

ЛАБОРАТОРНА РОБОТА № 2

Створення структури класу заданої предметної області.

Мета: 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 визначити 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.

					ДУ «Житомирська політехніка».25.121.00.000 – Лр2			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Ищук О.С.			Звіт з лабораторної роботи		Лім.	Арк.
Перевір.		Піонтьківський В.І.						Аркушів
Керівник								1
Н. контр.								10
Зав. каф.							ФІКТ Гр. ІПЗ-23-1	

- 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 оператора instanceof.

Програмний код програми:

```
package com.education.ztu;

// ===== ENUMS =====

// Перелічуваний тип для локацій
enum LocationTask2 {
    KYIV, LVIV, ODESSA,
}

// Перелічуваний тип для статі
enum Gender {
    MALE, FEMALE, OTHER
}

// ===== ІНТЕРФЕЙС =====
interface Human {
    void sayFullName();
    void sayAge();
    void sayLocation();
    void sayGender();

    // метод за замовчуванням
    default void whoIAm() {
        System.out.println("I am a human.");
    }
}

// ===== АБСТРАКТНИЙ КЛАС =====
abstract class Person implements Human {
    private String firstName;
    private String lastName;
    private int age;
    protected LocationTask2 location;
    protected Gender gender;

    // статичний лічильник екземплярів
    private static int counter = 0;

    // Блок ініціалізації
    {
        firstName = "Unknown";
        lastName = "Unknown";
        age = 18;
        location = LocationTask2.KYIV;
        gender = Gender.OTHER;
        counter++;
    }

    // Конструктор без аргументів
    public Person() {}

    // Конструктор з аргументами
    public Person(String firstName, String lastName, int age, LocationTask2
```

		Іщук О.С.			ДУ «Житомирська політехніка».25.121.00.000 – Лр2	Арк.
		Піонтиківський В.І.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

```

location, Gender gender) {
    this.firstName = firstName;
    this.lastName = lastName;
    this.age = age;
    this.location = location;
    this.gender = gender;
    counter++;
}

// Геттери та сеттери
public String getFirstName() { return firstName; }
public void setFirstName(String firstName) { this.firstName = firstName; }

public String getLastName() { return lastName; }
public void setLastName(String lastName) { this.lastName = lastName; }

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

public LocationTask2 getLocation() { return location; }
public void setLocation(LocationTask2 location) { this.location = location; }

public Gender getGender() { return gender; }
public void setGender(Gender gender) { this.gender = gender; }

// Методи інтерфейсу
@Override
public void sayFullName() {
    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);
}

// Статичний метод для показу лічильника
public static void showCounter() {
    System.out.println("Created persons: " + counter);
}

// Абстрактний метод
public abstract String getOccupation();

// Звичайний метод
public void getFullInfo() {
    sayFullName();
    sayAge();
    sayLocation();
    sayGender();
    System.out.println("Occupation: " + getOccupation());
}
}

```

		Іщук О.С.			ДУ «Житомирська політехніка».25.121.00.000 – Лр2	Арк.
		Піонтківський В.І.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```
// ===== KJAC STUDENT =====
class Student extends Person {
    private String university;
    private String specialty;

    public Student() {
        super();
        this.university = "Unknown University";
        this.specialty = "Unknown Specialty";
    }

    public Student(String firstName, String lastName, int age, LocationTask2
location, Gender gender,
        String university, String specialty) {
        super(firstName, lastName, age, location, gender);
        this.university = university;
        this.specialty = specialty;
    }

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

    public String getSpecialty() { return specialty; }
    public void setSpecialty(String specialty) { this.specialty = specialty; }

    @Override
    public String getOccupation() {
        return "Student";
    }

    public void study() {
        System.out.println(getFirstName() + " is studying " + specialty + " at " +
university);
    }
}

// ===== KJAC CAR =====
class Car {
    private String brand;
    private String model;

    private Engine engine;

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

    public void startCar() {
        engine.startEngine();
    }

    public void stopCar() {
        engine.stopEngine();
    }

    // Внутрішній клас Engine
    class Engine {
        private boolean works;

        public void startEngine() {
```

		Іщук О.С.			ДУ «Житомирська політехніка».25.121.00.000 – Лр2	Арк.
		Піонтківський В.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

```

        works = true;
        System.out.println("Engine started.");
    }

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

    public boolean isEngineWorks() {
        return works;
    }
}

public void showCarInfo() {
    System.out.println("Car: " + brand + " " + model);
}
}

// ===== KJIAC TEACHER =====
class Teacher extends Person {
    private String subject;
    private Car car;

    public Teacher() {
        super();
        this.subject = "Unknown Subject";
        this.car = new Car("NoBrand", "NoModel");
    }

    public Teacher(String firstName, String lastName, int age, LocationTask2
location, Gender gender,
        String subject, Car car) {
        super(firstName, lastName, age, location, gender);
        this.subject = subject;
        this.car = car;
    }

    @Override
    public String getOccupation() {
        return "Teacher";
    }

    public void teach() {
        System.out.println(getFirstName() + " is teaching " + subject);
    }

    public void driveCar() {
        car.showCarInfo();
        car.startCar();
    }
}

// ===== KJIAC EMPLOYEE =====
class Employee extends Person {
    private String company;
    private String position;
    private Car car;

    public Employee() {
        super();
        this.company = "Unknown Company";
        this.position = "Unknown Position";
    }
}

```

		Ицук О.С.			ДУ «Житомирська політехніка».25.121.00.000 – Лр2	Арк.
		Піонтківській В.І.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        this.car = new Car("NoBrand", "NoModel");
    }

    public Employee(String firstName, String lastName, int age, LocationTask2
location, Gender gender,
                    String company, String position, Car car) {
        super(firstName, lastName, age, location, gender);
        this.company = company;
        this.position = position;
        this.car = car;
    }

    @Override
    public String getOccupation() {
        return "Employee";
    }

    public void work() {
        System.out.println(getFirstName() + " works as " + position + " at " +
company);
    }

    public void driveCar() {
        car.showCarInfo();
        car.startCar();
    }
}

// ===== MAIN =====
public class Task2 {
    public static void main(String[] args) {
        // Створюємо студентів, викладачів та працівників
        Student st1 = new Student("Ivan", "Petrenko", 20, LocationTask2.KYIV,
Gender.MALE,
                                "KPI", "Computer Science");
        Teacher t1 = new Teacher("Olena", "Ivanova", 40, LocationTask2.LVIV,
Gender.FEMALE,
                                "Mathematics", new Car("Toyota", "Corolla"));
        Employee e1 = new Employee("Andriy", "Shevchenko", 30,
LocationTask2.ODESSA, Gender.MALE,
                                "SoftServe", "Developer", new Car("Tesla", "Model 3"));

        // Демонструємо методи
        st1.getFullInfo();
        st1.study();

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

        t1.getFullInfo();
        t1.teach();
        t1.driveCar();

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

        e1.getFullInfo();
        e1.work();
        e1.driveCar();

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

        // Демонстрація статичного лічильника
        Person.showCounter();
    }
}

```

		Іщук О.С.			ДУ «Житомирська політехніка».25.121.00.000 – Лр2	Арк.
		Піонтківській В.І.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```
// Демонстрація instanceof
if (st1 instanceof Person) {
    System.out.println("st1 is a Person");
}
if (t1 instanceof Teacher) {
    System.out.println("t1 is a Teacher");
}
if (e1 instanceof Human) {
    System.out.println("e1 implements Human");
}
}
```

Результат виконання:

```
Full name: Ivan Petrenko
Age: 20
Location: KYIV
Gender: MALE
Occupation: Student
Ivan is studying Computer Science at KPI
-----
Full name: Olena Ivanova
Age: 40
Location: LVIV
Gender: FEMALE
Occupation: Teacher
Olena is teaching Mathematics
Car: Toyota Corolla
Engine started.
-----
Full name: Andriy Shevchenko
Age: 30
Location: ODESSA
Gender: MALE
Occupation: Employee
Andriy works as Developer at SoftServe
Car: Tesla Model 3
Engine started.
-----
Created persons: 6
st1 is a Person
t1 is a Teacher
e1 implements Human
```

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.

Програмний код програми:

```
package com.education.ztu;

// Перерахування LocationTask3
enum LocationTask3 {
    КИЇВ,
    ЛЬВІВ,
    ОДЕСА,
    ХАРКІВ,
    ДНІПРО
}

// Клас з операціями
```

		Іщук О.С.			ДУ «Житомирська політехніка».25.121.00.000 – Лр2	Арк.
		Піонтківський В.І.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

```

class Operation {
    public static double addition(double... numbers) {
        double sum = 0;
        for (double n : numbers) sum += n;
        return sum;
    }

    public static double subtraction(double... numbers) {
        if (numbers.length == 0) return 0;
        double result = numbers[0];
        for (int i = 1; i < numbers.length; i++) result -= numbers[i];
        return result;
    }

    public static double multiplication(double... numbers) {
        double result = 1;
        for (double n : numbers) result *= n;
        return result;
    }

    public static double division(double... numbers) {
        if (numbers.length == 0) return 0;
        double result = numbers[0];
        for (int i = 1; i < numbers.length; i++) {
            if (numbers[i] != 0) result /= numbers[i];
            else {
                System.out.println("Ділення на нуль неможливе!");
                return 0;
            }
        }
        return result;
    }

    public static double average(double... numbers) {
        if (numbers.length == 0) return 0;
        return addition(numbers) / numbers.length;
    }

    public static double maximum(double... numbers) {
        if (numbers.length == 0) return Double.NaN;
        double max = numbers[0];
        for (double n : numbers) if (n > max) max = n;
        return max;
    }

    public static double minimum(double... numbers) {
        if (numbers.length == 0) return Double.NaN;
        double min = numbers[0];
        for (double n : numbers) if (n < min) min = n;
        return min;
    }
}

// Клас Task3 для демонстрації
public class Task3 {
    public static void main(String[] args) {
        double[] nums = {10, 5, 2};

        double sum = Operation.addition(nums);
        System.out.println("Сума чисел: " + String.format("%.2f", sum));

        double diff = Operation.subtraction(nums);
        System.out.println("Різниця чисел: " + String.format("%.2f", diff));
    }
}

```

		Іщук О.С.			ДУ «Житомирська політехніка».25.121.00.000 – Лр2	Арк.
		Піонтківській В.І.				8
Змн.	Арк.	№ докум.	Підпис	Дата		


```

double prod = Operation.multiplication(nums);
System.out.println("Добуток чисел: " + String.format("%.2f", prod));

double div = Operation.division(nums);
System.out.println("Частка чисел: " + String.format("%.2f", div));

double avg = Operation.average(nums);
System.out.println("Середнє значення: " + String.format("%.2f", avg));

double max = Operation.maximum(nums);
System.out.println("Максимальне значення: " + String.format("%.2f", max));

double min = Operation.minimum(nums);
System.out.println("Мінімальне значення: " + String.format("%.2f", min));

System.out.println("\nСписок локацій:");
for (LocationTask3 loc : LocationTask3.values()) {
    System.out.println("- " + loc);
}
}
}

```

Результат виконання:

```

Сума чисел: 17,00
Різниця чисел: 3,00
Добуток чисел: 100,00
Частка чисел: 1,00
Середнє значення: 5,67
Максимальне значення: 10,00
Мінімальне значення: 2,00

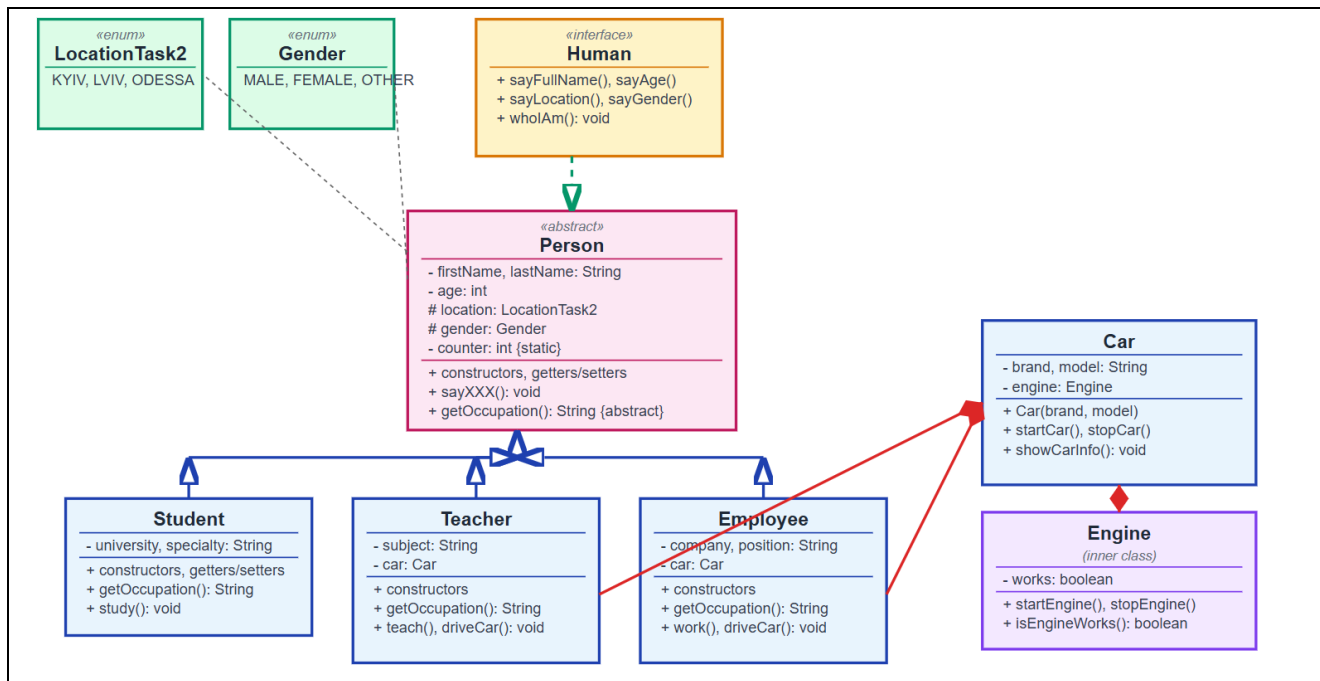
Список локацій:
- КИЇВ
- ЛЬВІВ
- ОДЕСА
- ХАРКІВ
- ДНІПРО

```

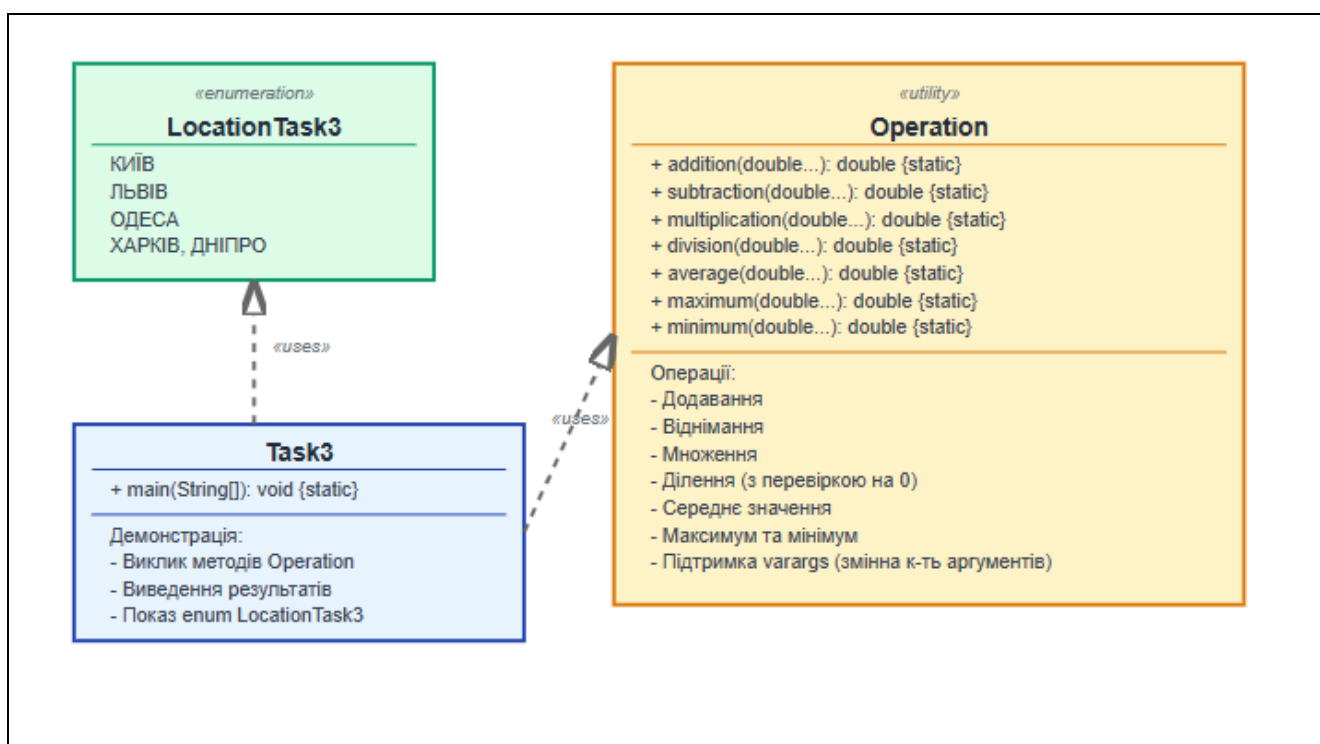
Завдання 4. Створити UML діаграму створеної структури ієрархії класів та зберегти як картинку.

UML-діаграма для завд. 2

		Іцук О.С.			ДУ «Житомирська політехніка».25.121.00.000 – Лр2	Арк.
		Піонтківський В.І.				9
Змн.	Арк.	№ докум.	Підпис	Дата		



UML-діаграма для завд. 3



Посилання на репозиторій: <https://github.com/Oleg-Ischuk/Java-IPZ-23-1>

Висновок: створив ієрархію класів заданої предметної області, попрацював з статичними методами.

		Ицук О.С.			ДУ «Житомирська політехніка».25.121.00.000 – Пр2	Арк.
		Піонтківський В.І.				10
Змн.	Арк.	№ докум.	Підпис	Дата		