

## Лабораторна робота №2

**Тема:** Створення структури класу заданої предметної області.

**Мета роботи:** створити ієрархію класів заданої предметної області, робота з статичними методами.

### Хід роботи:

**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 whoIAm (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).

					ДУ «Житомирська політехніка».22.121.16.000 – Лр2						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Кохан Т.О			Звіт з  лабораторної роботи №2			Літ.	Арк.	Аркушів	
Перевір.		Піонтківський В. І.								1	10
Керівник								ФІКТ Гр. ІПЗ-22-3			
Н. контр.											
Зав. каф.		Вакалюк Т.А.									

- в методі main класу Main створити об'єкти відповідних класів та продемонструвати роботу їх методів.
- продемонструвати роботу оператора instanceof.

### Лістинг програми:

#### Car.java:

```
package com.education.ztu.Task2;

public class Car {
    private String brand;
    private Engine engine;

    public Car(String brand) {
        this.brand = brand;
        this.engine = new Engine();
    }

    public Engine getEngine() {
        return engine;
    }

    public String getBrand() {
        return brand;
    }

    public boolean engineIsRunning() {
        return engine.isEngineWorks();
    }

    public class Engine {
        private boolean engineWorks = false;

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

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

        public boolean isEngineWorks() {
            return engineWorks;
        }
    }
}
```

#### Employee.java:

```
package com.education.ztu.Task2;

public class Employee extends Person {
    private String position;
    private String company;
    private Car car;
    private static int counter = 0;

    public Employee(String firstname, String lastname, int age, Gender gender,
```

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр2	Арк.
		Піонтківський В. І.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

```

Location location, String position, String company, Car car) {
    super(firstname, lastname, age, gender, location);
    this.position = position;
    this.company = company;
    this.car = car;
    counter++;
}

@Override
public void getOccupation() {
    System.out.println("I am an employee.");
}

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

public String getCompany() {
    return company;
}

public void setCompany(String company) {
    this.company = company;
}

public String getPosition() {
    return position;
}

public void setPosition(String position) {
    this.position = position;
}

public Car getCar() {
    return car;
}

public void setCar(Car car) {
    this.car = car;
}
}

```

## Gender.java:

```

package com.education.ztu.Task2;

public enum Gender {
    MALE, FEMALE;
}

```

## Human.java:

```

package com.education.ztu.Task2;

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

    default void whoIAm() {
        System.out.println(" I am a human!");
    }
}

```

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр2	Арк.
		Піонтківський В. І.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```
}  
}
```

## Location.java:

```
package com.education.ztu.Task2;  
  
public enum Location {  
    KIEV, ZHYTOMYR, VINNYTSYA, RIVNE;  
}
```

## Person.java:

```
package com.education.ztu.Task2;  
  
public abstract class Person implements Human {  
    private String firstname;  
    private String lastname;  
    private int age;  
    private Gender gender;  
    private Location location;  
    protected String fullInfo;  
    protected static int counter = 0;  
  
    {  
        firstname = "Tetiana";  
        lastname = "Kokhan";  
        age = 19;  
        gender = Gender.FEMALE;  
        location = Location.ZHYTOMYR;  
        fullInfo = "";  
        counter++;  
    }  
    public Person() {}  
  
    public Person(String firstname, String lastname, int age, Gender gender,  
Location location){  
        this.firstname = firstname;  
        this.lastname = lastname;  
        this.age = age;  
        this.gender = gender;  
        this.location = location;  
        this.fullInfo = firstname + " " + lastname + ", " + age + ", " + gender  
+ ", " + location;  
    }  
  
    public abstract void getOccupation();  
  
    @Override  
    public void sayName() {  
        System.out.println("Full Name: " + firstname + " " + lastname);  
    }  
    @Override  
    public void sayAge() {  
        System.out.println("Age: " + age);  
    }  
  
    @Override  
    public void sayLocation() {  
        System.out.println("Location: " + location);  
    }  
  
    @Override  
    public void sayGender() {  
        System.out.println("Gender: " + gender);  
    }  
}
```

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр2	Арк.
		Піонтківський В. І.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }

    public String getFullInfo() {
        return fullInfo;
    }

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

}

```

## Student.java:

```

package com.education.ztu.Task2;

public class Student extends Person {
    private String course;
    private String speciality;
    private String university;
    private static int counter = 0;

    public Student() {}

    public Student(String firstname, String lastname, int age, Gender gender,
Location location, String course, String speciality, String university) {
        super(firstname, lastname, age, gender, location);
        this.course = course;
        this.speciality = speciality;
        this.university = university;
        counter++;
    }

    @Override
    public void getOccupation() {
        System.out.println("I am a student.");
    }

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

    public String getCourse() {
        return course;
    }

    public void setCourse(String course) {
        this.course = course;
    }

    public String getSpeciality() {
        return speciality;
    }

    public void setSpeciality(String speciality) {
        this.speciality = speciality;
    }

    public String getUniversity() {
        return university;
    }

    public void setUniversity(String university) {
        this.university = university;
    }
}

```

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр2	Арк.
		Піонтківський В. І.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

```
}
```

## Teacher.java:

```
package com.education.ztu.Task2;

public class Teacher extends Person {
    private String subject;
    private String university;
    private Car car;
    private static int counter = 0;

    public Teacher() {}

    public Teacher(String firstname, String lastname, int age, Gender gender,
Location location, String subject, String university, Car car) {
        super(firstname, lastname, age, gender, location);
        this.subject = subject;
        this.university = university;
        this.car = car;
        counter++;
    }

    @Override
    public void getOccupation() {
        System.out.println("I am a teacher.");
    }

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

    public String getSubject() {
        return subject;
    }

    public void setSubject(String subject) {
        this.subject = subject;
    }

    public String getUniversity() {
        return university;
    }

    public void setUniversity(String university) {
        this.university = university;
    }

    public Car getCar() {
        return car;
    }

    public void setCar(Car car) {
        this.car = car;
    }
}
```

## Main.java:

```
package com.education.ztu.Task2;

public class Main {
```

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр2	Арк.
		Піонтківський В. І.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    public static void main(String[] args) {
        Car teacherCar = new Car("Nissan");
        Teacher teacher = new Teacher("Maria", "Swift", 30, Gender.FEMALE,
Location.KIEV, "English", "KPI", teacherCar);

        Car employeeCar = new Car("Volkswagen");
        Employee employee = new Employee("Ivan", "Ivanenko", 26, Gender.MALE,
Location.VINNYTSYA, "Lawyer", "Law firm", employeeCar);
        Student student = new Student("Denys", "Denysenko", 19, Gender.MALE,
Location.ZHYTOMYR, "Software engineering", "Software Engineering", "Zhytomyr
Polytechnic State University");

        System.out.println("==== Teacher =====");
        teacher.sayName();
        teacher.sayAge();
        teacher.sayLocation();
        teacher.sayGender();
        teacher.getOccupation();

        System.out.println("\n==== Employee =====");
        employee.sayName();
        employee.sayAge();
        employee.sayLocation();
        employee.sayGender();
        employee.getOccupation();

        System.out.println("\n==== Student =====");
        student.sayName();
        student.sayAge();
        student.sayLocation();
        student.sayGender();
        student.getOccupation();

        System.out.println("\n==== Teacher's car =====");
        System.out.println("Car: " + teacherCar.getBrand());
        System.out.println("Is the engine running? " +
teacherCar.engineIsRunning());
        teacherCar.getEngine().startEngine();
        System.out.println("Does the engine run after starting? " +
teacherCar.engineIsRunning());

        System.out.println("\n==== Count =====");
        Teacher.showCounter();
        Student.showCounter();
        Employee.showCounter();
        System.out.println("\n=====");
        if (teacher instanceof Person) {
            System.out.println("Teacher is a Person.");
        }
        if (employee instanceof Person) {
            System.out.println("Employee is a Person.");
        }
        if (student instanceof Person) {
            System.out.println("Student is a Person.");
        }
    }
}

```

### Результат виконання програми:

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр2	Арк.
		Піонтківський В. І.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

```

===== Teacher =====
Full Name: Maria Svift
Age: 30
Location: KIEV
Gender: FEMALE
I am a teacher.

===== Employee =====
Full Name: Ivan Ivanenko
Age: 26
Location: VINNYTSYA
Gender: MALE
I am an employee.

===== Student =====
Full Name: Denys Denysenko
Age: 19
Location: ZHYTOMYR
Gender: MALE
I am a student.

```

```

===== Teacher's car =====
Car: Nissan
Is the engine running? false
Engine started.
Does the engine run after starting? true

===== Count =====
Created teachers: 1
Created students: 1
Created employees: 1

=====
Teacher is a Person.
Employee is a Person.
Student is a Person.

```

Рис.1 Завдання 2

**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.

### Лістинг програми:

#### Location.java:

```

package com.education.ztu.Task3;

public enum Location {
    KIEV, ZHYTOMYR, RIVNE, Lviv;
}

```

#### Operation.java:

```

package com.education.ztu.Task3;

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];

```

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр2	Арк.
		Піонтківський В. І.				8
Змн.	Арк.	№ докум.	Підпис	Дата		



```

        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(double... numbers) {
        double result = numbers[0];
        for (int i = 1; i < numbers.length; i++) {
            result /= numbers[i];
        }
        return result;
    }
    public static double average(int... numbers) {
        int sum = addition(numbers);
        return (double) sum / numbers.length;
    }
    public static int maximum(int... numbers) {
        int max = numbers[0];
        for (int num : numbers) {
            if (num > max) {
                max = num;
            }
        }
        return max;
    }
    public static int minimum(int... numbers) {
        int min = numbers[0];
        for (int num : numbers) {
            if (num < min) {
                min = num;
            }
        }
        return min;
    }
}

```

## Main2.java:

```

package com.education.ztu.Task3;

public class Main2 {
    public static void main(String[] args) {
        System.out.println("Addition: " + Operation.addition(2, 4, 12));
        System.out.println("Subtraction: " + Operation.subtraction(100, 50));
        System.out.println("Multiplication: " + Operation.multiplication(4, 4,
4));
        System.out.println("Division: " + Operation.division(48.0, 2.0, 3.0));
        System.out.println("Average: " + Operation.average(2, 4, 6, 8));
        System.out.println("Maximum: " + Operation.maximum(7, 10, 1, 20, 16));
        System.out.println("Minimum: " + Operation.minimum(7, 10, 1, 20, 16));

        System.out.println("Locations:");
        for (Location location : Location.values()) {
            System.out.println(location);
        }
    }
}

```

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр2	Арк.
		Піонтківський В. І.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

## Результат виконання програми:

```
Addition: 18
Subtraction: 50
Multiplication: 64
Division: 8.0
Average: 5.0
Maximum: 20
Minimum: 1
Locations:
KIEV
ZHYTOMYR
RIVNE
Lviv
```

Рис.2 Завдання 3

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



Рис.3 UML діаграма до завдання 2

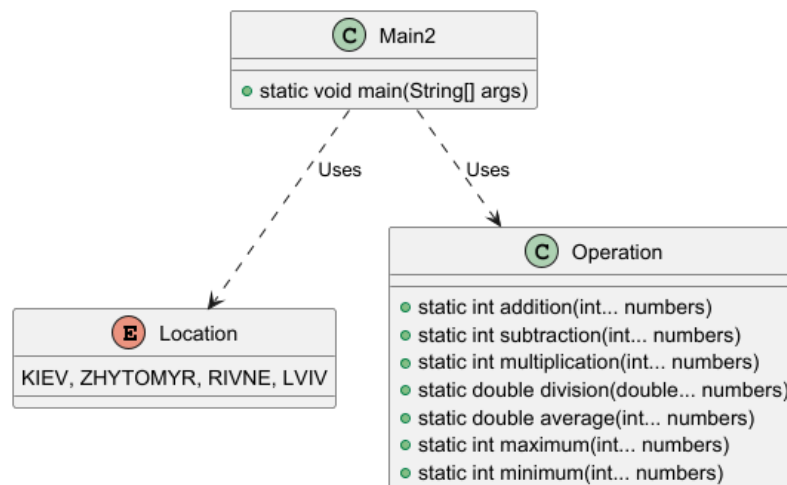


Рис.4 UML діаграма до завдання 3

**Висновок:** під час виконання лабораторної роботи я створила ієрархію класів заданої предметної області, попрацювала з статичними методами.

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр2	Арк.
		Піонтківський В. І.				10
Змн.	Арк.	№ докум.	Підпис	Дата		