисципліни студент повинен вміти: розробляти вимоги та специфікації компонентів інформаційних систем і об'єктів професійної діяльності; проектувати компоненти програмного забезпечення; проектувати людино-машинний інтерфейс інформаційних систем; реалізовувати прототипи архітектури програмного забезпечення; інтегрувати компоненти в систему; установлювати, налаштовувати та обслуговувати системне, інструментальне і прикладне програмне забезпечення та інформаційні системи.

В результаті вивчення дисципліни студент повинен отримати навички: програмування на крос-платформній мові програмування Java.

ЛАБОРАТОРНА РОБОТА 1

Тема: ЛІНІЙНІ ПРОГРАМИ В JAVA

Мета роботи: 1) Вивчити основні принципи побудови лінійних програм; 2) Освоїти навички складання арифметичних виразів.

Час виконання роботи 2 год.

1.1 Практична частина

При запуску Eclipse з'являється діалогове вікно, що дозволяє вибрати місце, де буде розташовуватися робочий простір. Робочий простір - каталог, в якому буде зберігатися ваша робота.

Після вибору робочого простору, на екрані з'явиться початкова сторінка, з пропозиціями подивитися підручники, приклади і т.д. Для переходу у вікно робочого середовища необхідно обрати Workbench, в якому і буде проходити ваша подальша робота.

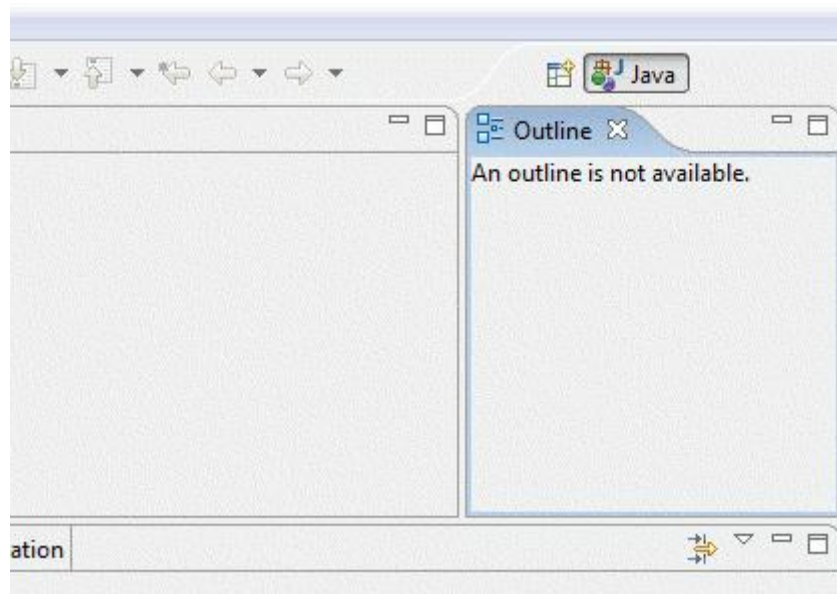


Рисунок 1.1 – Робоче середовище програми Eclipse

Основними складовими робочого середовища є уявлення (views), редактори (editors) і проекції або перспективи (perspectives).

Уявлення - це невеликий розділ всередині робочого середовища, який служить для навігації по деякій категорії об'єктів (такий, як ресурси або пакети), відкриття редакторів, відображення властивостей активних редакторів. Наприклад, уявлення Navigator показує проекти та інші ресурси, а уявлення Bookmarks відображає всі закладки в Workbench разом з іменами файлів, з якими ці закладки пов'язані. На рисунку зображений правий верхній кут робочого середовища з активним уявленням Outline.

Всі зміни, зроблені в уявленнях, негайно зберігаються. Інший тип візуальних компонентів Workbench-редактор, які використовуються для перегляду і редагування деякого ресурсу (наприклад, програмного коду). При виборі ресурсу з'являється відповідний редактор. Наприклад, при відкритті будь-якого текстового документу (з розширенням .txt) командою File -> Open File ... і ви побачите вбудований редактор простого тексту, який не форматований. Якщо що-небудь набрати в цьому редакторі, на його вкладці, де написано назву файлу, з'явиться зірочка. Вона означає, що редактор містить не збережені зміни. Вони збережуться, якщо натиснути Ctrl + S або вибрати команду File -> Save.

Існує безліч корисних уявлень, які додаються у вікно робочого середовища командою Window -> Show View. Однак замість того, щоб додавати їх по одному, зручніше буває перемикаати проекцію (perspective). Проекція (або перспектива) - це набір уявлень і редакторів, спеціально підібраний для виконання певної задачі. Після запуску в Eclipse відкривається перспектива Java, настроєна власне на написання програми. Для налагодження програми часто використовується проекція Debug. Переключити проекцію можна командою Window -> Open Perspective. Назва поточної проекції відображається в правому верхньому куті робочого середовища (див. рисунок 1.1).

Перш ніж приступити до програмування, необхідно створити проект, в якому Eclipse буде зберігати всі ресурси, пов'язані з вашою програмою.

Для створення проекту треба виконати команду File -> New -> Project. У вікні виберіть Java Project і натисніть «Далі». Вкажіть ім'я свого проекту. Зверніть увагу, що в директорії, зазначеної вами як робочий простір, буде створена папка з ім'ям вашого проекту (якщо, звичайно, не змінити налаштування в цьому вікні, чого для першого разу ми робити не треба). Після цього необхідно натиснути кнопку «Готово».

Тепер в поданні Package Explorer в лівій частині робочого середовища присутній ваш проект. У будь-який момент його можна видалити, клацнувши по його назві правою кнопкою миші і вибравши Delete. Після цього Eclipse запитає, знищити чи заодно папку з файлами проекту (при необхідності можна і знищити).

Якщо не видалити проект, до нього можна додати файли і папки за допомогою команд контекстного меню New -> File і New -> Folder відповідно. Якщо проект великий, то йому необхідна структура вкладених папок. Але в разі Java-проекту все трохи інакше. Справа в тому, що фрагменти Java-програми згруповані в пакети, а для кожного пакета створюється окрема папка. Paket створюється командою New -> Package. Для пакета теж треба придумати ім'я. В результаті в папці проекту буде створена нова папка з такою назвою. Можете перевірити.

Переглядати ресурси проекту може бути зручніше за допомогою уявлення Navigator. Відкрийте його командою Window -> Show View. Можно побачити, що крім директорій проекту і пакету Eclipse створив два допоміжних файли .classpath і .project. Вони легко відкриваються в редакторі, але особливого інтересу для нас вони зараз не представляють.

Програма на Java завжди складається з одного або декількох класів. Створити клас можна за допомогою команди New -> Class в контекстному меню уявлення Navigator (або Package Explorer, не має значення). При створенні класу необхідно вибрати пакет, до якого він буде ставитися

(виберіть тільки що створений вами пакет) і придумати йому ім'я. Імена класів прийнято починати з великої літери. Якщо не дотриматися цього правила, Eclipse виведе попередження, але нічого страшного не станеться.

Для цілей корисно поставити галочку в розділі «Які методи ви хочете створити в своєму класі?» Навпроти опції `public static void main (String [] args)`. В результаті в тілі класу буде згенеровано метод (функція) `main ()`. Java вимагає, щоб хоча б в одному з класів програми існував метод з таким заголовком. Саме він і буде виконаний при старті програми.

В результаті наших дій в папці пакета буде створений файл з ім'ям нашого класу і розширенням `.java`. Eclipse відкриє редактор коду, в якому відобразиться вміст цього файлу. Воно буде приблизно таким (імена пакета і класу, звичайно, можуть відрізнятися):

```
package mainPack;public class MyClass
{/** @param args*/public static
void main(String[] args) {// TODO Auto-generated method stub}}
```

Команди, що становлять тіло функції, можна написати замість автоматично згенерованого коментаря `// TODO Auto-generated method stub`. Ми напишемо тільки одну команду, яка буде виводити на екран класичний рядок «Hello, world!»:...

```
System.out.println("Hello, world!");
```

Залишилося програму запустити. Для цього виконаємо команду `Run -> Run` і отримаємо діалогове вікно з нетривіальними настройками запуску. У лівій частині цього вікна треба вибрати `Java Application` (додаток Java). Трохи подумавши, Eclipse знайде наш клас, що містить метод `main ()` і запропонує почати запуск програми саме з нього (в правій частині вікна на вкладці `Main` повинні з'явитися назви нашого проекту і нашого класу). Крім цього увазі програміста пропонується ще кілька закладок. Наприклад, на другий з них - `Arguments` - пропонується ввести параметри командного рядка (якщо програма розрахована на виклик з командного рядка з параметрами). Для нашої простої програми нічого додатково вказувати не потрібно. Просто натисніть кнопку `Run`.

В результаті роботи програми здійснюється виведення даних в так звану консоль. В операційній системі MS DOS консоллю служив весь екран монітора. Eclipse же відкриває нам уявлення Console, в якому (якщо все зроблено правильно) і відобразиться рядок "Hello, world!" - результат виведення нашої програми.

Тепер для повторного запуску програми (наприклад, якщо ми вирішили внести в неї якісь зміни або треба показати викладачеві), можна піти легшим шляхом - виконати команду Run -> Run Last Launched (ще раз запустити попередній додаток) або просто натиснути Ctrl + F11.

Приклад.

1. Скласти програму для обчислення функції F для завдання значення змінних a, b, x, z.

```
Package funk1;
Import java.util.Scanner;
Public class myFunk1 {
Private static Scanner in;
public static void main(String[] args) {
double a,b,x,z,f;
in = new Scanner(System in);
a=in.nextDouble();
b=in.nextDouble();
x=in.nextDouble();
z=in.nextDouble();
f=(a*Math.pow(Math.sin(b*x),2)-Math.sqrt(24*z))/(Math.pow99-
x+13).1/3)-b*Math.log(z*z+4));
System.out.println(f);
}
}
```

У ході виконання коду буде наступний результат у вікні «Console»:

8,365
6,0003
-0,1
0,2

2. Запустити створену програму на виконання:

Натискаємо клавіша CTRL+F11. Примітка – якщо з'явилося повідомлення про помилку необхідно внести відповідні виправлення в текст програми. Уводимо вихідні значення a, b, x, z.

1.2 Самостійна частина

Завдання 1.1. У відповідності зі своїм варіантом скласти програму обчислення функції y для x=6.37, a=2.56, b=7.18. Варіанти до завдання приведені в таблиці 1.1.

Таблиця 1.1. Варіанти до завдання 1.1

Варіант	Вид функції y	Варіант	Вид функції y
1	$y = \frac{\cos(x-a)}{e^{x-a} + \sqrt{ bx }}$	6	$y = \frac{0.8\cos(x+b)}{\sqrt{b^2 + x^3}}$
2	$y = \frac{e^{ x-a }}{\cos^2 bx}$	7	$y = \frac{\cos ax + 5.87}{\sqrt{b^2 + x^3}}$
3	$y = \frac{ax+b}{\ln^2 bx + \sqrt{a}}$	8	$y = \frac{\sin(x-ab)}{0.55\ln^2 ax + \frac{b}{x}}$
4	$y = \frac{3.1b - e^{ax}}{0.8\ln^3 ax}$	9	$y = \frac{\sqrt{a+x} + 0.25\ln x}{x - 4.87}$
5	$y = \frac{\arctan(\sqrt{b+x})}{0.3e^{-ax} - \cos bx}$	10	$y = \frac{x^3 + 5.8\sin ab}{2.1x + 3a + b}$

Завдання 1.2 У відповідності зі своїм варіантом скласти програму обчислення двох функцій y1 і y2 при a=3.56, b=5.86, x=2.28. Варіанти до завдання приведені в таблиці 1.2.

Таблиця 1.2. Варіанти до завдання 1.2.

Варіант	Вид функції y	Варіант	Вид функції y
1	$y_1 = (x-a)^3$ $y_2 = \frac{e^{x-a}}{\cos bx}$	6	$y_1 = \sqrt{a + \ln^2 x}$ $y_2 = \frac{\sin x + \sqrt{ab}}{\operatorname{arctg}(b-x)}$
2	$y_1 = a \cdot \ln^2 bx$ $y_2 = \sqrt[3]{\frac{ax+b}{\operatorname{tg} 0.25bx}}$	7	$y_1 = a^2 e^{-bx}$ $y_2 = \frac{\sqrt[5]{a+bx}}{\ln^2 x}$
3	$y_1 = \cos^2 ax$ $y_2 = \frac{\ln bx - a}{e^{\frac{x}{b}}}$	8	$y_1 = e^{ax} + \sqrt[3]{x}$ $y_2 = \frac{2ax-b}{b^2 + x^4}$
4	$y_1 = \sin^3(b+x)$ $y_2 = \frac{\ln x + \cos 5ab}{bx^3 - a^2}$	9	$y_1 = \operatorname{arctg}(bx - ae^{x-b})$ $y_2 = \frac{b \sin^2 ax}{\ln^3 bx - a^2 }$
5	$y_1 = \sqrt[5]{x^3 + a^3}$ $y_2 = \frac{\sin^2 ax}{\operatorname{tg}(x^2 - b)}$	10	$y_1 = a^3 + \cos^2 bx$ $y_2 = \frac{x^a + 20.5a^3b}{\operatorname{arctg}(ax + b - a^2)}$

Завдання 1.3 У відповідності зі своїм варіантом скласти програму для обчислення значення виразу. Вихідні дані – довільні числа. Варіанти до завдання приведені в таблиці 1.3.

1 При указівці величин, позначених грецькими буквами ($\acute{\alpha}$, β , γ , δ і т.д.) використовувати власні довільні імена змінних.

2 Для одержання нестандартних функцій (тангенс, десятковий логарифм і т.д.) необхідно використовувати формули перетворення, відомі з курсу вищої математики.

3 Щоб спростити запис складних виразів рекомендується використовувати проміжні змінні (вони будуть містити результат обчислення частини вираження).

Таблиця 1.3. Варіанти до завдання 1.3

Варіант	Вираження
1	$a = \ln \frac{3a^2 x^4 + ax^2 + y}{e^{x+y} + e^{-x-y}} + \sqrt{\frac{x^2 \sin(ax) - y^3 \cos(ax)}{1 + \sin^2(ax) + \cos^2(ay)}} - \operatorname{tg}^2 \frac{x}{2} + \frac{\cos x \cdot \operatorname{tgy}}{e^{ax}}$
2	$\beta = \sin(bx) \cdot \operatorname{tg} \frac{y}{2} - \sqrt[3]{\frac{btg^3 x - 2b^2 \cdot \operatorname{tg} x + 1}{\sin(x+y) + \cos(x-y)}} + \ln \left 2 \cos^3(bx) - \frac{\sqrt{\sin^2 y + 1}}{2 - \cos(bx)} \right $
3	$\chi = \sqrt{\frac{h^2 \operatorname{tg}^4 x - 3 \cos^2 y}{e^x + e^y}} + \log_2 \frac{ 2h \cdot x^3 \cdot x + y }{3 - \sin x - \cos y + 3} + \cos(hx) \cdot (1 + \operatorname{ctgy})$
4	$\delta = \sqrt{\frac{\operatorname{tg}^4 \frac{x}{d} + \operatorname{tg}^2 \frac{y}{d} + 1}{\sqrt{ 3 + \sin x + \cos^3 y }}} - \ln \left \frac{dx^3 - 3d^2 \cdot x + y}{3 - \sin(x+y) - \cos(x-y)} \right + d \cdot \operatorname{tg}^2 \frac{x}{2} + \frac{\cos x \cdot \operatorname{tgy}}{e^{dx}}$
5	$\varphi = \sin(fx) \cdot \operatorname{tg} \frac{y}{2} - \sqrt[3]{\frac{f \cdot \operatorname{tg}^3 x - 2f^2 \cdot \operatorname{tg} x + 1}{\sin(x+y) + \cos(x-y)}} - \log_2 \frac{e^{fx} + e^{fy} + 1}{\sqrt{\sin^4(x+y) + f^2(x-y) + 3}}$
6	$\gamma = \lg \frac{3g^2 x^4 + gx^2 + y}{e^{x+y} + e^{-x-y}} + \sqrt{\frac{x^2 \sin(gx) - y^4 \cos(gx)}{1 + \sin^2(gx) + \cos^2(gy)}} + \operatorname{tg}^3 \frac{x}{2} + \frac{\cos x \cdot \operatorname{tgy}}{e^{gx}}$
7	$\lambda = \cos(lx) \cdot (1 + \operatorname{ctgy}) + \ln \left \frac{lx^3 - 3l^2 x + y}{3 - \sin(x+y) - \cos(x-y)} \right + \sqrt{\frac{\operatorname{tg}^4 \frac{x}{l} + \operatorname{tg}^2 \frac{y}{d} + 1}{\sqrt{ 3 + \sin x - \cos^3 y }}}$
8	$\mu = m \cdot \operatorname{tg}^2 \frac{x}{2} + \frac{\cos x \cdot \operatorname{tgy}}{e^{mx}} + \sqrt{\frac{x^2 \sin(mx) - y^3 \cos(mx)}{1 + \sin^2(mx) + \cos^2(my)}} + \sin(mx) \cdot \operatorname{tg} \frac{y}{2}$
9	$\nu = \sqrt{\frac{n \cdot \operatorname{tg}^3 x + 3 \cos y}{e^x + e^y}} - \log_2 \frac{e^{nx} + e^{ny} + 1}{\sqrt{\sin^4 x + n \cdot \cos^2 y + 3}} - \ln \left 2 \cos^3 x - \frac{\sqrt{\sin^2 y + 1}}{2 - \cos x} \right $
10	$\sigma = \ln \left \frac{2s \cdot x^3 - 3s^2 x + y}{3 - \sin(x+y) - \cos(x-y)} \right + \sqrt[3]{\frac{\operatorname{tg}^4 \frac{x}{s} + \operatorname{tg}^2 \frac{y}{s} + 1}{\sqrt{ 3 + \sin x - \cos^3 y }}} - \operatorname{tg}^2 \frac{x}{2} + \frac{\cos x \cdot \operatorname{tgy}}{e^{sx}}$

1.3 Питання для перевірки знань

1. Завдяки чому мова Java є крос-платформною?
2. Перерахуйте основні переваги мови Java
3. Які основні напрями розвитку мови Java існують на сьогодні.
4. Дайте характеристику лексиці мови Java.
5. Назвіть правила найменування ідентифікаторів.
6. Розкрийте сенс Null-літералу
7. Перерахуйте примітивні типи даних.
8. Розкрийте сенс тип даних посилання.
9. Дайте характеристику цілим типам даних.
10. Дайте характеристику типам даних з плаваючою крапкою.

ЛАБОРАТОРНА РОБОТА 2

Тема: РОЗГАЛУЖЕНІ ПРОГРАМИ В JAVA

Мета роботи: 1) Вивчити основні принципи побудови лінійних програм. 2) Освоїти навички складання арифметичних виразів.

2.1 Практична частина

1. Увести програму для обчислення функції Y у залежності від умов. Вихідні дані: функція

Набираємо текст програми:

```
public class myTest1 {
    private static Scanner in;
    public static void main(String[] args) {
        double y,x;
        System.out.println("Введіть x");
        in = new Scanner(System.in);
        x=in.nextDouble();
        if(x<0)
            y=Math.pow(x, 2)+4.5;
        else
            y=5-Math.pow((3*x+4), 1/3);
        System.out.println("y="+y);
    }
}
```

2. Запустити створену програму на виконання: натискаємо клавішу CTRL + F11. Примітка - якщо з'явилось ПОВІДОМЛЕННЯ про помилку, то необхідно внести відповідні виправлення в текст програми. Вводимо вихідні значення x .

```
Введіть x
3,2
y=4.0
```


2.2 Самостійна частина

Завдання 2.1. Скласти програму для рішення квадратного рівняння

$$ax^2 + bx + c = 0.$$

Алгоритм роботи програми:

1) Увести з клавіатури значення коефіцієнтів a , b і c .

2) Обчислити дискримінант

3) Порівняти дискримінант із нулем. У залежності від результату порівняння виконати такі дії:

- якщо дискримінант від'ємний, то видати повідомлення «НЕМАЄ КОРЕНІВ» і закінчити виконання програми;

- якщо дискримінант дорівнює нулю, вивести повідомлення: «Один корінь», потім обчислити і вивести на екран значення x ;

- якщо дискримінант позитивний, то вивести повідомлення «Два кореня», потім обчислити і вивести на екран значення x_1 і x_2 ;

Зберегти програму в робочій папці під ім'ям KVUR.CPP. Вирішити з її допомогою рівняння $x^2 + 4x + 4 = 0$, $2x^2 + 6x - 8 = 0$ і $3x^2 + 2x + 4 = 0$.

Завдання 2.2. Скласти програму для обчислення значення функції Z у залежності від умов. Примітка – a і b – постійні величини (константи), значення перемінної x повинне вводитися з клавіатури. Зберегти програму в робочій папці під ім'ям FUNCT3.CPP.

$$Z = \begin{cases} \sin x & x \leq a \\ \cos x & a < x < b \\ \operatorname{tg} x & x \geq b \end{cases}$$

Вихідні дані: $a=5.1$ $b=19.3$ $x=2.6, 9.8, 45.8$.

Завдання 2.3. Скласти програму для обчислення значення функції Y у залежності від умов. Значення перемінної x повинно вводитися з клавіатури, параметри a і b – константи. Варіанти до завдання приведені в таблиці 2.1. Зберегти програму в робочій папці під ім'ям FUNCT2.CPP.

Таблиця 2.1. Варіанти до завдання 2.3

Варіант	Вихідні дані	Варіант	Вихідні дані
1	$y = \begin{cases} \sqrt{a} + \sqrt[3]{x} & x \leq b \\ e^{-bx} & x > b \\ \operatorname{tg}^3 0.1bx & x > b \end{cases}$ $a=0.54 \quad b=2.23 \quad x=1.85, 7.68$	6	$y = \begin{cases} \frac{\sqrt{a^2 + x}}{e^{bx} - a} & \ln x > a \\ \sqrt{\ln^5 a + x^5} & \ln x \leq a \end{cases}$ $a=2.8 \quad b=0.015 \quad x=4.5, 38.1$
2	$y = \begin{cases} \frac{2^3 \sqrt{\ln^2(x-a)}}{a^2 + x^2} & x \leq b \\ x^{\cos(a+b)} & x > b \end{cases}$ $a=1.32 \quad b=9.84 \quad x=4.36, 14,63$	7	$y = \begin{cases} \frac{\sqrt{b + \operatorname{tg}^2 ax}}{a + \sqrt[3]{x}} & \operatorname{tg} x < b \\ e^{x-b} \sin^3(x) & \operatorname{tg} x \geq b \end{cases}$ $a=1.14 \quad b=2.25 \quad x=0.45, 1.42$
3	$y = \begin{cases} \frac{\sqrt[4]{a + bx}}{\ln^2 x} & \ln x > a \\ a^2 e^{-bx} & \ln x \leq a \end{cases}$ $a=2.25 \quad b=0.86 \quad x=1.83, 17.74$	8	$y = \begin{cases} \frac{\sqrt[3]{x} + e^{-\frac{a}{x}}}{bx} & \operatorname{tg} x \leq a \\ \sin^3(bx) & \operatorname{tg} x > a \end{cases}$ $a=2.32 \quad b=0.18 \quad x=6.82, 7.73$
4	$y = \begin{cases} \frac{\ln^3 ax}{\sqrt{bx} + 1} & x \geq \ln b \\ \operatorname{tg}^2 ax + \sqrt{e^{xb}} & x < \ln b \end{cases}$ $a=2.35 \quad b=17.8 \quad x=0.38, 26.2$	9	$y = \begin{cases} \frac{\ln^2 x - \sin \frac{x}{b}}{\sqrt[4]{ax}} & e^x > a \\ a\sqrt{b^2 + e^x} & e^x \leq a \end{cases}$ $a=202 \quad b=17.8 \quad x=4.25, 6.38$
5	$y = \begin{cases} \frac{\operatorname{tg}^2 ax}{b + x^3} & \operatorname{tg} x > a \\ \sqrt{a^2 + e^{bx}} & \operatorname{tg} x \leq a \end{cases}$ $a=0.48 \quad b=2.63 \quad x=0.12, 2.35$	10	$y = \begin{cases} \frac{e^{ x-a }}{\cos^2 bx} & x > a \\ (x-a)^3 + b & x \leq a \end{cases}$ $a=4.8 \quad b=0.18 \quad x=1.36, 6.26$

2.3 Питання для перевірки знань

1. Перерахуйте основні оператори умови у Java.
2. Дайте характеристику конструкції if-else.
3. Дайте характеристику конструкції switch-case.

4. Чим оператори умов у Java відрізняються від відповідних операторів у мові програмування C++.
5. Перерахуйте основні оператори циклів у Java.
6. Дайте характеристику оператору for.
7. Що таке цикл з передумовою та с постумовою.
8. Дайте характеристику оператору while.
9. Чим оператори циклів у Java відрізняються від відповідних операторів у мові програмування C++.
10. Чим оператори введення-виведення у Java відрізняються від відповідних операторів у мові програмування C++.

ЛАБОРАТОРНА РОБОТА 3

Тема: КЛАСИ І ОБ'ЄКТИ

Мета: Вивчити основні принципи створення класів та об'єктів.

3.1 Практична частина

Java є об'єктно-орієнтованою мовою, тому такі поняття як "клас" і "об'єкт" грають в ньому ключову роль. Будь-яку програму на Java можна уявити як набір взаємодіючих між собою об'єктів.

Шаблоном або описом об'єкта є клас, а об'єкт являє екземпляр цього класу. Можна ще провести наступну аналогію. У нас у всіх є деяке уявлення про людину - наявність двох рук, двох ніг, голови, травної, нервової системи, головного мозку і т.д. Є деякий шаблон - цей шаблон можна назвати класом. Реально ж існуюча людина (фактично екземпляр даного класу) є об'єктом цього класу.

Клас визначається за допомогою ключового слова `class`:

```
class Book{  
  
}
```

Вся функціональність класу представлена його членами - полями (полями називаються змінні класу) і методами. Наприклад, клас `Book` міг би мати таке визначення:

```
class Book{  
    public String name;  
    public String author;  
    public int year;  
  
    public void Info(){  
        System.out.printf("Книга '%s' (автор %s) была издана в %d году  
\n", name, author, year);  
    }  
}
```

Таким чином, в класі `Book` визначені три змінних і один метод, який виводить значення цих змінних. Крім звичайних методів в класах використовуються також і спеціальні методи, які називаються

конструкторами. Конструктори потрібні для створення нового об'єкта даного класу і, як правило, виконують початкову ініціалізацію об'єкта. Назва конструктора має збігатися з назвою класу:

```
class Book{

    public String name;
    public String author;
    public int year;

    Book(){
        name = "неизвестно";
        author = "неизвестно";
        year = 0;
    }

    Book(String name, String author, int year){

        this.name = name;
        this.author = author;
        this.year = year;
    }

    public void Info(){
        System.out.printf("Книга '%s' (автор %s) была издана в %d году \n",
name, author, year);
    }
}
```

Тут у класу Book визначено два конструктора. Перший конструктор без параметрів привласнює невизначені початкові значення полів. Другий конструктор присвоює полям класу значення, які передаються через його параметри.

Так як імена параметрів і імена полів класу в даному випадку у нас збігаються - name, author, year, то ми використовуємо ключове слово this. Це ключове слово представляє посилання на поточний об'єкт. Тому в вираженні this.name = name; перша частина this.name означає, що name - це поле поточного класу, а не назву параметра name. Якби у нас параметри і поля називалися по-різному, то використовувати слово this було б необов'язково.

Ми можемо визначити кілька конструкторів для установки різної кількості параметрів і потім викликати один конструктор з іншого:

```

public class Book{

    public String name;
    public String author;
    public int year;

    Book(String name, String author){

        this.name = name;
        this.author = author;
    }

    Book(String name, String author, int year){

        this(name, author);
        this.year = year;
    }
}

```

Наприклад, у нас може скластися ситуація, коли нам потрібно встановити тільки два параметри або тільки три, проте встановлювати в конструктор з трьома параметрами все три поля класу не має сенсу, так як ми можемо передати дві з них в інший конструктор класу, де і відбудеться їх установка. Виклик конструктора проводиться за допомогою ключового слова `this`, після якого йде в дужках список параметрів.

Створення об'єкту

Щоб безпосередньо використовувати клас у програмі, треба створити його об'єкт. Процес створення об'єкта двоступеневий: спочатку оголошується змінна даного класу, а потім за допомогою ключового слова `new` і конструктора безпосередньо створюється об'єкт, на який і буде вказувати оголошена змінна:

```

Book b; // объявление переменной, которая еще не хранит ссылку на объект
b = new Book(); // виділення пам'яті під об'єкт Book

```

Після оголошення змінної `Book b`; ця змінна ще не посилається на жоден об'єкт і має значення `null`. Потім ми створюємо безпосередньо об'єкт класу `Book` за допомогою одного з конструкторів і ключового слова `new`.

Ініціалізатор

Крім конструктора початкову ініціалізацію об'єкта цілком можна було проводити за допомогою ініціалізатора об'єкту. Так, ми б могли замінити конструктор без параметрів наступним блоком:

```
public class Book{

    public String name;
    public String author;
    public int year;

    /*начало блока инициализатора*/
    {
        name = "неизвестно";
        author = "неизвестно";
        year = 0;
    }
    /*конец блока инициализатора*/

    Book(String name, String author, int year){

        this.name = name;
        this.author = author;
        this.year = year;
    }
}
```

Програма з класами

Тепер використовуємо клас в програмі. Створимо в eclipse новий проект. Назвемо його, наприклад, BookStore.

Отже, за замовчуванням у нас в проекті створюється пакет bookstore. Додамо в нього новий клас. Для цього натиснемо на пакет bookstore правою кнопкою миші і в контекстному меню виберемо Новий-> Клас Java ...

Потім у вікні створення нового класу назвемо його Book, а решта опції залишимо за замовчуванням. Після цього в пакет bookstore буде додано новий клас. Змінимо автосгенерований код у файлі Book.java на наступний:

```
package bookstore;

public class Book{

    public String name;
    public String author;
    public int year;
```

```

Book() {

    name = "неизвестно";
    author = "неизвестно";
    year = 0;
}

Book(String name, String author, int year){

    this.name = name;
    this.author = author;
    this.year = year;
}

public void info(){
    System.out.printf("Книга '%s' (автор %s) была издана в %d году \n",
name, author, year);
}
}

```

Тепер перейдемо до коду файлу BookStore.java і змінимо його наступним чином:

```

package bookstore;

public class BookStore {

    public static void main(String[] args) {

        Book b1 = new Book("Война и мир", "Л. Н. Толстой", 1869);
        b1.info();

        Book b2 = new Book();
        b2.info();
    }
}

```

Якщо ми запусимо код на виконання, то консоль виведе нам інформацію про книги b1 і b2:

```

Книга 'Війна і мир »(автор Л. Н. Толстой) була видана в 1869 році
Книга 'невідомо' (автор невідомо) була видана в 0 році

```

Приклад.

```

package lab3;

public class lab {
    // властивості класу
    public double x; // абсциса центру
}

```



```

        public double y; // ордината центру
        public double r; // радіус
        // методи класу
        // виводить на екран параметри кола
        public void printCircle () {
            System.out.println ( "Коло з центром (" + x + ";" + y +
") і радіусом" + r);
        }
        // переміщує центр, рух окружності

    }

package lab3;

public class test {
    public static void main (String [] args) {

        // Створюємо об'єкт (окружність класу Circle), у неї буде
        нульовою
        // радіус і центр в (0.0; 0.0), оскільки всі властивості
        отримають
        // значення за замовчуванням
        lab o1 = new lab ();
        // виводимо на екран параметри кола
        o1.printCircle ();
        // Міняємо абсциссу центру, звертається до властивості x
        o1.x = 3;
        // Міняємо радіус, звертається до властивості r
        o1.r = 12.3;
        // виводимо на екран оновлені параметри кола
        o1.printCircle ();
        // Створюємо інший об'єкт того ж класу
        lab o2 = new lab ();
        o2.r = 3.14;
        o2.zoomCircle (1.66);
        o2.printCircle (); // Окружність з центром (0,0; 0.0) і
        радіусом 5.2124
    }
}

```

Конструктори з параметрами

Спосіб конструювання об'єктів `Box` з різними розмірами. Найпростіше рішення цієї задачі - додавання до конструктору параметрів. Як легко здогадатися, це робить конструктор значно кориснішим. Наприклад, наступна версія класу `Box` визначає конструктор з параметрами, який встановлює розміри паралелепіпеду відповідно до значень цих параметрів. Зверніть особливу увагу на спосіб створення об'єктів `Box`.

3.2 Самостійна частина

Завдання 1. для всіх варіантів завдань створити клас з зазначеними двома полями (Поле 1, Поле 2) і трьома методами:

- конструктор для ініціалізації об'єкта;
- функція формування рядка з інформацією про об'єкт;
- функція обробки значень полів за індивідуальним варіантом.

В основній програмі вводити значення полів кожного об'єкта і виводити результати в консоль. Індивідуальні варіанти завдань наведені в таблиці 3.1.

Таблиця 3.1 - Варіанти індивідуальних завдань.

№ варіанту	Поле 1	Поле 2	Функція обробки полів
1	Номінал купюри (1,2, 5, 10 і т.д.)	Кількість купюр	Вирахувати суму купюр
2	Номінал монети (1,2, 5, 10 і т.д.)	Кількість монет	Вирахувати суму монет
3	Ціна товару	Кількість одиниць товару	Обчислити загальну вартість товару
4	Калорійність 100г продукту	Вага продукту в грамах	Обчислити загальну калорійність продукту

Продовження таблиці 3.1.

5	Дійсне число – ліва межа діапазону	Дійсне число - права межа діапазону	Квадрат довжини діапазону
6	Кількість хвилин	Кількість секунд	Обчислити загальну кількість секунд
7	Кількість годин	Кількість хвилин	Обчислити загальну кількість хвилин
8	Дійсне число - перший катет прямокутного трикутника	Дійсне число - другий катет прямокутного трикутника	Обчислити площу прямокутного трикутника
9	Дійсне число - швидкість руху (м / сек)	Ціле число - час руху в хвилинах	Обчислити пройдену відстань (в метрах)
10	Дійсне число - перший катет прямокутного трикутника	Дійсне число - другий катет прямокутного трикутника	Обчислити довжину гіпотенузи прямокутного трикутника
11	Ціле число - нижня частина трапеції	Ціле число - верхня підстава трапеції	Обчислити напів-суму підстав трапеції
12	Дійсне число - перший катет прямокутного трикутника	Дійсне число - другий катет прямокутного трикутника	Обчислити тангенс кута, протилежного другого катета прямокутного трикутника
13	Дійсне число	Дійсне число	Обчислити напів-різницю чисел

Продовження таблиці 3.1.

14	Дійсне число	Дійсне число	Обчислити корінь квадратний з добутку чисел
15	Ціле число - x	Ціле число - y	Обчислити цілу частину від ділення x на y
16	Ціле число - x	Ціле число - y	Обчислити квадрат меншого з чисел
17	Ціле число - x	Ціле число - y	Обчислити куб більшого з чисел
18	Тривалість телефонної розмови в хвилинах	Вартість однієї хвилини розмови	Обчислити загальну вартість розмови
19	Координата точки на площині (по горизонталі)	Координата точки на площині (по вертикалі)	Визначити периметр прямокутника, обмеженого координатами точки і вісями Ox і Oy
20	Дійсне число - a	Дійсне число - b	Обчислити різницю квадратів чисел $a^2 - b^2$
21	Дійсне число - a	Дійсне число - b	Обчислити різницю квадратів чисел $a^2 + b^2$
22	Координата точки на площині (по горизонталі) - x_1	Координата точки на площині (по горизонталі) - y_1	Визначити площу прямокутника, обмеженого координатами точки і вісями Ox і Oy

Продовження таблиці 3.1.

23	Координата точки на площині (по горизонталі) - x_1	Координата точки на площині (по горизонталі) - y_1	Обчислити відстань від точки до початку координат
24	Кількість годин роботи	Тариф оплати за годину роботи	Загальна вартість роботи
25	Радіус кола	Кут в радіанах	Обчислити довжину дуги
26	Радіус кола підстави	Висота циліндра	Обчислити площу поверхні циліндра
27	Радіус кола основи конуса	Висота конуса	Обчислити обсяг конуса
28	Напруга (в Вольтах)	Опір (в Омах)	Обчислити значення струму (в Амперах)
29	Струм в амперах	Опір резистора R_1 (в Омах)	Обчислити потужність на ділянці електричного кола (в Ватах)
30	Маса тіла - m (в грамах)	Швидкість руху - v (в м / с)	Обчислити кінетичну енергію тіла, що рухається $W_k = \frac{mv^2}{2}$

Завдання 2. Створити клас з полями, зазначеним в індивідуальному завданні (табл. 3.2, стовп 2).

Реалізувати в класі методи:

- конструктор за замовчуванням;
- конструктор перезавантаження з параметрами;
- деструктор для звільнення пам'яті (з повідомленням про знищення об'єкта);
- Функції Обробка даних (1 і 2), зазначені в індивідуальному завданні (Табл. 3.2, стовпці 3 і 4);
- функцію формування рядків інформації про об'єкт.

Таблиця 3.2 - Варіанти індивідуальних завдань

№ варіанту	Батьківський клас і його поля	Функція-метод 1 обробки даних	Функція-метод 2 обробки даних
1	Дата (три числа): день, місяць, рік	Визначити, чи є рік високосний (кратним 4)	Збільшити дату на 5 днів
2	Дата (три числа): день, місяць, рік	Збільшити рік на 1	Зменшити дату на 2 дні
3	Дата (три числа): день, місяць, рік	Визначити, чи збігаються номер місяця і число дня	Збільшити дату на один місяць
4	Час (три числа): години, хвилини, секунди	Обчислити кількість секунд в зазначеному часу	Збільшити час на 5 секунд
5	Час (три числа): години, хвилини, секунди	Обчислити кількість повних хвилин в зазначеному часі	Зменшити час на 10 хвилин

Продовження таблиці 3.2

6	Час (три числа): години, хвилини, секунди	Визначити кількість хвилин до півночі (24:00:00)	Збільшити час на 100 хвилин
7	Координати зображення прямокутника: x1, y1, x2, y2	Обчислити площу прямокутника в пікселях	Зобразити прямокутник на формі (Image) з товщиною лінії 2 пікселя
8	Координати зображення прямокутника: x1, y1, x2, y2	Обчислити довжину діагоналі прямокутника в пікселях	Зобразити прямокутник і його діагональ на формі (Image)
9	Координати зображення прямокутника: x1, y1, x2, y2	Визначити, чи є прямокутник квадратом?	Зобразити прямокутник на формі (Image), зафарбувати зеленим кольором
10	Правильна дріб: чисельник, знаменник	Висловити значення дроби в процентах	Знайти суму чисел значення знаменника
11	Комплексне число: дійсна (a1) і уявна (b1) частини числа	Обчислити модуль комплексного числа	Знайти комплексне число, зворотне заданому

Продовження таблиці 3.2

12	Комплексне число: дійсна і уявна частина числа	Обчислити добуток комплексного числа на число, що вводиться користувачем	Обчислити аргумент комплексного числа в градусах
13	Книга: назва, кількість сторінок, ціна	Обчислити середню вартість однієї сторінки	Збільшити ціну книги в два рази, якщо назва починається зі слова «Програмування»
14	Підручник: назва, автор, рік видання	Вирахувати, скільки років підручнику	Кількість днів, що минули після року видання книги
15	Працівник: прізвище, оклад, рік надходження на роботу	Обчислити стаж роботи працівника на даному підприємстві	Скільки днів минуло після року надходження на роботу
16	Працівник: прізвище, оклад, рік народження	Обчислити вік працівника	Скільки календарних днів до виповнення працівнику 50 років
17	Вектор на площині: координати вектора на площині (x1, y1, x2, y2)	Обчислити довжини вектора	Зобразити лінію вектора на формі (Image) з товщиною лінії 2 пікселя

Продовження таблиці 3.2

18	Вектор на площині: координати вектора на площині (x_1, y_1, x_2, y_2)	Обчислити координати середини вектора	Дорівнює чи кут нахилу вектора 45 градусів?
19	Вектор на площині: координати вектора на площині (x_1, y_1, x_2, y_2)	Обчислити координати вектора, подвоєної довжини	Обчислити площу прямокутного трикутника, утвореного вектором і прямими, паралельними вісями Ox, Oy .
20	Циліндр: діаметр основи, висота	Обчислити об'єм циліндру	Зобразити коло заданого діаметру на формі (Image), зафарбувати червоним кольором.
21	Паралелепіпед: довжини сторін	Обчислити об'єм паралелепіпеда	Обчислити довжину найбільшої діагоналі паралелепіпеду.
22	Паралелепіпед: довжини сторін	Обчислити площу поверхні	Обчислити суму довжин усіх ребер паралелепіпеду.
23	Чотири цілих числа: a, b, c, d	Обчислити середнє арифметичне чисел	Визначити максимальне з чисел

Продовження таблиці 3.2

24	Три речових числа x, y, z	Обчислити середнє геометричне чисел	Визначте, скільки цифр містить сума заданих трьох чисел.
25	Товар: найменування, ціна, рік випуску	Визначити, скільки років тому був випущений товар	Збільшити ціну товару на 20%, якщо в найменуванні товару є слово «TV».
26	Товар: найменування, ціна в гривні, виробник	Перерахувати ціну товару в доларах	Збільшити ціну товару в доларах, якщо назва товару містить слово «Toyota».
27	Координати зображення еліпса: x_1, y_1, x_2, y_2	Визначити, чи є еліпс окружністю?	Зобразити еліпс на формі (Image) зеленим кольором.
28	Книга: назва, кількість сторінок, ціна	Збільшити кількість сторінок на 10	Зменшити ціну в два рази, якщо кількість сторінок більше 100 (Після збільшення)
29	Кімната: довжина, ширина, висота (в метрах)	Площа стін (разом з вікнами і дверима)	Площа стін без вікна (розмір 2×15 м) і дверей (розмір 2×8 м).
30	Працівник: прізвище, посаду, оклад	Збільшити оклад на 15% (кожному працівникові)	Працівникам, у яких прізвище починається з поєднання букв «Іван», привласнити посаду «інженер».

3.3 Питання для перевірки знань

1. Чому об'єктно-орієнтоване програмування прийшло на зміну процедурному?
2. Який основний принцип, що забезпечив високу продуктивність ООП.
3. Розкрийте поняття клас з точки зору ООП.
4. Розкрийте поняття інкапсуляція.
5. Який клас стоїть у голові всіх класів Java.
6. Розкрийте поняття поле класу.
7. Розкрийте поняття метод класу.
8. Які основні модифікатори доступу існують у Java.
9. Які основні правила роботи зі змінними у якості параметрів методів?
10. Розкрийте поняття об'єкт у мові програмування Java.

ЛАБОРАТОРНА РОБОТА 4

Тема: УСПАДКУВАННЯ В JAVA

Мета: Вивчити основні принципи створення успадкування класів.

4.1 Практична частина

Нам потрібні деякі класи для участі в прикладі (рисунок 4.1):

Animal: Клас симулює клас Тварин.

Duck: Клас симулює клас качки, підклас **Animal**.

Cat: Клас симулює клас кішки, підклас **Animal**

Mouse: Клас симулює клас мишей, підклас **Animal**.

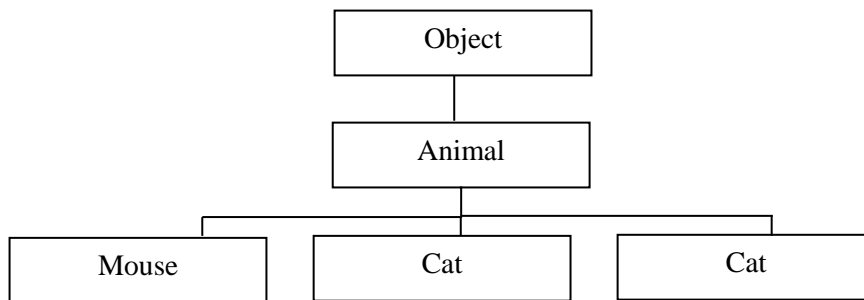


Рисунок 4.1 – Ієрархія успадкування класів

Тут у нас є клас **Animal**, з методом без змісту.

```
public abstract String getAnimalName ();
```

Це метод є абстрактним методом (**abstract**), в підкласах нам потрібно оголосити і застосувати його зміст. Цей метод означає повернення назви тварини.

Class **Animal** має 1 абстрактний метод, який потрібно оголосити як абстрактний (**abstract**). Абстрактний клас має конструктор (**constructor**), але ви не можете створити з нього об'єкт.

Тобто ви хочете створити об'єкт тваринного, вам потрібно створити з певного виду тварини, в даному випадку, ви належні створити з конструктора Cat, Mouse або Duck.

Створимо клас **Animal.java**.

```
package org.o7planning.tutorial.inheritance.animal;

// Клас має мінімум 1 абстрактний метод
// зобов'язаний бути оголошеним як абстрактний.
public abstract class Animal {
    // Ім'я, наприклад Кот Том, Миша Jerry.
    private String name;

    // Constructor за замовчуванням.
    public Animal () {
        // Прикріплює значення за замовчуванням для 'name'.
        this.name = this.getAnimalName ();
    }

    public Animal (String name) {
        this.name = name;
    }

    public String getName () {
        return name;
    }

    public void setName (String name) {
        this.name = name;
    }

    // Це абстрактний метод.
    // Повертає назву даної тварини.
    // Докладний зміст даного методу
    // буде виконант (implemented) в підкласах.
    public abstract String getAnimalName ();
}
```

Далі розглянемо class Cat, успадкований від Animal.

```
package org.o7planning.tutorial.inheritance.animal;

public class Cat extends Animal {

    private int age;
    private int height;

    public Cat (int age, int height) {
        // Викликається Constructor за замовчуванням
        батьківського класу (Animal).
        // З метою прикріпити значення полів (field)
        батьківського класу.
        super ();
        // Потім тільки прикріпити значення його полів (field).
        this.age = age;
        this.height = height;
    }

    public Cat (String name, int age, int height) {
        // Викликається Constructor батьківського класу (Animal)
        // З метою прикріпити значення полів (field)
        батьківського класу.
        super (name);
        // Потім тільки прикріпити значення його полів (field).
        this.age = age;
        this.height = height;
    }

    public int getAge () {
        return age;
    }

    public void setAge (int age) {
        this.age = age;
    }

    public int getHeight () {
        return height;
    }
}
```

```

    }

    public void setHeight (int height) {
        this.height = height;
    }

    // Виконати (implement) абстрактний метод оголошений в
    батьківському класі.
    @Override
    public String getAnimalName () {
        return "Cat";
    }

}

```

Створимо class Mouse, успадкований з Animal.

```

package org.o7planning.tutorial.inheritance.animal;

public class Mouse extends Animal {

    private int weight;

    // Constructor за замовчуванням.
    public Mouse () {
        // Gọi tới cấu tử Mouse (int)
        // Call to constructor Mouse (int)
        this (100);
    }

    // Constructor має 1 параметр
    public Mouse (int weight) {
        // Якщо ви не викликаєте ніякої super (...).
        // Java викличе super () за замовчуванням.
        // super ();
        this.weight = weight;
    }

    // Constructor має 2 параметра.
    public Mouse (String name, int weight) {
        super (name);
        this.weight = weight;
    }
}

```

```

    public int getWeight () {
        return weight;
    }

    public void setWeight (int weight) {
        this.weight = weight;
    }

    @Override
    public String getAnimalName () {
        return "Mouse";
    }
}

```

Для тестування коду створимо наступний файл InstanceofDemo.java.

```

package org.o7planning.tutorial.inheritance.demo;

import org.o7planning.tutorial.inheritance.animal.Animal;
import org.o7planning.tutorial.inheritance.animal.Cat;
import org.o7planning.tutorial.inheritance.animal.Mouse;

public class InstanceofDemo {

    public static void main (String [] args) {

        // Створити об'єкт тварина.
        // Animal є абстрактним класом,
        // ви не можете створити об'єкт з Constructor в Animal.
        Animal tom = new Cat ( "Tom", 3, 20);

        System.out.println ( "name:" + tom.getName ());
        System.out.println ( "animalName:" + tom.getAnimalName
());

        // Використовуючи оператор 'instanceof' перевірити
        // чи є об'єкт певним видом чи ні.
        boolean isMouse = tom instanceof Mouse;
        System.out.println ( "Tom is mouse?" + IsMouse);

        boolean isCat = tom instanceof Cat;
        System.out.println ( "Tom is cat?" + IsCat);

```



```

        boolean isAnimal = tom instanceof Animal;
        System.out.println ( "Tom is animal?" + IsAnimal);
    }
}

```

Результати запуску програми (рисунок 4.2).

```

<terminated> InheritanceDemo1 [Java Application] D:\DevProgra
name: Tom
animalName: Cat
Tom is mouse? false
Tom is cat? true
Tom is animal? true

```

Рисунок 4.2 – Результат роботи програми

Створимо файл **InheritMethodDemo.java**.

```

package org.o7planning.tutorial.inheritance.demo;

import org.o7planning.tutorial.inheritance.animal.Cat;

public class InheritMethodDemo {

    public static void main (String [] args) {

        // Створити об'єкт Cat.
        Cat tom = new Cat ("Tom", 3, 20);

        // Викликати методи успадковані від батьківського класу
        (Animal).

        System.out.println ("name:" + tom.getName ());
        System.out.println ("animalName:" + tom.getAnimalName
        ());

        System.out.println ("-----");
    }
}

```

```

        // Викликати методи оголошені в класі Cat.
        System.out.println ("Age:" + tom.getAge ());
        System.out.println ("Height:" + tom.getHeight ());
    }
}

```

Результат роботи файлу «InheritMethodDemo.java» (рисунок 4.3).

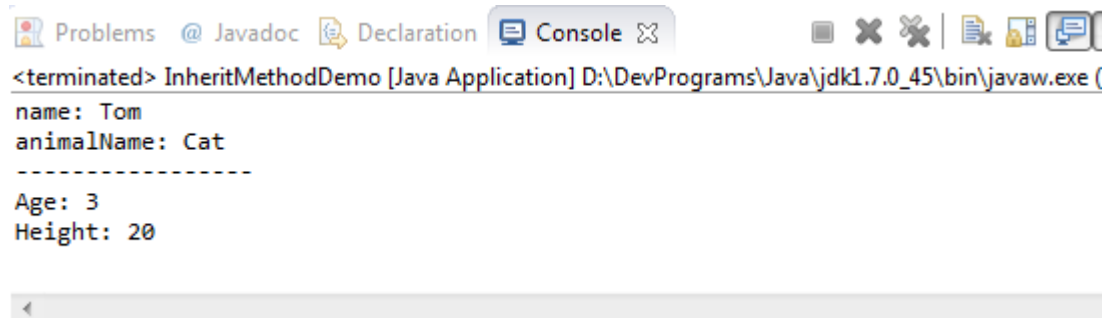


Рисунок 4.3 - Результат роботи файлу «InheritMethodDemo.java»

Створимо файл CastDemo.java.

```

package org.o7planning.tutorial.inheritance.demo;

import java.util.Random;
import org.o7planning.tutorial.inheritance.animal.Animal;
import org.o7planning.tutorial.inheritance.animal.Cat;
import org.o7planning.tutorial.inheritance.animal.Mouse;

public class CastDemo {

    // Цей метод повертає випадкове тварина.
    public static Animal getRandomAnimal () {
        // Повертає випадкове значення 0 або 1.
        int random = new Random (). nextInt (2);

        Animal animal = null;
        if (random == 0) {
            animal = new Cat ( "Tom", 3, 20);
        } Else {
            animal = new Mouse ( "Jerry", 5);
        }
        return animal;
    }
}

```

```

public static void main (String [] args) {

    Animal animal = getRandomAnimal ();

    if (animal instanceof Cat) {
        // Cast в вид Cat.
        Cat cat = (Cat) animal;

        // I викликає метод класу Cat.
        System.out.println ( "Cat height:" + cat.getHeight
());
    } Else if (animal instanceof Mouse) {
        // Cast в вид Mouse.
        Mouse mouse = (Mouse) animal;

        // I викликає метод класу Mouse.
        System.out.println ( "Mouse weight:" +
mouse.getWeight ());
    }

}
}
}

```

4.2 Самостійна частина

Завдання 1. побудувати клас 1-го рівня з зазначеними в індивідуальному завданні (табл. 4.1) полями і методами:

- конструктор;
- функція, яка визначає «якість» об'єкта - Q по заданій формулі (таблиця 4. 1, стовп 2);
- висновок інформації про об'єкт.

Побудувати клас 2-го рівня (клас-нащадок), який містить:

- додаткове поле P ;

- функція, яка визначає «якість» об'єкта класу 2-го рівня - Q_p , яка перебиває функцію якості класу 1-го рівня (Q), виконуючи обчислення за новою формулою (табл. 4.1, стовп 3).

Створити проект для демонстрації роботи: введення і виведення інформації про об'єкти класів 1-го і 2-го рівнів.

Таблиця 4.1. Варіанти індивідуальних завдань.

№ варіанту	Поля і функція «якості» (Q) класу 1-го рівня	Поле і функція «якості» Q_p класу 2-го рівня
1	комп'ютер: - найменування процесора; - тактова частота процесора (МГц); - обсяг оперативної пам'яті (Мб). $Q = (0,1 \cdot \text{частота}) + \text{пам'ять}$	P : обсяг вінчестера (Гб) $Q_p = Q + 0,5 \cdot P$
2	Оператор мобільного зв'язку: - назва оператора; - вартість 1 хвилини розмови; - площа покриття. $Q = 100 \cdot \text{площа покриття} / \text{вартість 1 хвилини розмови}$	P : наявність плати за кожне з'єднання $Q_p = 0,7 \cdot Q$, якщо P - істина, інакше $Q_p = 1,5 \cdot Q$
3	Товар на складі: - найменування; - ціна; - кількість. $Q = \text{ціна} / \text{кількість}$	P : рік випуску товару $Q_p = Q + 0,5 \cdot (T - P)$, де T - поточний рік

Продовження таблиці 4.1.

4	<p>Кабель:</p> <ul style="list-style-type: none"> - тип; - кількість жил кабелю; - діаметр. <p>$Q = \text{діаметр} / \text{кількість жив}$</p>	<p>P: наявність обплетення</p> <p>Q_p: якщо P - істина, то $Q_p = 2 \cdot Q$; інакше $Q_p = 0,7 \cdot Q$</p>
5	<p>Підручник з програмування:</p> <ul style="list-style-type: none"> - назва; - кількість сторінок; - ціна. <p>$Q = \text{ціна} / \text{кількість сторінок}$</p>	<p>P: рік видання</p> <p>$Q_p = Q - 0,2 \cdot (T - P)$, де T - поточний рік</p>
6	<p>Мобільний телефон:</p> <ul style="list-style-type: none"> - марка; - ціна; - обсяг пам'яті. <p>$Q = \text{обсяг пам'яті} / \text{ціна}$</p>	<p>P: кількість SIM карт</p> <p>$Q_p = Q \cdot P$</p>
7	<p>Супутникова антена (тарілка):</p> <ul style="list-style-type: none"> - діаметр; - матеріал; - ціна. <p>$Q = \text{діаметр} / \text{ціна}$</p>	<p>P: тип підвіски (азимутальна, полярна, тороїдальна)</p> <p>Q_p = Q, якщо тип підвіски азимутальний, Q_p = 2 · Q, якщо тип підвіски полярний, Q_p = 2,5 · Q, якщо тип підвіски тороїдальний</p>

Продовження таблиці 4.1.

8	<p>Іспит:</p> <ul style="list-style-type: none"> - дисципліна; - число студентів на іспиті; - тривалість іспиту (ч). <p>$Q = \text{число студентів} / \text{тривалість}$</p>	<p>P: відсоток двійок</p> $Q_p = Q \cdot (100 - P) / 100$
9	<p>Спортсмен:</p> <ul style="list-style-type: none"> - прізвище; - число змагань; - сума місць, зайнятих спортсменом в змаганнях. <p>$Q = (\text{число змагань}) / (\text{сума місць})$</p>	<p>P: займав чи хоча б раз перше місце</p> <p>$Q_p = 1,5 \cdot Q$, якщо P - істина, інакше - $Q_p = Q$</p>
10	<p>Програміст:</p> <ul style="list-style-type: none"> - прізвище; - число програм, написаних програмістом; - число мов програмування, якими він пише програми. <p>$Q = (\text{число програм}) * (\text{число мов})$</p>	<p>P: число програм, які працюють правильно</p> $Q_p = Q \cdot P / (\text{число всіх програм})$
11	<p>Вистава:</p> <ul style="list-style-type: none"> - назва; n_1 - число глядачів на початку; n_2 - число глядачів в кінці. <p>$Q = (n_2 - n_1) / n_1$</p>	<p>P: рік написання п'єси</p> $Q_p = Q \cdot (T - P + 1)$, де T - поточний рік
12	<p>Алмаз:</p> <ul style="list-style-type: none"> - назва; - вага (в каратах); - якість ограновування в балах (число). 	<p>P: колір (білий, блакитний, жовтий, і тп)</p> <p>Q_p: якщо колір блакитний, то $Q_p = Q + 1$; а якщо жовтий,</p>

Продовження таблиці 4.1.

	$Q = 0,4 \cdot \text{вага} + 0,6 \cdot \text{якість}$ ограновування	то $Q_p = Q - 0,5$ інакше $Q_p = Q$
13	Комп'ютерна мережа: - Назва організації; - число робочих станцій; - середня відстань між станціями (М). $Q = \text{число станцій} \cdot \text{середня відстань}$	Р: середня швидкість передачі даних в мережі (Мб / с) $Q_p = Q \cdot P$
14	Армія: - вид військ; - чисельність (тис людей); - озброєність (бали - число). $Q = 0,3 \cdot \text{чисельність} + 0,7 \cdot \text{озброєність}$	Р: досвід (кількість місяців, протягом яких армія вела бойові дії) $Q_p = Q \cdot (P + 1)$
15	Автомобіль: - Марка автомобіля; - потужність двигуна (кВт); - число, місць. $Q = 0,1 \cdot \text{потужність} \cdot \text{число, місць}$	Р: рік виготовлення $Q_p = Q - 1,5 \cdot (T - P)$, де Т - поточний рік
16	Партія: - назва; - чисельність (тис. членів); - відсоток голосів на останніх виборах. $Q = 0,3 \cdot \text{чисельність} + 0,7 \cdot \text{відсоток гол}$ Р: чисельність партії в минулому році	Q_p : якщо чисельність в поточному році збільшилася, то $Q_p = 1,2 \cdot Q$; а якщо скоротилася, то $Q_p = 0,8 \cdot Q$
17	Вищий навчальний заклад: - назва закладу; - кількість студентів, зарахованих на 1-й курс;	Р: відсоток випускників, які працюють за фахом $Q_p = P \cdot Q$

Продовження таблиці 4.1.

	<p>- кількість випускників. $Q = \text{кількість випускників} / \text{кількість зарахованих}$</p>	
18	<p>Солдат: - прізвище; - зріст (м); - вага, кг). $Q = \text{зріст} \cdot \text{вага}$</p>	<p>P: освіта (початкова, середня, вища) Qp: якщо освіта вища, то $Qp = 2 \cdot Q$; а якщо початкова, то $Qp = 0,5 \cdot Q$; інакше $Qp = Q$</p>
19	<p>Телевізор: - фірма; - діагональ екрану (дюйм); - звукова потужність (дб). $Q = \text{діагональ} + (0,05 \cdot \text{потужність})$</p>	<p>P: країна-виробник Qp: якщо країна - Японія, то $Qp = 2 \cdot Q$; а якщо Сінгапур або Корея, то $Qp = 1,5 \cdot Q$; інакше $Qp = Q$</p>
20	<p>Мітинг: - назва події; n1 - число ораторів; n2 - число учасників. $Q = n1 / n2$</p>	<p>P: число груп ораторів, які висловлювали однакові думки $Qp = Q + P / n2$</p>
21	<p>Будинок: - номер будинку; - число квартир; - рік споруди. $Q = (\text{число квартир}) + 2 \cdot (\text{T} - \text{рік споруди})$, де T - поточний рік</p>	<p>P: район (центр, околиця, і т.д.) Qp: якщо район - центр, то $Qp = 2 \cdot Q$; інакше $Qp = 0,5 \cdot Q$</p>

Продовження таблиці 4.1.

22	<p>Керівник:</p> <ul style="list-style-type: none"> - прізвище; - самооцінка (в балах - ціле число); - оцінка іншими людьми (в балах). <p>$Q = (\text{оцінка іншими}) / \text{самооцінка}$</p>	<p>Р: оцінка нащадками (в балах) $Q_p = 0,3 \cdot Q + 0,7 \cdot P$</p>
23	<p>Студент:</p> <ul style="list-style-type: none"> - прізвище; - середній бал; - курс. <p>$Q = 0,2 \cdot \text{Середній бал} \cdot \text{курс}$</p>	<p>Р: вивчає дисципліни англійською мовою</p> <p>$Q_p = 2 \cdot Q$, якщо Р - істина, інакше $Q_p = 0,9 \cdot Q$</p>
24	<p>антена:</p> <ul style="list-style-type: none"> - назва; - потужність; - висота (м). <p>$Q = \text{потужність} + 0,5 \cdot \text{висота}$</p>	<p>Р: коефіцієнт випромінювання</p> <p>$Q_p = Q - 0,1 \cdot P$</p>
25	<p>Літак:</p> <ul style="list-style-type: none"> - марка; - кількість двигунів; - висота польоту. <p>$Q = \text{к-ть двигунів} \cdot \text{висота польоту} / 1000$</p>	<p>Р: країна-виробник</p> <p>Q_p: якщо країна - Росія, то</p> <p>$Q_p = Q + 1$;</p> <p>а якщо Франція, то</p> <p>$Q_p = Q + 0,5$</p>
26	<p>Студент:</p> <ul style="list-style-type: none"> - прізвище; - число іспитів; - число оцінок «п'ять». <p>$Q = \text{число оцінок «п'ять»} / \text{число іспитів}$</p>	<p>Р: число оцінок «три»</p> <p>$Q_p = Q - 0,5P$</p>

Продовження таблиці 4.1.

27	<p>Фірма:</p> <ul style="list-style-type: none"> - назва; - дохід (тис \$); - рейтинг (в балах). <p>$Q = \text{дохід} \cdot \text{рейтинг}$</p>	<p>P: інвестиції в фірму (Тис \$)</p> <p>$Q_p = P_3 + Q$</p>
28	<p>Військовий корабель:</p> <ul style="list-style-type: none"> - назва; - довжина; - число гармат головного калібру. <p>$Q = (\text{число гармат}) / \text{довжина}$</p>	<p>P: крейсерська швидкість (В морських вузлах)</p> <p>$Q_p = 0.25Q + P$</p>
29	<p>Коробка сірників:</p> <ul style="list-style-type: none"> - фірма виробник; - число сірників в коробці; - час горіння одного сірника (с). <p>$Q = (\text{число сірників}) \cdot \text{час}$</p>	<p>P: середній % бракованих сірників в коробці</p> <p>$Q_p = (100-P) Q / 100$</p>
30	<p>Полководець:</p> <ul style="list-style-type: none"> - прізвище; - число битв; - число перемог. <p>$Q = (\text{число перемог})^2 / (\text{число битв})$</p>	<p>P: число перемог з меншими, ніж у супротивника, силами</p> <p>$Q_p = P_2 / \text{битви} + Q$</p>

Завдання 2. Написати програму відповідно до завдання. У всіх класах описати необхідні конструктори, за допомогою яких будуть створюватися об'єкти класів. Параметри створюваних об'єктів задавати константами або вводити з клавіатури (за бажанням студента) і передавати в конструктори об'єктів у вигляді параметрів. Висновок інформації повинен здійснюватися на консоль.

Таблиця 4.2. Варіанти індивідуальних завдань.

№ варіанту	Задача
1	<p>Створити клас Автомобіль з властивостями: Назва, Максимальна швидкість (в км / год). Визначити 2 віртуальних метода: метод «Вартість»-вартість автомобіля, яка розраховується за формулою. Максимальна швидкість * 100 і метод «Оновлення моделі», що збільшує максимальну швидкість на 10. Визначити також метод «Інформація», який повертає рядок, що містить інформацію про об'єкт: назва, максимальна швидкість і вартість. Створити також клас спадкоємець Представницький автомобіль, в якому перевизначити методи: метод «Вартість» повертає число, рівне. Максимальна швидкість * 250, а метод «Оновлення моделі» збільшує швидкість на 5 км / год. У головній програмі (або після натискання на кнопку) створити об'єкт класу Автомобіль з максимальною швидкістю 140 км / год і класу Представницький автомобіль з максимальною швидкістю 160 км / год.</p> <p>Вивести на екран (або форму) інформацію про автомобілі. Оновити моделі автомобілів і знову вивести інформацію про них.</p>
2	<p>Створити клас Трикутник, заданий значеннями довжин трьох сторін (a, b, c), з методами «Периметр» і «Площа». Визначити також метод «Інформація», який повертає рядок, що містить інформацію про трикутник: довжини сторін, периметр і площу.</p> <p>Створити також клас спадкоємець Чотирикутник, з додатковими параметрами - довжиною четвертого боку (d) і довжинами діагоналей (e, f) і перевизначити методи «Периметр» (сума всіх сторін) і «Площа». Площу обчислювати за наступною формулою</p> $S = \sqrt{\frac{4e^2 f^2 - (b^2 + d^2 - a^2 - c^2)^2}{16}}.$

Продовження таблиці 4.2.

	<p>У головній програмі (або після натискання на кнопку) створити об'єкт класу Трикутник і об'єкт класу Чотирикутник і вивести інформацію про них. Для спрощення перевірки рекомендується в якості конкретного об'єкту класу чотирикутник взяти квадрат.</p>
3	<p>Створити клас Комп'ютер з властивостями: Частота процесора (в МГц), кількість ядер, об'єм пам'яті (в МБ), обсяг жорсткого диску (в ГБ). Визначити два віртуальних методу: «Вартість», що повертає приблизну розрахункову вартість комп'ютера, яка розраховується за формулою: $\text{Частота процесора} * \text{кількість ядер} / 100 + \text{кількість пам'яті} / 80 + \text{обсяг жорсткого диска} / 20$ і логічний метод «Придатність», який повертає істину (true), якщо частота процесора не менше 2000 МГц, кількість ядер не менше 2, обсяг пам'яті не менше 2048 МБ, і обсяг жорсткого диску не менше 320 Гб. Визначити також метод «Інформація», який повертає рядок, що містить інформацію про комп'ютер: частоту процесора, кількість ядер, об'єм пам'яті, обсяг жорсткого диску, вартість і придатність для наших потреб. Створити також клас спадкоємець Ноутбук, з додатковою властивістю. Тривалість автономної роботи (в хвилинах) і перевизначити методи: метод «Вартість» повертає число, рівне вартості звичайного комп'ютера + кількість хвилин автономної роботи / 10, а метод «Придатність» повертає істину, тоді коли і ноутбук придатний як звичайний комп'ютер, і тривалість автономної роботи не менш 60 хвилин. У головній програмі (або після натискання на кнопку) створити звичайний комп'ютер і ноутбук, і вивести інформацію про них.</p>

Продовження таблиці 4.2.

4	<p>Створити клас Прямокутник, заданий значеннями довжин двох сторін (a і b), з віртуальними методами «Периметр» і «Площа», які повертають периметр і площу відповідно, а також віртуальний метод «Збільшити в два рази», що збільшує в два рази кожен зі сторін. Визначити також метод «Інформація», який повертає рядок, що містить інформацію про трикутник: довжини сторін, периметр і площу. Створити також клас спадкоємець Прямокутник з округленими кутами, з додатковим параметром радіус заокруглення (r). Для нього перевизначити. Периметр за формулою $p = 8 \cdot r + 2 \cdot \pi \cdot r$, де p - периметр звичайного прямокутника з тими ж сторонами, а Площа за формулою $S = 4 \cdot r^2 + \pi \cdot r^2$, де S - площа звичайного прямокутника. Також перевизначити метод «Збільшити в два рази» так, щоб він також збільшував в два рази радіус заокруглення (як і раніше збільшуючи боку в два рази). У головній програмі (або після натискання на кнопку) створити звичайний прямокутник і прямокутник з округленими кутами і вивести інформацію про них. Після цього збільшити обидва прямокутника в два рази і видати оновлену інформацію.</p>
5	<p>Створити клас Фотоапарат з властивостями: Модель, Оптичне збільшення (Zoom, дійсне число від 1 до 35) і матеріал корпусу (метал або пластик). Визначити віртуальний метод: метод «Вартість» - повертає число - вартість фотоапарата (в \$), що розраховується за формулою $(Zoom + 2) \cdot 10$, якщо корпус пластиковий і $(Zoom + 2) \cdot 15$, якщо матеріал металевий. Визначити також метод «Інформація», який повертає рядок, що містить інформацію про об'єкт: Модель, Zoom і Вартість. Також визначити логічний метод «Дорогий», який буде повертати істину (true), якщо вартість фотоапарата більше 200 \$.</p>

Продовження таблиці 4.2.

	<p>Створити також клас спадкоємець Цифровий фотоапарат, в якому буде додатковий цілий параметр - кількість мегапікселів і перевизначити метод «Вартість», який буде повертати число, рівне вартості звичайного фотоапарату помножити на кількість мегапікселів, а також визначити новий метод «Оновлення моделі», який збільшує кількість мегапікселів на 2.</p> <p>У головній програмі (або після натискання на кнопку) створити об'єкт класу Фотоапарат з 4-ми кратним оптичним збільшенням ($Zoom = 4$) і пластиковим корпусом, а також Цифровий фотоапарат з металевим корпусом, 8-ю мегапікселями і 3-кратним оптичним збільшенням. Вивести на екран (або форму) інформацію по фотоапаратах і про те, чи є вони дорогими. Оновити моделі цифрового фотоапарата і знову вивести інформацію про нього.</p>
6	<p>Створити клас Студент з властивостями: ПБ, факультет, курс, мінімальна оцінка за іспитами за останню сесію (за 5-ти бальною системою). Визначити віртуальні методи: «Перевести на наступний курс», що збільшує курс на 1, якщо мінімальна оцінка не менше 3, інакше не робить нічого, а також «Стипендія», який повертає стипендію (в грн): 0 грн, якщо мінімальна оцінка не вище 3, 200 грн, якщо мінімальна оцінка дорівнює 4 і 300 грн, якщо мінімальна оцінка дорівнює 5. Визначити також метод «Інформація», який повертає рядок, що містить інформацію про студента: ПБ, факультет, курс, мінімальна оцінка за іспитами та нараховану стипендію. Створити також клас спадкоємець Студент-контрактник, в якому буде додатковий логічний параметр - сплачений контракт і перевизначені методи «Перевести на наступний курс», що збільшує курс</p>

Продовження таблиці 4.2.

	<p>на 1, якщо мінімальна оцінка не менше 3 і за контракт сплачено, а також «Стипендія» повертає завжди 0 грн. В головній програмі (або після натискання на кнопку) створити об'єкт класу Студент та 2 об'єкти класу Студент-контрактник (один з яких сплатив за контракт, а інший ні). Видати інформацію про студентів, потім застосувати до них метод «Перевести на наступний курс» і знову видати інформацію про них.</p>
7	<p>Створити клас Коло заданий своїм радіусом (r), з віртуальним методом «Площа», що повертає площу кола, а також віртуальний метод «Збільшити» з одним речовим параметром - у скільки разів збільшити, що збільшує радіус в задану кількість разів. Визначити також метод «Інформація», який повертає рядок, що містить інформацію про коло: радіус і площа.</p> <p>Створити також клас спадкоємець Кільце, з додатковим параметром- внутрішнім радіусом (r_{in}), при цьому успадкований від батька радіус буде позначати зовнішній радіус. Перевизначити метод «Площа», як різницю між площею зовнішнього кола мінус площа внутрішнього кола. Також доопределити метод «Збільшити», щоб він збільшував також і внутрішній радіус. У головній програмі (або після натискання на кнопку) створити звичайне коло і кільце і вивести інформацію про них. Після цього збільшити обидва об'єкта в півтора рази і видати оновлену інформацію.</p>
8	<p>Створити клас Табуретка з властивостями: Висота (h, в см), Якість виробу (низька, середня, висока). Визначити два віртуальних методу: «кількість деревини», яке вимагає табуретка, за формулою $4 \cdot h + 12$, якщо якість низька, і $5 \cdot h + 14$, якщо якість середня або висока, а також «вартість», рівна $d \cdot 2$, для низької якості, $d \cdot 3$, для середньої якості, $d \cdot 4$, для високої якості,</p>

Продовження таблиці 4.2.

	<p>де d - кількість деревини, яку вимагає даний об'єкт. Визначити також метод «Інформація», який повертає рядок, що містить інформацію про об'єкт: Висоту, якість матеріалу, кількість деревини і вартість.</p> <p>Створити також клас спадкоємець Стілець з додатковою властивістю: висота спинки (h_2, в см), і перевизначити метод «кількість деревини», за формулою $d + 2h_2 + 5$, де d - кількість деревини, які вимагає табуретка з такими ж параметрами (Метод «вартість» не перевизначати). У головній програмі (або після натискання на кнопку) створити екземпляри класів Табуретка і Стілець, і надрукувати інформацію в такому вигляді: «табуретка» + інформація про табуретку і «стілець» + інформація про стільці.</p>
9	<p>Створити клас Фільм з властивостями: Назва, Режисер, тривалість (в хвиликах), кількість акторів. Визначити віртуальний метод: «Вартість», що повертає приблизну розрахункову вартість фільму (в тис. \$), Що розраховується за формулою тривалість * 20 + кількість акторів * 30, але якщо режисер = «Стівен Спілберг» або «Джеймс Кемерон », то вартість в два рази вище (у порівнянні з вищевказаної формулою). Визначити також метод «Інформація», який повертає рядок, що містить інформацію про фільм: Назва, режисер, тривалість, кількість акторів і вартість. Створити також клас спадкоємець Мультфільм, в якому перевизначити метод «Вартість» за формулою тривалість * 25 + кількість акторів * 10 (незалежно від режисера). У головній програмі (або після натискання на кнопку) створити 2 фільми з режисерами: «Стівен Спілберг» і «Єжи Гофман», а також мультфільм і вивести інформацію про них.</p>

Продовження таблиці 4.2.

10	Створити клас Літак з властивостями: Марка, Модель, Максимальна швидкість (в км / год), Максимальна висота (в метрах). Визначити віртуальний метод «Вартість» - вартість літака, яка розраховується за формулою: Максимальна швидкість * 1000 + Максимальна висота * 100. Визначити також метод «Інформація», який повертає рядок, що містить інформацію про об'єкт: Марка, Модель, Максимальна швидкість, Максимальна висота і Вартість . Створити також клас спадкоємець Бомбардувальник, в якому перевизначити метод «Вартість», який поверне подвоєну вартість щодо формули для класу Літак. Також створити клас Винищувач - спадкоємець класу Літак, для якого перевизначити метод «Вартість» як потрібну вартість, щодо формули вартості для Літака .В головній програмі (або після натискання на кнопку) створити об'єкт класу Літак, класу Бомбардувальник, класу Винищувач. Вивести на екран (або форму) інформацію про літаки.
----	--

Завдання 3. Описати кілька класів: абстрактний клас-батько з полями і методами, характерними для всіх об'єктів завдання, при цьому деякий метод або методи може бути чисто віртуальним і кілька класів-спадкоємців, що визначають додаткові поля (при необхідності) і перевизначити деякі віртуальний метод (методи) .

У головній програмі повинен бути оголошений масив (статичний або динамічний) об'єктів (батьківського класу), в який поміщаються класів-нащадків (В С ++ необхідно використання масиву покажчиків на об'єкти).

Таблиця 4.3. Варіанти індивідуальних завдань.

№ варіанту	Задача
1	<p>Програмно промоделювати розмови людей. Всього є 5 людей. Кожна людина має ім'я (рядок) і вік (число). Встановіть вік кожної людини - випадкове число від 20 до 40, а імена встановіть випадковим чином зі списку «Олександр», «Андрій», «Анастасія», «Ірина», «Наталя», «Павло», «Роман», «Світлана», «Сергій», «Тетяна».</p> <p>Будь-яка людина здатна виконувати дві дії:</p> <ul style="list-style-type: none"> - вітатися з іншою людиною; - розповідати про себе. <p>Люди діляться на 3 типи (різні класи):</p> <p>1 (Формалісти) Вітаються з усіма так: «Здрастуй, <ім'я>», де <Ім'я> - ім'я людини, з яким він вітається.</p> <p>2 (Неформали) З усіма вітаються: «Привіт, <ім'я>!»</p> <p>3 (Реалісти) Якщо вік співрозмовника менше або дорівнює або більше не більш ніж на 5 років, каже «Привіт, <ім'я>!», інакше «Здрастуй, <Ім'я> ».</p> <p>У програмній реалізації вітання повинне бути реалізоване як поліморфний метод, який приймає параметр - людина і повертає рядок. Розповідь про людину є рядком виду «Мене звать Вася, мій вік 21 років, я неформал »(замість Вася ім'я людини, замість 21 його вік, як бачите у людей з граматиною не все гаразд і вони кажуть «років» після будь-якого числа, якщо хочете, можете реалізувати правильну граматику, замість неформал може бути формаліст, або реаліст). Програма повинна показати інформацію про всіх людей. Потім всі люди повинні привітатися один з одним в такому порядку: перший вітається з другим, потім другий з першим, потім перший з третім, третій з першим, і так</p>

Продовження таблиці 4.3.

	<p>далі перший з усіма і всі з першим, потім другий з третім, третій з другим і т.д. Потрібно видати в окремих рядках ім'я людини, який вітається, двокрапка, вітання:</p> <p>Петя: Привіт, Вася!</p> <p>Вася: Здравствуй, Петя</p> <p>Петя: Привіт, Женя!</p> <p>...</p>
2	<p>Програмно промоделювати стрілянину по мішені групою осіб. Кожна людина має своє ім'я, вік (в роках) і стаж навчання стрільбі (в роках).</p> <p>Люди діляться на новачків, досвідчених і ветеранів (нащадки класу людина). Для кожної людини визначте поліморфний метод «Стріляти» без параметрів, які повертають логічне значення (потрапив - true, не потрапив - false). Попадання визначається випадковим чином, причому для новачка ймовірність потрапити дорівнює $0,01 * \text{стаж навчання}$; для досвідченого = $0,05 * \text{стаж навчання стрільбі}$; для ветерана = $0,9 - 0,01 * \text{вік}$.</p> <p>Люди стріляють по черзі, починаючи з першого, поки хтось не потрапить в мішень. Стрільба припиняється після того, як хтось потрапив або всі вистрілили по одному разу. Після кожного пострілу потрібно виводити на екран всю інформацію про стріляючого і результат стрільби. У головній програмі створіть масив з 7 людей в такому порядку: новачок, досвідчений, ветеран, досвідчений, новачок і зробіть стрілянину з виведенням її результатів.</p>
3	<p>Програмно промоделювати задачу заліку студентами. Кожен студент характеризується ПБ і кількістю відвіданих занять. Також відомо кількість загальних занять = 20 Студенти діляться</p>

Продовження таблиці 4.3.

	<p>на звичайних, кмітливих і геніїв (нащадки класу студент). Для кожної людини визначте поліморфний метод «Здати залік» без параметрів, повертають логічне значення (здав - true, не здав - false). Звичайні студенти точно здають залік, якщо відвідали всі заняття, якщо були більш ніж на половині занять, то здають з ймовірністю 0,5; інакше не здають. Кмітливі студенти теж точно здають залік, якщо відвідали всі заняття, якщо були більш ніж на половині занять, то здають з ймовірністю 0,7; інакше не здають залік, генії точно здають залік, якщо були хоча б на одному занятті, інакше не здають. Створіть масив з 10 студентів (5 звичайних, 4 кмітливих і 1 генія), задайте їх ПІБ та кількість відвіданих занять константами, зазначеними в конструкторах об'єктів і промоделюйте задачу заліку з висновком докладних результатів (всіх відомостей про студентів, а також результат задачі).</p>
4	<p>Промоделювати виліток листів сталі.</p> <p>Листи сталі характеризуються товщиною (в мм), і щільністю сталі (в кг / м³).</p> <p>Листи діляться на квадратні (додатково задаються одним числом - шириною і довжиною одночасно, в мм), прямокутні (задаються шириною і довжиною, в мм) і трикутні (у вигляді прямокутного трикутника, задаються двома катетами, в мм). Для кожного типу сталі визначте віртуальний метод «Площа» - повертає площу аркуша. Також визначте метод «Вага», який обчислює вагу листа, множачи площу листа на його товщину і щільність сталі. Також задайте метод «Інформація», який видаватиме інформацію про листи. У головній програмі створіть масив з 15 листів сталі,</p>

Продовження таблиці 4.3.

	створивши 5 квадратних, 7 прямокутних і 3 трикутних листа випадкових розмірів. Виведіть інформацію по листах і порахуйте сумарну площу і сумарну вагу всіх листів.
--	--

4.3 Питання для перевірки знань

1. Які основні парадигми ООП.
2. Рокрийте поняття успадкування.
3. Які стандартні бібліотеки Java ви знаєте.
4. Як додати до проекту існуючу бібліотеку.
5. Розкрийте сенс команди `super`.
6. Що таке затеняюча об'ява змінних.
7. Рокрийте поняття поліморфізм.
8. Для чого потрібен модифікатор `static`.
9. Що таке абстрактний клас.
10. Чи можливо у Java устаткування від декілька класів.

ЛАБОРАТОРНА РОБОТА 5

Тема: ПОЛІМОРФІЗМ

Мета: Одержання практичних навичок по створенню й використанню поліморфізму в Java.

5.1 Практична частина

Постановка завдання:

Введіть у головному класі прізвище спортсмена, число змагань та суму місць, які зайняв спортсмен у змаганнях. Розрахуйте число Q =число змагань/сума місць.

Створіть клас спадкоємець та додайте змінну P – чи займав спортсмен хоча б один раз перше місце. Розрахуйте $Q_p = 1,5 * Q$, якщо P – істина, інакше $Q_p = Q$.

Своримо головний клас Parent.

```
public class Parent { // створення класу «Parent»
    int kol,sum; //створюємо змінні кількості змагань
та суми місць,які зайняв спортсмен

    String name; // створюємо змінну прізвища
спортсмену
    Parent(String name, int kol, int sum){

        this.name = name; //ініціалізація змінної name
        this.kol = kol; //ініціалізація змінної kol
        this.sum = sum; //ініціалізація змінної sum
    }

    public double Qua() //створюємо метод (називаємо його
на власний розсуд)
    {
        double Q; //вказуємо тип даних змінної Q
```

```

        Q=kol/sum;           // розраховуємо за формулою
згідно завданню
        return Q;           //повертаємо дані змінної Q
    }

    void saySomething(){

        System.out.println("Имя-"+name+" Качество="+Qua());
    }
}

```

Створимо другий клас Child.

```

public class Child extends Parent {
    int P;           // створимо змінну P
    double Qr;       // створимо змінну Qr
    Child(String name, int kol, int sum,int P)
    {
        super(name, kol, sum); //наслідуємо змінні з
головного класу
        this.P=P;           //ініціалізуємо змінну P
    }
    @Override           // перезавантаження методу
    public double Qua(){

        if(P==1) {

            Qr= super.Qua()*1.5;
            return Qr;
        }
        else {
            Qr=super.Qua();

            return Qr;
        }
    }
}

```

```

    }
    void saySomething() {
        System.out.println("Имя-" + name + " Качество=" + Qua());
    }
}

```

Створимо клас Main.

```

public class Main {

    public static void main(String[] args) {
        Parent p1 = new Parent("Vasya1", 5, 5);
        Parent p2 = new Child("Vasya2", 5, 5, 1);
        Parent p3 = new Child("Vasya3", 5, 5, 0);
        p1.saySomething();
        p2.saySomething();
        p3.saySomething();
    }
}

```

5.2 Самостійна частина

Завдання 1. побудувати клас 1-го рівня з зазначеними в індивідуальному завданні (табл. 4.1) полями і методами:

- конструктор;
- функція, яка визначає «якість» об'єкта - Q по заданій формулі (таблиця 5.1, стовп 2);
- висновок інформації про об'єкт.

Побудувати клас 2-го рівня (клас-нащадок), який містить:

- додаткове поле P;
- функція, яка визначає «якість» об'єкта класу 2-го рівня - Qp, яка перекриває функцію якості класу 1-го рівня (Q), виконуючи обчислення за новою формулою (табл. 5.1, стовп 3).

Створити проект для демонстрації роботи: введення і виведення інформації про об'єкти класів 1-го і 2-го рівнів.

Таблиця 5.1. Варіанти індивідуальних завдань.

№ варіанту	Поля і функція «якості» (Q) класу 1-го рівня	Поле і функція «якості» Q _p класу 2-го рівня
1	<p>Комп'ютер:</p> <ul style="list-style-type: none"> - найменування процесора; - тактова частота процесора (МГц); - обсяг оперативної пам'яті (Мб). <p>$Q = (0,1 \cdot \text{частота}) + \text{пам'ять}$</p>	<p>P: обсяг вінчестера (Гб)</p> <p>$Q_p = Q + 0,5 \cdot P$</p>
2	<p>Оператор мобільного зв'язку:</p> <ul style="list-style-type: none"> - назва оператора; - вартість 1 хвилини розмови; - площа покриття. <p>$Q = 100 \cdot \text{площа покриття} / \text{вартість 1 хвилини розмови}$</p>	<p>P: наявність плати за кожне з'єднання</p> <p>$Q_p = 0,7 \cdot Q$, якщо P - істина, інакше $Q_p = 1,5 \cdot Q$</p>
3	<p>Товар на складі:</p> <ul style="list-style-type: none"> - найменування; - ціна; - кількість. <p>$Q = \text{ціна} / \text{кількість}$</p>	<p>P: рік випуску товару</p> <p>$Q_p = Q + 0,5 \cdot (T - P)$, де T - поточний рік</p>
4	<p>Кабель:</p> <ul style="list-style-type: none"> - тип; - кількість жил кабелю; - діаметр. <p>$Q = \text{діаметр} / \text{кількість жив}$</p>	<p>P: наявність обплетення</p> <p>Q_p: якщо P - істина, то $Q_p = 2 \cdot Q$; інакше $Q_p = 0,7 \cdot Q$</p>

Продовження таблиці 5.1

5	<p>Підручник з програмування:</p> <ul style="list-style-type: none"> - назва; - кількість сторінок; - ціна. <p>$Q = \text{ціна} / \text{кількість сторінок}$</p>	<p>P: рік видання</p> <p>$Q_p = Q - 0,2 \cdot (T-P)$, де T - поточний рік</p>
6	<p>Мобільний телефон:</p> <ul style="list-style-type: none"> - марка; - ціна; - обсяг пам'яті. <p>$Q = \text{обсяг пам'яті} / \text{ціна}$</p>	<p>P: кількість SIM карт</p> <p>$Q_p = Q \cdot P$</p>
7	<p>Супутникова антена (тарілка):</p> <ul style="list-style-type: none"> - діаметр; - матеріал; - ціна. <p>$Q = \text{діаметр} / \text{ціна}$</p>	<p>P: тип підвіски (азимутальна, полярна, тороїдальна)</p> <p>$Q_p = Q$, якщо тип підвіски азимутальний, $Q_p = 2 \cdot Q$, якщо тип підвіски полярний, $Q_p = 2,5 \cdot Q$, якщо тип підвіски тороїдальний</p>
8	<p>Іспит:</p> <ul style="list-style-type: none"> - дисципліна; - число студентів на іспиті; - тривалість іспиту (ч). <p>$Q = \text{число студентів} / \text{тривалість}$</p>	<p>P: відсоток двійок</p> <p>$Q_p = Q \cdot (100-P) / 100$</p>

Продовження таблиці 5.1

9	<p>Спортсмен:</p> <ul style="list-style-type: none"> - прізвище; - число змагань; - сума місць, зайнятих спортсменом в змаганнях. <p>$Q = (\text{число змагань}) / (\text{сума місць})$</p>	<p>P: займав чи хоча б раз перше місце</p> <p>$Q_p = 1,5 \cdot Q$, якщо P - істина, інакше - $Q_p = Q$</p>
10	<p>Програміст:</p> <ul style="list-style-type: none"> - прізвище; - число програм, написаних програмісти; - число мов програмування, якими він пише програми. <p>$Q = (\text{число програм}) * (\text{число мов})$</p>	<p>P: число програм, які працюють правильно</p> <p>$Q_p = Q \cdot P / (\text{число всіх програм})$</p>
11	<p>Вистава:</p> <ul style="list-style-type: none"> - назва; n1 - число глядачів на початку; n2 - число глядачів в кінці. <p>$Q = (n2 - n1) / n1$</p>	<p>P: рік написання п'єси</p> <p>$Q_p = Q \cdot (T - P + 1)$, де T - поточний рік</p>
12	<p>Алмаз:</p> <ul style="list-style-type: none"> - назва; - вага (в каратах); - якість ограновування в балах (число). <p>$Q = 0,4 \cdot \text{вага} + 0,6 \cdot \text{якість ограновування}$</p>	<p>P: колір (білий, блакитний, жовтий, і тп)</p> <p>Q_p: якщо колір блакитний, то $Q_p = Q + 1$; а якщо жовтий, то $Q_p = Q - 0,5$</p> <p>інакше $Q_p = Q$</p>

Продовження таблиці 5.1

13	<p>Комп'ютерна мережа:</p> <ul style="list-style-type: none"> - Назва організації; - число робочих станцій; - середня відстань між станціями (М). <p>$Q = \text{число станцій} \cdot \text{середня відстань}$</p>	<p>P: середня швидкість передачі даних в мережі (Мб / с)</p> <p>$Q_p = Q \cdot P$</p>
14	<p>Армія:</p> <ul style="list-style-type: none"> - вид військ; - чисельність (тис людей); - озброєність (бали - число). <p>$Q = 0,3 \cdot \text{чисельність} + 0,7 \cdot \text{озброєність}$</p>	<p>P: досвід (кількість місяців, протягом яких армія вела бойові дії)</p> <p>$Q_p = Q \cdot (P + 1)$</p>
15	<p>Автомобіль:</p> <ul style="list-style-type: none"> - Марка автомобіля; - потужність двигуна (кВт); - число місць. <p>$Q = 0,1 \cdot \text{потужність} \cdot \text{число місць}$</p>	<p>P: рік виготовлення</p> <p>$Q_p = Q - 1,5 \cdot (T - P)$, де T - поточний рік</p>
16	<p>Партія:</p> <ul style="list-style-type: none"> - назва; - чисельність (тис. членів); - відсоток голосів на останніх виборах. <p>$Q = 0,3 \cdot \text{чисельність} + 0,7 \cdot \text{відсоток гол}$</p> <p>P: чисельність партії в минулому році</p>	<p>Q_p: якщо чисельність в поточному році збільшилася, то $Q_p = 1,2 \cdot Q$; а якщо скоротилася, то $Q_p = 0,8 \cdot Q$</p>
17	<p>Вищий навчальний заклад:</p> <ul style="list-style-type: none"> - назва закладу; - кількість студентів, зарахованих на 1-й курс; 	<p>P: відсоток випускників, які працюють за фахом</p> <p>$Q_p = P * Q$</p>

Продовження таблиці 5.1

	<p>- кількість випускників. $Q = \text{кількість випускників} / \text{кількість зарахованих}$</p>	
18	<p>Солдат: - прізвище; - зріст (м); - вага, кг). $Q = \text{зріст} \cdot \text{вага}$</p>	<p>P: освіта (початкова, середня, вища) Qp: якщо освіта вища, то $Qp = 2 \cdot Q$; а якщо початкова, то $Qp = 0,5 \cdot Q$; інакше $Qp = Q$</p>
19	<p>Телевізор: - фірма; - діагональ екрану (дюйм); - звукова потужність (дБ). $Q = \text{діагональ} + (0,05 \cdot \text{потужність})$</p>	<p>P: країна-виробник Qp: якщо країна - Японія, то $Qp = 2 \cdot Q$; а якщо Сінгапур або Корея, то $Qp = 1,5 \cdot Q$; інакше $Qp = Q$</p>
20	<p>Мітинг: - назва події; n1 - число ораторів; n2 - число учасників. $Q = n1 / n2$</p>	<p>P: число груп ораторів, які висловлювали однакові думки $Qp = Q + P / n2$</p>
21	<p>Будинок: - номер будинку; - число квартир; - рік споруди. $Q = (\text{число квартир}) + 2 \cdot (\text{T} - \text{рік споруди})$, де T - поточний рік</p>	<p>P: район (центр, околиця, і т.д.) Qp: якщо район - центр, то $Qp = 2 \cdot Q$; інакше $Qp = 0,5 \cdot Q$</p>

Продовження таблиці 5.1.

22	<p>Керівник:</p> <ul style="list-style-type: none"> - прізвище; - самооцінка (в балах - ціле число); - оцінка іншими людьми (в балах). <p>$Q = (\text{оцінка іншими}) / \text{самооцінка}$</p>	<p>P: оцінка нащадками (в балах) $Q_p = 0,3 \cdot Q + 0,7 \cdot P$</p>
23	<p>Студент:</p> <ul style="list-style-type: none"> - прізвище; - середній бал; - курс. <p>$Q = 0,2 \cdot \text{Середній бал} \cdot \text{курс}$</p>	<p>P: вивчає дисципліни англійською</p> <p>$Q_p = 2 \cdot Q$, якщо P - істина, інакше $Q_p = 0,9 \cdot Q$</p>
24	<p>антена:</p> <ul style="list-style-type: none"> - назва; - потужність; - висота (м). <p>$Q = \text{потужність} + 0,5 \cdot \text{висота}$</p>	<p>P: коефіцієнт випромінювання</p> <p>$Q_p = Q - 0,1 \cdot P$</p>
25	<p>Літак:</p> <ul style="list-style-type: none"> - марка; - кількість двигунів; - висота польоту. <p>$Q = \text{к-ть двигунів} \cdot \text{висота польоту} / 1000$</p>	<p>P: країна-виробник</p> <p>Q_p: якщо країна - Росія, то</p> <p>$Q_p = Q + 1$;</p> <p>а якщо Франція, то</p> <p>$Q_p = Q + 0,5$</p>
26	<p>Студент:</p> <ul style="list-style-type: none"> - прізвище; - число іспитів; - число оцінок «п'ять». <p>$Q = \text{число оцінок «п'ять»} / \text{число іспитів}$</p>	<p>P: число оцінок «три»</p> <p>$Q_p = Q - 0,5P$</p>

Продовження таблиці 5.1.

27	<p>Фірма:</p> <ul style="list-style-type: none"> - назва; - дохід (тис \$); - рейтинг (в балах). <p>$Q = \text{дохід} \cdot \text{рейтинг}$</p>	<p>P: інвестиції в фірму (Тис \$)</p> <p>$Q_p = P_3 + Q$</p>
28	<p>Військовий корабель:</p> <ul style="list-style-type: none"> - назва; - довжина; - число гармат головного калібру. <p>$Q = (\text{число гармат}) / \text{довжина}$</p>	<p>P: крейсерська швидкість (В морських вузлах)</p> <p>$Q_p = 0.25 * Q + P$</p>
29	<p>Коробка сірників:</p> <ul style="list-style-type: none"> - фірма виробник; - число сірників в коробці; - час горіння одного сірника (с). <p>$Q = (\text{число сірників}) \cdot \text{час}$</p>	<p>P: середній % бракованих сірників в коробці</p> <p>$Q_p = (100 - P) Q / 100$</p>
30	<p>полководець:</p> <ul style="list-style-type: none"> - прізвище; - число битв; - число перемог. <p>$Q = (\text{число перемог})^2 / (\text{число битв})$</p>	<p>P: число перемог з меншими, ніж у супротивника, силами</p> <p>$Q_p = P_2 / \text{битви} + Q$</p>

Завдання 2. Написати програму відповідно до завдання. У всіх класах описати необхідні конструктори, за допомогою яких будуть створюватися об'єкти класів. Параметри створюваних об'єктів задавати константами або вводити з клавіатури (за бажанням студента) і передавати в конструктори об'єктів у вигляді параметрів. Висновок інформації повинен здійснюватися на консоль.

Таблиця 5.2. Варіанти індивідуальних завдань.

№ варіанту	Задача
1	<p>Створити клас Автомобіль з властивостями: Назва, Максимальна швидкість (в км / год). Визначити 2 віртуальних методу: метод «Вартість»-вартість автомобіля, яка розраховується за формулою. Максимальна швидкість * 100 і метод «Оновлення моделі», що збільшує максимальну швидкість на 10. Визначити також метод «Інформація», який повертає рядок, що містить інформацію про об'єкт: Назва, Максимальну швидкість і Вартість. Створити також клас спадкоємець Представницький автомобіль, в якому перевизначити методи: метод «Вартість» повертає число, рівне. Максимальна швидкість * 250, а метод «Оновлення моделі» збільшує швидкість на 5 км / год. У головній програмі (або після натискання на кнопку) створити об'єкт класу Автомобіль з максимальною швидкістю 140 км / год і класу Представницький автомобіль з максимальною швидкістю 160 км / год.</p> <p>Вивести на екран (або форму) інформацію про автомобілі. Оновити моделі автомобілів і знову вивести інформацію про них.</p>
2	<p>Створити клас Трикутник, заданий значеннями довжин трьох сторін (a, b, c), з методами «Периметр» і «Площа». Визначити також метод «Інформація», який повертає рядок, що містить інформацію про трикутнику: довжини сторін, периметр і площу.</p> <p>Створити також клас спадкоємець Чотирикутник, з додатковими параметрами - довжиною четвертого боку (d) і довжинами діагоналей (e, f) і перевизначити методи «Периметр» (сума всіх сторін) і «Площа». Площа обчислювати за такою формулою</p> $S = \sqrt{\frac{4e^2 f^2 - (b^2 + d^2 - a^2 - c^2)^2}{16}}.$

Продовження таблиці 5.2.

	<p>У головній програмі (або після натискання на кнопку) створити об'єкт класу Трикутник і об'єкт класу Чотирикутник і вивести інформацію про них. Для спрощення перевірки рекомендується в якості конкретного об'єкта класу чотирикутник взяти квадрат.</p>
3	<p>Створити клас Комп'ютер з властивостями: Частота процесора (в МГц), кількість ядер, об'єм пам'яті (в МБ), обсяг жорсткого диску (в ГБ). Визначити два віртуальних методу: «Вартість», що повертає приблизну розрахункову вартість комп'ютера, яка розраховується за формулою. Частота процесора * кількість ядер / 100 + кількість пам'яті / 80 + обсяг жорсткого диску / 20 і логічний метод «Придатність», який повертає істину (true), якщо частота процесора не менше 2000 МГц, кількість ядер не менше 2, обсяг пам'яті не менше 2048 МБ, і обсяг жорсткого диску не менше 320 Гб. Визначити також метод «Інформація», який повертає рядок, що містить інформацію про комп'ютер: частоту процесора, кількість ядер, об'єм пам'яті, обсяг жорсткого диску, вартість і придатність для наших потреб. Створити також клас спадкоємець Ноутбук, з додатковою властивістю. Тривалість автономної роботи (в хвилинах) і перевизначити методи: метод «Вартість» повертає число, рівне вартості звичайного комп'ютера + кількість хвилин автономної роботи / 10, а метод «Придатність» повертає істину, тоді коли і ноутбук придатний як звичайний комп'ютер, і тривалість автономної роботи не менш 60 хвилин. У головній програмі (або після натискання на кнопку) створити звичайний комп'ютер і ноутбук і вивести інформацію про них.</p>

Продовження таблиці 5.2.

4	<p>Створити клас Прямокутник, заданий значеннями довжин двох сторін (a і b), з віртуальними методами «Периметр» і «Площа», які повертають периметр і площу відповідно, а також віртуальний метод «Збільшити в два рази», що збільшує в два рази кожен зі сторін. Визначити також метод «Інформація», який повертає рядок, що містить інформацію по трикутнику: довжини сторін, периметр і площу. Створити також клас спадкоємець Прямокутник з округленими кутами, з додатковим параметром радіус заокруглення (r). Для нього перевизначити. Периметр за формулою $p = 8 \cdot r + 2 \cdot \pi \cdot r$, де p - периметр звичайного прямокутника з тими ж сторонами, а Площа за формулою $S = 4 \cdot r^2 + \pi \cdot r^2$, де S - площа звичайного прямокутника. Також перевизначити метод «Збільшити в два рази» так, щоб він також збільшував в два рази радіус заокруглення (як і раніше збільшуючи бока в два рази). У головній програмі (або після натискання на кнопку) створити звичайний прямокутник і прямокутник з округленими кутами і вивести інформацію про них. Після цього збільшити обидва прямокутника в два рази і видати оновлену інформацію.</p>
5	<p>Створити клас Фотоапарат з властивостями: Модель, Оптичне збільшення (Zoom, дійсне число від 1 до 35) і матеріал корпусу (метал або пластик). Визначити віртуальний метод: метод «Вартість» - повертає число - вартість фотоапарата (в \$), що розраховується за формулою $(Zoom + 2) \cdot 10$, якщо корпус пластиковий і $(Zoom + 2) \cdot 15$, якщо матеріал металевий. Визначити також метод «Інформація», який повертає рядок, що містить інформацію про об'єкт: Модель, Zoom і Вартість. Також визначити логічний метод «Дорогий», який буде повертати істину (true), якщо вартість фотоапарата більше 200 \$.</p>

Продовження таблиці 5.2.

	<p>Створити також клас спадкоємець Цифровий фотоапарат, в якому буде додатковий цілий параметр - кількість мегапікселів і перевизначити метод «Вартість», який буде повертати число, рівне вартості звичайного фотоапарата помножити на кількість мегапікселів, а також визначити новий метод «Оновлення моделі», який збільшує кількість мегапікселів на 2.</p> <p>У головній програмі (або після натискання на кнопку) створити об'єкт класу Фотоапарат з 4-ми кратним оптичним збільшенням (Zoom = 4) і пластиковим корпусом, а також Цифровий фотоапарат з металевим корпусом, 8-ю мегапікселями і 3-кратним оптичним збільшенням. Вивести на екран (або форму) інформацію по фотоапаратах і про те, чи є вони дорогими. Оновити моделі цифрового фотоапарата і знову вивести інформацію про нього.</p>
6	<p>Створити клас Студент з властивостями: ПІБ, факультет, курс, мінімальна оцінка за іспитами за останню сесію (за 5-ти бальною системою). Визначити віртуальні методи: «Перевести на наступний курс», що збільшує курс на 1, якщо мінімальна оцінка не менше 3, інакше не робить нічого, а також «Стипендія», який повертає стипендію (в грн): 0 грн, якщо мінімальна оцінка не вище 3, 200 грн, якщо мінімальна оцінка дорівнює 4 і 300 грн, якщо мінімальна оцінка дорівнює 5. Визначити також метод «Інформація», який повертає рядок, що містить інформацію про студента: ПІБ, факультет, курс, мінімальна оцінка за іспитами та нараховану стипендію. Створити також клас спадкоємець Студент-контрактник, в якому буде додатковий логічний параметр - сплачений контракт і перевизначені методи «Перевести на наступний курс», що збільшує курс</p>

Продовження таблиці 5.2.

	<p>на 1, якщо мінімальна оцінка не менше 3 і за контракт сплачено, а також «Стипендія» повертає завжди 0 грн. В головній програмі (або після натискання на кнопку) створити об'єкт класу Студент та 2 об'єкти класу Студент-контрактник (один з яких сплатив за контракт, а інший ні). Видати інформацію про студентів, потім застосувати до них метод «Перевести на наступний курс» і знову видати інформацію про них.</p>
7	<p>Створити клас Коло заданий своїм радіусом (r), з віртуальним методом «Площа», що повертає площу кола, а також віртуальний метод «Збільшити» з одним речовим параметром - у скільки разів збільшити, що збільшує радіус в задане число раз. Визначити також метод «Інформація», який повертає рядок, що містить інформацію по колу: радіус і площа.</p> <p>Створити також клас спадкоємець Кільце, з додатковим параметром- внутрішнім радіусом (r_{in}), при цьому успадкований від батька радіус буде позначати зовнішній радіус. Перевизначити метод «Площа», як різницю між площею зовнішнього кола мінус площа внутрішнього кола. Також довизначити метод «Збільшити», щоб він збільшував також і внутрішній радіус. У головній програмі (або після натискання на кнопку) створити звичайне коло і кільце і вивести інформацію про них. Після цього збільшити обидва об'єкта в півтора рази і видати оновлену інформацію.</p>
8	<p>Створити клас Табуретка з властивостями: Висота (h, в см), Якість виробу (низька, середня, висока). Визначити два віртуальних методу: «кількість деревини», яку вимагає табуретка, за формулою $4 \cdot h + 12$, якщо якість низька, і $5 \cdot h + 14$, якщо якість середня або висока, а також «вартість», рівна $d \cdot 2$, для низької якості, $d \cdot 3$, для середньої якості, $d \cdot 4$, для високої якості,</p>

Продовження таблиці 5.2.

	<p>де d - кількість деревини, яку вимагає даний об'єкт. Визначити також метод «Інформація», який повертає рядок, що містить інформацію про об'єкт: Висоту, якість матеріалу, кількість деревини і вартість.</p> <p>Створити також клас спадкоємець Стілець з додатковою властивістю: висота спинки (h_2, в см), і перевизначити метод «кількість деревини», за формулою $d + 2h_2 + 5$, де d - кількість деревини, які вимагає табуретка з такими ж параметрами (Метод «вартість» не перевизначати). У головній програмі (або після натискання на кнопку) створити екземпляри класів Табуретка і Стілець, і надрукувати інформацію в такому вигляді: «табуретка» + інформація по табуретці і «стілець» + інформація про стільці.</p>
9	<p>Створити клас Фільм з властивостями: Назва, Режисер, тривалість (в хвиликах), кількість акторів. Визначити віртуальний метод: «Вартість», що повертає приблизну розрахункову вартість фільму (в тис. \$), Що розраховується за формулою тривалість * 20 + кількість акторів * 30, але якщо режисер = «Стівен Спілберг» або «Джеймс Кемерон », то вартість в два рази вище (у порівнянні з вищевказаної формулою). Визначити також метод «Інформація», який повертає рядок, що містить інформацію про фільм: Назва, режисера, тривалість, кількість акторів і вартість. Створити також клас спадкоємець Мультфільм, в якому перевизначити метод «Вартість» за формулою тривалість * 25 + кількість акторів * 10 (незалежно від режисера). У головній програмі (або після натискання на кнопку) створити 2 фільми з режисерами: «Стівен Спілберг» і «Єжи Гофман», а також мультфільм і вивести інформацію про них.</p>

Продовження таблиці 5.2.

10	Створити клас Літак з властивостями: Марка, Модель, Максимальна швидкість (в км / год), Максимальна висота (в метрах). Визначити віртуальний метод «Вартість» - вартість літака, яка розраховується за формулою $\text{Максимальна швидкість} * 1000 + \text{Максимальна висота} * 100$. Визначити також метод «Інформація», який повертає рядок, що містить інформацію про об'єкт: Марка, Модель, Максимальну швидкість, Максимальну висоту і Вартість . Створити також клас спадкоємець Бомбардувальник, в якому перевизначити метод «Вартість», який поверне подвоєну вартість щодо формули для класу Літак. Також створити клас Винищувач - спадкоємець класу Літак, для якого перевизначити метод «Вартість» як потрібну вартість, щодо формули вартості для Літака. В головній програмі (або після натискання на кнопку) створити об'єкт класу Літак, класу Бомбардувальник, класу Винищувач. Вивести на екран (або форму) інформацію про літаки.
----	--

Завдання 3. Описати кілька класів: абстрактний клас-батько з полями і методами, характерними для всіх об'єктів завдання, при цьому деякий метод або методи може бути чисто віртуальним і кілька класів-спадкоємців, що визначають додаткові поля (при необхідності) і перевизначити деякі віртуальний метод (методи) .

У головній програмі повинен бути оголошений масив (статичний або динамічний) об'єктів (батьківського класу), в який поміщаються класів-нащадків (В С ++ необхідно використання масиву покажчиків на об'єкти).

Таблиця 5.3. Варіанти індивідуальних завдань.

№ варіанту	Задача
1	<p>Програмно промоделювати розмови людей. Всього є 5 людей. Кожна людина має ім'я (рядок) і вік (число). Встановіть вік кожної людини - випадкове число від 20 до 40, а імена встановіть випадковим чином зі списку «Олександр», «Андрій», «Анастасія», «Ірина», «Наталя», «Павло», «Роман», «Світлана», «Сергій», «Тетяна».</p> <p>Будь-яка людина здатна виконувати дві дії:</p> <ul style="list-style-type: none"> - вітатися з іншою людиною; - розповідати про себе. <p>Люди діляться на 3 типи (різні класи):</p> <p>1 (Формалісти) Вітаються з усіма так: «Здрастуй, <ім'я>», де <Ім'я> - ім'я людини, з яким він вітається.</p> <p>2 (Неформали) З усіма вітаються: «Привіт, <ім'я>!»</p> <p>3 (Реалісти) Якщо вік співрозмовника менше або дорівнює або більше не більш ніж на 5 років, каже «Привіт, <ім'я>!», інакше «Здрастуй, <Ім'я> ».</p> <p>У програмній реалізації вітання повинне бути реалізоване як поліморфний метод, який приймає параметр - людина і повертає рядок. Розповідь про людину є рядком виду «Мене звать Вася, мій вік 21 років, я неформал »(замість Вася ім'я людини, замість 21 його вік, як бачите у людей з граматиною не все гаразд і вони кажуть «років» після будь-якого числа, якщо хочете, можете реалізувати правильну граматику, замість неформал може бути формаліст, або реаліст). Програма повинна показати інформацію про всіх людей. Потім всі люди повинні привітатися один з одним в такому порядку: перший вітається з другим, потім другий з першим, потім перший з третім, третій з першим, і так</p>

Продовження таблиці 5.3.

	<p>далі перший з усіма і всі з першим, потім другий з третім, третій з другим і т.д. Потрібно видати в окремих рядках ім'я людини, який вітається, двокрапка, вітання:</p> <p>Петя: Привіт, Вася!</p> <p>Вася: Здрастуй, Петя</p> <p>Петя: Привіт, Женя!</p> <p>...</p>
2	<p>Програмно промоделювати стрілянину по мішені групою осіб. Кожна людина має своє ім'я, вік (в роках) і стаж навчання стрільбі (в роках).</p> <p>Люди діляться на новачків, досвідчених і ветеранів (нащадки класу людина). Для кожної людини визначте поліморфний метод «Стріляти» без параметрів, які повертають логічне значення (потрапив - true, не потрапив - false). Попадання визначається випадковим чином, причому для новачка ймовірність потрапити дорівнює $0,01 * \text{стаж навчання}$; для досвідченого = $0,05 * \text{стаж навчання стрільбі}$; для ветерана = $0,9 - 0,01 * \text{вік}$.</p> <p>Люди стріляють по черзі, починаючи з першого, поки хтось не потрапить в мішень. Стрільба припиняється після того, як хтось потрапив або всі вистрілили по одному разу. Після кожного пострілу потрібно виводити на екран всю інформацію по стріляючому і результат стрільби. У головній програмі створіть масив з 7 людей в такому порядку: новачок, досвідчений, ветеран, досвідчений, новачок і зробіть стрілянину з виведенням її результатів.</p>
3	<p>Програмно промоделювати задачу заліку студентами. Кожен студент характеризується ПБ і кількістю відвіданих занять. Також відомо кількість загальних занять = 20 Студенти діляться на</p>

Продовження таблиці 5.3.

	<p>звичайних, кмітливих і геніїв (нащадки класу студент). Для кожного людини визначте поліморфний метод «Здати залік» без параметрів, повертають логічне значення (здав - true, не здав - false). Звичайні студенти точно здають залік, якщо відвідали всі заняття, якщо були більш ніж на половині занять, то здають з ймовірністю 0,5; інакше не здають. Кмітливі студенти теж точно здають залік, якщо відвідали всі заняття, якщо були більш ніж на половині занять, то здають з ймовірністю 0,7; інакше не здають залік. Генії точно здають залік, якщо були хоча б на одному занятті, інакше не здають. Створіть масив з 10 студентів (5 звичайних, 4 кмітливих і 1 генія), задайте їх ПІБ та кількість відвіданих занять константами, зазначеними в конструкторах об'єктів і промоделюйте задачу заліку з висновком докладних результатів (всіх відомостей про студентів, а також результат задачі).</p>
4	<p>Промоделювати виліток листів сталі. Листи стали характеризуються товщиною (в мм), і щільністю сталі (в кг / м³). Листи діляться на квадратні (додатково задаються одним числом - шириною і довжиною одночасно, в мм), прямокутні (задаються шириною і довжиною, в мм) і трикутні (у вигляді прямокутного трикутника, задаються двома катетами, в мм). Для кожного типу сталі визначте віртуальний метод «Площа» - повертає площу листа. Також визначте метод «Вага», який обчислює вага листа, множачи площа листа на його товщину і щільність стали. Також задайте метод «Інформація», який видаватиме інформацію про листи.</p> <p>У головній програмі створіть масив з 15 листів сталі,</p>

Продовження таблиці 5.3.

	створивши 5 квадратних, 7 прямокутних і 3 трикутних листа випадкових розмірів. Виведіть інформацію по листах і порахуйте сумарну площу і сумарну вагу всіх листів.
--	--

5.3 Питання для перевірки знань

1. Які основні парадигми ООП.
2. Рокрийте поняття поліморфізму.
3. Які стандартні бібліотеки Java ви знаєте.
4. Як додати до проекту існуючу бібліотеку.
5. Розкрийте сенс команди `super`.
6. Що таке затеняюча об'ява змінних.
7. Рокрийте поняття поліморфізм.
8. Для чого потрібен модифікатор `static`.
9. Що таке абстрактний клас?
10. Чи можливо у Java устаткування від декілька класів.

ЛАБОРАТОРНА РОБОТА 6

Тема: ІНТЕРФЕЙСИ

Мета: Одержання практичних навичок по створенню й використанню інтерфейсів в Java.

6.1 Практична частина

Інтерфейс - це конструктор мови програмування Java, в рамках якої можуть описуватися тільки абстрактні публічні (abstract public) методи і статичні константні властивості (final static). Тобто так само, як і на основі абстрактних класів, на основі інтерфейсів можна породжувати об'єкти.

Один інтерфейс може бути спадкоємцем іншого інтерфейсу.

Класи можуть реалізовувати інтерфейси (тобто отримувати від інтерфейсу список методів і описувати реалізації кожного з них), при тому, що особливо важливо, один клас може реалізовувати відразу кілька інтерфейсів.

Перед описом інтерфейсу вказується ключове слово interface. Коли клас реалізує інтерфейс, то після його імені вказується ключове слово implements і далі через кому перераховуються імена тих інтерфейсів, методи яких будуть повністю описані в класі.

Приклад:

```
interface Instruments {
    final static String key = "До мажор";
    public void play();
}
class Drum implements Instruments {
    public void play() {
        System.out.println("бум бац бац бум бац бац");
    }
}
class Guitar implements Instruments {
```

```

public void play() {
    System.out.println("до ми соль до ре до");
}
}

```

Оскільки всі властивості інтерфейсу повинні бути константними і статичними, а всі методи загальнодоступними, то відповідні модифікатори перед властивостями і методами дозволяється не вказувати. Тобто інтерфейс можна було описати так:

```

interface Instruments {
    static public String key = "До мажор";
    void play ();
}

```

Але коли метод `play ()` буде описуватися в класі інтерфейсу, який реалізується, перед ним все одно необхідно буде явно вказати модифікатор `public`.

Множинне спадкування інтерфейсів

Java не підтримує множинне успадкування класів. Це пояснюється тим, що таке спадкування породжує певні проблеми.

Найчастіше вказуються неоднозначності, що виникають при так званому «ромбоподібному» спадкуванні, коли один клас «А» є спадкоємцем двох інших класів «В» і «С», які в свою чергу обидва є спадкоємцями класу «D». Проблема допустимості множинного спадкоємства криється в наступному. Припустимо, що в батьку А визначався якийсь метод `m1 ()`. І цей же метод ми викликаємо для об'єкта класу D. А що якщо `m1 ()` був перевизначений в класах В і С. Запитання: яка реалізація з трьох буде працювати при виклику методу `m1 ()` для об'єкта класу D? Від цього можна було б позбутися зажадавши в описаному випадку при виклику уточнювати при виклику, який з методів потрібно (так і зроблено в деяких мовах), але в Java від множинного спадкоємства класів вирішили відмовитися.

Замість множинного успадкування класів в Java введено множинне спадкування інтерфейсів, яке частково вирішує проблеми (але, як буде показано в прикладі далі, на жаль, не всі).

Розглянемо приклад, де реалізовано два інтерфейси з методами доступними для вантажного та для легкового транспорту. Клас Pickup (пікап) повинен володіти як можливістю перевезення вантажів, так і пасажирів, тому він реалізує відразу обидва інтерфейсу:

```
interface PassangersAuto {
    void transportPassangers();
}
interface CargoAuto {
    void transportCargo();
}
class Truck implements CargoAuto {
    final static int a = 1;
    public void transportCargo() {
        System.out.println("Везу груз");
    }
}
class Sedan implements PassangersAuto {
    public void transportPassangers() {
        System.out.println("Везу пасажиров");
    }
}
class Pickup implements CargoAuto, PassangersAuto {
    public void transportCargo() {
        System.out.println("Везу груз");
    }
    public void transportPassangers() {
        System.out.println("Везу пасажиров");
    }
}
```

Найчастіше, відкриту частину класу (тобто загальнодоступні методи) зумовлює як раз в інтерфейсі. Тоді глянувши на один лише інтерфейс можна зрозуміти, які ж методи повинні використовуватися для взаємодії з об'єктами даного класу. Тобто інтерфейси цілком відповідають принципам інкапсуляції. Як, втім, і принципу поліморфізму. Адже в декількох класах

метод деякого інтерфейсу може бути реалізований по-різному, хоча і з одним і тим же ім'ям.

Але інтерфейси, як було зазначено вище, не є досконалим інструментом позбавленим певних недоліків. Розглянемо приклад, коли у нас є два інтерфейси, в кожному з яких є властивості з однаковими іменами (але, можливо, різними значеннями) і методи з однаковими іменами.

Успадкувавши клас від пари цих інтерфейсів ми не зможемо звертатися до властивості його об'єктів безпосередньо, без вказівки того, який з двох інтерфейсів ми мали на увазі. Це обмеження існує тому, що в інтерфейсах властивостям може даватися різне початкове значення і, відповідно, програма не зможе визначити яке ж значення вибрати.

Також до властивості можна звернутися як до статичної властивості одного з інтерфейсів.

Проблема зникне, якщо перед зверненням до властивості ми наведемо об'єкт до одного з батьківських інтерфейсів (нагадаємо, що будь-який об'єкт можна явно привести до класу або інтерфейсу його батька прямого або транзитивного).

На жаль, створити окремі реалізації для двох однойменних методів з різних інтерфейсів в класі спадкоємця не вийде (щоб потім ними можна було користуватися через той же приведення об'єктів до потрібного інтерфейсу). Якщо клас реалізує кілька інтерфейсів, в яких є однойменні методи, то в ньому може задаватися лише одна загальна для всіх реалізація цих методів (і це вже обмежує поліморфізм при множині спадкування через інтерфейси в Java).

Отже, код прикладу:

```
interface Interfacel {
    int someField = 100;
    String someMethod ();
}
interface Interface2 {
    int someField = 200;
    String someMethod ();
```

```

}
class SomeClass implements Interface1, Interface2 {
    public String someMethod () {
        return "It Works";
    }
}
public class Main {
    public static void main (String [] args) {
        SomeClass a = new SomeClass ();
        System.out.println (a.someMethod ()); // It works
        System.out.println (a.someField); // помилка
        System.out.println (((Interface1) a) .someField); // 100
        System.out.println (Interface1.someField); // 100
    }
}

```

6.2 Самостійна частина

Створити на Java ієрархію класів:

1.Graph

1.1Point

1.1.1Circle

1.1.2Rectangle

1.1.2.1 Ellipse

1.2Background

Клас Graph повинен містити абстрактний метод draw (). Інші класи повинні його реалізовувати, відображаючи атрибути об'єктів у вигляді рядка, наприклад:

точка: $x = 10, y = 20$

прямокутник: $x = 2, y = 5, w = 3, h = 4$

Клас Point і всі дочірні класи повинні містити метод moveTo (x, y), що задає координати об'єкта, а також метод move (dx, dy), що змінює координати на задану величину.

Описати в дочірніх класах всі необхідні атрибути, конструктори і методи (об'єкти повинні мати координати і кольори, у класу Background є кольори і назва текстури).

Інтерфейс

В ході роботи необхідно реалізувати наступну ієрархію класів:

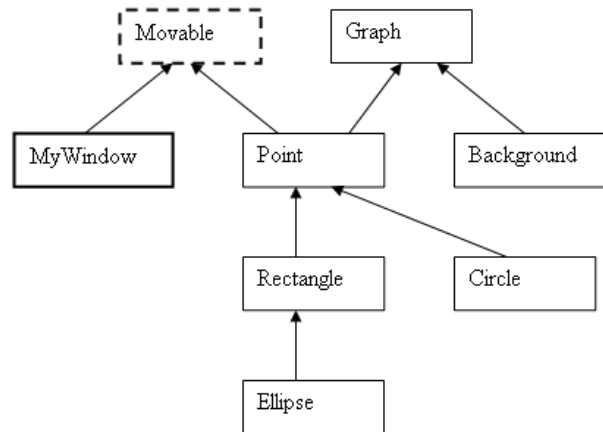


Рисунок 5.1 – Ієрархія класів

Створіть інтерфейс Movable, що оголошує наступні методи:

- int getX ();
- int getY ();
- void moveTo (int x, int y);

Підключіть створений інтерфейс до класу Point. Якщо описані методи не були реалізовані в попередніх лабораторних роботах - опишіть їх.

Створіть новий клас MyWindow без вказівки батьківського класу (за замовчуванням він буде успадкований від класу Object).

Підключіть інтерфейс Movable до класу MyWindow і реалізуйте описані методи. В даному класі метод moveTo () крім зміни координат повинен виводити повідомлення про їх зміну.

6.3 Питання для перевірки знань

1. Яка мова відноситься до мов опису документів?
2. Які мови належать до аплікаційних мов програмування?

3. Що таке функціональні інтерфейси?
4. Для чого потрібен функціональний інтерфейс `biconsumer <T,U>`?
5. Для чого потрібен функціональний інтерфейс `bifunction <T,U,R>`?
6. Для чого потрібен функціональний інтерфейс `binaryoperator <T>`?
7. Для чого потрібен функціональний інтерфейс `bipredicate <T,U>`?
8. Для чого потрібен функціональний інтерфейс `booleansupplier`?
9. Для чого потрібен функціональний інтерфейс `consumer <T>`?
10. Для чого потрібен функціональний інтерфейс `doublebinaryoperator`?

ЛАБОРАТОРНА РОБОТА 7

Тема: МАСИВИ

Мета: Отримання практичних навичок по створенню та використанню масивів.

7.1 Практична частина

Оголошення масиву в Java

При створенні масиву в Java, насамперед, його потрібно оголосити. Це можна зробити наступним чином:

```
int [] myFirstArray;
```

Можна також оголосити масив так:

```
int mySecondArray [];
```

Однак, це не є правильним в Java, оскільки дужки позначають те, що ми маємо справу з масивом і логічніше, коли вони знаходяться поруч з позначенням типу. Виходячи з цього прикладу, ми оголосили 2 масива з іменами `myFirstArray` і `mySecondArray`. Обидва масива будуть містити елементи типу `int`.

Подібним чином можна оголосити масив будь-якого типу:

```
byte [] anArrayOfBytes;  
short [] anArrayOfShorts;  
long [] anArrayOfLongs;  
float [] anArrayOfFloats;  
double [] anArrayOfDoubles;  
boolean [] anArrayOfBooleans;  
char [] anArrayOfChars;  
String [] anArrayOfStrings;  
...
```

Тип масиву задається наступним чином `type []`, де `type` це тип даних містяться в ньому елементів. Дужки є спеціальним позначенням того, що змінні містяться в масиві. Ім'я масиву може бути будь-яким, проте, воно

повинно відповідати правилам іменування змінних. Масиви можна створювати не тільки з змінних базових типів, але і з довільних об'єктів.

При оголошенні масиву в мові Java не вказується його розмір і не резервується пам'ять для нього. Відбувається лише створення посилання на масив.

Резервація пам'яті для масиву і його ініціалізація

Далі, для оголошеного `myFirstArray`, зарезервуємо пам'ять за допомогою ключового слова `new`.

```
int [] myFirstArray;  
myFirstArray = new int [15];
```

У нашому прикладі ми створили масив з 15 елементів типу `int` і привласнили його раніше оголошеній змінній `myFirstArray`.

Оголошувати ім'я масиву і резервувати для нього пам'ять також можна на одному рядку.

```
int [] myArray = new int [10];
```

При створенні масиву за допомогою ключового слова `new`, всі елементи масиву автоматично ініціалізовані нульовими значеннями. Для того, щоб привласнити елементам масиву свої початкові значення, необхідно провести його ініціалізацію. Ініціалізацію можна проводити як поелементно:

```
myFirstArray [0] = 10; // ініціалізація першого елемента  
myFirstArray [1] = 20; // ініціалізація другого елемента  
myFirstArray [2] = 30; // і т.д.
```

так і в циклі, за допомогою індексу проходячи всі елементи масиву і привласнюючи їм значення.

```
for (int i = 0; i <15; i ++ ) {  
    myFirstArray [i] = 10;  
}
```

Як видно з попередніх прикладів, для того, щоб звернутися до елемента масиву, потрібно вказати його ім'я, після чого в квадратних дужках - індекс елемента. Елемент масиву з конкретним індексом поводитьься так само, як і змінна.

Розглянемо створення і ініціалізацію масиву на наступному прикладі. У ньому ми створюємо масив, що містить цифри 0-9 і виводимо значення на консоль.

```
// створення і ініціалізація масиву
int [] numberArray = new int [10];
for (int i = 0; i <10; i ++) {
    numberArray [i] = i;
}
// вивід значень на консоль
for (int i = 0; i <10; i ++) {
    System.out.println ((i + 1) + "-й елемент масиву =" +
numberArray [i]);
}
```

Спрощена форма запису

Для створення і ініціалізації масиву можна також використовувати спрощений запис. Вона не містить слово `new`, а в дужках перераховуються початкові значення масиву.

```
int [] myColor = {255, 255, 0};
```

Тут довжина масиву визначається числом значень, розташованих між дужками і розділених комами. Такий запис більше підходить для створення невеликих масивів.

Визначення розміру масиву

Розмір масиву не завжди очевидний, тому для того, щоб його дізнатися слід використовувати властивість `length`, яке повертає довжину масиву.

```
myColor.length;
```

Даний код допоможе нам дізнатися, що довжина масиву `myColor` дорівнює 3.

Приклад: Визнач 4 числа, необхідно знайти мінімальне

```
int [] numbers = {-9, 6, 0, -59};
int min = numbers [0];
for (int i = 0; i <numbers.length; i ++) {
    if (min> numbers [i])
        min = numbers [i];
}
```

```
System.out.println (min);
```

7.2 Самостійна частина

Завдання 1. Створити програму рішення задачі за індивідуальним варіантом (таблиця 7.1).

Таблиця 7.1. Варіанти індивідуальних завдань.

№ варіанту	Задача
1	Ввести масив, що складається з 14 елементів цілого типу. Знайти кількість елементів парних за значенням.
2	Ввести масив, що складається з 12 елементів цілого типу. Отримати новий масив, замінивши значення п'ятого елемента середньоарифметичним вихідним масивом.
3	Заданий цілочисельний масив, що складається з 11 елементів. Знайти кількість елементів, абсолютне значення яких більше середнього арифметичного.
4	Ввести цілочисельний масив, що складається з 10 елементів. Поміняти місцями максимальний і перший елементи.
5	Ввести цілочисельний масив, що складається з 9 елементів. Поміняти місцями максимальний і мінімальний елементи масиву.
6	Ввести масив, що складається з 20 елементів цілого типу. Визначити яких елементів більше парних або непарних за значенням.
7	Задан масив, що складається з 15 елементів дійсного типу. Визначити кількість елементів, значення яких більше першого елемента.
8	Задан масив, що складається з 16 елементів дійсного типу. Визначити індекси (місце розташування) максимального і мінімального елементів.

Продовження таблиці 7.1

9	Дан масив, що складається з 15 елементів цілого типу. Отримати новий масив, як різницю між елементами вихідного масиву і його середнього арифметичного.
10	Ввести цілочисельний масив, що складається з 17 елементів (додатних і від'ємних). Знайти суму елементів, абсолютне значення яких більше середньоарифметичного модулів від'ємних елементів.
11	Ввести цілочисельний масив, що складається з 17 елементів (додатних і від'ємних). Знайти суму елементів, абсолютне значення яких більше середньоарифметичного модулів від'ємних елементів.
12	Ввести масив, що складається з 12 елементів дійсного типу. Розташувати елементи в порядку убутання. Обчислити суму максимального і мінімального елементів масиву.
13	Ввести цілочисельний масив, що складається з 15 елементів. Визначити суму і різницю максимального і мінімального елементів.
14	Ввести цілочисельний масив, що складається з 17 елементів. Замінити елементи кратні трьом на суму непарних за значенням елементів.
15	Ввести масив, що складається з 14 елементів дійсного типу. Розташувати елементи с 1 по 7 по зростанню, а з 8 по 14 - в порядку убутання.
16	Ввести масив, що складається з 12 елементів дійсного типу. Визначити кількість чисел, що стоять між максимальним і мінімальним елементами.

Продовження таблиці 7.1

17	Ввести масив, що складається з 15 елементів цілого типу. Визначити кількість додатних і від'ємних, добуток і кількість нульових елементів.
18	Ввести масив, що складається з 12 елементів дійсного типу. Визначити межі інтервалу, в якому знаходяться значення елементів масиву.
19	Дан масив - 19 елементів цілого типу. Знайти суму елементів, розташованих до першого від'ємного елементу. Якщо від'ємних елементів немає, то видати відповідне повідомлення.
20	Ввести масив, що складається з 16 елементів цілого типу. замінити всі елементи, кратні трьом, нулями. Визначити кількість замін.
21	В заданому масиві М (12) дійсних чисел збільшити мінімальний елемент в три рази і поміняти місцями з останнім.
22	Дан масив М (15) дійсних чисел. Розташувати елементи в зворотному порядку.
23	Ввести масив, що складається з 14 елементів цілого типу. Визначити суму елементів парних за індексом і добуток елементів непарних за значенням.
24	Ввести масив, що складається з 12 елементів дійсного типу. Визначити кількість і суму чисел, значення яких менше значення останнього елемента.
25	Дан масив, що складається з 15 елементів цілого типу (додатних і від'ємних). Отримати новий масив, елементи якого визначаються як різниця між елементами вихідного масиву і сумою додатних елементів заданого масиву.

Продовження таблиці 7.1

26	Дан масив, що складається з 15 елементів дійсного типу. Визначити різницю між добутком всіх додатних елементів і добутком модулів всіх і від'ємних.
27	У масиві цілих чисел з кількістю елементів 19 визначити максимальне число і замінити їм все парні за значенням елементи.
28	Ввести цілочисельний масив, що складається з 17 елементів. знайти суму і кількість елементів, абсолютне значення яких більше середньоарифметичного додатних елементів
29	Дан масив, що складається з 18 елементів дійсного типу. Визначити частку від ділення добутки всіх додатних елементів і суми модулів всіх від'ємних.
30	Дан масив дійсних чисел M (12). Визначити суму мінімального елемента і його індексу.

Завдання 2. Створити програму рішення задачі за індивідуальним варіантом (таблиця 7.2).

Таблиця 7.2. Варіанти індивідуальних завдань.

№ варіанту	Задача
1	Ввести два цілочисельних масива - по 10 елементів в кожному. Сформувати новий масив, на парних місцях якого будуть елементи з непарними індексами з першого масиву, а на непарних - з парними індексами з другого.
2	Ввести масив, що складається з 8 елементів (вісім двозначних чисел) цілого типу. Отримати новий масив, що складається з цифр, що знаходяться в молодших розрядах елементів вихідного масиву

Продовження таблиці 7.2

3	Ввести цілочисельний масив, що складається з 17-ти елементів (двозначні цілі числа). Обчислити суму чисел цього масиву.
4	Ввести два масиви дійсних чисел, що складаються з 9 і 7 елементів. Сформувати третій масив з упорядкованих за зростанням значень обох масивів.
5	Ввести два масиви X і Y, що складаються з 10-ти елементів цілого типу. Сформувати масив S, що складається з однакових елементів вихідних масивів.
6	Розрахувати значення 12-ти елементів масиву Y за формулою $y_i = i^2 - 2i + 19.3 \cos i$. Вивести на екран цей масив і новий, розмістивши в ньому спочатку елементи, значення яких менше середнього арифметичного, а потім інші, не змінюючи їх послідовності.
7	Дан масив дійсних чисел Z (16). Визначити різницю між сумою елементів с парними індексами і сумою елементів, індекси яких кратні трьом.
8	У заданому цілочисленому масиві R (9) визначити індекс найбільшої з непарних за значенням додатних елементів.
9	Ввести з клавіатури масив X, що складається з 15 елементів цілого типу. Розрахувати елементи масиву Y за формулою $y_i = \cos x_i^2 + 2.7 \lg^2 i^2$. Сформувати третій масив з упорядкованих по спадаючій значень обох масивів.
10	Ввести з клавіатури масив X, що складається з 17 елементів цілого типу. Розрахувати елементи масиву Y за формулою: $y_i = \begin{cases} (x_i)^3 - 7.5, & \text{если } \cos(x_i) > 0 \\ x_i^2 - 5e^{\sin(x_i)}, & \text{если } \cos(x_i) \leq 0 \end{cases}$ Впорядкувати масив Y по зростанню, масив X по спадаючій і сформувати новий масив R, елементами якого є парні за індексом елементи масиву X і Y.

Продовження таблиці 7.2

11	Ввести масив, що складається з 9 елементів (дев'ять двозначних чисел) цілого типу. Отримати новий масив, що складається з сум цифр елементів вихідного масиву.
12	Ввести масив, що складається з 12 елементів дійсного типу. Розташувати елементи в порядку убутання. Визначити кількість що відбулися при цьому перестановок.
13	Ввести з клавіатури цілочисельний масив, що складається з 11 елементів. Обчислити суму непарних за значенням від'ємних елементів і замінити елементи кратні трьом на цю суму.
14	Ввести масив, що складається з 14 елементів дійсного типу. Поміняти місцями першу половину з другої. Визначити кількість вироблених при цьому перестановок.
15	Дан масив дійсних чисел. Визначити елемент масиву (значення і індекс), який найбільш віддалений від заданого дійсного числа S.
16	Ввести цілочисельний масив, що складається з 10 елементів. Визначити суму і кількість елементів, розташованих до першого від'ємного числа.
17	Визначити кількість локальних мінімумів в заданому числовому масиві. (Локальний мінімум в числовому масиві - це послідовність трьох чисел, що стоять поруч, в якій середнє число менше стоїть ліворуч і праворуч від нього).
18	Визначити кількість локальних максимумів в заданому числовому масиві. (Локальний максимум в числовому масиві - це послідовність трьох чисел, що стоять поруч, в якій середнє число більше стоїть ліворуч і праворуч від нього).

Продовження таблиці 7.2

19	У заданому цілочисельному масиві Z (15) додатних, від'ємних і нульових чисел визначити суму і вивести послідовність значень елементів, які розташовані між першим від'ємними і нульовим елементами.
20	У заданому числовому масиві визначити і вивести індекси послідовностей чисел, які монотонно зменшуються (кожне наступне число менше попереднього).
21	У заданому цілочисельному масиві видалити елементи, які зустрічаються більше двох разів.
22	Ввести масив, що складається з 10-ти елементів цілого типу. Сформуванати новий, розташувавши спочатку всі від'ємні елементи і нулі, після чого - додатні, зберігаючи порядок їх проходження.
23	Ввести масив X , що складається з 10-ти елементів цілого типу. Обчислити елементи масиву Y за формулою: $y_i = x_i^2 + 0.3$ та знайти P . $P = \frac{x_1 y_1 \cdot x_3 y_3 \cdot \dots \cdot x_9 y_9}{x_0 y_0 \cdot x_2 y_2 \cdot \dots \cdot x_8 y_8}$. Визначити залишок від ділення.
24	Ввести масив, що складається з 10 елементів (десять двозначних чисел) цілого типу. Отримати новий масив, що складається з різниць чисел елементів вихідного масиву.
25	Ввести масив, що складається з 15 елементів цілого типу. Впорядкувати масив так, щоб всі від'ємні числа були розташовані спочатку по зростанню, а всі додатні - в кінці спаданням.
26	Дано два масиви дійсних чисел по 12 елементів в кожному. Замінити нулями ті елементи першого масиву, які є в другому.
27	Заданий цілочисельний масив. Визначити кількість ділянок масиву, на якому елементи монотонно зростають (кожне наступне число більше попереднього).

Продовження таблиці 7.2

28	Заданий цілочисельний масив. Визначити залишок від ділення суми елементів з парними індексами на суму елементів з непарними індексами.
29	Заданий цілочисельний масив. Визначити процентний вміст елементів, що перевищують середнє арифметичне всіх елементів масиву.
30	Ввести два масиви дійсних чисел. визначити максимальні елементи в кожному масиві і поміняти їх місцями.

Завдання 3. Створити програму рішення задачі за індивідуальним варіантом (таблиця 7.3).

Таблиця 7.3. Варіанти індивідуальних завдань.

№ варіанту	Задача
1	Дано ціле число в двійковій системі числення, тобто послідовність цифр 0 і 1. Скласти програму перекладу цього числа в десяткову систему числення.
2	Дано ціле число в двійковій системі числення, тобто послідовність цифр 0 і 1. Скласти програму перекладу цього числа в вісімкову систему числення.
3	Дано ціле число в двійковій системі числення, тобто послідовність цифр 0 і 1. Скласти програму перекладу цього числа в шістнадцяткову систему числення.
4	Дано дробове число в двійковій системі числення, тобто послідовність цифр 0 і 1, між якими ставиться крапка. Скласти програму перекладу цього числа в десяткову систему числення.

Продовження таблиці 7.3

5	Дано дробове число в двійковій системі числення, тобто послідовність цифр 0 і 1, між якими ставиться крапка. Скласти програму перекладу цього числа в вісімкову систему числення.
6	Дано дробове число в двійковій системі числення, тобто послідовність цифр 0 і 1, між якими ставиться крапка. Скласти програму перекладу цього числа в шістнадцяткову систему числення.
7	Ввести масив, що складається з 15-ти елементів (двозначні цілі числа). Змінити розрядність цифр, що утворюють елементи вихідного масиву і, таким чином, сформувати новий масив. Наприклад, вихідний масив: 25 71 84 ..., новий масив: 52 17 48
8	Ввести масив, що складається з 9 елементів (дев'ять двозначних чисел в вісімковій системі числення). Сформувати новий масив шляхом перекладу значень елементів вихідного масиву в десяткову систему числення.
9	Ввести цілочисельний масив, що складається з 7 елементів (сім двозначних чисел). Отримати новий масив, що складається з цифр елементів вихідного масиву, що стоять в старших розрядах
10	Ввести два масиви дійсних чисел, що складаються з 7 і 9 елементів. Сформувати третій масив з упорядкованих по спадаючій значень обох масивів.
11	Дан масив, що складається з 12 двійкових чисел. Видалити елементи, які зустрічаються більше двох разів.
12	Ввести масив, в якому тільки два однакових елемента. Визначити їх місце розташування.
13	Дано ціле число в двійковій системі числення, тобто послідовність цифр 0 і 1. Здійснити циклічний зсув елементів масиву вліво на дві позиції. Визначити різницю початкового і одержаного після зсуву числа.

Продовження таблиці 7.3

14	Заданий масив чисел в двійковій системі числення. Впорядкувати елементи масиву по спадаючій. Визначити суму чисел.
15	Заданий масив чисел в двійковій системі числення. Впорядкувати елементи масиву по зростанню. Визначити середнє значення чисел.
16	Заданий масив чисел в двійковій системі числення. Поміняти місцями максимальний і мінімальний елементи.
17	Дано ціле число в двійковій системі числення, тобто послідовність цифр 0 і 1. Здійснити циклічний зсув елементів масиву вправо на одну позицію. Визначити суму початкового і одержаного після зсуву числа.
18	Заданий цілочисельний масив. Визначити різницю між сумою значень елементів масиву на ділянках, де елементи монотонно зростають (кожне наступне число більше попереднього) і сумою значень елементів масиву на ділянках, де елементи монотонно зменшується (кожне наступне число менше попереднього).
19	Заданий цілочисельний масив. Визначити, чи утворюють значення його елементів арифметичну прогресію. Якщо «так» - вивести різниця прогресії, якщо «ні» - відповідь «не утворюють».
20	Заданий цілочисельний масив. Визначити, чи утворюють значення його елементів геометричну прогресію. Якщо «так» - вивести знаменник прогресії, якщо «ні» - відповідь «не утворюють».
21	Заданий цілочисельний масив. Вивести індекси тих елементів, значення яких більше, ніж у вартях праворуч від нього. Визначити кількість таких чисел.
22	З заданого целочисленного масиву вивести номер останнього з тих його елементів, які задовольняють нерівності $A_{i-1} < A_i < A_{i+1}$. Якщо таких елементів немає - вивести «таких немає».

Продовження таблиці 7.3

23	У заданому масиві двійкових чисел визначити кількість чисел, що стоять між максимальним і мінімальним елементами. Вивести повідомлення, якщо таких чисел немає.
24	У заданому масиві двійкових чисел виконати циклічний зсув всіх елементів вправо на одну позицію
25	У заданому масиві двійкових чисел виконати циклічний зсув всіх елементів вліво на одну позицію. Визначити суму чисел до i після операції зсуву.
26	У заданому масиві двійкових чисел збільшити значення кожного елемента на двійкове число 1010.
27	Дан масив дійсних чисел. Визначити елемент масиву (значення і індекс), який найбільш близький до заданого дійсного числа R .
28	Дан масив двійкових чисел. Визначити елемент масиву (значення і індекс), який найбільш віддалений від заданого двійкового числа D .
29	Задані додатне і від'ємне число в двійковій системі (дві послідовності нулів і одиниць). Скласти програму обчислення суми цих чисел.
30	Заданий цілочисельний масив - три десяткових числа. Сформувати масив, що складається з чисел вихідного масиву, представлених в двійковій системі числення.

7.3 Питання для перевірки знань

1. Що таке масив?
2. Які є види масивів?
3. В яких випадках розумно використовувати масив, а не ArrayList?
4. Чому масив символів краще рядка для зберігання пароля?
5. Як одним викликом скопіювати елементи з будь-якої Collection в масив?

6. Як створити і ініціалізувати масив?
7. Як задати тип масиву?
8. Яким чином відбувається звернення до елемента масива?
9. Як визначити розмір масиву?
10. За допомогою якого слова всі елементи масиву автоматично ініціалізовані нульовими значеннями?

ЛАБОРАТОРНА РОБОТА 8

Тема: РЯДКИ

Мета: Отримання практичних навичок по роботі з класами String та StringBuilder.

8.1 Практична частина

Об'єкти класу String є незмінними (immutable). У Java є спеціальні класи StringBuffer і StringBuilder, які допускають зміни в рядку.

Класи String, StringBuffer, StringBuilder визначені в пакеті java.lang і доступні автоматично без оголошення імпорту. Всі три класи реалізують інтерфейс CharSequence.

Створити рядок дуже просто. Наприклад, можна так:

```
String aboutCat = "Кот - це звучить гордо, а якщо наступити на хвіст, то голосно";
```

Можна створити масив рядків:

```
String [] cats = { "Васька", "Барсик", "Мурзик" };
```

Можна створити порожній об'єкт класу String:

```
String str = new String ();
```

Можна створити рядок через масив символів:

```
char [] chars = { 'c', 'a', 't' };  
String str = new String (chars);
```

Є ще конструктор, що дозволяє задати діапазон символівного масиву.

Вам потрібно вказати початок діапазону і кількість символів для використання:

```
char [] chars = { 'c', 'a', 't', 'a', 'm', 'a', 'r', 'a', 'n' };  
String str = new String (chars, 0, 3);
```

Можна створити об'єкт класу String з об'єкта класів StringBuffer і StringBuilder за допомогою наступних конструкторів:

```
String (StringBuffer об'єкт_StrBuf)  
String (StringBuilder об'єкт_StrBuild)
```

Оператори + і += для String

На мові Java знак плюс (+) означає конкатенацію рядків (concatenation), іншими словами - об'єднання рядків.

```
String cat = "Кот";
String name = "Васька";
// складаємо два рядки і пробіл між ними, щоб слова НЕ злиплися
String fullname = cat + " " + name; // вийде Кот Васька
```

Якщо один з операндів у виразі містить рядок, то інші операнди також повинні бути рядками. Тому Java сама може привести змінні до строковому поданням, навіть якщо вони не є рядками.

```
int digit = 4;
String paws = "лапи";
String aboutcat = digit + paws; // хоча ми складаємо число і
рядок, але все одно отримуємо рядок
```

За лаштунками Java за нас перетворило число 4 в рядок "4".

Форматування рядків

Припустимо у нас є строковий ресурс:

```
<String name = "aboutcat"> У kota на ім'я Барсик чотири лапи,
один хвіст. Йому 5 років </ string>
```

Щоб вивести цей рядок програмно в елементі TextView, можна використовувати код:

```
TextView tvCatsInfo = (TextView) findViewById (R.id.textView1);
tvCatsInfo.setText (R.string.aboutcat);
```

Уявімо, що є кілька котів. Можна, звичайно, завести для кожного kota свій рядок. Але ж рядки дуже схожі, змінюються лише імена і вік. Також можна змінювати число лап і хвостів.

У таких випадках можна застосувати форматування рядків. Потрібно визначити слова, які ми будемо міняти і замінити їх на спеціальний набір символів, які починаються з символу відсотка, потім йде число, що збільшується на одиницю, далі \$ s для рядків або \$ d для чисел. Отже, змінимо наш строкової ресурс так:

```
<String name = "aboutcat"> У kota на ім'я% 1 $ s% 2 $ s лапи,% 3
$ s хвіст. Йому% 4 $ d років </ string>
```

Внесемо зміни в код:

```
String strBarsik = "Барсик";
String strPaws = "чотири";
String strTail = "один";
int year = 5;
String strCats = getResources (). GetString (R.string.aboutcat);
String strFinal = String.format (strCats, strBarsik, strPaws,
strTail, year);
tvCatsInfo.setText (strFinal);
```

Якщо є кіт Васька і йому шість років, то додаємо дві змінні і форматуємо рядок:

```
String strVaska = "Васька";
year = 6;
String strFinal = String.format (strCats, strVaska, strPaws,
strTail, year);
tvCatsInfo.setText (strFinal);
```

Тут показаний найпростіший приклад з форматуванням. Треба пам'ятати про нього і застосовуйте в потрібних місцях.

Строковий ресурс

Рядки бажано зберігати в ресурсах (по ресурсам є окрема стаття).

Програмно доступ до строковому ресурсу робиться так:

```
String catName = getResources (). GetString (R.string.barsik);
```

Витягти рядки з строкових масивів в ресурсах

Припустимо, у нас є строковий масив, визначений у файлі strings.xml під ім'ям cats_array. Тоді отримати доступ до рядків з ресурсів можна так:

```
Resources res = getResources ();
String [] cats = res.getStringArray (R.array.cats_array);
```

Методи

```
public char charAt (int index)
```

Повертає символ із зазначеним зміщенням в цьому рядку. Відлік йде від 0. Не треба використовувати негативні і неіснуючі значення, будьте серйозніше. Для вилучення одного символу використовуйте getChars ().

```
String testString = "Кошеня";
char myChar = testString.charAt (2);
```

```
tv.setText (Character.toString (myChar)); // виводить третій  
СИМВОЛ - Т
```

Public int codePointAt (int index)

Повертає Unicode-символ в заданому індексі

```
String testString = "Кошеня";  
int myChar = testString.codePointAt (3);  
tv.setText (String.valueOf (myChar)); // повертає 1105
```

Public int codePointBefore (int index)

Повертає Unicode-символ, який передає даному індексу

```
String testString = "Кошеня";  
int myChar = testString.codePointBefore (4);  
tv.setText (String.valueOf (myChar)); // повертає 1105
```

Public int codePointCount (int start, int end)

Обчислює кількість Unicode-символів між позиціями start і end

```
String testString = "Кошеня";  
int myChar = testString.codePointCount (0, 3);  
tv.setText (String.valueOf (myChar)); // повертає 3
```

public int compareTo (String string)

Порівнює зазначений рядок, використовуючи значення символів Unicode і обчислює, який з рядків менше, дорівнює або більше наступного. Може використовуватися при сортуванні. Регістр враховується. Якщо рядки співпадають, то повертається 0, якщо менше нуля, то викликає рядок менше рядка string, якщо більше нуля, то викликає рядок більше рядка string. Слова з великим регістром стоять вище слова з малими.

```
String testString = "Кошеня";  
if (testString.compareTo ( "кошеня") == 0) {  
    tvInfo.setText ( "Рядки рівні");  
} Else {  
    tvInfo.setText ( "Рядки не рівні. Повернено"  
        + TestString.compareTo ( "кошеня")); // повертає -32  
}
```

Відсортуємо масив рядків через сортування.

```
String [] poem = { "Ми", "веземо", "з", "собою", "кота"};  
for (int j = 0; j <poem.length; j ++)  
for (int i = j + 1; i <poem.length; i ++)
```

```

if (poem [i] .compareTo (poem [j]) <0) {
String temp = poem [j];
poem [j] = poem [i];
poem [i] = temp;
}
}
System.out.println (poem [j]);
}

```

В результаті ми отримаємо:

```

ми
веземо
кота
з
собою

```

Як бачите, від зміни місць доданків сума сортування коти не змінюються.

Public int compareToIgnoreCase (String string)

Порівнює зазначену рядок, використовуючи значення символів Unicode, без урахування регістру.

```

String testString = "Кошеня";

if (testString.compareToIgnoreCase ("кошеня") == 0) {
    tv.setText ("Рядки рівні"); // слова однакові, якщо не
враховувати регістр
} Else {
    tv.setText ("Рядки не рівні. Повернено"
        + TestString.compareTo ("кошеня"));
}

```

Public String concat (String string)

Об'єднує рядок з вказаним рядком. Повертається новий рядок, який містить об'єднання двох рядків. Зверніть увагу, що саме ім'я методу містить кота!

```

String testString = "Сук";

```

```
String newString = testString.concat ("кіт");  
tv.setText (newString);
```

Метод виконує ту ж функцію, що і оператор + і можна було написати Сук + кіт. Але справжній кошатнік буде використовувати "котячий" метод.

Public boolean contains (CharSequence cs)

Визначає, чи містить рядок послідовність символів в CharSequence

```
String testString = "кошеня";
```

```
if (testString.contains ("кіт")) {  
    infoTextView.setText ("У слові кошеня міститься слово  
кіт!");  
}
```

public boolean contentEquals (CharSequence cs) - порівнює CharSequence з цим рядком.

public boolean contentEquals (StringBuffer strbuf) - порівнює StringBuffer з цим рядком

public static String copyValueOf (char [] data, int start, int length) - створює новий рядок, що містить зазначені символи з масиву Data починаючи з позиції start (нумерація починається з нуля) довгою length.

public static String copyValueOf (char [] data) - створює новий рядок, що містить символи із зазначеного масиву. Зміна масиву після створення рядка не змінює створений рядок.

public boolean endsWith (String suffix) - перевіряє, чи закінчується рядок символами suffix.

```
String str1 = "Суккот";  
if (str1.endsWith ("кіт"))  
    infoTextView.setText ("Слово закінчується на котика");  
else  
    infoTextView.setText ("Погане слово. Немає сенсу його  
використовувати");
```

public boolean equals (Object string)

Порівнює вказаний об'єкт і рядок і повертає true, якщо порівнювані рядки рівні, тобто містить ті ж символи і в тому ж порядку з урахуванням регістру.

```
String str1 = "Кот";
String str2 = "Кішка";
if (str1.equals (str2))
    infoTextView.setText ("Рядки збігаються");
else
    infoTextView.setText ("Рядки не збігаються");
```

Не плутати метод з оператором ==, який порівнює два посилання на об'єкти і визначає, чи посилаються вони на один і той же екземпляр.

public boolean equalsIgnoreCase (String string)

Порівнює зазначений рядок з вихідним рядком без урахування регістру і повертає true, якщо вони рівні. Діапазон A-Z вважається рівним діапазону a-z.

```
String str1 = "Кот";
String str2 = "кіт";

if (str1.equalsIgnoreCase (str2))
    infoTextView.setText ("Рядки збігаються");
else
    infoTextView.setText ("Рядки не збігаються");
```

public static String format (Locale locale, String format, Object ... args)

Повертає відформатований рядок, використовуючи доданий формат і аргументи, локалізованих в даній області. Наприклад дату або час

```
// виводимо число типу float з двома знаками після коми
String.format ( "%. 2f", floatValue);
```

Склеюємо два слова, які виводяться з нового рядка. При цьому друге слово виводиться у верхньому регістрі.

```
String str1 = "Кот";
String str2 = "васька";
String strResult = String.format ( "% s \ n% S", str1, str2);
// виводимо результат в TextView
infoTextView.setText (strResult);
```

Конвертуємо число в вісімкову систему.

```
String str1 = "8";
int inInt = Integer.parseInt (str1); // конвертуємо рядок в число
String strResult = String.format ( "(Вісімкове значення):% o \n", inInt);
```

За аналогією виводимо в шістнадцятковій системі

```
String str1 = "255";
int inInt = Integer.parseInt (str1);
String strResult = String.format ( "(Шістнадцяткове значення):% x \n", inInt);
```

```
// число 255 буде виведено як ff
infoTextView.setText (strResult);
```

Для верхнього регістру використовуйте% X, тоді замість ff буде FF.

Для десяткової системи використовуйте% d.

Дату теж можна виводити по різному.

```
Date now = new Date ();
Locale locale = Locale.getDefault ();
infoTextView.setText (
String.format (locale, "% tD \n", now) + // (MM / DD / YY)
String.format (locale, "% tF \n", now) + // (YYYY-MM-DD)
String.format (locale, "% tr \n", now) + // Full 12-hour time
String.format (locale, "% tz \n", now) + // Time zone GMT offset
String.format (locale, "% tZ \n", now)); // Localized time zone
bbreviation
```

public byte [] getBytes (String charsetName) - повертає відформатований рядок, використовуючи доданий формат.

public void getBytes (int start, int end, byte [] data, int index) та інші **перевантажені версії** - метод зберігає символи в масив байтів, альтернатива методу getChars (). Часто використовується при експорті рядків з різних джерел, де використовуються інші символи Unicode. Наприклад, Java за замовчуванням працює з 16-бітовими символами Unicode, а в інтернеті часто рядки використовують 8-бітовий код Unicode, ASCII і ін.

public void getChars (int start, int end, char [] buffer, int index) - метод для вилучення декількох символів з рядка. Вам треба вказати індекс початку підрядка (start), індекс символу, наступного за кінцем витягується підрядка

(end). Масив, який приймає виділені символи знаходиться в параметрі `buffer`. Індекс в масиві, починаючи з якого буде записуватися підстрока, передається в параметрі `index`. Слідкуйте, щоб масив був достатнього розміру, щоб в ньому помістилися всі символи зазначеної підрядка.

```
String unusualCat = "Кошеня на ім'я Гав";
int start = 5;
int end = 12;
char [] buf = new char [end - start];
unusualCat.getChars (start, end, buf, 0);
infoTextView.setText (new String (buf));
```

public int hashCode () - повертає ціле число - хеш-код для даного об'єкта.

public int indexOf (int c) - повертає номер першої зустрічної позиції з вказаним індексом `c`.

```
String testString = "кошеня";
// символ е зустрічається в четвертій позиції (index = 3)
infoTextView.setText (String.valueOf (testString.indexOf (
"e")));
```

public int indexOf (int c, int start) - шукає індекс `c`, починаючи з позиції `start`

```
String testString = "кошеня";
// поверне -1, так як після 5 символу букви е немає
infoTextView.setText (String.valueOf (testString.indexOf ("e",
4)));
```

public int indexOf (String string) - шукає ланцюжок символів `subString`

```
String testString = "У віконця";
infoTextView.setText (String.valueOf (testString.indexOf (
"кішка")));
```

public int indexOf (String subString, int start) - шукає ланцюжок символів `subString`, починаючи з позиції `start`

```
String testString = "У віконця";
infoTextView.setText (String.valueOf (testString.indexOf (
"кішка", 2)));
```

public String intern () - «хешує» (перетворення масиву вхідних даних довільної довжини в (вихідну) бітовий рядок встановленої довжини, що виконується певним алгоритмом) рядок.

public boolean isEmpty () - перевіряє, чи є рядок порожній

```
if (catname.isEmpty ()) {  
    // тут ваш код  
}
```

Даний метод з'явився в API 9 (Android 2.1). Для старих пристроїв використовуйте `String.length () == 0`

public int lastIndexOf (String string) та інші перевантажені версії - повертає номер останньої зустрічної позиції з вказаним індексом. Наприклад, отримати ім'я файлу без розширення можна так:

```
filename.toString (). substring (0, filename.getString ().  
lastIndexOf ( "." ));
```

У цьому прикладі ми отримуємо позицію останньої точки і отримуємо підстроку до неї.

public int length () - повертає довжину рядка

```
String testString = "кошеня";  
infoTextView.setText (String.valueOf (testString.length ())); //  
повертає 7 (сім символів)
```

public boolean matches (String regularExpression) - перевіряє, чи відповідає рядок регулярними виразами.

public int offsetByCodePoints (int index, int codePointOffset) - повертає позицію, розташовану на відстані `codePointOffset` після початкової позиції, заданої параметром `index`.

public boolean regionMatches (int thisStart, String string, int start, int length) - метод порівнює вказану частину рядка з іншою частиною рядка. Потрібно поставити індекс початку діапазону рядка викликає об'єкта класу `String`. Рядок для порівнювання передається в параметрі `string`. Індекс символу, починаючи з якого потрібно виконувати порівняння передається в параметрі `start`, а довжина порівнює підрядок в параметрі `length`.

public boolean regionMatches (boolean ignoreCase, int thisStart, String string, int start, int length) - переважана версія. Метод порівнює вказану частину рядка з іншою частиною рядка, ігноруючи регістр.

public String replace (CharSequence target, CharSequence replacement) та інші переважені версії - змінює знак або послідовність символів target на replacement

```
String testString = "кит";
// міняємо і на о
infoTextView.setText (testString.replace ( "i", "o")); //
повертається кіт
```

public String replaceAll (String regularExpression, String replacement)

```
int i = Integer.parseInt ( "kitty123" .replaceAll ( "[\\ D]",
""));
int j = Integer.parseInt ( "123kitty" .replaceAll ( "[\\ D]",
""));
int k = Integer.parseInt ( "1k2it3ty" .replaceAll ( "[\\ D]",
""));
```

```
// поверне
```

```
i = 123;
```

```
j = 123;
```

```
k = 123;
```

public String replaceFirst (String regularExpression, String replacement) - видаляє перші символи за допомогою регулярного виразу.

Наприклад, якщо потрібно видалити нулі на початку чисел 001, 007, 000 024, то можна використовувати такий виклик.

```
String s = "001234-cat";
```

```
String s = s.replaceFirst ( "^ 0 *", ""); // залишиться 1234-cat
```

public String [] split (String regularExpression) та інші переважені версії - розбиває рядок на масив з слів.

Наприклад, є рядок Васька Рижик Мурзик Барсик і ми хочемо отримати масив імен котів:

```
String catnames = "Васька Рижик Мурзик Барсик";
```

```
String aCats [] = catnames.split ( ""); // по пропуску
```

отримаємо:

```
aCats [0] = Васька  
aCats [1] = Рижик  
aCats [2] = Мурзик  
aCats [3] = Барсик
```

public boolean startsWith (String prefix) - перевіряє, чи починається рядок символами prefix з початку рядка

```
String str1 = "котлета";  
if (str1.startsWith ( "кіт"))  
    infoTextView.setText ( "Слово містить kota");  
else  
    infoTextView.setText ( "Погане слово. Немає сенсу його  
використовувати");
```

public boolean startsWith (String prefix, int start) - перевіряє, чи починається заданий рядок символами prefix з вказаної позиції.

```
String str1 = "Суккот";  
if (str1.startsWith ( "кіт", 3))  
    infoTextView.setText ( "Слово містить kota");  
else  
    infoTextView.setText ( "Погане слово. Немає сенсу його  
використовувати");
```

public CharSequence subSequence (int start, int end) - аналогічний методу substring (), але може використовуватися для CharSequence.

public String substring (int start) і інші перевантажені версії - створює нову послідовність / рядок з символами з цього рядка починаючи з позиції start до кінця рядка / закінчуючи символом з позиції end. Новий рядок містить символи від start до end - 1, тому беремо на один символ більше.

```
String testString = "скотина";  
infoTextView.setText (testString.substring (1, 4)); //  
повертається кіт
```

public char [] toCharArray () - копіює символи в цьому рядку в масив символів. Той же результат можна отримати через метод getChars (). Документація не рекомендує використовувати даний метод, пропонуючи метод charAt ().

```
String unusualCat = "Кошеня на ім'я Гав";
```

```
char [] yomoe = unusualCat.toCharArray ();
infoTextView.setText (String.valueOf (yomoe [3]));
```

public String toLowerCase () і інші перевантажені версії - перетворює рядок в нижній регістр. Перетворенням управляє задана за замовчуванням регіональна мова.

```
String cat = "Кот";
String lower = cat.toLowerCase ();
infoTextView.setText (lower);
```

public String toString () - повертає рядок.

public String toUpperCase () - перетворює рядок у верхній регістр.

Перетворенням управляє задана за замовчуванням регіональна мова.

```
String cat = "Кот";
String upper = cat.toUpperCase ();
infoTextView.setText (upper);
```

public String trim () - видаляє пробіли на початку і в кінці рядка.

```
String str = "Hello Kitty" .trim ();
infoTextView.setText (str);
```

public static String valueOf (long value) та інші перевантажені версії - конвертує вміст (числа, об'єкти, символи, масиви символів) в рядок.

```
int catAge = 7; // це число
infoTextView.setText (String.valueOf (catAge)); // перетворено в
рядок
```

Генеруємо випадкову рядок

Припустимо, нам потрібен випадковий рядок із заданих символів.

```
private static final String mCHAR =
"ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890";
private static final int STR_LENGTH = 9; // довжина генерується
рядка
Random random = new Random ();

public void onClick (View view) {
    TextView infoTextView = (TextView) findViewById
(R.id.textViewInfo);
    infoTextView.setText (createRandomString ());
}
```

```

public String createRandomString () {
    StringBuilder builder = new StringBuilder ();
    for (int i = 0; i <STR_LENGTH; i ++) {
        int number = random.nextInt (mCHAR.length ());
        char ch = mCHAR.charAt (number);
        builder.append (ch);
    }
    return builder.toString ();
}

```

Порівняння рядків: equals () або ==?

Розглянемо приклад.

```

String str1 = "Мурзик";
String str2 = new String (str1);
boolean isCat = str1 == str2;
infoTextView.setText (str1 + "==" + str2 + "->" + isCat);

```

Хоча в двох змінних міститься одне і те ж слово, ми маємо справу з двома різними об'єктами і оператор == поверне false.

Мова Java у випадку з логічним оператором == (а також! =) порівнює посилання на об'єкти (при роботі з примітивами такої проблеми немає):

```

String s1 = "hello";
String s2 = "hello";
String s3 = s1;
String s4 = "h" + "e" + "l" + "l" + "o";
String s5 = new String ( "hello");
String s6 = new String (new char [] { 'h', 'e', 'l', 'l', 'o'});

infoTextView.setText (s1 + "==" + s2 + ":" + (s1 == s2));
// спробуйте і інші варіанти
// infoTextView.setText (s1 + "equals" + s2 + ":" + (s1.equals
(s2)));
// infoTextView.setText (s1 + "==" + s3 + ":" + (s1 == s3));
// infoTextView.setText (s1 + "equals" + s3 + ":" + (s1.equals
(s3)));
// infoTextView.setText (s1 + "==" + s4 + ":" + (s1 == s4));
// infoTextView.setText (s1 + "equals" + s4 + ":" + (s1.equals
(s4)));

```

```

        // infoTextView.setText (s1 + "==" + s5 + ":" + (s1 == s5)); //
false
        // infoTextView.setText (s1 + "equals" + s5 + ":" + (s1.equals
(s5)));
        // infoTextView.setText (s1 + "==" + s6 + ":" + (s1 == s6)); //
false
        // infoTextView.setText (s1 + "equals" + s6 + ":" + (s1.equals
(s6)));

```

8.2 Самостійна частина

Завдання 1. Написати програму згідно варіанту (Таблиця 8.1).

Таблиця 8.1. Варіанти до завдання 1.

№ варіану	Задача
1	У заданому рядку порахувати кількість російських букв «А».
2	У заданому рядку поміняти місцями перший і останній символ рядка.
3	У заданому рядку замість першого символу поставити пробіл, а замість останнього - крапку.
4	У заданому рядку видалити перший символ.
5	У заданому рядку додати в кінець рядка символ «>».
6	У заданому рядку порахувати кількість символів цифр (від «0» до «9»).
7	Визначити, чи є в заданому рядку символ цифра. Вивести повідомлення про це.
8	У заданому рядку замість кожного пропуску поставити символ «>».
9	У заданому рядку видалити останній символ.
10	У заданому рядку видалити всі пробіли.
11	У заданому рядку видалити другий і четвертий за рахунком символи.

Продовження таблиці 8.1.

12	Знайти суму кодів всіх символів заданого рядка.
13	Видалити перше слово заданого рядка. Роздільником слів вважається пробіл.
14	У заданому рядку порахувати кількість крапок і ком.
15	У заданому рядку порахувати кількість круглих і квадратних дужок.
16	Взяти заданий рядок в лапки виду «».
17	В кінці заданого рядка дописати символи «_», довівши довжину рядка до 25.
18	Для заданого рядка визначити чи входить в неї хоча б одна російська буква.
19	Для заданого рядка визначити чи входить в неї хоча б одна латинська буква.
20	У заданому рядку знайти місце розташування першого символу «*».
21	Видалити в заданому рядку все, крім першого і останнього символу.
22	Для заданого рядка, визначити які символи зустрічаються частіше «!» або «?».
23	У заданому рядку визначити, чи збігається кількість круглих відкриваються і круглих закриваються дужок.
24	У заданому рядку після кожної крапки поставити ще одну точку.
25	У заданому рядку після кожної крапки поставити пробіл.
26	У рядку замінити кожен символ «!» Символом «» (пропуск).
27	Визначити, чи входить до цього рядка пробіл.
28	У заданому рядку порахувати кількість поєднань символів «- +».
29	У заданому рядку видалити всі російські літери.
30	У заданому рядку видалити всі латинські літери.

Завдання 2. Написати програму згідно варіанту (Таблиця 8.2).

Таблиця 8.2. Варіанти до завдання 2.

№ варіану	Задача
1	У заданому рядку замінити пробіли, які йдуть поспіль - на один пробіл.
2	У заданому рядку порахувати кількість слів. Роздільником слів вважається один або кілька пробілів.
3	У заданому рядку замінити кожну російську літеру символом «*».
4	У заданому рядку видалити всі латинські літери.
5	Дан рядок. Дописати в кінець рядка його довжину.
6	У заданому рядку дописати після кожного символу «*» символ «».
7	У заданому рядку вставити перед кожним символом «!» Символ «,»
8	Задані два однакові по довжині рядка. Побудувати новий рядок, в якому на парних місцях розташовані елементи першого рядка, а на непарних - елементи другого рядка.
9	У заданому рядку замінити кожен пробіл двома пробілами.
10	Задан рядок. Побудувати новий рядок, в якій всі символи записані в зворотному порядку.
11	Побудувати рядок, що складається з малих букв латинського алфавіту (за алфавітом).
12	Побудувати рядок, що складається з великих літер російського алфавіту (за алфавітом).
13	У заданому рядку поміняти місцями символи між собою, щостоять поруч.
14	У заданому рядку видалити перший символ «», що знаходиться в рядку.
15	У заданому рядку видалити останній символ «» (пропуск), що знаходиться в рядку.
16	У заданому рядку розташувати всі символи по зростанню їх кодів.

Таблиця 8.2. Варіанти до завдання 2.

17	У заданому рядку замінити кожен символ «№» рядком «номер один по одному».
18	У заданому рядку перед кожною великою латинською літерою поставити «» (точка, пробіл).
19	Розрядити пробілами заданий рядок.
20	У заданому рядку замінити поєднання символів «- +» символом «0».
21	У заданому рядку замінити кожен символ рядка наступним за кодом символом. Букву «я» замінити пробілом.
22	У заданому рядку поміняти кожен символ рядка попереднім по таблиці кодування символом.
23	У заданому рядку після кожного символу вставити число, що відповідає коду цього символу.
24	Отримати з заданого рядка два рядки, що складаються з символів першого рядка, що мають відповідно парні і непарні індекси.
25	Задані два рядки однакової довжини. Побудувати новий рядок, що складається з символів як однієї, так і іншого рядка, що чергуються між собою.
26	У рядку після кожного слова дописати його довжину.
27	Задан рядок. Отримати передостаннє слово цього рядка. Роздільником слів вважаються один або кілька пробілів.
28	В заданому рядку замінити парну кількість поспіль пробілів символом «Ч», а непарну - символом «Н».
29	Для заданого рядка після кожної російської літери поставити її код, а після кожної латинської - символ «».
30	У заданому рядку, після кожного слова типу «sin», «cos» або «log», поставити дужку «(».

Завдання 3. Написати програму згідно варіанту (Таблиця 8.3).

Таблиця 8.3. Варіанти до завдання 3.

№ варіанту	Задача
1	У заданому рядку поміняти місцями перше й останнє слово рядка. Роздільниками слів вважаються пробіли.
2	Задані два рядки. Побудувати новий рядок, що складається з символів, які входять як в одну, так і в інший рядок.
3	Задані два рядки. Побудувати новий рядок, що складається з символів, які входять в перший рядок, але не входять в другу.
4	У заданому рядку замінити кожен символ «*» числом, відповідним номером один по одному входження цього символу в рядок.
5	У заданому рядку замінити кожен символ «!» Числом, рівним індексу цього символу в рядку.
6	У заданому рядку замінити кожен символ рядка наступним за кодом символом. Букву «я» замінити пробілом.
7	У заданому рядку порахувати кількість різних символів, що входять до цього рядка.
8	У заданому рядку розташувати в зворотному порядку всі слова. Роздільниками слів вважаються пробіли.
9	У заданому рядку порахувати кількість слів, що містять тільки рядкові російські букви. Роздільниками слів вважаються пробіли.
10	За заданим рядком отримати масив слів, що входять до цього рядка. Роздільниками слів вважаються пробіли.

8.3 Питання для перевірки знань

1. Який метод повертає строкове представлення об'єкту?
2. Які "строкові" класи ви знаєте?
3. Які основні властивості "строкових" класів (їх особливості)?
4. Чи можна успадкувати строковий тип, чому?

5. Дайте визначення поняттю конкатенація рядків.
6. Як перетворити рядок в число?
7. Як порівняти значення двох рядків?
8. Як перевернути рядок?
9. Як працює порівняння двох рядків?
10. Як обрізати прогалини в кінці рядка?
11. Як замінити символ в рядку?
12. Як отримати частину рядка?
13. Дайте визначення поняттю "пул рядків".
14. Який метод дозволяє виділити підрядок в рядку?
15. Як розбити рядок на підрядки по заданому роздільнику?
16. Який метод викликається для перетворення змінної в рядок?
17. Як дізнатися значення конкретного символу рядка, знаючи його порядковий номер у рядку?
18. Як знайти необхідний символ в рядку?
19. Чи можна синхронізувати доступ до рядка?
20. Що робить метод `intern ()`?
21. Чим відрізняються і що спільного у класів `String`, `StringBuffer` і `StringBuilder`?
22. Як правильно порівняти значення рядків двох різних об'єктів типу `String` і `StringBuffer`?
23. Чому масив символів краще рядка для зберігання пароля?
24. Чому рядок є популярним ключем в `HashMap` в `Java`?
25. Напишіть метод видалення даного символу з рядка.

ЛАБОРАТОРНА РОБОТА 9

Тема: РОБОТА З КЛАСАМИ-КОЛЕКЦІЯМИ JAVA

Мета: 1) Вивчити особливості роботи з класами-колекціями Java.
2) Навчитися створювати програми обробки даних використовуючи класи-колекції.

9.1 Практична частина

Основна ідея при розгляді цих інтерфейсів має бути така - є список методів, які вкрай важливі для певних типів колекцій - списків, множин, черг і інше. Список має свої особливості, безліч - свої, чергу - свої. Набір методів для списку і для безлічі буде відрізнятися, тому що ці типи колекцій (список і безліч) мають деякі важливі відмінності. Їх треба розглядати як спеціалізовані інструменти - наприклад, для закручування шурупів потрібен шуруповерт, для бетонних стін - перфоратор, для сверління лунки - ледобур. Зауважте, що вони всі мають «одну природу», але кожен має деяку спеціалізацію:

java.util.Collection - основний інтерфейс, який описує базові методи, якими повинна володіти будь-яка колекція. Тобто якщо якийсь клас претендує на звання «колекція» - він повинен реалізувати ті методи, які описані в цьому інтерфейсі. Проводячи аналогію з нашим набором свердлильних інструментів - інтерфейс **java.util.Collection** їх загальний батько - у нього є можливість сверdlити. Можливо, що Java версії 8 (і вище) здасться вам важкувато, тому для початку раджу зайти на документацію по Java версії 7. [java.util.Collection](#).

java.util.List - інтерфейс для операцій з колекцією, яка є списком. Список може включати однакові елементи. Елементи в списку зберігаються в тому порядку, в якому вони туди занесені. Мимовільних переміщень елементів в ньому не відбувається - тільки з вашого відома. Наприклад, ви

можете додати елемент на якусь позицію і тоді відбудеться зсув інших елементів.

Можна отримати доступ до будь-якого елемента по його порядковому номеру / індексу всередині списку.

Тобто якщо вам потрібно, щоб колекція володіла такими властивостями - вибирайте клас, який реалізує інтерфейс `java.util.List`.

`java.util.Set` - інтерфейс для зберігання безлічі. На відміну від `java.util.List` цей інтерфейс не може мати однакові елементи (дивимося методи `equals` і `hashCode` в статті Рішення на основі класів) і порядок зберігання елементів у множині може змінюватися при додаванні / видаленні / зміні елемента. Може виникнути питання, навіщо така колекція потрібна - це зручно в разі, коли ви створюєте набір унікальних елементів з якоїсь групи елементів.

`java.util.SortedSet` - це спадкоємець інтерфейсу `java.util.Set` і його додатковим функціоналом є автоматичне вибудовування елементів всередині безлічі по порядку.

`java.util.Queue` - інтерфейс пропонує працювати з колекцією як з чергою, тобто колекція має метод для додавання елементів в один кінець і метод для отримання елемента з іншого кінця - тобто справжня черга за принципом FIFO - First In First Out - якщо першим прийшов, то першим і підеш. Для широкого кола завдань така конструкція роботи з колекцією буває досить зручною структурою.

`java.util.Map` - дуже зручна конструкція, яка зберігає дані не у вигляді списку значень, а в вигляді пари ключ-значення. Це дуже затребувана форма, в якій ви отримуєте доступ до значення в колекції по ключу. Наприклад, доступ до даних користувача на сайті може бути здійснений за логіном (по email наприклад). Самих даних може бути досить багато, але для пошуку можна використовувати дуже коротку рядок-ключ.

Колекція дозволяє вам працювати з групою об'єктів і спеціалізація колекції визначається вимогами до самих даних і до тих операцій, які потрібно використовувати при роботі з даними.

Приклад (демонстрацію) використання основних методів інтерфейсу `java.util.Collection`.

```
package edu.javacourse.collection;

import java.util.ArrayList;
import java.util.Collection;
import java.util.Iterator;

public class ExampleCollection
{
    public static void main (String [] args) {
        // Створюємо колекції для демонстрації
        Collection coll = createFirstCollection ();
        Collection col2 = createSecondCollection ();

        // Демонстрація проходу по колекції
        System.out.println ( "===== Прохід по колекції");
        for (Object o: coll) {
            System.out.println ( "Item:" + o);
        }

        System.out.println ();
        // Демонстрація проходу по колекції через ітератор
        System.out.println ( "===== Прохід по колекції
через ітератор");
        for (Iterator it = coll.iterator (); it.hasNext ();) {
            String s = (String) it.next ();
            System.out.println ( "Item:" + s);
        }
        System.out.println ();

        // Демонстрації групових операцій
        System.out.println ();
        System.out.println ( "===== Групові операції");
```

```

// Можна перевірити містяться чи ВСІ елементи col2 в col
if (coll.containsAll (col2)) {
    System.out.println ( "Колекція col містить всі від
col2");
}

System.out.println ( "===== Додавання всіх
елементів в col1 з col2");
// Можна додати елементи з col2 в col1
coll.addAll (col2);
for (Object o: coll) {
    System.out.println ( "Item:" + o);
}

System.out.println ( "===== Видалення всіх
елементів col2, які є в col1");
// Можна видалити ВСІ елементи col2, які є в col1
coll.removeAll (col2);
for (Object o: coll) {
    System.out.println ( "Item:" + o);
}

// перестворює колекції для дпальнейшей демонстрації
coll = createFirstCollection ();
col2 = createSecondCollection ();
System.out.println ( "===== Видалення елементів з
coll, яких немає в col2");
coll.retainAll (col2);
for (Object o: coll) {
    System.out.println ( "Item:" + o);
}
System.out.println ( "===== Очищення колекції -
не буде елементів");
coll.clear ();
for (Object o: coll) {
    System.out.println ( "Item:" + o);
}
System.out.println ();

```



```

// Видалення елемента колекції
// Знову створюємо колекцію для демонстрації
coll = createFirstCollection ();
// Видаляємо один елемент
coll.remove ( "1");
System.out.println ( "===== Видаляємо елемент 1
'- його не буде в списку");
for (Object o: coll) {
    System.out.println ( "Item:" + o);
}

// Видалення колекції через ітератор
// Знову створюємо колекцію для демонстрації
coll = createFirstCollection ();
System.out.println ( "===== Видалення через
итератор");
while (! coll.isEmpty ()) {
    Iterator it = coll.iterator ();
    Object o = it.next ();
    System.out.println ( "Видаляємо:" + o);
    // Видаляємо елемент
    it.remove ();
}
}

// Перша колекція для прикладу
private static Collection createFirstCollection () {
    // Створити колекцію на основі стандартного класу
ArrayList
Collection col = new ArrayList ();
// Додавання в колекцію
col.add ( "1");
col.add ( "2");
col.add ( "3");
col.add ( "4");
col.add ( "5");
col.add ( "6");
col.add ( "7");
return col;

```

```

    }

    // Друга колекція для прикладу
    private static Collection createSecondCollection () {
        // Створити колекцію на основі стандартного класу
ArrayList
        Collection col2 = new ArrayList ();
        col2.add ( "1");
        col2.add ( "2");
        col2.add ( "3");
        return col2;
    }
}

package edu.javacourse.collection;

import java.util.ArrayList;
import java.util.Collection;
import java.util.Iterator;

public class ExampleCollection
{
    public static void main (String [] args) {
        // Створюємо колекції для демонстрації
        Collection col1 = createFirstCollection ();
        Collection col2 = createSecondCollection ();

        // Демонстрація проходу по колекції
        System.out.println ( "===== Прохід по колекції");
        for (Object o: col1) {
            System.out.println ( "Item:" + o);
        }
    }
}

```

Якщо дивитися приклад рядок за рядком, то можна побачити, які саме функції використовуються і при запуску можна подивитися результат виконання цих функцій. В принципі все досить просто - є можливість додавати в колекцію елемент, є можливість його видаляти, є можливість

пройти по всьому списку елементів і деякі інші операції. Для початку розглянемо два методи, де ми створюємо колекції.

```
// Перша колекція для прикладу
private static Collection createFirstCollection () {
    // Створити колекцію на основі стандартного класу
ArrayList
    Collection col = new ArrayList ();
    // Додавання в колекцію
    col.add ("1");
    col.add ("2");
    col.add ("3");
    col.add ("4");
    col.add ("5");
    col.add ("6");
    col.add ("7");
    return col;
}

// Друга колекція для прикладу
private static Collection createSecondCollection () {
    // Створити колекцію на основі стандартного класу
ArrayList
    Collection col2 = new ArrayList ();
    col2.add ("1");
    col2.add ("2");
    col2.add ("3");
    return col2;
}
```

Тобто, ми беремо потрібний клас і створюємо його екземпляр. Для прикладу оберемо `java.util.ArrayList`. Оскільки цей клас реалізує (імплементує) інтерфейс `java.util.Collection`, то у нього є всі методи, які в інтерфейсі описані. Додавання в колекцію відбувається тоді, коли ми викликаємо метод `add`. Викликали - тепер ваш об'єкт вже в колекції. Додавайте скільки завгодно рядків або інші об'єкти.

В колекцію можна помістити об'єкт будь-якого класу, але не можна туди помістити елементарний тип - `int`, `char`, `long`.

При побудові великих систем дуже вигідно інтегрувати окремі частини через інтерфейси - контракти. Як один модуль реалізує для іншого цей контракт - це особиста справа модуля-реалізатора. Головне - виконати контракт в повному обсязі. І чим менше треба виконувати - тим простіше. Значить є сенс свідомо використовувати тільки те, що треба і не писати зайвий код.

```
// Створюємо колекції для демонстрації
Collection coll = createFirstCollection ();
Collection col2 = createSecondCollection ();

// Демонстрація проходу по колекції
System.out.println ("===== Прохід по колекції");
for (Object o: coll) {
    System.out.println ("Item:" + o);
}
```

В даному лістингу коду демонструється конструкція, яка дозволяє звернутися до кожного елементу колекції. Зверніть увагу - що всі класи успадковуються від класу Object, то будь-який елемент в колекції може розглядатися як Object.

Тут при кожному циклі ми поміщаємо в змінну про посилання на наступний об'єкт в колекції. Тобто у нас виходить два посилання - одна всередині самої колекції, друга - наша змінна.

Така конструкція з'явилася тільки в Java 1.5. До цього для проходу по колекції треба було використовувати ітератор:

```
// Демонстрація проходу по колекції через ітератор
System.out.println ("===== Прохід по колекції
через ітератор");
for (Iterator it = coll.iterator (); it.hasNext ();) {
    String s = (String) it.next ();
    System.out.println ("Item:" + s);
}
```

java.util.Iterator - це інтерфейс, який дозволяє переміщатися за списком елементів. При виклику методу iterator () ви отримуєте показник на початок колекції, але - **УВАГА** - немає на перший елемент. Метод ітератора

hasNext () повертає true у випадку, якщо ітератор може переміститися до наступного елемента (є наступний за поточним), якщо отримуємо false - значить елементів більше немає.

Метод ітератора next () переміщується на наступний елемент і повертає його значення - об'єкт типу.

Нагадаємо, що з об'єктами ми працюємо через посилання і тому реальний об'єкт в колекції має тип String, то приведення не викличе помилок. Таке приведення дозволить нам працювати тепер з об'єктів як з рядком - там багато всяких цікавих методів є, яких немає у Object.

Роздивимось фрагмент коду:

```
// Видалення елемента колекції
    // Знову створюємо колекцію для демонстрації
    coll = createFirstCollection ();
    // Видаляємо один елемент
    coll.remove ("1");
    System.out.println ("===== Видаляємо елемент 1
'- його не буде в списку");
    for (Object o: coll) {
        System.out.println ("Item:" + o);
    }
```

Тут ми видаляємо елемент з колекції. Але що дуже важливо відзначити - ми передаємо інший об'єкт. Елемент, який ми передали методу remove і об'єкт, який знаходиться в колекції - це однакові, але різні об'єкти. По суті з колекції видаляється об'єкт, для якого метод equals повертає true.

```
// Видалення колекції через ітератор
    // Знову створюємо колекцію для демонстрації
    coll = createFirstCollection ();
    System.out.println ("===== Видалення через
итератор");
    while (! coll.isEmpty ()) {
        Iterator it = coll.iterator ();
        Object o = it.next ();
        System.out.println ("Видаляємо:" + o);
        // Видаляємо елемент
        it.remove ();
    }
```

```
}
```

Приклад 2. Обчислити скільки разів кожна буква зустрічається в тексті.

```
import java.util.HashMap;
import java.util.*;
public class Main {
    public static void main (String [] args) {
        String txt = "лабораторна робота";
        HashMap <Character, Integer> map = new HashMap <Character,
Integer> (40);
        for (int i = 0; i <txt.length (); ++ i) {
            char c = txt.charAt (i);
            // перевіряємо чи є символ буквою
            if (Character.isLetter (c)) {
                if (map.containsKey (c)) {
                    map.put (c, map.get (c) + 1);
                } Else {
                    map.put (c, 1);
                }
            }
            // вивід на екран букв з частотою їх появи
            for (Entry <Character, Integer> entry: map.entrySet ()) {
                System.out.println ("буква:" + entry.getKey () + "кол - у:" +
entry.getValue ());
            }
        }
    }
}
```

Приклад 3. ArrayList Example in Java

```
import java.util.*;
public class ArrayListExample {
    public static void main (String args []) {
        /* Creation of ArrayList: I'm going to add String
        * Elements so I made it of string type */
        ArrayList <String> obj = new ArrayList <String> ();
        /* This is how elements should be added to the array list */
        obj.add ( "Ajeet");
        obj.add ( "Harry");
        obj.add ( "Chaitanya");
        obj.add ( "Steve");
        obj.add ( "Anuj");
    }
}
```

```

    / * Displaying array list elements * /
    System.out.println ( "Currently the array list has following
elements:" + obj);
    / * Add element at the given index * /
    obj.add (0, "Rahul");
    obj.add (1, "Justin");
    / * Remove elements from array list like this * /
    obj.remove ( "Chaitanya");
    obj.remove ( "Harry");
System.out.println ( "Current array list is:" + obj);
    / * Remove element from the given index * /
    obj.remove (1);
System.out.println ( "Current array list is:" + obj);
    }
}

```

Приклад 3. Set in Java

```

Set setA = new HashSet ();
setA.add ( "element 0");
setA.add ( "element 1");
setA.add ( "element 2");
// access via Iterator
Iterator iterator = setA.iterator ();
while (iterator.hasNext () {
    String element = (String) iterator.next ();
}
// access via new for-loop
for (Object object: setA) {
    String element = (String) object;
}

```

Приклад 4. HashMapExample.java

```

import java.util.HashMap;
import java.util.Map;
public class HashMapExample {
public static void main (String [] args) {
Map vehicles = new HashMap ();
// Add some vehicles.
vehicles.put ( "BMW", 5);
vehicles.put ( "Mercedes", 3);
vehicles.put ( "Audi", 4);

```

```

vehicles.put ( "Ford", 10);
System.out.println ( "Total vehicles:" + vehicles.size ());
// Iterate over all vehicles, using the keySet method.
for (String key: vehicles.keySet ())
System.out.println (key + "-" + vehicles.get (key));
System.out.println ();
String searchKey = "Audi";
if (vehicles.containsKey (searchKey))
System.out.println ( "Found total" + vehicles.get (searchKey)
+ ""
+ searchKey + "cars! \ N");
// Clear all values.
vehicles.clear ();
// Equals to zero.
System.out.println ( "After clear operation, size:" +
vehicles.size ());
}
}

```

9.2 Самостійна частина

Завдання 1. Програму відповідно до індивідуального завданням.
Обираємо два завдання: варіант = номер по журналу та номер + 1.

Таблиця 9.1. Варіанти до завдання 1.

№ варіанту	Завдання
1	Створити стек з цілих чисел. Обчислити твір непарних значень елементів стека. Організувати перегляд даних стека.
2	Створити чергу з дійсних чисел. Визначити кількість позитивних значень елементів черги. Організувати перегляд даних черзі.
3	Дано число $N (> 0)$ і набір з N чисел. Створити стек, що містить вихідні числа (останнє число буде вершиною стека), і вивести покажчик на його вершину.

Продовження таблиці 9.1

4	Створити чергу, інформаційними полями якого є: комп'ютер і обсяг його оперативної пам'яті. Видалити з черги відомості про комп'ютер, які були введені першими. Організувати перегляд даних черзі і обчислити загальний обсяг пам'яті комп'ютерів, записаних в черзі.
5	Дано число D і покажчики $P1$ і $P2$ на початок і кінець черги, що містить не менше двох елементів. Додати елемент зі значенням D в кінець черги і витягти з черги перший (початковий) елемент. Вивести значення витягнутого елемента і вміст черги. Після вилучення елемента з черги звільнити пам'ять, займану цим елементом.
6	Створити стек з дійсних чисел. Визначити максимальний елемент в стеці. Організувати перегляд даних стека.
7	Створити стек, інформаційними полями якого є: прізвище та середній бал студента. Додати в стек відомості про нового студента. Організувати перегляд даних стека.
8	Створити чергу, інформаційними полями якої є: телефон і його ціна. Видалити з черги Ознайомлення з телефоном, які були введені першими. Організувати перегляд даних черзі.
9	Створити стек, інформаційними полями якого є: назва гори і висота. Додати в стек відомості про нову гору. Організувати перегляд даних стека і визначити середню висоту гір.
10	Створити стек, інформаційними полями якого є: назва книги і кількість сторінок. Додати в стек відомості про нову книгу. Організувати перегляд даних стека і визначити кількість книг в стеці.

Продовження таблиці 9.1

11	Створити чергу, інформаційними полями якої є: прізвище та середній бал студента. Додати в чергу відомості про нового студента. Організувати перегляд даних черзі.
12	Створити чергу, інформаційними полями якої є: довжини катетів прямокутного трикутника (два дійсних числа). Додати в чергу відомості про новий трикутник. Організувати перегляд даних черзі. Визначити периметр трикутника на початку черги.
13	Створити стек, інформаційними полями якого є: вулиця, номер будинку і номер квартири. Додати в стек відомості про нову квартиру. Організувати перегляд даних стека і визначити кількість будинків на вулиці «Дерибасівська».
14	Дано число $N (> 0)$ і покажчики $P1$ і $P2$ на початок і кінець непорожньої черги. Витягти з черги N початкових елементів і вивести їх значення (якщо черга містить менше N елементів, то витягти всі її елементи). Після вилучення елементів з черги звільняти пам'ять, яку вони займали.
15	Створити чергу з дійсних чисел. Визначити мінімальний елемент черги. Організувати перегляд даних черзі.
16	Створити стек, інформаційними полями якого є: найменування товару і його ціна. Додати в стек відомості про новий товар. Організувати перегляд даних стека і обчислити середню ціну товарів.
17	Створити чергу, інформаційними полями якої є: найменування товару та його вартість. Додати в чергу відомості про новий товар. Організувати перегляд даних черзі і обчислити загальну вартість товарів з найменуванням «Ручка кулькова».

Продовження таблиці 9.1

18	Створити стек з цілих чисел. Обчислити середнє арифметичне парних значень елементів стека. Організувати перегляд даних стека.
19	Створити чергу, інформаційними полями якої є: найменування процесора і його тактова частота і кількість ядер. Додати в чергу відомості про новий процесор. Організувати перегляд даних черзі і роздрукувати дані про багатоядерні процесори (кількість ядер більше 1).
20	Створити чергу з цілих чисел. Визначити кількість парних значень елементів черги. Організувати перегляд даних черзі.
21	Створити чергу з цілих чисел. Визначити середнє значення елементів черги. Організувати перегляд даних черзі.
22	Створити стек, інформаційними полями якого є: книга і її ціна. Додати в стек відомості про нову книгу. Організувати перегляд даних стека і обчислити середню ціну книг.
23	Створити чергу з цілих чисел. Визначити кількість позитивних елементів черги. Організувати перегляд даних черзі.
24	Створити чергу, інформаційними полями якого є: книга і її ціна. Додати в чергу відомості про нову книгу. Організувати перегляд даних черзі і обчислити середню ціну книг.
25	Створити чергу з відомостей про клієнтів банку: прізвища та суми на рахунку. Визначити кількість клієнтів банку, у яких сума на рахунку більше 10000 гр. Організувати перегляд даних черзі.
26	Створити стек, інформаційними полями якого є: диск і його обсяг. Додати в стек відомості про новий диск. Організувати перегляд даних стека і обчислити диск з максимальним об'ємом.
27	Створити чергу з цілих чисел. Визначити кількість елементів черги менших 10. Організувати перегляд даних черзі.

Продовження таблиці 9.1

28	Створити стек, інформаційними полями якого є: прізвище працівника і його оклад. Додати в стек відомості про нового працівника. Організувати перегляд даних стека і обчислити середній оклад.
29	Створити чергу з дійсних чисел. Визначити кількість негативних чисел черзі. Організувати перегляд даних черзі.
30	Створити стек, інформаційними полями якого є: монітор, діагональ і його ціна. Додати в стек відомості про новий монітор. Організувати перегляд даних стека і визначити кількість моніторів з діагоналлю більше 20 дюймів.

Завдання 2. Написати програму відповідно до варіанту.

Таблиця 9.2. Варіанти до завдання 2.

№ варіанту	Завдання
1	Створити стек цілочисельних значень, для реалізації використовуючи одиночні списки. Реалізувати операції додавання (push) і видалення (pop) елемента з стека. Додайте в стек числа 4, 3, 1, 2, 4 і роздрукуйте вміст стека. Видаліть один елемент з стека, і роздрукуйте вміст стека ще раз. Знайдіть мінімальний елемент, що належить стеку.
2	Створити чергу речових значень, для реалізації використовуючи одинзв'язні списки. Реалізувати операції додавання (enqueue) і видалення (dequeue) елемента з черги. Додайте в чергу числа: -2.2, 2.3, 2.2, 5.1, 6.7 і роздрукуйте вміст черги. Видаліть 3 елементи з черги, потім додайте в чергу число 1.9 і роздрукуйте чергу ще раз. Знайдіть твір елементів, що належать черзі.

Продовження таблиці 9.2

3	Створити стек строкових значень, для реалізації використовуючи одиночні списки. Реалізувати операції додавання (push) і видалення (pop) елемента з стека. Додайте в стек рядки «abc», «fx», «glc», «hi», «gogo» і роздрукуйте вміст стека. Видаліть один елемент з стека, потім додайте рядок «the end» і роздрукуйте вміст стека ще раз. Знайдіть кількість рядків в стеці, що складаються з 2 символів.
4	Створити чергу строкових значень, для реалізації використовуючи одинзв'язні списки. Реалізувати операції додавання (enqueue) і видалення (dequeue) елемента з черги. Додайте в чергу рядки «one», «two», «three», «four» і роздрукуйте вміст черги. Видаліть 2 елементи з черги, потім додайте в чергу рядок «inf» і роздрукуйте чергу ще раз. Знайдіть сумарну довжину рядків, що належать черзі, крім першого рядка черзі.
5	Створити стек цілочисельних значень, для реалізації використовуючи одинзв'язні списки. Реалізувати операції додавання (push) і видалення (pop) елемента з стека. Додайте в стек числа 1, 2, 3, 4, 5 і роздрукуйте вміст стека. Видаліть 3 елементи з стека, і роздрукуйте вміст стека ще раз. Знайдіть суму елементів стека.
6	Створити чергу речових значень, для реалізації використовуючи одинзв'язні списки. Реалізувати операції додавання (enqueue) і видалення (dequeue) елемента з черги. Додайте в чергу числа 2.1, 2.1, 5.3 і роздрукуйте вміст черги. Видаліть 1 елемент з черги, потім додайте в чергу число 4.9 і роздрукуйте чергу ще раз. Знайдіть суму елементів черги.

Продовження таблиці 9.2

7	Створити чергу речових значень, для реалізації використовуючи одинзв'язні списки. Реалізувати операції додавання (enqueue) і видалення (dequeue) елемента з черги. Додайте в чергу числа 2.2, 1.2, 2.0, 5.2 і роздрукуйте вміст черги. Видаліть 2 елементи з черги, потім додайте в чергу число 2.9 і роздрукуйте чергу ще раз. Знайдіть суму елементів черги.
8	Створити стек строкових значень, для реалізації використовуючи одинзв'язні списки. Реалізувати операції додавання (push) і видалення (pop) елемента з стека. Додайте в стек рядки «sdf», «2», «ssd4», «hello» і роздрукуйте вміст стека. Видаліть 2 елементи з стека, і роздрукуйте вміст стека ще раз. Знайдіть рядок мінімальної довжини, що належить стеку.
9	Створити чергу строкових значень, для реалізації використовуючи одинзв'язні списки. Реалізувати операції додавання (enqueue) і видалення (dequeue) елемента з черги. Додайте в чергу рядки «one», «two», «three», «four» і роздрукуйте вміст черги. Видаліть 1 елемент з черги, потім додайте в чергу рядок «five» і роздрукуйте чергу ще раз. * Знайдіть сумарну довжину всіх рядків, що належать черзі.
10	Створити стек цілочисельних значень, для реалізації використовуючи одинзв'язні списки. Реалізувати операції додавання (push) і видалення (pop) елемента з стека. Додайте в стек числа -5, 3, -4, 5 і роздрукуйте вміст стека. Видаліть один елемент з стека, додайте число 10 в стек, і надрукуйте вміст стека ще раз. * Знайдіть суму всіх позитивних елементів, що належать стеку.

Продовження таблиці 9.2

11	Створити чергу речових значень, для реалізації використовуючи одинзв'язні списки. Реалізувати операції додавання (enqueue) і видалення (dequeue) елемента з черги. Додайте в чергу числа 2.2, 3.2, 2.4, -3.2 і роздрукуйте вміст черги. Видаліть 1 елемент з черги, потім додайте в чергу число 0.04 і роздрукуйте чергу ще раз. * Знайдіть суму чисел по модулю менших 1, що належать черзі.
12	Створити стек строкових значень, для реалізації використовуючи одинзв'язні списки. Реалізувати операції додавання (push) і видалення (pop) елемента з стека. Додайте в стек рядки «Students», «of», «the», «group», «TE» і роздрукуйте вміст стека. Видаліть один елемент з стека, і роздрукуйте вміст стека ще раз. * Надрукуйте всі рядки, що починаються з малої літери «t», що належать стеку.
13	Створити чергу строкових значень, для реалізації використовуючи одинзв'язні списки. Реалізувати операції додавання (enqueue) і видалення (Dequeue) елемента з черги. Додайте в чергу рядки «one», «two», «three», «four», «five», «six», «seven» і роздрукуйте вміст черги. Видаліть 1 елемент з черги, потім додайте в чергу рядок «eight» і роздрукуйте чергу ще раз. * Знайдіть кількість рядків починаються з літер «s» або «t».
14	Створити стек строкових значень, для реалізації використовуючи одинзв'язні списки. Реалізувати операції додавання (push) і видалення (pop) елемента з стека. Додайте в стек рядки «abc», «de», «f», «g», «hi», «jk» і роздрукуйте вміст стека. Видаліть один елемент з стека, і роздрукуйте вміст стека ще раз. * Знайдіть кількість односимвольних рядків в стеці.
15	Створити чергу строкових значень, для реалізації використовуючи одинзв'язні списки. Реалізувати операції додавання (enqueue) і видалення (dequeue) елемента з черги.

Продовження таблиці 9.2

15	Додайте в чергу рядки «one», «two», «three», «four» і роздрукуйте вміст черги. Видаліть 1 елемент з черги, потім додайте в чергу рядок «five» і роздрукуйте чергу ще раз. * Знайдіть сумарну довжину рядків, що належать черзі.
16	Створити стек цілочисельних значень, для реалізації використовуючи одинзв'язні списки. Реалізувати операції додавання (push) і видалення (pop) елемента з стека. Додайте в стек числа -5, 3, -4, 5 і роздрукуйте вміст стека. Видаліть один елемент з стека, додайте в стек число 10, і роздрукуйте стек ще раз. * Знайдіть суму всіх позитивних елементів, що належать стеку.
17	Створити чергу речових значень, для реалізації використовуючи одинзв'язні списки. Реалізувати операції додавання (enqueue) і видалення (dequeue) елемента з черги. Додайте в чергу числа 46.5, 3.4, 32.4, -3.21 і роздрукуйте вміст черги. Видаліть 2 елементи з черги, потім додайте в чергу число 5.0 і роздрукуйте чергу ще раз. * Знайдіть суму елементів, по модулю великих 12, що належать черзі.
18	Створити стек строкових значень, для реалізації використовуючи одинзв'язні списки. Реалізувати операції додавання (push) і видалення (pop) елемента з стека. Додайте в стек рядки «Students», «of», «the», «group», «TE», «3» і роздрукуйте вміст стека. Видаліть один елемент з стека, і роздрукуйте вміст стека ще раз. * Надрукуйте всі рядки, які складаються з двох символів з стека.
19	Створити чергу строкових значень, для реалізації використовуючи одинзв'язні списки. Реалізувати операції додавання (enqueue) і видалення (dequeue) елемента з черги. Додайте в чергу рядки «one», «two», «three», «four», «five», «six», «seven» і роздрукуйте

Продовження таблиці 9.2

19	вміст черги. Видаліть 4 елементи з черги, потім додайте в чергу рядки «eight», «nine» і роздрукуйте чергу ще раз. * Знайдіть кількість рядків, що складаються з 4 символів.
20	Створити стек цілочисельних значень, для реалізації використовуючи одинзв'язні списки. Реалізувати операції додавання (push) і видалення (pop) елемента з стека. Додайте в стек числа 1, 4, 2 і роздрукуйте вміст стека. Додайте число 4 в стек, і роздрукуйте вміст стека ще раз. * Знайдіть кількість чисел, більших числа 3, що належить стеку.
21	Створити чергу речових значень, для реалізації використовуючи одинзв'язні списки. Реалізувати операції додавання (enqueue) і видалення (dequeue) елемента з черги. Додайте в чергу числа -2.1, 1.3, -1.34, 3.3 і роздрукуйте вміст черги. Видаліть 1 елемент з черги, потім додайте в чергу число 2.9 і роздрукуйте чергу ще раз. * Знайдіть суму негативних елементів черги.
22	Створити стек строкових значень, для реалізації використовуючи одинзв'язні списки. Реалізувати операції додавання (push) і видалення (pop) елемента з стека. Додайте в стек рядки «lll», «2», «sdf4», «bye» і роздрукуйте вміст стека. Видаліть 1 елемент з стека, і роздрукуйте вміст стека ще раз. * Знайдіть рядок максимальної довжини, що належить стеку.
23	Створити чергу строкових значень, для реалізації використовуючи одинзв'язні списки. Реалізувати операції додавання (enqueue) і видалення (dequeue) елемента з черги. Додайте в чергу рядки «one», «two», «three», «four» і роздрукуйте вміст черги. Додайте в чергу рядок «five» і роздрукуйте чергу ще раз. * Знайдіть сумарну довжину всіх рядків, що належать черзі, крім останнього рядка черзі.

Продовження таблиці 9.2

24	Створити стек цілочисельних значень, для реалізації використовуючи одинзв'язні списки. Реалізувати операції додавання (push) і видалення (pop) елемента з стека. Додайте в стек числа 5, 3, 44, 555 і роздрукуйте вміст стека. Видаліть 2 елементи з стека, додайте числа 20 і 30 в стек, і надрукуйте вміст стека ще раз. * Знайдіть кількість позитивних двозначних чисел, що належать стеку.
25	Створити чергу речових значень, для реалізації використовуючи одинзв'язні списки. Реалізувати операції додавання (enqueue) і видалення (dequeue) елемента з черги. Додайте в чергу числа -2.2, 5.5, 4.3, -4.5 і роздрукуйте вміст черги. Видаліть 1 елемент з черги і роздрукуйте чергу ще раз. * Знайдіть суму чисел по модулю більше 4, що належать черзі.
26	Створити стек строкових значень, для реалізації використовуючи одинзв'язні списки. Реалізувати операції додавання (push) і видалення (pop) елемента з стека. Додайте в стек рядки «Student», «of», «the», «OSAT» і роздрукуйте вміст стека. Видаліть один елемент з стека, додайте рядок «ONAT» і роздрукуйте вміст стека ще раз. * Порахуйте кількість рядків, що складаються не менше ніж з трьох символів, що належать стеку.
27	Створити чергу строкових значень, для реалізації використовуючи одинзв'язні списки. Реалізувати операції додавання (enqueue) і видалення (dequeue) елемента з черги. Додайте в чергу рядки «one», «two», «three», «four», «five», «six», і роздрукуйте вміст черги. Видаліть 2 елементи з черги, потім додайте в чергу рядок «seven» і роздрукуйте чергу ще раз. * Знайдіть кількість рядків починаються з літер «f» або «t».

Продовження таблиці 9.2

28	Створити стек цілочисельних значень, для реалізації використовуючи одинзв'язні списки. Реалізувати операції додавання (push) і видалення (pop) елемента з стека. Додайте в стек числа 1, 2, -3, 4 і роздрукуйте вміст стека. Видаліть один елемент з стека, і роздрукуйте вміст стека ще раз. * Знайдіть максимальний елемент, що належить стеку.
29	Створити чергу речових значень, для реалізації використовуючи одинзв'язні списки. Реалізувати операції додавання (enqueue) і видалення (dequeue) елемента з черги. Додайте в чергу числа -2.0, 2.0, 2.1, -2.1, 3.7 і роздрукуйте вміст черги. Видаліть 2 елементи з черги, потім додайте в чергу число 1.1 і роздрукуйте чергу ще раз. * Знайдіть твір позитивних елементів, що належать черзі.
30	Дано дві непусті черзі; адреси початку і кінця першої рівні P1 і P2, а другий - P3 і P4. Черги містять однакову кількість елементів. Об'єднати черги в одну, в якій елементи вихідних черг чергуються (починаючи з першого елемента першої черги). Вивести покажчики на початок і кінець отриманої черзі. Операції виділення і звільнення пам'яті не використовувати.

Завдання 3. Написати програму відповідно до індивідуального завдання.

Таблиця 9.3. Варіанти до завдання 3.

№ варіанту	Завдання
1	Дано дві непусті черзі; адреси початку і кінця першої рівні P1 і P2, а другий - P3 і P4. Елементи кожної з черг впорядковані за зростанням (в напрямку від початку черги до кінця).

Продовження таблиці 9.3

1	<p>Об'єднати черги в одну зі збереженням упорядкованості елементів. Вивести покажчики на початок і кінець отриманої черзі. Операції виділення і звільнення пам'яті не використовувати, поля з даними (Data) не зраджувати.</p>
2	<p>Арифметичний вираз можна представити в зворотній польській записи, де знаки операції слідують за операндами (а не ставляться між ними, як у звичайній запису виразів). Зворотній польський запис не вимагає дужок. Наприклад, висловом «1 + 2» відповідає запис «1 2 +», висловом «1 + 2 * 3» запис «1 2 3 * +» (спочатку множаться 2 на 3, а потім 1 складається з результатом), «(2 + 3) * (3 - 1)» записується як «2 3 + 3 1 - * ». Здається рядок - вираз в зворотній польській записи (числа і знаки +, -, * розділені пробілами). Використовуючи стек, обчисліть значення виразу. Підказка: потрібно послідовно перебрати всі числа і знаки з рядка, числа потрібно заносити в стек, а як зустрінеться знак операції, виймати 2 числа з стека, застосовувати до них поточну операцію, а результат заносити в стек.</p>
3	<p>Дана послідовність дужок виду «(», «)», «[», «]», «{», «}». Правильна дужкова послідовність називаються порожня послідовність, а також «(P)», «[P]», «{P}», де P - деяка правильна послідовність. Наприклад «{ } () []» і «{[] [() ()} ()» - правильні дужкові послідовності, а «(]», «[()» і «({)» - неправильні. Визначте є заданий рядок правильним Дужковий виразом. Підказка: обробіть по черзі всі символи вхідного рядка, поміщаючи відкривають дужки в стек, а для закривають дужок виймайте відкриває дужку з стека і перевіряйте, чи відповідають вони один одному.</p>

Продовження таблиці 9.3

4	<p>Реалізуйте чергу, використовуючи два стека. Підказка: стек інвертує порядок елементів (тобто якщо додати 1 2 3 4 5, виймаючи елементи, отримаємо 5 4 3 2 1), тому якщо елементи пройдуть два стека, то відновиться їх вихідний порядок. При приміщенні в чергу розміщуйте елемент в перший стек, при добуванні елементу з черги, виймайте елемент з другого стека, а якщо при цьому другий стек виявиться порожнім, перенесіть все елементи з першого стека в другій (діставайте елементи першого стека, поки вони є, і переносите в другій).</p>
5	<p>Дано число $N (> 0)$ і дві непусти черзі; адреси початку і кінця першої рівні $P1$ і $P2$, а другий - $P3$ і $P4$. Перемістити N початкових елементів першої черги в кінець другої черги. Якщо перша черга містить менше N елементів, то перемістити з першої черги в другу все елементи. Вивести нові адреси початку і кінця першої, а потім другої черги (для порожній черзі двічі вивести nil). Операції виділення і звільнення пам'яті не використовувати.</p>
6	<p>Дан набір з 10 чисел. Створити дві черги: перша повинна містити числа з вихідного набору з непарними номерами (1, 3, ..., 9), а друга - з парними (2, 4, ..., 10); порядок чисел в кожній черзі повинен збігатися з порядком чисел в початковому наборі. Вивести покажчики на початок і кінець першої, а потім другої черги.</p>
7	<p>Дан набір з 10 чисел. Створити дві черги: перша повинна містити всі непарні, а друга - всі парні числа з вихідного набору (порядок чисел в кожній черзі повинен збігатися з порядком чисел в початковому наборі). Вивести покажчики на початок і кінець першої, а потім другої черги (одна з черг може виявитися марною; в цьому випадку вивести для неї дві константи nil).</p>

Продовження таблиці 9.3

8	Дано покажчики P1 і P2 на початок і кінець непорожній черги. Витягувати з черги елементи, поки значення початкового елемента черги не стане парних, і виводити значення витягнутих елементів (якщо чергу не містить елементів з парними значеннями, то витягти всі її елементи). Вивести також нові адреси початку і кінця черги (для порожній черзі двічі вивести nil). Після вилучення елементів з черги звільняти пам'ять, яку вони займали.
9	Дано дві черги; адреси початку і кінця першої рівні P1 і P2, а другий - P3 і P4 (якщо чергу є марною, то відповідні адреси рівні nil). Перемістити всі елементи першої черги (в порядку від початку до кінця) в кінець другої черги і вивести нові адреси початку і кінця другої черги. Операції виділення і звільнення пам'яті не використовувати.
10	Дано дві непусти черзі; адреси початку і кінця першої рівні P1 і P2, а другий - P3 і P4. Переміщати елементи з початку першої черги в кінець другої, поки значення початкового елемента першої черги не стане парних (якщо перша черга не містить парних елементів, то перемістити з першої черги в другу все елементи). Вивести нові адреси початку і кінця першої, а потім другої черги (для порожній черзі двічі вивести nil). Операції виділення і звільнення пам'яті не використовувати.

9.3 Питання для перевірки знань

1. Дайте визначення поняттю "колекція".
2. Назвіть переваги використання колекцій.
3. Які дані можуть зберігати колекції?
4. Яка ієрархія колекцій?
5. Що ви знаєте про колекції типу List?

6. Що ви знаєте про колекції типу Set?
7. Що ви знаєте про колекції типу Queue?
8. Що ви знаєте про колекції типу Map, в чому їх принципова відмінність?
9. Назвіть основні реалізації List, Set, Map.
10. Які реалізації SortedSet ви знаєте і в чому їх особливість?
11. У чому відмінності / подібності List і Set?
12. Що різного / спільного у класів ArrayList і LinkedList, коли краще використовувати ArrayList, а коли LinkedList?
13. В яких випадках розумно використовувати масив, а не ArrayList?
14. Чим відрізняється ArrayList від Vector?
15. Що ви знаєте про реалізацію класів HashSet і TreeSet?
16. Чим відрізняються HashMap і TreeMap? Як вони влаштовані і працюють? Що згодом доступу до об'єктів, які залежності?
17. Що таке Hashtable, чим вона відрізняється від HashMap? На сьогоднішній день вона deprecated, як все-таки використовувати потрібну функціональність?
18. Що буде, якщо в Map покласти два значення з однаковим ключем?
19. Як задається порядок проходження об'єктів в колекції, як впорядкувати колекцію?
20. Дайте визначення поняттю "итератор".
21. Яку функціональність представляє клас Collections?
22. Як одержати не модифікуються колекцію?
23. Які колекції синхронізовані?
24. Як отримати синхронізовану колекцію з не синхронізовані?
25. Як отримати колекцію тільки для читання?
26. Чому Map не успадковується від Collection?
27. У чому різниця між Iterator і Enumeration?
28. Як реалізований цикл foreach?

29. Чому немає методу `iterator.add ()` щоб додати елементи в колекцію?
30. Чому в класі `iterator` немає методу для отримання наступного елемента без пересування курсора?
31. У чому різниця між `Iterator` і `ListIterator`?
32. Які є способи перебору всіх елементів `List`?
33. У чому різниця між `fail-safe` і `fail-fast` властивостями?
34. Що робити, щоб не виникло виключення `ConcurrentModificationException`?
35. Що таке стек і чергу, розкажіть в чому їх відмінності?
36. У чому різниця між інтерфейсами `Comparable` і `Comparator`?

ЛАБОРАТОРНА РОБОТА 10

Тема: ПАКЕТ JAVA.IO, ФАЙЛИ

Мета: 1) Вивчити особливості роботи з файлами у Java. 2) Навчитися створювати програми обробки даних використовуючи файли.

10.1 Практична частина

Приклад:

```
String fileName = "d: \\ file.txt";
FileWriter fw = null;
BufferedWriter bw = null;
FileReader fr = null;
BufferedReader br = null;
// Рядок, яка буде записана в файл
String data = "Some data to be written and readed \n";
try {
    fw = new FileWriter (fileName);
    bw = new BufferedWriter (fw);
    System.out.println ("Write some data to file:" + fileName);
    // Кілька разів записати рядок
    for (int i = (int) (Math.random () * 10); - i > = 0;) bw.write
(data);
    bw.close ();
    fr = new FileReader (fileName);
    br = new BufferedReader (fr);
    String s = null;
    int count = 0;
    System.out.println ("Read data from file:" + fileName);
    // Зчитувати дані, відображаючи на екран
    while ((s = br.readLine ()) != null)
        System.out.println ("row" + ++ count + "read:" + s);
    br.close ();
} Catch (Exception e) {
    e.printStackTrace ();
}
```

10.2 Самостійна частина

Завдання 1. Створити текстовий файл з інформацією. Організувати перегляд вмісту файлу. Організувати читання і обробку даних з файлу відповідно до індивідуального завдання (Таблиця 10.1). Для обробки даних із файлів використовувати класи. Зберегти отримані результати в новий текстовий файл.

Таблиця 10.1. Індивідуальні завдання до завдання 1.

№ варіанту	Завдання
1	"Людина": прізвище; ім'я; по батькові; стать; національність; зріст; вага; дата народження (Рік, місяць число); номер телефону; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира). Вивести відомості про саму молодшу людину.
2	«Школяр»: прізвище; ім'я; по батькові; стать; національність; зріст; вага; дата народження (рік, місяць число); номер телефону; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира); школа; клас. Вивести відомості про всіх учнів п'ятих класів.
3	«Студент»: прізвище; ім'я; по батькові; стать; національність; зріст; вага; дата народження (рік, місяць число); номер телефону; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира); ВНЗ; курс; група; середній бал; спеціальність. Вивести відомості про всіх студентів у яких середній бал нижче 70 балів.
4	«Покупець»: прізвище; ім'я; по батькові; стать; національність; зріст; вага; дата народження (рік, місяць число); номер телефону; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира); номер кредитної картки; банківського

Продовження таблиці таблиці 10.1.

	рахунку. Вивести дані про покупців з міста Одеси.
5	«Пацієнт»: прізвище; ім'я; по батькові; стать; національність; зріст; вага; дата народження (рік, місяць число); номер телефону; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира); номер лікарні; відділення; номер медичної карти; діагноз; група крові. Вивести дані про пацієнтів з 18 відділення.
6	«Власник автомобіля»: прізвище; ім'я; по батькові; номер телефону; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира) марка автомобіля; номер автомобіля; номер техпаспорта. Вивести дані про автомобілі марки "Ваз".
7	«Військовослужбовець»: прізвище; ім'я; по батькові; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира); національність; дата народження (рік, місяць число); посаду; звання. Вивести дані про військовослужбовців у званні "лейтенант".
8	«Робочий»: прізвище; ім'я; по батькові; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира); національність; дата народження (рік, місяць число); № цеху; табельний номер; освіта; рік надходження на роботу. Вивести дані про робітників, які надійшли на роботу в 2010 році.
9	«Власник телефону»: прізвище; ім'я; по батькові; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира); № телефону. Вивести дані про власників телефону номер, якого починається на 720.

Продовження таблиці таблиці 10.1.

10	«Абітурієнт»: прізвище; ім'я; по батькові; стать; національність; дата народження (рік, місяць число); домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира); оцінки по іспитів; прохідний бал. Вивести дані про абітурієнтів, прохідний бал яких дорівнює більше 4.
11	«Держава»: назва держави; столиця; Державна мова; населення; площа території; грошова одиниця; державний лад; Глава держави. Вивести дані про держави, населення яких більше 20 млн. Жителів.
12	«Автомобіль»: марка; колір; серійний номер; реєстраційний номер; рік випуску; рік техогляду; ціна. Вивести дані про автомобілі, яким більше 2 років.
13	«Товар»: найменування; вартість; термін зберігання; сорт; дата випуску; термін придатності. Вивести дані про товари термін придатності яких закінчується в цьому році.
14	«Кінострічка»: назва; режисер (прізвище та ім'я); рік виходу; країна; вартість; дохід; прибуток. Вивести дані про фільми режисера Єжи Гофмана.
15	«Рейс»: марка автомобіля; номер автомобіля; пункт призначення; вантажопідйомність (в тонах); вартість одиниці вантажу; загальна вартість вантажу. Вивести дані про автомобілі, вантажопідйомність яких більше 2 тонн.
16	«Книга»: назва; автор (прізвище та ім'я); рік виходу; видавництво; собівартість; ціна; прибуток. Вивести дані про книги авторів, прізвище яких починається на букву "К".
17	«Будівля»: адреса; тип будівлі; кількість поверхів; кількість квартир; термін експлуатації; термін до капітального ремонту (25 років - термін експлуатації). Вивести дані про будівлі термін експлуатації, яких більше 50 років.

Продовження таблиці таблиці 10.1.

18	«Програміст»: прізвище; ім'я; по батькові; стать; національність; дата народження (рік, місяць, число); освіта; номер телефону. Вивести відомості про програмістів, яким менше 25 років.
19	«Вчений»: прізвище; ім'я; по батькові; стать; національність; дата народження (рік, місяць, число); вчена ступінь, посада, номер телефону; домашня адреса (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира). Вивести відомості про вчених кандидатів технічних наук.
20	«Пенсіонер»: прізвище; ім'я; по батькові; стать; національність; дата народження (рік, місяць, число); номер телефону; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира). Вивести відомості про всіх пенсіонерів, які на пенсії більше 5 років.
21	«Футболіст»: прізвище; ім'я; по батькові; стать; національність; зріст; вага; дата народження (рік, місяць, число); номер телефону; назва команди; номер в команді; амплуа; результативність (кількість голів); кількість ігор. Вивести відомості про футболістів, які провели за свою команду більше 50 матчів.
22	«Манекенщица»: прізвище; ім'я; по батькові; стать; національність; зріст; вага; дата народження (рік, місяць, число); домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира). Вивести дані про наймолодшу манекенщицю.
23	«Міжнародна компанія»: назва; інтернет сайт; адреса головного офісу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира); тривалість перебування на світовому ринку; кількість співробітників; кількість філій в Європі. Вивести міжнародні компанії, кількість співробітників у яких більше 10000.

Продовження таблиці таблиці 10.1.

24	«Охоронець»: прізвище; ім'я; по батькові; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира); дата народження (рік, місяць, число). Вивести дані про старшого охоронця ".
25	«Зоопарк»: Назва тварини; кількість виду; адреса зоопарку (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира); загальна кількість тварин, кількість працівників. Вивести відомості про зоопарки, в яких є уссурійські тигри.
26	«Програмне забезпечення»: назва; назва компанії виробника; рік виходу; ціна. Вивести дані про програмне забезпечення, яке дорожче 2000 гривень.
27	«Мультфільм»: назва; режисер (прізвище та ім'я); рік виходу; країна; вартість; дохід; прибуток. Вивести дані про мультфільми компанії "Walt Disney".
28	«Баскетболіст»: прізвище; ім'я; по батькові; стать; національність; зріст; вага; дата народження (рік, місяць число); номер телефону; назва команди; номер в команді; ампула; результативність (кількість очок); кількість ігор. Вивести відомості про баскетболістів, які закинули за свою команду більше 150 очей.
29	«Область»: назва області; обласний центр; населення; площа території; губернатор. Вивести дані про області, населення яких менше 2 млн. жителів.
30	«Мотоцикл»: марка; колір; серійний номер; реєстраційний номер; рік випуску; рік техогляду; ціна. Вивести дані про мотоцикли марки "Harley Davidson".

Завдання 2. Організувати переглядання вмісту файлу. Організувати читання та обробку даних з файлу відповідно до індивідуального завдання.

Для обробки даних із файлів використовувати класи. Зберегти отримані результати в новий текстовий файл.

Таблиця 10.2. Індивідуальні завдання до завдання 2.

№ варіанту	Завдання
1	<p>Створити бінарний файл зі списком про студентів:</p> <ul style="list-style-type: none"> - прізвище та ініціали студентів; - номер групи; - успішність (масив з трьох дисциплін за 100-бальною системою); - розмір стипендії. <p>Вивести список студентів, які мають з усіх предметів позитивні оцінки і роздрукувати всі відомості про них.</p>
2	<p>Створити бінарний файл, компонентами якого є структура, містить наступні поля:</p> <ul style="list-style-type: none"> - найменування товару; - вартість одиниці товару; - кількість кожного товару. <p>Визначити загальну вартість товару, запропонованого для реалізації, і його середню ціну.</p>
3	<p>Створити бінарний файл, компонентами якого є структура, містить наступні поля:</p> <ul style="list-style-type: none"> - найменування товару; - вартість товару; - одиниця виміру. <p>Визначити найдорожчий товар на складі та відомості про нього.</p>
4	<p>Створити бінарний файл, компонентами якого є структура з наступними полями:</p> <ul style="list-style-type: none"> - список працівників заводу; - посаду працівника;

Продовження таблиці 10.2.

	<p>- заробітна плата працівника.</p> <p>Визначити середню зарплату слюсарів - працівників заводу та їх кількість.</p>
5	<p>Створити бінарний файл з інформацією про запропоновані до реалізації моніторів:</p> <ul style="list-style-type: none"> - назва фірми; - розмір по діагоналі; - вартість. <p>Визначити середню ціну моніторів, розміром не менше 19 дюймів і роздрукувати відомості про них.</p>
6	<p>Створити бінарний файл з інформацією про запропоновані до реалізації телевізорів:</p> <ul style="list-style-type: none"> - назва фірми; - розмір по діагоналі; - вартість. <p>Визначити кількість телевізорів фірми «Samsung», розміром більше 32 дюйма і роздрукувати відомості про них.</p>
7	<p>Створити бінарний файл з інформацією про наявність комп'ютерів для продажу:</p> <ul style="list-style-type: none"> - назва комп'ютера; - частота процесора; - обсяг оперативної пам'яті; - обсяг жорсткого диска; - тип монітора; - розмір монітора; - ціна.

Продовження таблиці 10.2.

	<p>Визначити комп'ютер з найбільшою продуктивністю: з найбільшою швидкістю процесора при обсязі оперативної пам'яті не менше 2 Гбайт і надрукувати його характеристики.</p>
8	<p>Створити бінарний файл з інформацією про наявність комп'ютерів для продажу:</p> <ul style="list-style-type: none"> - назва комп'ютера; - частота процесора; - обсяг оперативної пам'яті; - обсяг жорсткого диска; - ціна. <p>Визначити загальну вартість запропонованих до продажу комп'ютерів, у яких частота процесора більше 2 ГГц / сек фірми «Asus» і надрукувати інформацію про них.</p>
9	<p>Створити бінарний файл з інформацією про футболістів клубу:</p> <ul style="list-style-type: none"> - прізвище; - амплуа; - вік; - кількість ігор; - дата проведення гри; - кількість голів. <p>Визначити кращого форварда, який забив найбільше голів за останні 5 ігор.</p>
10	<p>Створити бінарний файл з інформацією про автора та його книг:</p> <ul style="list-style-type: none"> - автор; - назва книги; - тираж; - ціна примірника книги;

Продовження таблиці 10.2.

10	<p>- рік видання.</p> <p>Роздрукувати інформацію про всіх авторів, які в своїх назвах використовують ключове слово «Вбивство».</p>
11	<p>Створити бінарний файл з інформацією про працівників підприємства:</p> <ul style="list-style-type: none"> - прізвище; - вік; - освіту; - посаду; - стать. <p>Роздрукувати інформацію про всіх працівників жіночої статі без вищого освіти, яким в цьому році необхідно оформляти пенсію.</p>
12	<p>Створити бінарний файл з інформацією про виконавців джазової музики:</p> <ul style="list-style-type: none"> - виконавець; - назва альбому; - тираж; - рік випуску альбому; - вартість альбому. <p>Роздрукувати відомості про виконавця, який в поточному році випустив альбом кількістю понад 1000 примірників за ціною не менше 50 грн.</p>
13	<p>Створити бінарний файл з інформацією про співробітників фірми:</p> <ul style="list-style-type: none"> - прізвище; - посада; - зарплата; - дата народження. <p>Вивести відомості про співробітників, у яких зарплата вище середньої і вік яких менше 30-ти років.</p>

Продовження таблиці 10.2.

14	<p>Створити бінарний файл з інформацією про легкові автомобілі:</p> <ul style="list-style-type: none"> - марка автомобіля; - виробник; - тип; - рік випуску; - вартість. <p>Вивести відомості про всі автомобілі, термін випуску яких не менше 5 років і вироблених не в Китаї.</p>
15	<p>Створити бінарний файл з інформацією про працівників телестудії:</p> <ul style="list-style-type: none"> - прізвище; - посаду; - освіту; - дата прийому на роботу; - стать. <p>Вивести відомості про працівників телестудії, які працюють на посадах інженерів, але які не мають вищої освіти.</p>
16	<p>Створити бінарний файл з інформацією про автомобілі:</p> <ul style="list-style-type: none"> - марка автомобіля; - виробник; - рік випуску; - тип двигуна; - загальний пробіг в км; - ціна. <p>Вивести відомості про автомобілі, у яких пробіг становить менше 100 км з ціною менше 200000 грн.</p>

Продовження таблиці 10.2.

17	<p>Створити бінарний файл з інформацією про працівників заводу:</p> <ul style="list-style-type: none"> - прізвище; - посаду; - стать; - рік народження. <p>Вивести відомості про працівників заводу як для чоловіків, так і для жінок, яким в поточному році повинен оформляти пенсію.</p>
18	<p>Створити бінарний файл з інформацією про продовольчі товари, що зберігаються на складі:</p> <ul style="list-style-type: none"> - найменування товару; - рік надходження на склад; - кількість; - виробник; - ціна. <p>Визначити суму втрат при списанні товарів, термін зберігання яких перевищує 5 років.</p>
19	<p>Створити бінарний файл з інформацією про озера:</p> <ul style="list-style-type: none"> - найменування озера; - країна розташування озера; - глибина озера; - солоність озера в процентах. <p>Вивести відомості про озера, глибина яких менше 50 м, а солоність більше 20%.</p>
20	<p>Створити бінарний файл з інформацією про населені пункти:</p> <ul style="list-style-type: none"> - назва населеного пункту; - кількість населення; - відстань до поштового відділення в км. <p>Обчислити середній кілометраж, який проходить житель населеного пункту до поштового відділення.</p>

Продовження таблиці 10.2.

21	<p>Створити бінарний файл з інформацією про річки в різних регіонах країни:</p> <ul style="list-style-type: none"> - найменування річки; - довжина річки в км; - середня глибина в м. <p>Визначити загальну довжину річок, у яких глибина менше 50 м.</p>
22	<p>Створити бінарний файл з інформацією про клієнтів кабельного телебачення:</p> <ul style="list-style-type: none"> - прізвище клієнта; - вартість базового пакету в місяць; - вартість соціального пакета на місяць; - кількість місяців оплати за користування кабельним телебаченням. <p>Визначити різницю в оплаті за ту кількість місяців, яку клієнтом було оплачено, якби клієнт захотів перейти від базового пакета до соціального.</p>
23	<p>Створити бінарний файл з інформацією про футболістів клубу:</p> <ul style="list-style-type: none"> - прізвище; - амплуа; - рік народження; - кількість ігор, проведених гравцем; - кількість голів, забитих гравцем. <p>Вивести відомості про футболістів не старше 20 років і які забили не менше 5 голів за сезон.</p>
24	<p>Створити бінарний файл з інформацією про книги, присвячених програмуванню:</p> <ul style="list-style-type: none"> - автор; - назва книги;

Продовження таблиці 10.2.

24	<ul style="list-style-type: none"> - кількість тиражу; - вартість. <p>Роздрукувати відомості про кількість книг і їх загальну вартість, якщо книга присвячена програмуванню на мові C ++.</p>
25	<p>Створити бінарний файл з інформацією про хвороби і про ліки:</p> <ul style="list-style-type: none"> - назва хвороби; - назва ліків; - вартість ліків. <p>Роздрукувати відомості про ліки, здатних лікувати грип чи ГРЗ.</p>
26	<p>Створити бінарний файл з інформацією про предмети, що читаються на різних курсах:</p> <ul style="list-style-type: none"> - назва предмета; - курс, на якому читається даний предмет; - число годин, що відводяться під даний курс. <p>Роздрукувати інформацію про предмети, що читаються на другому курсі.</p>
27	<p>Створити бінарний файл з інформацією про бібліотеку та її читачів:</p> <ul style="list-style-type: none"> - прізвище читача; - назва книги, взятої читачем; - дата видачі книги; - термін, на який видана книга; - реальна дата здачі читачем взятої книги. <p>Роздрукувати відомості про читачів, які не повернули книгу в зазначений термін.</p>
28	<p>Створити бінарний файл з інформацією про молочну продукцію, що надійшла в магазин:</p> <ul style="list-style-type: none"> - назва молочного продукту;

Продовження таблиці 10.2.

28	<ul style="list-style-type: none">- дата надходження товару в магазин;- термін зберігання продукту за накладною;- дата перевірки наявності продукту в магазині. <p>Визначити перелік продуктів, які зберігаються в магазині більше відведеного терміну.</p>
29	<p>Створити бінарний файл з інформацією про книги в бібліотеці:</p> <ul style="list-style-type: none">- назва книги;- жанр: для дітей, для дорослих, для закоханих, для зневірених. <p>Визначити, який жанр викликає у читачів найбільший інтерес.</p>
30	<p>Створити бінарний файл з інформацією про кінофільми:</p> <ul style="list-style-type: none">- назва фільму;- скільки серій;- тривалість однієї серії;- по яких днях проходить серіал. <p>Визначити, скільки днів буде демонструватися самий багатосерійний фільм.</p>

10.3 Питання для перевірки знань

1. Які існують види потоків введення / виведення?
2. Назвіть основні предки потоків введення / виведення.
3. Що спільного й відмінного в наступних потоках: `InputStream`, `OutputStream`, `Reader`, `Writer`?
4. Що ви знаєте про `RandomAccessFile`?
5. Які є режими доступу до файлу?
6. У яких пакетах лежать класи-потоки?
7. Що ви знаєте про класи-надбудови?
8. Який клас-надбудова дозволяє читати дані з вхідного байтового потоку в форматі примітивних типів даних?

9. Який клас-надбудова дозволяє прискорити читання / запис за рахунок використання буфера?
10. Які класи дозволяють перетворити байтові потоки в символні і назад?
11. Який клас призначений для роботи з елементами файлової системи (ЕФС)?
12. Який символ є роздільником при вказівці шляху до ЕФС?
13. Як вибрати все ЕФС певного каталогу за критерієм (наприклад, з певним розширенням)?
14. Що ви знаєте про інтерфейс `FilenameFilter`?
15. Що таке серіалізація?
16. Які умови правильної серіалізації об'єкта?
17. Які класи дозволяють архівувати об'єкти?

ЛАБОРАТОРНА РОБОТА 11

Тема: ПАКЕТИ JAVA.AWT ТА JAVA.SWING

Мета: 1) Вивчити особливості роботи з формами у Java. 2) Навчитися створювати програми обробки даних використовуючи форми.

11.1 Практична частина

Створення фреймів (вікон)

Вікно верхнього рівня (таке що не містить в середині іншого вікна) в мові Java називається фреймом (frame - каркас). В бібліотеці AWT для цього вікна призначений клас Frame, а в бібліотеці Swing — JFrame.

Наступний код демонструє, як можна створити фрейм розміром 300x200 пікселів:

```
import javax.swing.JFrame;
public class SimpleFrame
{
    public static void main(String[] args)
    {
        JFrame frame = new JFrame(); // створити фрейм
        frame.setSize(300, 200); // задаємо ширину і висоту
фрейму
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); //
задаємо дії при закритті фрейму
        frame.setVisible(true); // показати фрейм на екрані
(зробити видимим)
    }
}
```

Проте розміщувати увесь код в main() поганий стиль, оскільки, наприклад, коли доведеться додавати нові фрейми та інші елементи, код може стати доволі заплутаним. Крім того метод main статичний і потрібно враховувати деякі особливості роботи у статичному контексті. Тому в main()

часто намагаються залишити лише саме необхідне. І взагалі, якщо якийсь графічний елемент потребує значного коду, то бажано роботу з ним розмістити окремо. В нашому випадку усю роботу з фреймом можна доручити класу, який розширюватиме клас JFrame.

Наступний переписаний код робить теж саме, що і попередній. На перший погляд може здатися, що такий спосіб є доволі незручним, насправді ж у великих проектах - це виправдовується.

```
import javax.swing.*;
public class SimpleFrame
{
    public static void main(String[] args)
    {
        MyFrame frame = new MyFrame();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
class MyFrame extends JFrame
{
    public MyFrame()
    {
        setSize(FRAME_WIDTH, FRAME_HEIGHT);
    }

    public static final int FRAME_WIDTH = 350;
    public static final int FRAME_HEIGHT = 200;
}
```

Дана програма складається з двох класів, хоча метод MyFrame() можна б було розмістити і в класі SimpleFrame. Проте, на думку ряду програмістів, краще відділяти клас який запускає програму на виконання від класу, в якому описується інтерфейс користувача. Одною з причин такого є більш краща читабельність коду програми.

Розглянемо детальніше нашу програму.

Перший рядок `import javax.swing.*`; робить доступними класи бібліотеки Swing. Далі в методі `main()` ми задаємо розмір нашому фреймові викликаючи конструктор класу `MyFrame`. Метод `setSize()` встановлює розмір вікна 350x200 пікселів. По замовчуванню фрейм буде 0x0 пікселів. Рядок `frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE)`; — вказує програмі, що потрібно зробити, якщо користувач захоче закрити фрейм. В даному випадку програма просто завершить свою роботу. І, насамкінець, ми робимо наш фрейм видимим викликаючи метод `setVisible()`.

Робота з фреймами

Фрейм - це контейнер, в який поміщають всі інші компоненти (меню, кнопки, прапорці та інші елементи графічного елементу). Сам клас `JFrame`, який реалізує фрейм складається з чотирьох областей (`pane`), що накладаються одна на одну: коренева область (`root pane`), область шару (`layered pane`), прозора область (`glass pane`) та область вмісту (`content pane`). Перші три застосовують для створення та обслуговування меню. Для роботи з графічними елементами застосовується область вмісту, в яку і додають компоненти. Додати компонент можна наступним чином:

```
Container contentPane = frame.getContentPane();
Component c=....;
contentPane.add (c)

// з версії JDK 1.5 реалізований метод JFrame.add()
// що переадресовує виклик методу області вмісту
contentPane.add()
// і тепер можна просто писати: add (c)
```

Щоправда напряму у фреймі не прийнято вставляти компоненти. Для цього використовується спеціальний компонент-контейнер - панель (`panel`), що добавляється до фрейму. Після цього у панель можна додавати різні графічні компоненти.

Для початку спробуємо намалювати прямокутник. Щоб додати відповідну панель, в якій буде здійснюватись малювання, необхідно:

Визначити клас, що розширює клас JPanel;

Замістити (перевизначити) в цьому класі метод paintComponent().

```
class MyPanel extends JPanel{
    public void paintComponent(Graphics g){
        .....// код, що здійснює малювання
    }
}
```

Програмний код:

```
import javax.swing.*;
import java.awt.*;
public class SimpleFrame2
{
    public static void main(String[] args)
    {
        MyFrame frame = new MyFrame();

        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}

class MyFrame extends JFrame
{
    public MyFrame()
    {
        setSize(FRAME_WIDTH, FRAME_HEIGHT);
        PaintPanel panel=new PaintPanel();
        add(panel);
    }

    public static final int FRAME_WIDTH = 350;
    public static final int FRAME_HEIGHT = 200;
}

class PaintPanel extends JPanel
{
    @Override // вказує, що даний метод заміщується (не
обов'язково)
    public void paintComponent(Graphics g)
```

```

    {
        super.paintComponent (g);
        g.drawString("Малюємо прямокутник:", 10, 20);
        g.drawRect(10, 40, 300, 100);
    }
}

```

Обробка подій

Натиснення кнопки, закриття вікна, клацання мишкою – все це є прикладами подій, які отримує операційна система і передає відповідній програмі на обробку. Програміст повинен передбачити як потрібно обробляти дані події. Розглянемо як обробка подій реалізовується в Java.

В Java запропонована, так звана, модель делегування подій (event delegation model). Джерело події (event source) породжує подію, після чого вона передається в обробник подій (event listener – дослівно слухач події). При цьому будь-який об'єкт може бути призначеним як обробник деякої події. Така модель доволі гнучка, оскільки кожен програміст може вибрати зручний для нього спосіб обробки події (де саме її обробляти), проте інколи текст програми може бути дещо заплутаним для тих хто не звик до такої моделі.

Інформація про подію інкапсулюється у об'єкті події (event object). Всі події описуються підкласами `java.util.EventObject`. Як приклади, можна навести підкласи `ActionEvent` та `WindowEvent`. Перші об'єкти породжують кнопки, а другі вікна.

Джерела подій містять методи, які дозволяють зв'язати його з обробниками подій. Коли виникає подія, джерело повідомляє про неї усіх зареєстрованих обробників. Обробники подій на основі інформації у об'єкті події визначає як реагувати на ту чи іншу подію.

При розробці графічного інтерфейсу розробнику необхідно здійснити наступне:

- Створити клас, який оброблятиме подію чи ряд подій і, який реалізуватиме відповідний інтерфейс

- Створити джерела подій (вікно, кнопки, смугу прокрутки тощо)
- Зв'язати джерела подій з обробниками подій
- Сказане демонструє наступний фрагмент програми:

```

ActionListener listener = . . .; // створити обробник подій
JButton button = new JButton("Ok"); // створюємо кнопку
button.addActionListener(listener); // зв'язуємо кнопку з
обробником подій

```

Клас, який реалізовуватиме інтерфейс `ActionListener` повинен мати метод `actionPerformed()` який в якості параметру отримуватиме об'єкт `ActionEvent`.

```

class MyListener implements ActionListener
{
    . . .
    public void actionPerformed(ActionEvent event)
    {
        // тут відбувається реакція на натиснення кнопки
        . . .
    }
}

```

Реалізувати такий механізм можна декількома способами. Розглянемо поступово на прикладах все вище сказане.

```

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JButton;
import java.awt.event.*;
import javax.swing.JOptionPane;

public class EventTest {

    public static void main(String[] args) {

        SimpleFrame frame = new SimpleFrame();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}

```

```

    }
    class SimpleFrame extends JFrame
    {
        public SimpleFrame()
        {
            setSize(DEFAULT_WIDTH, DEFAULT_HEIGHT); // задаємо розміри
фрейму
            JPanel panel = new JPanel(); // створюємо панель
            this.add(panel); // додаємо панель у фрейм
            JButton button = new JButton("OK!"); // створюємо кнопку
            ОК!
            panel.add(button); // додаємо кнопку на панель
            ActionListener buttonAction = new ActionListener(); //
створюємо екземпляр обробника подій
            button.addActionListener(buttonAction); //
зв'язуємо обробника подій з кнопкою "OK!"
        }
        public static final int DEFAULT_WIDTH = 300;
        public static final int DEFAULT_HEIGHT = 200;
    }
    class EventListener implements ActionListener{
        @Override // не обов'язково; вказуємо, що ми перевизначаємо
метод actionPerformed
        public void actionPerformed(ActionEvent event)
        {
            // Виводимо повідомлення у діалоговому вікні
            // зверніть увагу, що для цього ми створюємо новий фрейм
            JOptionPane.showMessageDialog(new JFrame(), "Ви натиснули
кнопку <<OK!>>");
        }
    }
}

```

Можна також зробити, що один і той же обробник буде обслуговувати декілька кнопок. В такому разі необхідно передбачити механізм розрізнення, яка кнопка натиснута. Найпростіше це зробити через конструктор обробника. Наступна програма виводитиме в фреймі дві кнопки і при натисненні на якусь з них буде виводитись повідомлення, яку кнопку натиснуто.

```

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JButton;
import java.awt.event.*;
import javax.swing.JOptionPane;

public class EventTest2 {

    public static void main(String[] args) {

        SimpleFrame frame = new SimpleFrame();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }

}

class SimpleFrame extends JFrame
{
    public static final int DEFAULT_WIDTH = 300;
    public static final int DEFAULT_HEIGHT = 200;

    public SimpleFrame()
    {
        setSize(DEFAULT_WIDTH, DEFAULT_HEIGHT); // задаємо розміри
фрейму

        JPanel panel = new JPanel();           // створюємо панель
        this.add(panel);                       // додаємо панель у
фрейм

        JButton button1 = new JButton("Кнопка 1"); // створюємо
кнопку 1
        panel.add(button1);                   // додаємо кнопку на панель
        ActionListener button1Action = new ActionListener(1); //
створюємо екземпляр обробника подій
        button1.addActionListener(button1Action); //
зв'язуємо обробника подій з кнопкою 1

        JButton button2 = new JButton("Кнопка 2"); // створюємо
кнопку 2

```



```

        panel.add(button2);                // додаємо кнопку на панель
        ActionListener button2Action = new ActionListener(2); //
створюємо екземпляр обробника подій
        button2.addActionListener(button2Action);                //
зв'язуємо обробника подій з кнопкою 2
    }
}
class EventListener implements ActionListener{
    private int buttonNumber;
    public EventListener (int number){
        buttonNumber=number;
    }
    @Override // не обов'язково, вказуємо, що ми перевизначаємо
метод actionPerformed
    public void actionPerformed(ActionEvent event)
    {
        // Виводимо повідомлення у діалоговому вікні, яку кнопку
натиснуто
        if (buttonNumber==1) JOptionPane.showMessageDialog(new
JFrame(), "Ви натиснули кнопку 1");
        else JOptionPane.showMessageDialog(new JFrame(), "Ви
натиснули кнопку 2");
    }
}

```

Часто також замість створення окремого класу обробника використовують внутрішні класи і навіть внутрішні не йменовані класи. Також в якості обробника події може виступати і сам клас, в якому описується графічний інтерфейс користувача.

11.2 Самостійна частина

Завдання 1. Згідно свого варіанту в таблиці 11.1 додати інтерфейс користувача у наступному вигляді (зразок рисунок 11.1):

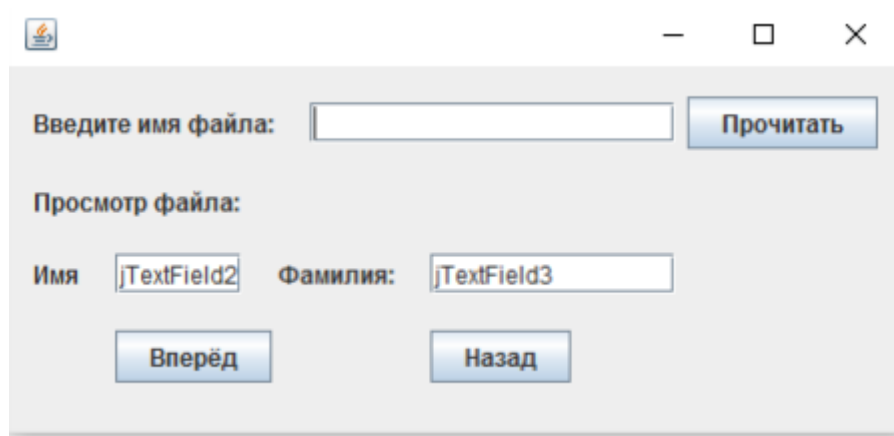


Рисунок 11.1 – Зразок виконання інтерфейсу

Таблиця 11.1 – Індивідуальні завдання до завдання 1

№ варіанту	Завдання
1	"Людина": прізвище; ім'я; по батькові; стать; національність; зріст; вага; дата народження (Рік, місяць число); номер телефону; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира). Вивести відомості про саму молодшу людину.
2	«Школяр»: прізвище; ім'я; по батькові; стать; національність; зріст; вага; дата народження (рік, місяць число); номер телефону; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира); школа; клас. Вивести відомості про всіх учнів п'ятих класів.
3	«Студент»: прізвище; ім'я; по батькові; стать; національність; зріст; вага; дата народження (рік, місяць число); номер телефону; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира); ВНЗ; курс; група; середній бал; спеціальність. Вивести відомості про всіх студентів у яких середній бал нижче 70 балів.

Продовження таблиці 11.1.

4	«Покупець»: прізвище; ім'я; по батькові; стать; національність; зріст; вага; дата народження (рік, місяць число); номер телефону; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира); номер кредитної картки, банківського рахунку. Вивести дані про покупців з міста Одеси.
5	«Пацієнт»: прізвище; ім'я; по батькові; стать; національність; зріст; вага; дата народження (рік, місяць число); номер телефону; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира); номер лікарні; відділення; номер медичної карти; діагноз; група крові. Вивести дані про пацієнтів з 18 відділення.
6	«Власник автомобіля»: прізвище; ім'я; по батькові; номер телефону; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира) марка автомобіля; номер автомобіля; номер техпаспорта. Вивести дані про автомобілі марки "Ваз".
7	«Військовослужбовець»: прізвище; ім'я; по батькові; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира); національність; дата народження (рік, місяць число); посаду; звання. Вивести дані про військовослужбовців у званні "лейтенант".
8	«Робочий»: прізвище; ім'я; по батькові; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира); національність; дата народження (рік, місяць число); № цеху; табельний номер; освіта; рік надходження на роботу. Вивести дані про робітників, які надійшли на роботу в 2010 році.
9	«Власник телефону»: прізвище; ім'я; по батькові; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок,

Продовження таблиці 11.1.

	квартира); № телефону. Вивести дані про власників телефону номер, якого починається на 720.
10	«Абітурієнт»: прізвище; ім'я; по батькові; стать; національність; дата народження (рік, місяць число); домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира); оцінки по іспитів; прохідний бал. Вивести дані про абітурієнтів, прохідний бал яких дорівнює більше 4.
11	«Держава»: назва держави; столиця; державна мова; населення; площа території; грошова одиниця; державний лад; Глава держави. Вивести дані про держави, населення яких більше 20 млн. Жителів.
12	«Автомобіль»: марка; колір; серійний номер; реєстраційний номер; рік випуску; рік техогляду; ціна. Вивести дані про автомобілі, яким більше 2 років.
13	«Товар»: найменування; вартість; термін зберігання; сорт; дата випуску; термін придатності. Вивести дані про товари термін придатності яких закінчується в цьому році.
14	«Кінострічка»: назва; режисер (прізвище та ім'я); рік виходу; країна; вартість; дохід; прибуток. Вивести дані про фільми режисера Єжи Гофмана.
15	«Рейс»: марка автомобіля; номер автомобіля; пункт призначення; вантажопідйомність (в тоннах); вартість одиниці вантажу; загальна вартість вантажу. Вивести дані про автомобілі, вантажопідйомність яких більше 2 тонн.
16	«Книга»: назва; автор (прізвище, а ім'я); рік виходу; видавництво; собівартість; ціна; прибуток. Вивести дані про книги авторів, прізвище яких починається на букву "К".
17	«Будівля»: адреса; тип будівлі; кількість поверхів; кількість квартир; термін експлуатації; термін до капітального ремонту (25

Продовження таблиці 11.1.

	років - термін експлуатації). Вивести дані про будівлі термін експлуатації, яких більше 50 років.
18	«Програміст»: прізвище; ім'я; по батькові; стать; національність; дата народження (рік, місяць число); освіта; номер телефону. Вивести відомості про програмістів, яким менше 25 років.
19	«Вчений»: прізвище; ім'я; по батькові; стать; національність; дата народження (рік, місяць число); вчена ступінь, посада, номер телефону; Домашня адреса (Поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира). Вивести відомості про вчених кандидатів технічних наук.
20	«Пенсіонер»: прізвище; ім'я; по батькові; стать; національність; дата народження (рік, місяць число); номер телефону; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира). Вивести відомості про всіх пенсіонерів, які на пенсії більше 5 років.
21	«Футболіст»: прізвище; ім'я; по батькові; стать; національність; зріст; вага; дата народження (рік, місяць число); номер телефону; назва команди; номер в команді; амплуа; результативність (кількість голів); кількість ігор. Вивести відомості про футболістів, які провели за свою команду більше 50 матчів.
22	«Манекенниця»: прізвище; ім'я; по батькові; стать; національність; зріст; вага; дата народження (рік, місяць число); домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира). Вивести дані про наймолодшу манекенницю.
23	«Міжнародна компанія»: назва; інтернет сайт; адреса головного офісу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира) тривалість перебування на світовому ринку; кількість співробітників; кількість філій в Європі. Вивести

Продовження таблиці 11.1.

	міжнародні компанії, кількість співробітників у яких більше 10000.
24	«Охоронець»: прізвище; ім'я; по батькові; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира); дата народження (рік, місяць число). Вивести дані про старшого охоронця ".
25	«Зоопарк»: Назва тварини; кількість виду; адреса зоопарку (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира); загальна кількість тварин, кількість працівників. Вивести відомості про зоопарки, в яких є уссурійські тигри.
26	«Програмне забезпечення»: назва; назва компанії виробника; рік виходу; ціна. Вивести дані про програмне забезпечення, яке дорожче 2000 гривень.
27	«Мультфільм»: назва; режисер (прізвище та ім'я); рік виходу; країна; вартість; дохід; прибуток. Вивести дані про мультфільми компанії "Walt Disney".
28	«Баскетболіст»: прізвище; ім'я; по батькові; стать; національність; зріст; вага; дата народження (рік, місяць число); номер телефону; назва команди; номер в команді; амплуа; результативність (кількість очок); кількість ігор. Вивести відомості про баскетболістів, яких закинули за свою команду більше 150 очей.
29	«Область»: назва області; обласний центр; населення; площа території; губернатор. Вивести дані про області, населення яких менше 2 млн. жителів.
30	«Мотоцикл»: марка; колір; серійний номер; реєстраційний номер; рік випуску; рік техогляду; ціна. Вивести дані про мотоцикли марки "Harley Davidson".

11.3 Питання для перевірки знань

1. У чому різниця між Swing і AWT?
2. Як центрувати вікно в Java?
3. Як принести вікно на фронт?
4. Чи є зберігання об'єктів Graphics гарною ідеєю?
5. Як відбувається завантаження ресурсу зображення?
6. Що робить `SwingUtilities.invokeLater`?
7. Чи безпечно створювати віджети Swing / AWT NOT на тему

Dispatch Event?

8. Як змінити розмір шрифту в `drawString` Java?
9. Що робить `.pack ()`?
10. Чи є Java Swing все ще у використанні?

ЛАБОРАТОРНА РОБОТА 12

Тема: ПАКЕТИ JAVA.AWT ТА JAVA.SWING

Мета: 1) Вивчити особливості роботи з формами у Java. 2) Навчитися створювати програми обробки даних використовуючи форми.

12.1 Практична частина

При роботі з файлами з програми виникає необхідність рано чи пізно використовувати діалог для вибору файлів. `JFileChooser` з бібліотеки Java Swing якраз є таким діалогом. Крім того діалог `JFileChooser` дозволяє виробляти навігацію по файловій системі. `JFileChooser` тільки надає можливість вибору файлу або директорії, більше нічого з ними не робить. Все інше - завдання розробника. Працювати з ним просто і легко. Давайте розберемо невеликий приклад і подивимося, що необхідно зробити, щоб показати `JFileChooser` і отримати інформацію про те, який вибір зробив користувач. А крім цього подивимося, як налаштувати `JFileChooser` належним чином. Далі ми будемо дивитися, як використовувати `JFileChooser` в самому простому випадку.

Для того, щоб показати діалог досить написати два рядки:

```
JFileChooser fileopen = new JFileChooser ();  
int ret = fileopen.showDialog (null, "Відкрити файл");
```

За допомогою першого рядка створюємо новий об'єкт `JFileChooser`, а за допомогою другого рядка відображаємо діалог користувачеві. Як тільки користувач закриває діалог, то в змінну `ret` заноситься певне значення. За допомогою цього значення можна зрозуміти, що власне сталося з `JFileChooser`: користувач натиснув скасування, або просто закрив діалог або ж забажав відкрити вибраний файл.

У фрагменті, представленому нижче припустимо, що користувач в діалозі `JFileChooser` вибрав файл і відкрив його. За допомогою методу

`getSelectedFile` ми отримуємо посилання на об'єкт `File` і потім продовжуємо з ним працювати.

```
JFileChooser fileopen = new JFileChooser ();
int ret = fileopen.showDialog (null, "Відкрити файл");
if (ret == JFileChooser.APPROVE_OPTION) {
    File file = fileopen.getSelectedFile ();
    / *
    * Якісь дії.
    * /
}
```

Приклад створення діалогу за допомогою команди **Jfilechooser**.

```
import java.awt.Dimension;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;

import javax.swing.Box;
import javax.swing.BoxLayout;
import javax.swing.JButton;
import javax.swing.JDialog;
import javax.swing.JFileChooser;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;

public class TestFrame extends JFrame {

    public TestFrame () {
        super ( "Тестове вікно");
        setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);

        JPanel panel = new JPanel ();
        panel.setLayout (new BoxLayout (panel,
        BoxLayout.Y_AXIS));

        panel.add (Box.createVerticalGlue ());

        final JLabel label = new JLabel ( "Обраний файл");
```

```

label.setAlignmentX (CENTER_ALIGNMENT);
panel.add (label);

panel.add (Box.createRigidArea (new Dimension (10, 10)));

JButton button = new JButton ( "Показати JFileChooser");
button.setAlignmentX (CENTER_ALIGNMENT);

button.addActionListener (new ActionListener () {
    public void actionPerformed (ActionEvent e) {
        JFileChooser fileopen = new JFileChooser ();
        int ret = fileopen.showDialog (null, "Відкрити
файл");

        if (ret == JFileChooser.APPROVE_OPTION) {
            File file = fileopen.getSelectedFile ();
            label.setText (file.getName ());
        }
    }
});

panel.add (button);
panel.add (Box.createVerticalGlue ());
getContentPane (). add (panel);

setPreferredSize (new Dimension (260, 220));
pack ();
setLocationRelativeTo (null);
setVisible (true);
}

public static void main (String [] args) {
    javax.swing.SwingUtilities.invokeLater (new Runnable () {
        public void run () {
            JFrame.setDefaultLookAndFeelDecorated (true);
            JDialog.setDefaultLookAndFeelDecorated (true);
            new TestFrame ();
        }
    });
}
}
}

```

12.2 Самостійна частина

Переробити **завдання 1** з лабораторної роботи 11. Додати кнопку «Огляд» та розробити форму перегляду файлової системи та вибору файлу (рисунок 11.1). Шлях до вибраного файлу повинен відображатися у текстовому полі. Розробити механізм збереження даних вказаних за завданням у л.р. 10 у новий файл.

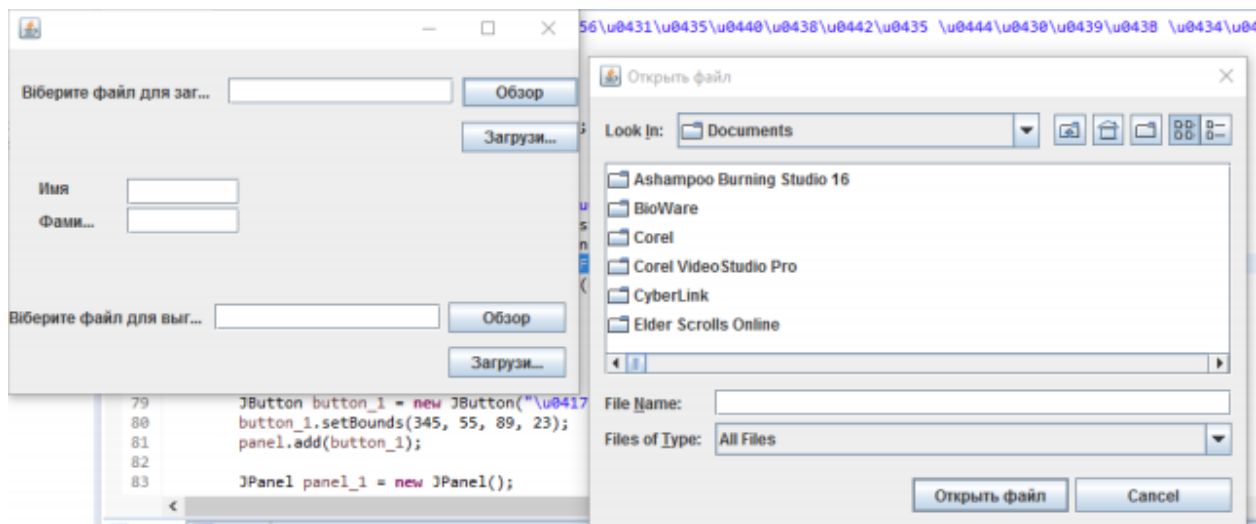


Рисунок 12.1 – Вигляд готового діалогового вікна

12.3 Питання для перевірки знань

1. Що таке `java.awt.Component.getName ()` і `setName ()`?
2. Як використовувати `FileDialog`?
3. Як можна вловити виключення AWT-потоків в Java?
4. Як отримати реальну висоту рядка в Java?
5. Java і GUI - Де `ActionListeners` належать відповідно до шаблону MVC?
6. SWT і AWT - в чому їх різниця?
7. Як `java.awt.dialog` досягає модальності?
8. Чи є простий спосіб порівняти екземпляри `BufferedImage`?
9. Як правильно вивести java повний екран?
10. Яка довжина рядка в пікселях в Java?

ЛАБОРАТОРНА РОБОТА 13

Тема: ПАКЕТ JAVA.AWT. АНІМАЦІЯ

Мета: 1) Вивчити особливості роботи з формами у Java. 2) Навчитися створювати програми обробки даних використовуючи форми.

13.1 Практична частина

Розглянемо код, Який анімує рух червоного кола по полотну. Уважно придивляйтесь до коментарів.

```
package lr13;
import java.applet.Applet;
import java.awt.Color;
import java.awt.Graphics;
import java.awt.Rectangle;
// Створюємо власний клас, який успадковується від класу Applet
// Даний клас реалізує методи інтерфейсу Runnable
public class BouncingCircle extends Applet implements Runnable {
    int x = 150, y = 50, r = 50; // Центр і радіус кола
    int dx = 11, dy = 7; // Рух кола по горизонт і вертикалі Thread
    animator
    Thread animator; // Нитка, яка здійснює анімацію
    volatile boolean pleaseStop; // Прапор зупинки руху
    // Метод для малювання кола червоним кольором
    public void paint (Graphics g) {
        g.setColor (Color.red); // установка кольору для g
        g.fillOval (x-r, y-r, r * 2, r * 2); // промальовування кола
    }
    / * Метод рухає коло і "відображає" його при ударі кола про
    стінку, потім викликає перерисовку. Даний метод викликається
    багаторазово анімаційним
    потоком * /
    public void animate () {
        Rectangle bounds = getBounds (); // Отримання розміру вікна
    програми
        if ((x - r + dx < 0) || (x + r + dx > bounds.width)) dx = -dx;
```

```

if ((y - r + dy < 0) || (y + r + dy > bounds.height)) dy = -dy;
// Зміна координат кола, по суті - рух.
x += dx; y += dy;
repaint ();
}
public void run () {
while (! pleaseStop) {
animate ();
try {Thread.sleep (100); } // Чекаємо 100 мілісекунд
catch (InterruptedException e) {}
}
}
public void start () {
animator = new Thread (this);
pleaseStop = false;
animator.start ();
}
public void stop ()
{
pleaseStop = true;
}
}

```

13.2 Самостійна частина

Завдання 1: Модифікувати код прикладу таким чином, щоб рух здійснювали відразу 10 об'єктів типу Rectangle. Всі створені об'єкти повинні зберігатися в одному масиві.

Завдання 2: Модифікувати код прикладу таким чином, щоб рух здійснювали відразу 10 об'єктів типу Rectangle різного кольору. Колір повинен вибиратися випадково. Всі створені об'єкти повинні зберігатися в масивах.

Завдання 3: Модифікувати код прикладу таким чином, щоб рух здійснювали відразу 10 об'єктів різного типу фігур (квадрат, коло, трикутник,

зірка, шестикутник) різного кольору. Колір та тип фігури повинен вибиратися випадково. Всі створені об'єкти повинні зберігатися в масивах.

12.3 Питання для перевірки знань

1. Як потрібно налагоджувати Java-аплети?
2. Як підписати аплет Java для використання в браузері?
3. Як розгорнути Java-аплет для сьогоdnішніх браузерів (аплет, вставка, об'єкт)?
4. Що означає попередження Java Applet про безпеку "JAR файл маніфесту не містить атрибут Permissions"?
5. Як обчислити медіану масиву?
6. Як об'єднати два файли Jar?
7. Чи існують гарні, зручні Java-аплети?
8. Як працює `URLConnection.setUseCaches ()`?
9. Як запустити Java-аплет з великим об'ємом пам'яті?
10. Чи використовують аплети браузер для запитів HTTP?

ЛАБОРАТОРНА РОБОТА 14

Тема: РОБОТА З БАЗАМИ ДАНИХ. JDBC

Мета: 1) Вивчити особливості роботи з формами у Java. 2) Навчитися створювати програми обробки даних використовуючи форми.

14.1 Практична частина

Як використовувати базу даних MySQL в Java

Для того, щоб отримати доступ до бази даних, ми будемо використовувати JDBC (Java Database Connectivity) API, який входить в стандартну бібліотеку Java. JDBC дозволяє підключитися до будь-якої бази даних: Postgres, MySQL, SQL Server, Oracle і т. Д. - за наявності відповідної реалізації драйвера, необхідного для підключення. Для бази даних MySQL ми будемо використовувати драйвер Type 4 JDBC з пакета `mysql-connector-java-5.1.23-bin.jar`. Він написаний на чистій Java, а значить, нам не знадобляться будь-які бібліотеки або ODBC-міст. Все, що нам треба буде зробити - це покласти JAR-файл в директорію, що міститься в CLASSPATH. JAR-файл містить клас `com.mysql.jdbc.Driver`, необхідний для підключення до MySQL. Якщо його не виявиться в CLASSPATH, під час виконання програми являється винятком `java.lang.ClassNotFoundException`, тому переконайтеся, що ви правильно налаштували шлях.

До речі, зверніть увагу на *Practical Database Programming with Java* (Ying Bai). Це відносно нова книга, і в ній розглядаються дві найпопулярніші бази даних: Oracle і SQL Server 2008. У книзі використовується IDE NetBeans для прикладів і описуються всі інструменти, необхідні для роботи з базами даних в Java. Це відмінна книга для початківців і досвідчених програмістів.

Підключаємо базу даних MySQL за допомогою JDBC

Для того, щоб підключити базу даних MySQL, нам буде потрібно чотири речі:

Рядок підключення JDBC (наприклад: jdbc: mysql: // localhost: 3306 / test).

Ім'я користувача (root).

Пароль (root).

База даних з деякою кількістю таблиць для прикладу (наприклад, база даних книг).

Рядок підключення для MySQL починається з jdbc: mysql. Ця назва протоколу з'єднання, за яким слідує хост і порт підключення, на яких запущена база даних. У нашому випадку це localhost з портом за замовчуванням 3306 (якщо ви його не поміняли при установці). Наступна частина - test - ім'я бази даних, яка вже існує в MySQL. Ми можемо створити таблицю Books:

```
CREATE TABLE `books` (  
  `Id` int (11) NOT NULL,  
  `Name` varchar (50) NOT NULL,  
  `Author` varchar (50) NOT NULL,  
  PRIMARY KEY ( `id` )  
) ENGINE = InnoDB DEFAULT CHARSET = latin1
```

і наповнити її хорошими книгами:

```
INSERT INTO test.books (id, `name`, author)  
  VALUES (1, 'Effective Java', 'Joshua Bloch');  
INSERT INTO test.books (id, `name`, author)  
  VALUES (2, 'Java Concurrency in Practice', 'Brian  
Goetz');
```

Програма на Java, яка використовує базу даних

Тепер напишемо програму на Java, яка буде підключатися до нашої бази даних, запущеної на localhost. Важливо пам'ятати про те, що необхідно закривати з'єднання, запити та результат виконання після завершення роботи з ними. Також важливо закривати їх в finally-блоці, зі своєю try / catch обгорткою, оскільки сам метод close () може кинути виняток, що призведе до витоку ресурсів. Можна використовувати обгортку try-with-resource, яка

з'явилася в Java 7. Більш того, це стандартний спосіб роботи з ресурсами в Java 1.7.

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

/ **
 * Simple Java program to connect to MySQL database running
on localhost and
 * Running SELECT and INSERT query to retrieve and add data.
 * @author Javin Paul
 * /
public class JavaToMySQL {

    // JDBC URL, username and password of MySQL server
    private static final String url = "jdbc: mysql: //
localhost: 3306 / test";
    private static final String user = "root";
    private static final String password = "root";

    // JDBC variables for opening and managing connection
    private static Connection con;
    private static Statement stmt;
    private static ResultSet rs;

    public static void main (String args []) {
        String query = "select count (*) from books";

        try {
            // opening database connection to MySQL server
            con = DriverManager.getConnection (url, user,
password);
```



```
at JavaToMySQL.main (JavaToMySQL.java:29)
Exception in thread "main" java.lang.NullPointerException
at JavaToMySQL.main (JavaToMySQL.java:46)
Java Result: 1
```

Додамо потрібний JAR-файл в шлях і знову запусимо програму. Інша поширена помилка являється вказання таблиці в рядку з'єднання: jdbc: mysql:// localhost: 3306 / test / book. В цьому випадку вилетить наступний виняток:

```
com.mysql.jdbc.exceptions.jdbc4.MySQLSyntaxErrorException:
Unknown          database          'test/book          at
sun.reflect.NativeConstructorAccessorImpl.newInstance0 (Native
Method)
at
sun.reflect.NativeConstructorAccessorImpl.newInstance (NativeCons
tructorAccessorImpl.java:62)
at
sun.reflect.DelegatingConstructorAccessorImpl.newInstance (Delega
tingConstructorAccessorImpl.java:45)
at
java.lang.reflect.Constructor.newInstance (Constructor.java:408)
at com.mysql.jdbc.Util.handleNewInstance (Util.java:411)
at com.mysql.jdbc.Util.getInstance (Util.java:386)
at
com.mysql.jdbc.SQLException.createSQLException (SQLException.java:1053)
```

Успішний запуск програми виведе на екран наступне:

```
Total number of books in the table: 2
```

Результат вірний, оскільки у нас в таблиці тільки дві книги: «Effective Java» і «Java Concurrency in Practice» . Якщо при розробці використовувався драйвер при компіляції, але він був відсутній при запуску, можна отримати виключення java.lang.ClassNotFoundException: com.mysql.jdbc.Driver.

JDBC - приклад з'єднання і простого виклику

Спробуємо подивитися на нескладному прикладі, як використовується JDBC-драйвер. У ньому ж ми познайомимося з деякими важливими інтерфейсами і класами.

Попередньо нам необхідно завантажити JDBC-драйвер для PostgreSQL. На даний момент це можна зробити зі сторінки PostgreSQL JDBC Download.

Якщо ви не знайшли цю сторінку, то просто наберіть в пошуковик "PostgreSQL JDBC download" і в перших же рядках знайдете потрібну сторінку.

Якщо ви виконали SQL-скрипт з розділу Установка PostgreSQL, який створював таблицю JC_CONTACT і вставив туди пару рядків, то ця програма дозволить вам "витягнути" ці дані і показати їх на екрані. Це звичайно ж дуже проста програма, але на ній ми зможемо подивитися дуже важливі моменти. Отже, ось код:

```
package edu.javacourse.database;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

public class SimpleDb
{
    public static void main (String [] args) {
        SimpleDb m = new SimpleDb ();
        m.testDatabase ();
    }

    private void testDatabase () {
        try {
            Class.forName ( "org.postgresql.Driver" );
            String url = "jdbc: postgresql: // localhost:
5432 / contactdb";
            String login = "postgres";
            String password = "postgres";
            Connection con = DriverManager.getConnection
(url, login, password);
            try {
                Statement stmt = con.createStatement ();
```


Наступним важливим викликом являється `DriverManager.getConnection(url, login, password)` ;.

Параметри `login` і `password` - це логін і пароль для підключення до СУБД. Параметр - **url** розглянемо детально.

Параметр **url** є рядком і його можна розбивати на дві частини. Перша частина `jdbc: postgresql:` дозволяє ідентифікувати, до якого типу СУБД ви підключаєтеся - Oracle, MySQL, PostgreSQL, IBM DB2, MS SQL Server. У нашому випадку тип бази даних - PostgreSQL.

Друга частина - `// localhost: 5432 / contactdb` - визначає конкретний екземпляр обраної бази даних. Тобто, якщо перша частина `url` вказує, що ми хочемо працювати з PostgreSQL, то друга частина вказує на якому хості і на якому порту (знову згадуємо основи TCP / IP) працює конкретний екземпляр PostgreSQL. Ще раз - перша частина поределяєт тільки тип, друга частина - параметри з'єднання з конкретним екземпляром СУБД.

Друга частина, крім IP-адреси і порту (`localhost: 3306`), включає ім'я бази даних, з якою ви будете з'єднуватися.

Повертаємося до теми інтерфейсу `java.sql.Driver`. Досить очевидно, що складний додаток на Java може працювати з декількома типами СУБД і одночасно в додатку беруть участь кілька JDBC-драйверів для різних типів СУБД.

Більшість класів в момент свого завантаження виконують дуже важливий крок - вони реєструються у класі `DriverManager`.

Серед методів можна знайти: `registerDriver (Driver driver)`. Цей метод статичний і створювати екземпляр `DriverManager` не треба. Таким чином драйвер під конкретний тип СУБД реєструється у `DriverManager`. У цього класу (можна глянути в вихідники) створюється список драйверів, кожен з яких реалізує інтерфейс `java.sql.Driver`. Далі зайдемо в документацію `java.sql.Driver`. Там є два дуже цікавих метода:

```
boolean acceptsURL (String url)
Connection connect (String url, Properties info)
```

Перший метод якраз і дозволяє класу DriverManager пройти по всьому списку зареєстрованих у ньому драйверів і у кожного запитати - "ти вмієш працювати з цим URL". Відзначимо, що драйвер під конкретний тип СУБД працює з унікальним набором - MySQL приймає рядок "jdbc: mysql:", PostgreSQL - "jdbc: postgresql:" і т.д. Тобто перша частина параметра url, дозволяє класу DriverManager вибрати драйвер для певного типу СУБД. Перший крок зроблено - ми вибрали потрібний драйвер.

Другий метод - дозволяє створити з'єднання - повертає екземпляр класу, який реалізує ще один важливий інтерфейс - java.sql.Connection. Другий метод використовує другу частину url з адресою, портом і ім'ям бази, а також використовується логін і пароль. Зверніть увагу на той факт, що реальний клас буде спеціальним, під конкретний тип СУБД, але він обов'язково повинен реалізувати інтерфейс java.sql.Connection.

java.sql.Connection - це реальне з'єднання з конкретним екземпляром СУБД певного типу. Наше з'єднання готове.

Наступні фрагмент коду вже буде простіше:

```
Statement stmt = con.createStatement ();
ResultSet rs = stmt.executeQuery ("SELECT * FROM
JC_CONTACT");
```

Перший рядок створює ще один важливий елемент - запит, який реалізує інтерфейс java.sql.Statement. Крім цього інтерфейсу використовуються також java.sql.PreparedStatement і java.sql.CallableStatement, але про них ми поговоримо трохи пізніше.

Важливо відзначити - створення запиту робиться через звернення до методу об'єкта java.sql.Connection - createStatement. І знову звертаю вашу увагу, що кожен виробник СУБД пише свою реалізацію всіх інтерфейсів.

Оскільки реалізація java.sql.Connection буде під певний тип СУБД, то і реалізація java.sql.Statement теж буде під певний тип СУБД.

Другий рядок за допомогою об'єкта-запиту `java.sql. Statement` робить запит в таблицю `st_student` і отримує ще один важливий елемент - об'єкт `java.sql.ResultSet`.

Після отримання даних у вигляді об'єкта `ResultSet`, ми можемо через його методи пройти по всьому набору даних (це дуже схоже на ітератори в колекціях) і вибрати поля з цього набору.

```
while (rs.next()) {
    String str = rs.getString("contact_id") + ":" +
rs.getString(2);
    System.out.println("Contact:" + str);
}
```

`ResultSet` можна розглядати, як показник на рядок в таблиці. Метод `rs.next ()` робить спробу пересунути на наступний запис. У разі успіху він повертає `true` і пересуває показник на наступну рядок. Якщо рядки закінчилися (або їх не було взагалі), повертається `false`.

Коли ми пересунулися на наступний рядок, то за допомогою набору методів можна отримати значення колонок в рядку - ми використовували метод `getString ()` в двох варіантах - один знаходить колонку на ім'я, другий - за індексом. Врахуйте, що номери колонок починаються з 1, а не з 0, як це робиться в масивах і колекціях. Крім методу `getString ()` для отримання рядків, `ResultSet` має методи для отримання чисел (цеих і речових), дат і т.і.

І зверніть увагу, що ми викликаємо у всіх об'єктів метод `close ()`. Особливо важливим є закриття `Connection`. Закриття `Statement` теж є досить хорошим рішенням, але не настільки критичним. В цьому випадку швидше звільняється пам'ять від ресурсів, які створювалися при запиті. Врахуйте, що `Statement` закривається автоматично при знищенні об'єкта. Що ж стосується `ResultSet`, то він автоматично закривається в момент закриття `Statement`.

Зверніть увагу на спосіб побудови обробки винятків. Спочатку ми створюємо `connect` в зовнішньому блоці **try ... catch** і потім вже у внутрішньому блоці **try ... catch** виконуємо запит і отримуємо з результату дані. У цьому ж блоці в розділі `finally` відбувається закриття з'єднання.

Така побудова дає впевненість, що незалежно від результату виконання запиту та отримання даних, з'єднання буде обов'язково закрито.

Оскільки при роботі додатка виключення при виконанні запиту не повинні бути частими, то "незакриття" Statement не повинно спричинити будь-яких ускладнень.

Запити на отримання даних і запити на оновлення

SQL-запити можна умовно розділити на дві групи:

1. отримання даних - до них відноситься оператор **SELECT**;
2. зміна даних - до них відносяться оператори **INSERT, UPDATE і DELETE**.

Для першої групи використовується вже знайомий нам метод інтерфейсу Statement - `executeQuery ()`. В принципі для початку цього методу цілком достатньо. Він покриває дуже великий відсоток запитів, які розробляються для реальних систем. Пізніше ми познайомимося з додатковими можливостями, але на даних момент раджу запам'ятати - якщо треба отримати дані з таблиці, то `executeQuery` в переважній більшості випадків буде самим правильним вибором.

Для другої групи запитів (знову ж таки в більшості випадків) можна використовувати інший метод інтерфейсу Statement - `executeUpdate ()`. На відміну від `executeQuery ()` (який повертає `ResultSet`) цей метод повертає ціле число, яке говорить скільки рядків у таблиці було змінено при виконанні вашого запиту.

Наприклад, можна оператором `DELETE FROM JC_CONTACT` видалити всі рядки (тому будьте дуже обережні). У цьому випадку метод `executeUpdate ()` поверне кількість вилучених рядків. У деяких ситуаціях знання про кількість змінених рядків буває зручним для побудови алгоритмів роботи з даними.

Висновок: для вибірки даних - `executeQuery ()`. Для зміни даних - `executeUpdate ()`.

Різновиди Statement

Найпростіший інтерфейс Statement ми вже бачили. І хоча він цілком придатний для роботи, для складних запитів він підходить не так добре. У деяких джерелах висловлюється думка, що використовувати Statement взагалі не треба - замість нього підходять більш складні і більш функціонально насичені інтерфейси.

1. PreparedStatement
2. CallableStatement

Для початку ми розглянемо інтерфейс PreparedStatement і інші можливості JDBC. До інтерфейсу CallableStatement звернемося пізніше - його використання по-перше не так часто зустрічається, і по-друге - після всього розглянутого, про нього розмова вже можна робити досить коротким.

PreparedStatement

Якщо перекласти назву, то можна отримати щось на зразок «підготовлений оператор». Найважливішим тут є слово «підготовлений». У чому ж полягає «підготовленість»?

Перш ніж ми розглянемо це питання, подивимось досить важливий з точки зору зручності момент, який виникає дуже часто. В будь-якому додатку нам треба вставити дані про контакт в таблицю JC_CONTACT. Для цього нам треба підготувати запит на зразок такого:

```
INSERT INTO JC_CONTACT (FIRST_NAME, LAST_NAME, PHONE, EMAIL)
VALUES ('Peter', 'Belgy', '+ 79112345678', 'peter@pisem.net');
```

Треба написати код, який буде генерувати потрібний нам рядок з параметрів - ім'я, прізвище, адресу та телефон. Треба тільки не забути, що всі строкові дані треба виділити символом «' »». Якщо ми це робимо в окремій функції, то виходить такий результат:

```
public String buildInsert (String firstName, String
lastName, String phone, String email) {
    String sql = "INSERT INTO JC_CONTACT (FIRST_NAME,
LAST_NAME, PHONE, EMAIL) VALUES ('" + firstName + "', '" +
lastName + "', '" + phone + "', '" + email + "');" ;
```

```
        return sql;
    }
}
```

Ми передаємо в функцію у вигляді параметрів ім'я, прізвище, телефон і адресу і з них складаємо рядок SQL-запиту.

Якщо нам треба використовувати параметри з запитом, то в «ручному» режимі побудова запиту стає важкою. Цей запит дозволяє вам зробити дві речі:

1. Заздалегідь підготувати запит із зазначенням місць, де будуть підставлятися параметри
2. Встановити параметри певного типу і виконати після цього запит з уже встановленими параметрами

Конструкція для `PreparedStatement` для нашого варіанта установки параметрів буде виглядати ось так:

```
// Змінні для прикладу
String firstName = "Dmitry";
String lastName = "Chekhov";
String phone = "+79871112233";
String email = "dmitry@pisem.net";

// Запит із зазначенням місць для параметрів у вигляді знака
"?"
String sql = "INSERT INTO JC_CONTACT (FIRST_NAME, LAST_NAME,
PHONE, EMAIL) VALUES (?, ?, ?, ?)";

// Створення запиту. Мінлива con - це об'єкт типу Connection
PreparedStatement stmt = con.prepareStatement (sql);

// Установка параметрів
stmt.setString (1, firstName);
stmt.setString (2, lastName);
stmt.setString (3, phone);
stmt.setString (4, email);

// Виконання запиту
```

```
stmt.executeUpdate ();
```

Приклад використання об'єкта JDBC PreparedStatement - пакетне оновлення

При необхідності виконати безліч однотипних запитів до бази даних використовуйте технології пакетного поновлення даних. Загальний механізм виглядає наступним чином:

1. ми формуємо безліч запитів, які хочемо виконати. При цьому після кожного запиту необхідно викликати метод `addBatch ()`, який фіксує черговий запит в послідовності;
2. ми виконуємо всі раніше додані запити за допомогою виклику методу `executeBatch ()`.

Можно послідовно викликати методи `executeUpdate ()` для виконання кожного запиту окремо, але в цьому випадку у нас може виникнути проблема з продуктивністю. Пов'язано це з тим, що при кожному виклику методу `executeUpdate ()` буде ініціюватися нове з'єднання з базою даних, а ця операція вимоглива до ресурсів (особливо, якщо додаток і СУБД фізично знаходяться на різних машинах). Таким чином, якщо потрібно виконати 1000 операцій вставки даних, то буде 1000 разів ініційовано з'єднання. При використанні пакетного поновлення для виконання всієї 1000 запитів з'єднання буде ініційовано тільки один раз.

Клас `JDBCPreparedStatementBatchExample`:

```
package  
ru.j4web.examples.java.jdbc.jdbcpreparedstatementbatchexample;  
  
import java.sql.Connection;  
import java.sql.Date;  
import java.sql.DriverManager;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.util.logging.Level;
```

```

import java.util.logging.Logger;

public class JDBCPreparedStatementBatchExample {

    private static final String DB_URL = "jdbc: mysql: //
dev-server / sampledb"
        + "? User = sampleuser & password =
samplepassword";

    private static final String INSERT_STATEMENT = "INSERT
INTO users"
        + "(Username, created_by, creation_date) VALUES
(?,?,?)";

    private static final String SELECT_STATEMENT = "SELECT
user_id, username,"
        + "Created_by, creation_date FROM users";

    public static void main (String [] args) {
        // Дивимося, що було в базі даних
        printUsers ();
        // Виконуємо пакетне оновлення даних
        insertUsers ();
        // Дивимося, що стало в базі даних
        printUsers ();
    }

    private static void printUsers () {

        try          (Connection          connection          =
DriverManager.getConnection (DB_URL);
                    PreparedStatement statement
                    = Connection.prepareStatement (SELECT_STATEMENT))
        {

            ResultSet result = statement.executeQuery ();

            while (result.next ()) {

```

```

        System.out.println (
"=====");
        Integer userId = result.getInt ( "user_id");
        String  username  =  result.getString (
"username");
        String  createdBy  =  result.getString (
"created_by");
        Date  creationDate  =  result.getDate (
"creation_date");
        System.out.println ( "Next row: user_id =" +
userId
        + "; Username =" + username + ";
created_by ="
        + CreatedBy + "; creation_date =" +
creationDate);
    }

    } Catch (SQLException ex) {
        Logger.getLogger
(JDBCPreparedStatementBatchExample.class
        .getName ()). log (Level.SEVERE, null,
ex);
    }
}

private static void insertUsers () {

    try (Connection connection =
DriverManager.getConnection (DB_URL);
        PreparedStatement statement
        = Connection.prepareStatement (INSERT_STATEMENT))
    {

        for (int i = 11; i <= 20; i ++) {

            statement.setString (1, "user_" + i);

```

```

        statement.setString (2, "j4web");
        statement.setDate (3, new Date (new
java.util.Date (). getTime ()));

        statement.addBatch ();
    }

    statement.executeBatch ();

    System.out.println ( "Дані додані");

    } Catch (SQLException ex) {
        Logger.getLogger
(JDBCPreparedStatementBatchExample.class
.getName ()). log (Level.SEVERE, null,
ex);
    }
}
}
}

```

Результат роботи програми:

```

--- exec-maven-plugin: 1.2.1: exec (default-cli) @
JDBCPreparedStatementBatchExample ---
=====
Next row: user_id = 6; username = user_6; created_by = j4web.ru;
creation_date = 2016-03-02
=====
Next row: user_id = 7; username = user_7; created_by = j4web.ru;
creation_date = 2016-03-02
=====
Next row: user_id = 8; username = user_8; created_by = j4web.ru;
creation_date = 2016-03-02
=====

```

Next row: user_id = 9; username = user_9; created_by = j4web.ru;
creation_date = 2016-03-02

=====

Next row: user_id = 10; username = user_10; created_by = j4web.ru;
creation_date = 2016-03-02

дані додані

=====

Next row: user_id = 6; username = user_6; created_by = j4web.ru;
creation_date = 2016-03-02

=====

Next row: user_id = 7; username = user_7; created_by = j4web.ru;
creation_date = 2016-03-02

=====

Next row: user_id = 8; username = user_8; created_by = j4web.ru;
creation_date = 2016-03-02

=====

Next row: user_id = 9; username = user_9; created_by = j4web.ru;
creation_date = 2016-03-02

=====

Next row: user_id = 10; username = user_10; created_by = j4web.ru;
creation_date = 2016-03-02

=====

Next row: user_id = 11; username = user_11; created_by = j4web;
creation_date = 2016-03-09

=====

Next row: user_id = 12; username = user_12; created_by = j4web;
creation_date = 2016-03-09

=====

Next row: user_id = 13; username = user_13; created_by = j4web;
creation_date = 2016-03-09


```
=====
Next row: user_id = 14; username = user_14; created_by = j4web;
creation_date = 2016-03-09
```

```
=====
Next row: user_id = 15; username = user_15; created_by = j4web;
creation_date = 2016-03-09
```

```
=====
Next row: user_id = 16; username = user_16; created_by = j4web;
creation_date = 2016-03-09
```

```
=====
Next row: user_id = 17; username = user_17; created_by = j4web;
creation_date = 2016-03-09
```

```
=====
Next row: user_id = 18; username = user_18; created_by = j4web;
creation_date = 2016-03-09
```

```
=====
Next row: user_id = 19; username = user_19; created_by = j4web;
creation_date = 2016-03-09
```

```
=====
Next row: user_id = 20; username = user_20; created_by = j4web;
creation_date = 2016-03-09
```

BUILD SUCCESS

14.2 Самостійна частина

Використовуючи завдання за своїм варіантом з лабораторної роботи, розробити такий алгоритм роботи програми:

1. Читання файлу з даними.
2. Загрузка даних у класи.

3. Підключення до бази даних.
4. Перевірка наявності зчитаних даних у БД.
5. Якщо даних немає – додати дані до БД.
6. Форма повинна читати дані з БД.
7. Розробити фільтрацію даних на формі згідно з завданням у таблиці 14.1.

Тобто на формі повинні виводитися тільки ті дані, які вказані у фільтрі, або при пустому фільтрі – всі дані.

Приклад інтерфейсу на рисунку 14.1.

Рисунок 14.1 – Приклад інтерфейсу програми

Таблиця 14.1 – Індивідуальні завдання до завдання

№ варіанту	Завдання
1	"Людина": прізвище; ім'я; по батькові; стать; національність; зріст; вага; дата народження (Рік, місяць число); номер телефону; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира). Вивести відомості про саму молодшу людину.

Продовження таблиці 14.1.

2	«Школяр»: прізвище; ім'я; по батькові; стать; національність; зріст; вага; дата народження (рік, місяць, число); номер телефону; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира); школа; клас. Вивести відомості про всіх учнів п'ятих класів.
3	«Студент»: прізвище; ім'я; по батькові; стать; національність; зріст; вага; дата народження (рік, місяць, число); номер телефону; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира); ВНЗ; курс; група; середній бал; спеціальність. Вивести відомості про всіх студентів у яких середній бал нижче 70 балів.
4	«Покупець»: прізвище; ім'я; по батькові; стать; національність; зріст; вага; дата народження (рік, місяць, число); номер телефону; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира); номер кредитної картки; банківського рахунку. Вивести дані про покупців з міста Одеси.
5	«Пацієнт»: прізвище; ім'я; по батькові; стать; національність; зріст; вага; дата народження (рік, місяць, число); номер телефону; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира); номер лікарні; відділення; номер медичної карти; діагноз; група крові. Вивести дані про пацієнтів з 18 відділення.
6	«Власник автомобіля»: прізвище; ім'я; по батькові; номер телефону; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира) марка автомобіля; номер автомобіля; номер техпаспорта. Вивести дані про автомобілі марки "Ваз".

Продовження таблиці 14.1.

7	«Військовослужбовець»: прізвище; ім'я; по батькові; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира); національність; дата народження (рік, місяць, число); посаду; звання. Вивести дані про військовослужбовців у званні "лейтенант".
8	«Робочий»: прізвище; ім'я; по батькові; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира); національність; дата народження (рік, місяць, число); № цеху; табельний номер; освіта; рік надходження на роботу. Вивести дані про робітників, які надійшли на роботу в 2010 році.
9	«Власник телефону»: прізвище; ім'я; по батькові; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира); № телефону. Вивести дані про власників телефону номер, якого починається на 720.
10	«Абітурієнт»: прізвище; ім'я; по батькові; стать; національність; дата народження (рік, місяць, число); домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира); оцінки по іспитів; прохідний бал. Вивести дані про абітурієнтів, прохідний бал яких дорівнює більше 4.
11	«Держава»: назва держави; столиця; державна мова; населення; площа території; грошова одиниця; державний лад; глава держави. Вивести дані про держави, населення яких більше 20 млн. жителів.
12	«Автомобіль»: марка; колір; серійний номер; реєстраційний номер; рік випуску; рік техогляду; ціна. Вивести дані про автомобілі, яким більше 2 років.
13	«Товар»: найменування; вартість; термін зберігання; сорт; дата випуску; термін придатності. Вивести дані про товари термін придатності яких закінчується в цьому році.

Продовження таблиці 14.1.

14	«Кінострічка»: назва; режисер (прізвище та ім'я); рік виходу; країна; вартість; дохід; прибуток. Вивести дані про фільми режисера Єжи Гофмана.
15	«Рейс»: марка автомобіля; номер автомобіля; пункт призначення; вантажопідйомність (в тоннах); вартість одиниці вантажу; загальна вартість вантажу. Вивести дані про автомобілі, вантажопідйомність яких більше 2 тонн.
16	«Книга»: назва; автор (прізвище та ім'я); рік виходу; видавництво; собівартість; ціна; прибуток. Вивести дані про книги авторів, прізвище яких починається на букву "К".
17	«Будівля»: адреса; тип будівлі; кількість поверхів; кількість квартир; термін експлуатації; термін до капітального ремонту (25 років - термін експлуатації). Вивести дані про будівлі термін експлуатації, яких більше 50 років.
18	«Програміст»: прізвище; ім'я; по батькові; стать; національність; дата народження (рік, місяць число); освіта; номер телефону. Вивести відомості про програмістів, яким менше 25 років.
19	«Вчений»: прізвище; ім'я; по батькові; стать; національність; дата народження (рік, місяць, число); вчена ступінь, посада, номер телефону; домашня адреса (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира). Вивести відомості про вчених кандидатів технічних наук.
20	«Пенсіонер»: прізвище; ім'я; по батькові; стать; національність; дата народження (рік, місяць, число); номер телефону; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира). Вивести відомості про всіх пенсіонерів, які на пенсії більше 5 років.

Продовження таблиці 14.1.

21	«Футболіст»: прізвище; ім'я; по батькові; стать; національність; зріст; вага; дата народження (рік, місяць, число); номер телефону; назва команди; номер в команді; амплуа; результативність (кількість голів); кількість ігор. Вивести відомості про футболістів, які провели за свою команду більше 50 матчів.
22	«Манекенщиця»: прізвище; ім'я; по батькові; стать; національність; зріст; вага; дата народження (рік, місяць, число); домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира). Вивести дані про наймолодшу манекенщицю.
23	«Міжнародна компанія»: назва; інтернет сайт; адреса головного офісу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира) тривалість перебування на світовому ринку; кількість співробітників; кількість філій в Європі. Вивести міжнародні компанії, кількість співробітників у яких більше 10000.
24	«Охоронець»: прізвище; ім'я; по батькові; домашню адресу (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира); дата народження (рік, місяць, число). Вивести дані про старшого охоронця " .
25	«Зоопарк»: назва тварини; кількість виду; адреса зоопарку (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира); загальна кількість тварин, кількість працівників. Вивести відомості про зоопарки, в яких є уссурійські тигри.
26	«Програмне забезпечення»: назва; назва компанії виробника; рік виходу; ціна. Вивести дані про програмне забезпечення, яке дорожче 2000 гривень.
27	«Мультфільм»: назва; режисер (прізвище та ім'я); рік виходу; країна; вартість; дохід; прибуток. Вивести дані про мультфільми компанії "Walt Disney".

Продовження таблиці 14.1.

28	«Баскетболіст»: прізвище; ім'я; по батькові; стать; національність; зріст; вага; дата народження (рік, місяць, число); номер телефону; назва команди; номер в команді; амплуа; результативність (кількість очок); кількість ігор. Вивести відомості про баскетболістів, яких закинули за свою команду більше 150 очей.
29	«Область»: назва області; обласний центр; населення; площа території; губернатор. Вивести дані про області, населення яких менше 2 млн. жителів.
30	«Мотоцикл»: марка; колір; серійний номер; реєстраційний номер; рік випуску; рік техогляду; ціна. Вивести дані про мотоцикли марки "Harley Davidson".

14.3 Питання для перевірки знань

1. Які основні елементи баз даних - таблиці, процедури, функції, констрейнтів і т.д ..?
2. Як ви розумієте null в базах даних?
3. Агрегатні функції, як вони працюють з null. Не забудьте про group by і having.
4. Яким чином краще додавати велику кількість записів в таблицю?
5. Що таке перша нормальна форма і процес нормалізації? Які бувають нормальні форми?
6. У чому сенс індексу СУБД, як вони влаштовані, як зберігаються? Як би ви реалізували той же функціонал?
7. Що таке JDBC API і коли його використовують?
8. Що таке JDBC Driver і які різні типи драйверів JDBC ви знаєте?
9. Як JDBC API допомагає досягти слабкою зв'язку між Java програмою і JDBC Drivers API?

10. Що таке JDBC Connection? Покажіть кроки для підключення програми до бази даних.
11. Як використовується JDBC DriverManager class?
12. Як отримати інформацію про сервер бази даних з java програми?
13. Що таке JDBC Statement?
14. Які відмінності між execute, executeQuery, executeUpdate?

СПИСОК ЛІТЕРАТУРИ

1. Альфред, В. Ахо Компиляторы. Принципы, технологии и инструментарий / Альфред В. Ахо и др. - М.: Вильямс, 2015. - 689 с.
2. Берд, Б. Java для чайников / Берд. Барри - М.: Диалектика / Вильямс, 2013. - 521 с.
3. Гарнаев, А. WEB-программирование на Java и JavaScript / А.Гарнаев, С. Гарнаев. - Москва: СПб., 2017. - 718 с.
4. Гонсалвес, Э. Изучаем Java EE 7 / Э. Гонсалвес. - М.: Питер, 2016. - 640 с.
5. Гупта, А. Java EE 7. Основы / А. Гупта. - М.: Вильямс, 2014. - 336 с.
6. Монахов, В. Язык программирования Java и среда NetBeans (+ CD-ROM) / В. Монахов. - М.: БХВ-Петербург, 2012. - 720 с.
7. Савитч, У. Язык Java. Курс программирования / У. Савитч. - М.: Вильямс, 2015. - 928 с.
8. Хабибуллин, И. Самоучитель Java / И. Хабибуллин. - М.: БХВ-Петербург, 2014. - 768 с.
9. Шилдт, Г. Java 8. Руководство для начинающих / Г. Шилдт. - М.: Вильямс, 2015. - 720 с.
10. Эккель, Б. Философия Java / Б. Эккель. - М.: Питер, 2016. - 809 с.
11. Бібліотека ТДАТУ, м.Мелітополь, пр.Б.Хмельницького, 18.
12. Oracle [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – [Росія : Російський Інтернет холдинг Oracle 1998-2015]. – Режим доступу: <http://www.java.com/ru/>

Навчальне видання

**ЛИТВИН ЮЛІЯ ОЛЕКСАНДРІВНА
СЦИЛІЩИН ЮРІЙ ОЛЕКСАНДРОВИЧ
СТРОКАНЬ ОКСАНА ВІКТОРІВНА**

КРОС-ПЛАТФОРМНЕ ПРОГРАМУВАННЯ

Лабораторний практикум для виконання лабораторних робіт

Надруковано з оригіналів макетів замовника
Підписано до друку __.__.2019 р. формат 60x84 1/16
Папір офсетний. Наклад 100 примірників
Замовлення № ____

**Виготовлювач ПП Верескун В.М.
Видавничо-поліграфічний центр «Люкс»
М. Мелітополь, вул. М.Грушевського,10 тел. (0619) 44-45-11**

Свідоцтво про внесення суб'єкта видавничої справи
до Державного реєстру видавців, виробників
і розповсюджувачів видавничої продукції
від 11.06.2002 р. серія ДК № 1125

