

Лабораторна робота №2

Тема: Створення структури класу заданої предметної області.

Завдання 1. Створити консольний Java проект java_lab_2_oop з пакетом com.education.ztu

Завдання 2. Створити ієрархію класів відповідно до UML діаграми:

- поля класів повинні бути приховані модифікаторами доступу private, protected;
- створити конструктор без аргументів та з аргументами;
- створити блок ініціалізації, в якому ініціалізуються значення полів за замовчуванням у разі, якщо викликається конструктор без аргументів;
- створити геттери та сеттери для полів;
- створити статичну змінну counter для підрахунку створених екземплярів даного класу та статичний метод showCounter для відображення значення змінної counter.
- створити enum Location та Gender і використати їх в полях класів.
- створити інтерфейс Human з методами sayFullName, sayAge, sayLocation, sayGender та whoAm (default)
- створити абстрактний клас Person з абстрактним методом getOccupation та звичайним методом getFullInfo, що імплементує Human;
- створити класу Student, Teacher, Employee, що наслідують Person та перевизначити необхідні методи та створити свої.
- для Teacher, Employee додати поле Car, що є об'єктом відповідного класу.
- створити в Car внутрішній клас Engine з методами startEngine, stopEngine, isEngineWorks та реалізувати їх логіку.
- додати до описаної функціональності свою (нові поля та методи).
- в методі main класу Main створити об'єкти відповідних класів та продемонструвати роботу їх методів.
- продемонструвати роботу оператору instanceof.

Завдання 3. Створити клас Operation з статичними методами addition, subtraction, multiplication, division, average, maximum, minimum, що приймають необмежену кількість аргументів через varargs.
в методі main класу Main2 продемонструвати роботу методів класу Operation

- вивести всі значення enum Location.

Завдання 4. Створити UML діаграму створеної структури ієрархії класів та зберегти як картинку.

					ДУ «Житомирська політехніка».24.121.03.000 – Лр			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Бекіров А.Е.			Звіт з лабораторної роботи		Літ.	Арк.
Перевір.		Піонтківський В.І						1
Керівник							Аркушів	6
Н. контр.							ФІКТ Гр. ВТ-22-1	
Зав. каф.								

Лістинг:

```
package com.education.ztu;
```

```
public class Operation {
```

```
    public static int addition(int... numbers) {
```

```
        int sum = 0;
```

```
        for (int num : numbers) {
```

```
            sum += num;
```

```
        }
```

```
        return sum;
```

```
    }
```

```
    public static int subtraction(int... numbers) {
```

```
        int result = numbers[0];
```

```
        for (int i = 1; i < numbers.length; i++) {
```

```
            result -= numbers[i];
```

```
        }
```

```
        return result;
```

```
    }
```

```
    public static int multiplication(int... numbers) {
```

```
        int result = 1;
```

```
        for (int num : numbers) {
```

```
            result *= num;
```

```
        }
```

```
        return result;
```

```
    }
```

```
    public static double division(int a, int b) {
```

```
        return (double) a / b;
```

```
    }
```

```
    public static double average(int... numbers) {
```

```
        return (double) addition(numbers) / numbers.length;
```

```
    }
```

```
    public static int maximum(int... numbers) {
```

```
        int max = Integer.MIN_VALUE;
```

```
        for (int num : numbers) {
```

```
            max = Math.max(max, num);
```

```
        }
```

```
        return max;
```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Пр	Арк.
		Піонтківський В.І.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }

    public static int minimum(int... numbers) {
        int min = Integer.MAX_VALUE;
        for (int num : numbers) {
            min = Math.min(min, num);
        }
        return min;
    }
}

package com.education.ztu;

public class Car {
    private String model;

    public Car(String model) {
        this.model = model;
    }

    public class Engine {
        private boolean isWorking = false;

        public void startEngine() {
            isWorking = true;
            System.out.println("Engine started.");
        }

        public void stopEngine() {
            isWorking = false;
            System.out.println("Engine stopped.");
        }

        public boolean isEngineWorks() {
            return isWorking;
        }
    }

    @Override
    public String toString() {
        return "Машина{" + "модель=" + model + "\" + '\"';
    }
}

```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Лр	Арк.
		Піонтківський В.І.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
}
package com.education.ztu;
import com.education.ztu.enums.Gender;
import com.education.ztu.enums.Location;

public class Employee extends Person {
    private Car car;

    public Employee(String firstName, String lastName, int age, Location location,
Gender gender, Car car) {
        super(firstName, lastName, age, location, gender);
        this.car = car;
    }

    @Override
    public String getOccupation() {
        return "Працівник за машиною: " + car;
    }
}
package com.education.ztu;

import com.education.ztu.enums.Gender;
import com.education.ztu.enums.Location;

public abstract class Person implements Human {
    private static int counter = 0;

    protected String firstName;
    protected String lastName;
    protected int age;
    protected Location location;
    protected Gender gender;

    static {
        counter = 0;
    }

    {
        firstName = "Олег";
        lastName = "Іванько";
        age = 18;
    }
}

```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Лр	Арк.
		Піонтківський В.І.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        location = Location.URBAN;
        gender = Gender.MALE;
        counter++;
    }

    public Person() {
    }

    public Person(String firstName, String lastName, int age, Location location, Gender
gender) {
        this.firstName = firstName;
        this.lastName = lastName;
        this.age = age;
        this.location = location;
        this.gender = gender;
    }

    public static int getCounter() {
        return counter;
    }

    public static void showCounter() {
        System.out.println("Instances created: " + counter);
    }

    public abstract String getOccupation();

    public String getFullInfo() {
        return String.format("Name: %s %s, Age: %d, Location: %s, Gender: %s",
            firstName, lastName, age, location, gender);
    }

    @Override
    public void sayFullName() {
        System.out.println("Full Name: " + firstName + " " + lastName);
    }

    @Override
    public void sayAge() {
        System.out.println("Age: " + age);
    }

```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Лр	Арк.
		Піонтківський В.І.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

```

@Override
public void sayLocation() {
    System.out.println("Location: " + location);
}

@Override
public void sayGender() {
    System.out.println("Gender: " + gender);
}
}

package com.education.ztu;
import com.education.ztu.enums.Gender;
import com.education.ztu.enums.Location;

public class Student extends Person {
    private String university;

    public Student(String firstName, String lastName, int age, Location location, Gender
gender, String university) {
        super(firstName, lastName, age, location, gender);
        this.university = university;
    }

    @Override
    public String getOccupation() {
        return "Студент в " + university;
    }
}

package com.education.ztu;
import com.education.ztu.enums.Gender;
import com.education.ztu.enums.Location;

public class Teacher extends Person {
    private Car car;

    public Teacher(String firstName, String lastName, int age, Location location, Gender
gender, Car car) {
        super(firstName, lastName, age, location, gender);
        this.car = car;
    }

    @Override

```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Лр	Арк.
		Піонтківський В.І.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    public String getOccupation() {
        return "Машина викладача: " + car;
    }
}

package com.education.ztu;

public interface Human {
    void sayFullName();
    void sayAge();
    void sayLocation();
    void sayGender();

    default void whoIam() {
        System.out.println("Я - людина");
    }
}

```

Результат:

```

Engine started.
Engine status: Working
Engine stopped.
Full Name: Коля Петренко
Full Name: Лена Акименко
Full Name: Олег Бульба
Студент в ZTU

```

Лабораторна робота №3

Тема: Використання узагальнень (generics). Клонування та порівняння об'єктів.

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Лр	Арк.
		Піонтківський В.І.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 1. Відкрити заготовлений проект з реалізованою базовою функціональністю.

Завдання 2. За допомогою узагальнень (generics) встановити такі обмеження:

- до команди можна додавати тільки учасників, що відносяться до одної ліги (Schoolar, Student або Employee).
- грати між собою можуть тільки команди з учасниками одної ліги (тобто команда студентів може грати тільки іншою командою студентів).
- продемонструвати створення команд, гравців, додавання гравців до команд, гри між ними.

Завдання 3. Клонування:

- для класу Participant імплементувати інтерфейс Cloneable та перевизначити метод clone.
- для класу Participant перевизначити методи hashCode та equals.
- для класу Participant та його підкласів перевизначити метод toString.
- для класу Team Реалізувати глибоке клонування через статичний метод або конструктор копіювання.
- продемонструвати клонування та використання методів hashCode, equals та toString.

Завдання 4. Порівняння:

- для класу Participant імплементувати інтерфейс Comparable та перевизначити метод compareTo для сортування учасників по імені.
- створити Comparator для порівняння учасників по віку.
- *створити компаратор з пріоритетом використовуючи можливості Java 8 (спочатку порівняння по імені, а потім по віку).
- продемонструвати роботу порівнянь на прикладі сортування учасників команд.

Завдання 5. В GitLab проекті Java_labs_ztu, створити директорію Lab_3 та запусити в Lab_3 виконану лабораторну роботу. Надати доступ для перевірки викладачу.

Лістинг:

```
package com.education.ztu.game;
```

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

```
        Schoolar schoolar1 = new Schoolar("Ivan", 13);  
        Schoolar schoolar2 = new Schoolar("Mariya", 15);  
        Student student1 = new Student("Mykola", 20);  
        Student student2 = new Student("Viktoria", 21);
```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Лр	Арк.
		Піонтківський В.				8
Змн.	Арк.	№ докум.	Підпис	Дата		


```
Employee employee1 = new Employee("Andriy", 28);
Employee employee2 = new Employee("Oksana", 25);
```

```
Team<Scholar> scholarTeam = new Team<>("Dragon");
scholarTeam.addNewParticipant(scholar1);
scholarTeam.addNewParticipant(scholar2);
```

```
Team<Student> studentTeam = new Team<>("Vpered");
studentTeam.addNewParticipant(student1);
studentTeam.addNewParticipant(student2);
```

```
Team<Employee> employeeTeam = new Team<>("Robotyagi");
employeeTeam.addNewParticipant(employee1);
employeeTeam.addNewParticipant(employee2);
```

```
Team<Scholar> scholarTeam2 = new Team<>("Rozumnyky");
Scholar scholar3 = new Scholar("Sergey", 12);
Scholar scholar4 = new Scholar("Olga", 14);
scholarTeam2.addNewParticipant(scholar3);
scholarTeam2.addNewParticipant(scholar4);
```

```
scholarTeam.playWith(scholarTeam2);
scholarTeam.playWith(employeeTeam);
```

```
}
```

```
}
```

```
package com.education.ztu;
```

```
import com.education.ztu.game.*;
```

```
import java.util.Arrays;
```

```
import java.util.List;
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        Scholar scholar1 = new Scholar("Ivan", 13);
```

```
        Scholar scholar2 = new Scholar("Mariya", 15);
```

```
        Scholar clonedScholar = (Scholar) scholar1.clone();
```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Лр	Арк.
		Піонтківський В.І.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

```
System.out.println("Original: " + scholar1);
System.out.println("Cloned: " + clonedScholar);
```

```
System.out.println("hashCode of Ivan: " + scholar1.hashCode());
System.out.println("hashCode of cloned Ivan: " + clonedScholar.hashCode());
System.out.println("Are they equal? " + scholar1.equals(clonedScholar));
```

```
List<Participant> participants = Arrays.asList(
    new Scholar("Ivan", 13),
    new Scholar("Mariya", 15),
    new Student("Mykola", 20),
    new Student("Viktoria", 21),
    new Employee("Andriy", 28),
    new Employee("Oksana", 25)
);
```

```
participants.sort(Participant::compareTo);
System.out.println("\nParticipants sorted by name:");
participants.forEach(System.out::println);
```

```
participants.sort(Participant.ageComparator);
System.out.println("\nParticipants sorted by age:");
participants.forEach(System.out::println);
```

```
participants.sort(Participant.nameAndAgeComparator);
System.out.println("\nParticipants sorted by name and age:");
participants.forEach(System.out::println);
}
}
package com.education.ztu;
```

```
import com.education.ztu.game.*;
```

```
import java.util.List;
```

```
public class Main2 {
    public static void main(String[] args) {
```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Лр	Арк.
		Піонтківський В.І.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

```
Team<Schoolar> schoolarTeam = new Team<>("Dragon");
Team<Schoolar> scollarTeam2 = new Team<>("Rozumnyky");
```

```
Schoolar schoolar1 = new Schoolar("Ivan", 13);
Schoolar schoolar2 = new Schoolar("Mariya", 15);
schoolarTeam.addNewParticipant(schoolar1);
schoolarTeam.addNewParticipant(schoolar2);
```

```
Schoolar schoolar3 = new Schoolar("Sergey", 12);
Schoolar schoolar4 = new Schoolar("Olga", 14);
scollarTeam2.addNewParticipant(schoolar3);
scollarTeam2.addNewParticipant(schoolar4);
```

```
System.out.println("\nSchoolar team sorted by name:");
schoolarTeam.getParticipants().sort(Participant::compareTo);
schoolarTeam.getParticipants().forEach(participant ->
System.out.println(participant.getName()));
```

```
System.out.println("\nSchoolar team sorted by age:");
schoolarTeam.getParticipants().sort(Participant.ageComparator);
schoolarTeam.getParticipants().forEach(participant ->
System.out.println(participant.getName()));
```

```
Team<Employee> employeeTeam = new Team<>("Robotyagi");
Employee employee1 = new Employee("Andriy", 28);
Employee employee2 = new Employee("Oksana", 25);
employeeTeam.addNewParticipant(employee1);
employeeTeam.addNewParticipant(employee2);
```

```
System.out.println("\nEmployee team sorted by name and age:");
employeeTeam.getParticipants().sort(Participant.nameAndAgeComparator);
employeeTeam.getParticipants().forEach(participant ->
System.out.println(participant.getName()));
```

```
schoolarTeam.playWith(scollarTeam2);
```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Лр	Арк.
		Піонтківський В.І.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        scholarTeam.playWith(employeeTeam);
    }
}

```

Результат:

```

To the team Dragon was added participant Ivan
To the team Dragon was added participant Mariya
To the team Rozumnyky was added participant Sergey
To the team Rozumnyky was added participant Olga

Scholar team sorted by name:
Ivan
Mariya

Scholar team sorted by age:
Ivan
Mariya

To the team Robotyagi was added participant Andriy
To the team Robotyagi was added participant Oksana

```

Лабораторна робота №4

Тема: Класи String, StringBuffer та StringBuilder. Локалізація та інтернаціоналізація.
Робота з датами.

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Пр	Арк.
		Піонтківський В.І.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 1. Створити консольний Java проект java_lab_4 з пакетом com.education.ztu

Завдання 2. Практика методів класу String:

- Напишіть метод, який приймає як параметр будь-який рядок, наприклад "I learn Java!!!".
- Роздрукувати останній символ рядка.
- Перевірити, чи закінчується ваш рядок підрядком "!!!".
- Перевірити, чи починається ваш рядок підрядком "I learn ".
- Перевірити, чи містить ваш рядок підрядок "Java".
- Знайти позицію підрядка "Java" у рядку "I learn Java!!!".
- Замінити всі символи "a" на "o".
- Перетворіть рядок на верхній регістр.
- Перетворіть рядок на нижній регістр.
- Вирізати рядок Java.

Завдання 3. Створити рядок за допомогою класу StringBuilder або StringBuffer та його методів:

- Дано два числа, наприклад, 4 і 36, необхідно скласти наступні рядки:
 $4 + 36 = 40$
 $4 - 36 = -32$
 $4 * 36 = 144$
- Використати метод `StringBuilder.append()`.
- Замініть символ "=" на слово "рівно". Використати методи `StringBuilder.insert()`, `StringBuilder.deleteCharAt()`.
- Замініть символ "=" на слово "рівно". Використати метод `StringBuilder.replace()`.
- Змінити послідовність розташування символів в рядку на протилежну. Використати метод `StringBuilder.reverse()`.
- Визначити довжину та `capacity`.

Завдання 4. Вивести у форматованому вигляді чек з купленими товарами використовуючи можливості класу `Formatter`:

Лістинг:

```
package com.education.ztu;
```

```
import java.util.Locale;
```

```
public class Main {  
    public static void main(String[] args) {  
        StringMethods.stringOperations("I learn Java!!!");  
        StringBuilderMethods.stringBuilderOperations(4, 36);  
        Receipt.printReceipt();  
    }  
}
```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Лр	Арк.
		Піонтківський В.І.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        Localization.printLocalizedReceipt(new Locale("ua", "UA"));
        DateTimeOperations.dateOperations();
    }
}
package com.education.ztu;

public class StringMethods {
    public static void stringOperations(String input) {
        System.out.println("Останній символ: " + input.charAt(input.length() - 1));
        System.out.println("Закінчується на '!!!': " + input.endsWith("!!!"));
        System.out.println("Починається на 'I learn ': " + input.startsWith("I learn "));
        System.out.println("Містить 'Java': " + input.contains("Java"));
        System.out.println("Позиція 'Java': " + input.indexOf("Java"));
        System.out.println("Замінено 'a' на 'o': " + input.replace('a', 'o'));
        System.out.println("Верхній регістр: " + input.toUpperCase());
        System.out.println("Нижній регістр: " + input.toLowerCase());
        System.out.println("Вирізати 'Java': " + input.substring(input.indexOf("Java"),
input.indexOf("Java") + 4));
    }
}
package com.education.ztu;

public class StringBuilderMethods {
    public static void stringBuilderOperations(int a, int b) {
        StringBuilder sb = new StringBuilder();

        sb.append(a).append(" + ").append(b).append(" = ").append(a + b);
        System.out.println(sb);

        sb.setLength(0);
        sb.append(a).append(" - ").append(b).append(" = ").append(a - b);
        System.out.println(sb);

        sb.setLength(0);
        sb.append(a).append(" * ").append(b).append(" = ").append(a * b);
        System.out.println(sb);

        sb.replace(sb.indexOf("="), sb.indexOf("=") + 1, "рівно");
        System.out.println(sb);

        sb.reverse();
        System.out.println("Реверс: " + sb);
    }
}

```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Лр	Арк.
		Піонтківський В.І.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        System.out.println("Довжина: " + sb.length() + ", capacity: " + sb.capacity());
    }
}
package com.education.ztu;

import java.util.Formatter;

public class Receipt {

    public static void printReceipt() {
        Formatter formatter = new Formatter();
        System.out.println("Дата та час покупки: 28.03.2019 13:25:12");
        System.out.println("=====");
        System.out.println("№    Товар    Категорія    Ціна");
        System.out.println("=====");
        formatter.format("%d.  %-10s  %-15s  %.2f €\n", 1, "Джинси", "Жіночий одяг",
1500.78);
        formatter.format("%d.  %-10s  %-15s  %.2f €\n", 2, "Спідниця", "Жіночий одяг",
1000.56);
        formatter.format("%d.  %-10s  %-15s  %.2f €\n", 3, "Краватка", "Чоловічий одяг",
500.78);
        System.out.println(formatter);
        System.out.println("=====");
        System.out.println("Разом: 3002,34 €");
    }
}

receipt.date = Дата та час покупки: 28.03.2019 13:25:12
receipt.header = =====
receipt.item = %d.  %-10s  %-15s  %s €
receipt.footer = Разом: 3002,34 €
receipt.date = Date et heure d'achat : 28.03.2019 13:25:12
receipt.header = =====
receipt.item = %d.  %-10s  %-15s  %s €
receipt.footer = Total : 3002,34 €
receipt.date = Purchase date and time: 28.03.2019 13:25:12
receipt.header = =====
receipt.item = %d.  %-10s  %-15s  %s
receipt.footer = Total: 3002.34

```

Лабораторна робота №6

Тема: Обробка виключних ситуацій. Потoki вводу-виводу. Робота з файлами.
Лістинг:

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Лр	Арк.
		Піонтківський В.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 1. Створити консольний Java проект java_lab_6 з пакетом com.education.ztu. Створіть в корені проекту папку directory_for_files. Всі файли з якими ви будете працювати при виконанні завдань повинні знаходитись в ній.

Завдання 2. Перевірка логіну та паролю:

- Створити статичний метод checkCredentials, який приймає на вхід три параметри: login, password і confirmPassword.
- Login повинен містити лише латинські літери, цифри та знак підкреслення. Довжина login має бути меншою за 20 символів. Якщо login не відповідає цим вимогам, необхідно викинути WrongLoginException.
- Password повинен містити лише латинські літери, цифри та знак підкреслення. Довжина password має бути менше 20 символів. Також password і confirmPassword повинні бути рівними. Якщо password не відповідає цим вимогам, необхідно викинути WrongPasswordException.
- WrongPasswordException і WrongLoginException - користувацькі класи виключення з двома конструкторами - один за замовчуванням, другий приймає повідомлення виключення і передає його в конструктор класу Exception.
- Обробка винятків проводиться усередині методу.
- Використовуємо multi-catch block.
- Метод повертає true, якщо значення є вірними або false в іншому випадку.

Завдання 3. Запис звіту про покупки в текстовий файл та читання з нього:

- Перевикористати код для формування звіту з покупок з лабораторної роботи 4.
- Після покупки, записати звіт у файл, який містить інформацію про вміст кошика.
- Використовуємо клас FileWriter або PrintWriter для запису звіту.
- Використовуємо FileReader для читання звіту та відображення в консолі.
- Не використовувати try-with-resources.

Лістинг:

```
package com.education.ztu;
```

```
import java.io.*;
import java.util.zip.*;
import java.nio.file.*;
import java.util.regex.*;
```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Лр	Арк.
		Піонтківський В.І.				16
Змн.	Арк.	№ докум.	Підпис	Дата		


```

public class Lab6 {

    public static boolean checkCredentials(String login, String password, String
confirmPassword) {
        try {
            if (!login.matches("[a-zA-Z0-9_]{1,20}")) {
                throw new WrongLoginException("Login не відповідає вимогам.");
            }
            if (!password.matches("[a-zA-Z0-9_]{1,20}") ||
!password.equals(confirmPassword)) {
                throw new WrongPasswordException("Password не відповідає вимогам або
не співпадає з confirmPassword.");
            }
            return true;
        } catch (WrongLoginException | WrongPasswordException e) {
            System.out.println("Помилка: " + e.getMessage());
            return false;
        }
    }

    public static void writeReport(String content, String filePath) throws IOException {
        FileWriter writer = new FileWriter(filePath);
        writer.write(content);
        writer.close();
    }

    public static String readReport(String filePath) throws IOException {
        FileReader reader = new FileReader(filePath);
        BufferedReader bufferedReader = new BufferedReader(reader);
        StringBuilder content = new StringBuilder();
        String line;
        while ((line = bufferedReader.readLine()) != null) {
            content.append(line).append("\n");
        }
        reader.close();
        return content.toString();
    }

    public static void copyFile(String sourcePath, String destPath) throws IOException {
        try (FileInputStream fis = new FileInputStream(sourcePath);

```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Лр	Арк.
		Піонтківський В.І.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        FileOutputStream fos = new FileOutputStream(destPath)) {
        byte[] buffer = new byte[1024];
        int bytesRead;
        while ((bytesRead = fis.read(buffer)) != -1) {
            fos.write(buffer, 0, bytesRead);
        }
    }
}

```

```

public static void modifyFile(String filePath, long position, String text) throws
IOException {
    try (RandomAccessFile raf = new RandomAccessFile(filePath, "rw")) {
        raf.seek(position);
        raf.write(text.getBytes());
    }
}

```

```

public static void manageFiles(String directoryPath) throws IOException {
    File innerDir = new File(directoryPath + "/inner_directory");
    if (innerDir.mkdir()) {
        System.out.println("Папка створена: " + innerDir.getAbsolutePath());
    }
}

```

```

File file1 = new File(innerDir, "file1.txt");
File file2 = new File(innerDir, "file2.txt");
file1.createNewFile();
file2.createNewFile();

```

```

file1.delete();
File renamedDir = new File(directoryPath + "/renamed_inner_directory");
innerDir.renameTo(renamedDir);

```

```

File parentDir = new File(directoryPath);
for (File file : parentDir.listFiles()) {
    System.out.println(file.getName() + (file.isDirectory() ? " (папка)" : " (файл)") +
    ", розмір: " + file.length() + " байт");
}
}

```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Пр	Арк.
		Піонтківський В.І.				18
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public static void archiveFiles(String directoryPath, String zipFilePath) throws
IOException {
    try (ZipOutputStream zos = new ZipOutputStream(new
FileOutputStream(zipFilePath))) {
        Files.walk(Paths.get(directoryPath)).filter(Files::isRegularFile).forEach(path -> {
            try (FileInputStream fis = new FileInputStream(path.toFile())) {
                ZipEntry entry = new ZipEntry(path.getFileName().toString());
                zos.putNextEntry(entry);
                byte[] buffer = new byte[1024];
                int bytesRead;
                while ((bytesRead = fis.read(buffer)) != -1) {
                    zos.write(buffer, 0, bytesRead);
                }
            } catch (IOException e) {
                e.printStackTrace();
            }
        });
    }
}

```

```

public static void main(String[] args) {

```

```

    try {
        String directoryPath = "./directory_for_files";
        new File(directoryPath).mkdir();

```

```

        System.out.println("Перевірка логіну: " + checkCredentials("valid_login",
"password123", "password123"));

```

```

        String reportPath = directoryPath + "/report.txt";
        writeReport("Звіт про покупки", reportPath);
        System.out.println(readReport(reportPath));

```

```

        String copyPath = directoryPath + "/copy.txt";
        copyFile(reportPath, copyPath);

```

```

        modifyFile(copyPath, 10, "Додатковий текст");

```

```

        manageFiles(directoryPath);

```

```

        String zipPath = directoryPath + "/archive.zip";
        archiveFiles(directoryPath, zipPath);

```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Пр	Арк.
		Піонтківський В.І.				19
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

```

class WrongLoginException extends Exception {
    public WrongLoginException(String message) {
        super(message);
    }
}





```

```

class WrongPasswordException extends Exception {
    public WrongPasswordException(String message) {
        super(message);
    }
}

```

Результат:

 renamed_inner_directory
 archive.zip
 copy.txt
 report.txt

Лабораторна робота №7

Тема: Багатопоточне програмування в Java

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Пр	Арк.
		Піонтківський В.І.				20
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 1. Створити консольний Java проект java_lab_7 з пакетом com.education.ztu.

Завдання 2. Створити клас, що розширює Thread:

- Створити клас MyThread, що розширює Thread.
- Перевизначити метод run(). У циклі for вивести на консоль повідомлення «Я люблю програмувати!!!» 100 разів.
- Створити екземпляр класу та запустити новий потік.
- Вивести ім'я створеного потоку, його пріоритет, перевірити чи він живий, чи є потоком демоном.
- Змінити ім'я, пріоритет створеного потоку та вивести в консоль оновлені значення.
- Після завершення роботи створеного потоку (використати метод join()) вивести ім'я головного потоку, та його пріоритет.
- Відобразити в консолі, коли ваш потік буде в стані NEW, RUNNUNG, TERMINATED.

Завдання 3. Створити клас, що реалізує інтерфейс Runnable для виводу в консоль чисел від 0 до 10000, що діляться на 10 без залишку:

- Створити клас MyRunnable, який реалізує інтерфейс Runnable.
- Імплементувати метод run().
- Визначити умову, якщо потік хочуть перервати, то завершити роботу потоку та вивести повідомлення «Розрахунок завершено!!!»
- Створити три потоки, які виконують завдання друку значень.
- Використовуємо статичний метод Thread.sleep(), щоб зробити паузу на 2 секунди для головного потоку, а після цього викликати для створених потоків метод interrupt().

Завдання 4. Створити клас, що реалізує інтерфейс Runnable для вививедення арифметичної прогресії від 1 до 100 з кроком 1:

- Створити клас, який реалізує інтерфейс Runnable.
- Створити об'єкт зі статичною змінною result для збереження значення арифметичної прогресії.
- Перевизначити метод run(). Створити цикл for. У циклі виводимо через пробіл значення змінної result. Та додаємо наступне значення до змінної result та чекаємо 0,2 секунду.
- Забезпечити коректну роботу використовуючи синхронізований метод.

Лістинг:

```
package com.education.ztu;
```

```
import java.util.Scanner;
```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Пр	Арк.
		Піонтківський В.І.				21
Змн.	Арк.	№ докум.	Підпис	Дата		

```

class SharedData {
    private String message;
    private boolean hasMessage = false;

    public synchronized void writeMessage(String message) {
        while (hasMessage) {
            try {
                wait();
            } catch (InterruptedException e) {
                Thread.currentThread().interrupt();
            }
        }
        this.message = message;
        hasMessage = true;
        notify();
    }

    public synchronized String readMessage() {
        while (!hasMessage) {
            try {
                wait();
            } catch (InterruptedException e) {
                Thread.currentThread().interrupt();
            }
        }
        hasMessage = false;
        notify();
        return message;
    }
}

class Reader implements Runnable {
    private final SharedData sharedData;

    public Reader(SharedData sharedData) {
        this.sharedData = sharedData;
    }

    @Override
    public void run() {
        Scanner scanner = new Scanner(System.in);
        while (true) {

```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Пр	Арк.
		Піонтківський В.І.				22
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        System.out.print("Введіть повідомлення: ");
        String input = scanner.nextLine();
        if (input.equalsIgnoreCase("exit")) {
            sharedData.sendMessage("Кінець програми");
            break;
        }
        sharedData.sendMessage(input);
    }
    scanner.close();
}

class Printer implements Runnable {
    private final SharedData sharedData;

    public Printer(SharedData sharedData) {
        this.sharedData = sharedData;
    }

    @Override
    public void run() {
        while (true) {
            String message = sharedData.readMessage();
            if (message.equals("Кінець програми")) {
                System.out.println("Printer завершує роботу.");
                break;
            }
            System.out.println("Printer отримав: " + message);
        }
    }
}

package com.education.ztu;

class ArithmeticProgression implements Runnable {
    private static int result = 1;

    public synchronized void printProgression() {
        for (int i = 1; i <= 100; i++) {
            System.out.print(result + " ");
            result += 1;
            try {
                Thread.sleep(200); // чекаємо 0.2 секунди
            }
        }
    }
}

```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Пр	Арк.
		Піонтківський В.І.				23
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        } catch (InterruptedException e) {
            Thread.currentThread().interrupt();
        }
    }
}

@Override
public void run() {
    printProgression();
}

public static void main(String[] args) {
    ArithmeticProgression task = new ArithmeticProgression();

    Thread thread1 = new Thread(task);
    Thread thread2 = new Thread(task);
    Thread thread3 = new Thread(task);

    thread1.start();
    thread2.start();
    thread3.start();
}
}

package com.education.ztu;

import java.util.Random;
import java.util.concurrent.*;

public class ArraySum {
    private static final int ARRAY_SIZE = 1_000_000;
    private static final int THREAD_COUNT = 5;
    private static final int[] numbers = new int[ARRAY_SIZE];

    public static void main(String[] args) {

        Random random = new Random();
        for (int i = 0; i < ARRAY_SIZE; i++) {
            numbers[i] = random.nextInt(100);
        }

        long startTime = System.currentTimeMillis();

```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Пр	Арк.
		Піонтківський В.І.				24
Змн.	Арк.	№ докум.	Підпис	Дата		


```

int singleThreadSum = calculateSumSingleThread();
long endTime = System.currentTimeMillis();
System.out.println("Однопотокова сума: " + singleThreadSum);
System.out.println("Час виконання (однопотоковий): " + (endTime - startTime) +
" мс");

```

```

startTime = System.currentTimeMillis();
int multiThreadSum = calculateSumMultiThread();
endTime = System.currentTimeMillis();
System.out.println("Багатопотокова сума: " + multiThreadSum);
System.out.println("Час виконання (багатопотоковий): " + (endTime - startTime)
+ " мс");
}

```

```

private static int calculateSumSingleThread() {
    int sum = 0;
    for (int number : numbers) {
        sum += number;
    }
    return sum;
}

```

```

private static int calculateSumMultiThread() {
    ExecutorService executor = Executors.newFixedThreadPool(THREAD_COUNT);
    int chunkSize = ARRAY_SIZE / THREAD_COUNT;
    Future<Integer>[] futures = new Future[THREAD_COUNT];

    for (int i = 0; i < THREAD_COUNT; i++) {
        final int start = i * chunkSize;
        final int end = (i == THREAD_COUNT - 1) ? ARRAY_SIZE : start + chunkSize;
        futures[i] = executor.submit(() -> calculatePartialSum(start, end));
    }
}

```

```

int totalSum = 0;

```

```

try {
    for (Future<Integer> future : futures) {
        totalSum += future.get();
    }
}

```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Лр	Арк.
		Піонтківський В.І.				25
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
    } catch (InterruptedException | ExecutionException e) {
        e.printStackTrace();
    } finally {
        executor.shutdown();
    }

    return totalSum;
}

private static int calculatePartialSum(int start, int end) {
    int sum = 0;
    for (int i = start; i < end; i++) {
        sum += numbers[i];
    }
    return sum;
}
}

package com.education.ztu;

class MyRunnable implements Runnable {
    @Override
    public void run() {
        for (int i = 0; i <= 10000; i++) {
            if (i % 10 == 0) {
                System.out.println(i);
            }
            if (Thread.currentThread().isInterrupted()) {
                System.out.println("Розрахунок завершено!!!");
                return;
            }
        }
    }
}

public static void main(String[] args) throws InterruptedException {
    MyRunnable task = new MyRunnable();

    Thread thread1 = new Thread(task);
    Thread thread2 = new Thread(task);
    Thread thread3 = new Thread(task);

```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Лр	Арк.
		Піонтківський В.І.				26
Змн.	Арк.	№ докум.	Підпис	Дата		

```

thread1.start();
thread2.start();
thread3.start();

Thread.sleep(2000);

thread1.interrupt();
thread2.interrupt();
thread3.interrupt();
}
}
package com.education.ztu;

class MyThread extends Thread {
    public MyThread(String name) {
        super(name);
    }

    @Override
    public void run() {
        for (int i = 0; i < 100; i++) {
            System.out.println("Я люблю програмувати!!!");
        }
    }

    public static void main(String[] args) throws InterruptedException {

        MyThread myThread = new MyThread("InitialThread");
        System.out.println("Статус потоку: NEW");

        myThread.start();
        System.out.println("Ім'я: " + myThread.getName());
        System.out.println("Пріоритет: " + myThread.getPriority());
        System.out.println("Чи живий: " + myThread.isAlive());
        System.out.println("Чи активний: " + myThread.isDaemon());
        System.out.println("Статус потоку: RUNNING");

        myThread.setName("UpdatedThread");
        myThread.setPriority(Thread.MAX_PRIORITY);
        System.out.println("Оновлене ім'я: " + myThread.getName());
    }
}

```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Пр	Арк.
		Піонтківський В.І.				27
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        System.out.println("Оновлений пріоритет: " + myThread.getPriority());

        myThread.join();
        System.out.println("Статус потоку: TERMINATED");

        Thread mainThread = Thread.currentThread();
        System.out.println("Ім'я головного потоку: " + mainThread.getName());
        System.out.println("Пріоритет головного потоку: " + mainThread.getPriority());
    }
}
package com.education.ztu;

import java.util.Scanner;

class SharedData {
    private String message;
    private boolean hasMessage = false;

    public synchronized void writeMessage(String message) {
        while (hasMessage) {
            try {
                wait();
            } catch (InterruptedException e) {
                Thread.currentThread().interrupt();
            }
        }
        this.message = message;
        hasMessage = true;
        notify();
    }

    public synchronized String readMessage() {
        while (!hasMessage) {
            try {
                wait();
            } catch (InterruptedException e) {
                Thread.currentThread().interrupt();
            }
        }
        hasMessage = false;
    }
}

```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Пр	Арк.
		Піонтківський В.І.				28
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        notify();
        return message;
    }
}

class Reader implements Runnable {
    private final SharedData sharedData;

    public Reader(SharedData sharedData) {
        this.sharedData = sharedData;
    }

    @Override
    public void run() {
        Scanner scanner = new Scanner(System.in);
        while (true) {
            System.out.print("Введіть повідомлення: ");
            String input = scanner.nextLine();
            if (input.equalsIgnoreCase("exit")) {
                sharedData.writeMessage("Кінець програми");
                break;
            }
            sharedData.writeMessage(input);
        }
        scanner.close();
    }
}

```

```

class Printer implements Runnable {
    private final SharedData sharedData;

    public Printer(SharedData sharedData) {
        this.sharedData = sharedData;
    }

    @Override
    public void run() {
        while (true) {
            String message = sharedData.readMessage();
            if (message.equals("Кінець програми")) {
                System.out.println("Printer завершує роботу.");
                break;
            }
        }
    }
}

```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Пр	Арк.
		Піонтківський В.І.				29
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
    System.out.println("Printer отримав: " + message);
  }
}

```

Результат:

Лабораторна робота №8

Тема: Лямбда вирази. Функціональні інтерфейси. Посилання на методи. Stream API.

Лістинг:

```
package com.education.ztu;
```

```
import java.util.*;
```

```
import java.util.function.*;
```

```
import java.util.stream.*;
```

```
public class Lab8 {
```

```
    @FunctionalInterface
```

```
    interface Printable {
```

```
        void print();
```

```
    }
```

```
    public static void main(String[] args) {
```

```
        Printable printable = () -> System.out.println("Лямбда вираз для інтерфейсу Printable.");
```

```
        printable.print();
```

```
        Predicate<String> isNumeric = s -> {
```

```
            try {
```

```
                Double.parseDouble(s);
```

```
                return true;
```

```
            } catch (NumberFormatException e) {
```

```
                return false;
```

```
            }
```

```
        };
```

```
        System.out.println("123 is numeric: " + isNumeric.test("123"));
```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Лр	Арк.
		Піонтківський В.І.				30
Змн.	Арк.	№ докум.	Підпис	Дата		

```

Consumer<String> startLesson = s -> System.out.println("Пара почалася о 8:30: "
+ s);
Consumer<String> endLesson = s -> System.out.println("Пара закінчилася о 9:50:
" + s);
startLesson.andThen(endLesson).accept("Java Programming");

```

```

Supplier<String> upperCaseSupplier = () -> "Це речення у верхньому
перістрі".toUpperCase();
System.out.println(upperCaseSupplier.get());

```

```

Function<String, Integer> multiplyNumbers = input -> {
    return Arrays.stream(input.split(" "))
        .mapToInt(Integer::parseInt)
        .reduce(1, (a, b) -> a * b);
};
System.out.println("Добуток чисел: " + multiplyNumbers.apply("2 3 4"));

```

```

List<Product> products = Arrays.asList(
    new Product("Phone", "Apple", 999, 10),
    new Product("Laptop", "Lenovo", 1200, 5),
    new Product("Headphones", "JBL", 300, 20),
    new Product("Monitor", "Samsung", 400, 7)
);

```

```

products.stream().map(Product::getBrand).distinct().forEach(System.out::println);

```

```

products.stream().filter(p -> p.getPrice() <
1000).limit(2).forEach(System.out::println);

```

```

int totalCount = products.stream().mapToInt(Product::getCount).sum();
System.out.println("Сума всіх товарів: " + totalCount);

```

```

Map<String, List<Product>> groupedByBrand =
products.stream().collect(Collectors.groupingBy(Product::getBrand));
System.out.println(groupedByBrand);

```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Пр	Арк.
		Піонтківський В.І.				31
Змн.	Арк.	№ докум.	Підпис	Дата		

```
List<Product> sortedByPrice = products.stream()
    .sorted(Comparator.comparingInt(Product::getPrice))
    .collect(Collectors.toList());
System.out.println(sortedByPrice);
```

```
products.stream().map(Product::getName).forEach(System.out::println);
```

```
Integer maxNumber = Arrays.asList(1, 2, 3, 4,
5).stream().max(Integer::compare).orElseThrow(() -> new RuntimeException("Числа
відсутні"));
System.out.println("Максимальне число: " + maxNumber);
}
```

```
static class Product {
    private String name;
    private String brand;
    private int price;
    private int count;

    public Product(String name, String brand, int price, int count) {
        this.name = name;
        this.brand = brand;
        this.price = price;
        this.count = count;
    }

    public String getName() {
        return name;
    }

    public String getBrand() {
        return brand;
    }

    public int getPrice() {
        return price;
    }

    public int getCount() {
```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Лр	Арк.
		Піонтківський В.І.				32
Змн.	Арк.	№ докум.	Підпис	Дата		


```

        return count;
    }

    @Override
    public String toString() {
        return "Product{" +
            "name='" + name + '\'' +
            ", brand='" + brand + '\'' +
            ", price=" + price +
            ", count=" + count +
            '}';
    }
}

```

Результат:

```

Лямбда вираз для інтерфейсу Printable.
123 is numeric: true
Пара почалася о 8:30: Java Programming
Пара закінчилася о 9:50: Java Programming
ЦЕ РЕЧЕННЯ У ВЕРХНЬОМУ РЕГІСТРІ
Добуток чисел: 24
Apple
Lenovo
JBL
Samsung
Product{name='Phone', brand='Apple', price=999, count=10}
Product{name='Headphones', brand='JBL', price=300, count=20}
Сума всіх товарів: 42

```

Лабораторна робота №9

Тема: Регулярні вирази. Рефлексія. Анотації.

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Лр	Арк.
		Піонтківський В.І.				33
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 1. Створити консольний Java проект `java_lab_9` з пакетом `com.education.ztu`.

Завдання 2. Робота з регулярними виразами:

- Використати власний текст, що містить дані 5-10 співробітників компанії (ПІБ, вік, посада, досвід роботи, адреса, email, телефон і т. д.)
- Знайти в тексті всі номери телефонів та emailи.
- Змінити формати відображення дат народження (наприклад: 20.05.1995 на 1995-05-20)
- Змінити посади кільком співробітникам.
- Результати роботи відобразити в консолі.

Завдання 3. Робота з користувацьким класом методами Reflection API:

- Створити власний клас в якому міститимуться публічні та приватні поля, конструктори і методи з аргументами та без.
- Отримати об'єкт класу `Class` для користувацького класу трьома способами.
- Отримати всі поля, методи, конструктори, що визначені тільки в цьому класі (не враховувати наслідування) та вивести ці значення в консоль (назву, типи параметрів та значення, що повертається).
- Створити екземпляр класу.
- Викликати метод класу.
- Попрацювати з приватним полем (встановити та отримати значення)
- Результати роботи відобразити в консолі.

Завдання 4. Створення власної анотації:

- Створити власну анотацію, задати їй необхідні поля та значення за замовчуванням для них.
- Встановити їй обмеження застосування через анотацію `@Target`
- Встановити їй політику утримання через анотацію `@Retention`
- Додати анотацію до відповідного об'єкту в коді.
- Отримати дані анотації з об'єкту та вивести в консоль.

Лістинг:

```
package com.education.ztu;
```

```
import java.lang.annotation.*;
```

```
import java.lang.reflect.*;
```

```
public class Task4 {  
    public static void run() {
```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Пр	Арк.
		Піонтківський В.І.				34
Змн.	Арк.	№ докум.	Підпис	Дата		

```

Class<?> clazz = AnnotatedClass.class;

if (clazz.isAnnotationPresent(Info.class)) {
    Info info = clazz.getAnnotation(Info.class);
    System.out.println("Дані анотації:");
    System.out.println("Автор: " + info.author());
    System.out.println("Дата: " + info.date());
}
}
}

@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.TYPE)
@interface Info {
    String author() default "Unknown";
    String date();
}

@Info(author = "Student", date = "2024-06-10")
class AnnotatedClass {
    public void doSomething() {
        System.out.println("Doing something...");
    }
}

package com.education.ztu;

import java.lang.reflect.*;

public class Task3 {
    public static void run() {
        try {
            Class<?> clazz = Employee.class;

            System.out.println("Поля класу:");
            for (Field field : clazz.getDeclaredFields()) {
                System.out.println(field.getName() + " : " + field.getType());
            }

            System.out.println("\nМетоди класу:");
            for (Method method : clazz.getDeclaredMethods()) {

```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Пр	Арк.
		Піонтківський В.І.				35
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        System.out.println(method.getName());
    }

    Constructor<?> constructor = clazz.getConstructor(String.class, int.class);
    Object employee = constructor.newInstance("Gosha Yra", 35);

    Method displayMethod = clazz.getMethod("displayInfo");
    displayMethod.invoke(employee);

    Field ageField = clazz.getDeclaredField("age");
    ageField.setAccessible(true);
    ageField.set(employee, 40);

    System.out.println("\nПісля зміни приватного поля:");
    displayMethod.invoke(employee);

    } catch (Exception e) {
        e.printStackTrace();
    }
}

class Employee {
    private String name;
    private int age;

    public Employee(String name, int age) {
        this.name = name;
        this.age = age;
    }

    public void displayInfo() {
        System.out.println("Name: " + name + ", Age: " + age);
    }
}

package com.education.ztu;

import java.util.regex.*;

```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Пр	Арк.
		Піонтківський В.І.				36
Змн.	Арк.	№ докум.	Підпис	Дата		

```
public class Task2 {  
    public static void run() {  
        String text = ""  
            Гоша Іваненко, 30 років, розробник, email: gosha.ivanenko@example.com,  
            телефон: +38093658565, дата: 21.01.1994  
            Юля Стенаненко, 52 років, дизайнер, email: yulia.stepanenko@example.com,  
            телефон: +380730274020, дата: 20.09.1988  
            "";  
  
        System.out.println("Оригінальний текст:");  
        System.out.println(text);  
  
        System.out.println("\nНомери телефонів:");  
        Pattern phonePattern = Pattern.compile("\\+380\\d{9}");  
        Matcher phoneMatcher = phonePattern.matcher(text);  
        while (phoneMatcher.find()) {  
            System.out.println(phoneMatcher.group());  
        }  
  
        System.out.println("\nEmail адреси:");  
        Pattern emailPattern = Pattern.compile("\\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]  
[+\\.][A-Za-z]{2,}\\b");  
        Matcher emailMatcher = emailPattern.matcher(text);  
        while (emailMatcher.find()) {  
            System.out.println(emailMatcher.group());  
        }  
  
        String updatedText = text.replaceAll("(\\d{2})\\.\\.\\. (\\d{2})\\.\\.\\. (\\d{4})", "$3-$2-$1");  
        System.out.println("\nТекст з оновленими датами:");  
        System.out.println(updatedText);  
  
        String updatedRoles = updatedText.replaceAll("розробник", "тестувальник")  
            .replaceAll("дизайнер", "аналітик");  
        System.out.println("\nТекст з оновленими посадами:");  
        System.out.println(updatedRoles);  
    }  
}
```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Лр	Арк.
		Піонтківський В.А.				37
Змн.	Арк.	№ докум.	Підпис	Дата		

```
package com.education.ztu;
```

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("=== Виконання Завдання 2 ===");  
        Task2.run();  
  
        System.out.println("\n=== Виконання Завдання 3 ===");  
        Task3.run();  
  
        System.out.println("\n=== Виконання Завдання 4 ===");  
        Task4.run();  
    }  
}
```

Результат:

=== Виконання Завдання 2 ===

Оригінальний текст:

Гоша Іваненко, 30 років, розробник, email: gosha.ivanenko@example.com, телефон: +38093658565, дата: 21.01.1994

Юля Стенаненко, 52 років, дизайнер, email: yulia.stepanenko@example.com, телефон: +380730274020, дата: 20.09.1988

Номери телефонів:

+380730274020

Email адреси:

gosha.ivanenko@example.com

yulia.stepanenko@example.com

Текст з оновленими датами:

Гоша Іваненко, 30 років, розробник, email: gosha.ivanenko@example.com, телефон: +38093658565, дата: 1994-01-21

Юля Стенаненко, 52 років, дизайнер, email: yulia.stepanenko@example.com, телефон: +380730274020, дата: 1988-09-20

Текст з оновленими посадами:

Гоша Іваненко, 30 років, тестувальник, email: gosha.ivanenko@example.com, телефон: +38093658565, дата: 1994-01-21

Юля Стенаненко, 52 років, аналітик, email: yulia.stepanenko@example.com, телефон: +380730274020, дата: 1988-09-20

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Пр	Арк.
		Піонтківський В.І.				38
Змн.	Арк.	№ докум.	Підпис	Дата		

=== Виконання Завдання 3 ===

Поля класу:

name : class java.lang.String

age : int

Методи класу:

displayInfo

Name: Gosha Yra, Age: 35

Після зміни приватного поля:

Name: Gosha Yra, Age: 40

=== Виконання Завдання 4 ===

Дані анотації:

Автор: Student

Дата: 2024-06-10

Лабораторна робота №12

Тема: Інструменти тестування Junit та Mockito

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Пр	Арк.
		Піонтківський В.І.				39
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 1. Створити maven Java проект java_lab_12 з пакетом com.education.ztu. Додайте необхідні залежності до проекту (в pom.xml): junit 4.12, mockito-core 1.9.5, assertj-core 3.11.1, JUnitParams 1.1.0
Посилання: <https://mvnrepository.com/>

Завдання 2. Додати до створеного проекту код, який треба покрити тестами. Можете використати код з лабораторних робіт 2 чи 3 або код якогось іншого вашого java проекту.

Завдання 3. Тестування за допомогою бібліотеки Junit.
Створіть відповідні тестові класи та напишіть тест-кейси.
Ваші тест-кейси повинні використовувати такі анотації: @BeforeClass, @AfterClass, @Before, @After, @Test, @Ignore
Ваші тест-кейси повинні використовувати такі методи: fail, assertTrue, assertsEquals, assertNull, assertNotNull, assertSame, assertNotSame,
Перевірте позитивні, негативні випадки та випадки, які генерують виключні ситуації.
Використайте Rules (TemporaryFolder, ExpectedException, TestName, Timeout, ErrorCollector, Verifier)
Використайте групування тестів @Suite.SuiteClasses та параметризацію @Parameters

Завдання 4. Тестування за допомогою фреймворка Mockito.
Створіть відповідні тестові класи та проанотуйте їх анотацією @RunWith(MockitoJUnitRunner.class)
Напишіть тест-кейси. Ваші тест-кейси повинні використовувати такі методи та анотації:

- @Mock, @Spy, @Captor, @InjectMocks або аналогічні їм статичні методи класу Mockito
- when(mock).thenReturn(value), when(mock).thenThrow(), doReturn(value).when(mock).methodName(params)
- verify() в комбінації з atLeast, atLeastOnce, atMost, times, never

Лістинг:

```
package com.education.ztu;
```

```
import static org.junit.Assert.*;
import org.junit.*;
import org.junit.rules.*;
import java.util.*;
import java.util.stream.*;
```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Лр	Арк.
		Піонтківський В.І.				40
Змн.	Арк.	№ докум.	Підпис	Дата		


```

public class Lab8Test {

    @Rule
    public ExpectedException expectedException = ExpectedException.none();

    private Lab8.Product product;

    @BeforeClass
    public static void setupBeforeClass() {
        System.out.println("Тести починаються...");
    }

    @AfterClass
    public static void tearDownAfterClass() {
        System.out.println("Тести завершилися.");
    }

    @Before
    public void setup() {
        product = new Lab8.Product("Phone", "Apple", 999, 10);
    }

    @After
    public void tearDown() {
        product = null;
    }

    @Test
    public void testProductCreation() {
        assertNotNull("Об'єкт продукту не повинен бути null", product);
        assertEquals("Phone", product.getName());
        assertEquals("Apple", product.getBrand());
    }

    @Test
    public void testMultiplyNumbersFunction() {
        String input = "2 3 4";
        int result = Arrays.stream(input.split(" "))
            .mapToInt(Integer::parseInt)
            .reduce(1, (a, b) -> a * b);
        assertEquals(24, result);
    }
}

```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Пр	Арк.
		Піонтківський В.І.				41
Змн.	Арк.	№ докум.	Підпис	Дата		

```

@Test
public void testOptionalMaxNumber() {
    Integer max = Arrays.asList(1, 2, 3, 4, 5).stream()
        .max(Integer::compare)
        .orElseThrow(() -> new RuntimeException("Числа відсутні"));
    assertEquals(Integer.valueOf(5), max);
}

@Test
public void testStreamGrouping() {
    List<Lab8.Product> products = Arrays.asList(
        new Lab8.Product("Phone", "Apple", 999, 10),
        new Lab8.Product("Laptop", "Dell", 1200, 5)
    );
    Map<String, List<Lab8.Product>> grouped = products.stream()
        .collect(Collectors.groupingBy(Lab8.Product::getBrand));
    assertTrue(grouped.containsKey("Apple"));
    assertTrue(grouped.containsKey("Dell"));
}

@Test(expected = RuntimeException.class)
public void testOptionalException() {
    List<Integer> emptyList = Collections.emptyList();
    emptyList.stream().max(Integer::compare).orElseThrow(() -> new
RuntimeException("Числа відсутні"));
}
}

package com.education.ztu;

import org.junit.*;
import org.mockito.*;

import static org.junit.Assert.assertEquals;
import static org.mockito.Mockito.*;
import java.util.*;

public class Lab8MockitoTest {

    @Mock
    private Lab8.Product mockProduct;

```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Лр	Арк.
		Піонтківський В.І.				42
Змн.	Арк.	№ докум.	Підпис	Дата		

```

@Captor
private ArgumentCaptor<String> captor;

@InjectMocks
private List<Lab8.Product> productList = new ArrayList<>();

@Before
public void setup() {
    MockitoAnnotations.initMocks(this);
    productList.add(new Lab8.Product("Laptop", "Dell", 1200, 5));
}

@Test
public void testMockProduct() {
    when(mockProduct.getName()).thenReturn("MockPhone");
    assertEquals("MockPhone", mockProduct.getName());
    verify(mockProduct, times(1)).getName();
}

@Test
public void testSpyProductList() {
    List<Lab8.Product> spyList = spy(productList);
    spyList.add(new Lab8.Product("Monitor", "Samsung", 400, 7));
    verify(spyList, times(1)).add(any(Lab8.Product.class));
    assertEquals(2, spyList.size());
}
}

package com.education.ztu;

import java.util.*;
import java.util.function.*;
import java.util.stream.*;

public class Lab8 {

    // Завдання 2
    @FunctionalInterface
    interface Printable {
        void print();
    }

```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Пр	Арк.
		Піонтківський В.І.				43
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public static void main(String[] args) {

    Printable printable = () -> System.out.println("Лямбда вираз для інтерфейсу
Printable.");
    printable.print();

    Predicate<String> isNumeric = s -> {
        try {
            Double.parseDouble(s);
            return true;
        } catch (NumberFormatException e) {
            return false;
        }
    };
    System.out.println("123 is numeric: " + isNumeric.test("123"));

    Consumer<String> startLesson = s -> System.out.println("Пара почалася о 8:30: "
+ s);
    Consumer<String> endLesson = s -> System.out.println("Пара закінчилася о 9:50:
" + s);
    startLesson.andThen(endLesson).accept("Java Programming");

    Supplier<String> upperCaseSupplier = () -> "Це речення у верхньому
перістрі".toUpperCase();
    System.out.println(upperCaseSupplier.get());

    Function<String, Integer> multiplyNumbers = input -> {
        return Arrays.stream(input.split(" "))
            .mapToInt(Integer::parseInt)
            .reduce(1, (a, b) -> a * b);
    };
    System.out.println("Добуток чисел: " + multiplyNumbers.apply("2 3 4"));

    List<Product> products = Arrays.asList(
        new Product("Phone", "Apple", 999, 10),
        new Product("Laptop", "Dell", 1200, 5),
        new Product("Headphones", "Sony", 300, 20),
        new Product("Monitor", "Samsung", 400, 7)
    );

    products.stream().map(Product::getBrand).distinct().forEach(System.out::println);

```

```

products.stream().filter(p -> p.getPrice() <
1000).limit(2).forEach(System.out::println);

int totalCount = products.stream().mapToInt(Product::getCount).sum();
System.out.println("Сума всіх товарів: " + totalCount);

Map<String, List<Product>> groupedByBrand =
products.stream().collect(Collectors.groupingBy(Product::getBrand));
System.out.println(groupedByBrand);

List<Product> sortedByPrice = products.stream()
    .sorted(Comparator.comparingInt(Product::getPrice))
    .collect(Collectors.toList());
System.out.println(sortedByPrice);

products.stream().map(Product::getName).forEach(System.out::println);

Integer maxNumber = Arrays.asList(1, 2, 3, 4,
5).stream().max(Integer::compare).orElseThrow(() -> new RuntimeException("Числа
відсутні"));
System.out.println("Максимальне число: " + maxNumber);
}

static class Product {
    private String name;
    private String brand;
    private int price;
    private int count;

    public Product(String name, String brand, int price, int count) {
        this.name = name;
        this.brand = brand;
        this.price = price;
        this.count = count;
    }

    public String getName() {
        return name;
    }

    public String getBrand() {

```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Лр	Арк.
		Піонтківський В.І.				45
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return brand;
    }

    public int getPrice() {
        return price;
    }

    public int getCount() {
        return count;
    }

    @Override
    public String toString() {
        return "Product{" +
            "name=\"" + name + "\" +
            ", brand=\"" + brand + "\" +
            ", price=" + price +
            ", count=" + count +
            '}';
    }
}

```

Результат:

```

Лямбда вираз для інтерфейсу Printable.
123 is numeric: true
Пара почалася о 8:30: Java Programming
Пара закінчилася о 9:50: Java Programming
ЦЕ РЕЧЕННЯ У ВЕРХНЬОМУ РЕГІСТРІ
Добуток чисел: 24
Apple
Dell
Sony
Samsung
Product{name='Phone', brand='Apple', price=999, co
Product{name='Headphones', brand='Sony', price=300
Сума всіх товарів: 42
{Sony=[Product{name='Headphones', brand='Sony', pr
[Product{name='Headphones', brand='Sony', price=300
Phone
Laptop
Headphones
Monitor
Максимальне число: 5

```

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Пр	Арк.
		Піонтківський В.І.				46
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновок: в ході ЛР ми навчилися працювати з інструментами тестування junit та Mockito

Посилання на репозиторій з лабораторними:

<https://github.com/Alikhan9900/Java-Labs>

		Бекіров А.Е.			ДУ «Житомирська політехніка».24.121.03.000 – Лр	Арк.
		Піонтківський В.І.				47
Змн.	Арк.	№ докум.	Підпис	Дата		