



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота
№8
Основи ООП

Виконали
Студенти групи ІА-23:
Ковальський К.В.

Перевірів:
Колеснік В.М.

Київ 2022

Хід роботи:

1. Повторити теоретичні відомості
2. Проаналізувати предметну область завдання свого варіанту (табл.1)
3. Розробити базовий клас (відповідно до завдання можливо абстрактний клас або інтерфейс), клас-нащадок, а також допоміжні класи та/або інтерфейси за необхідністю. Відповідно до предметної області завдання передбачити відповідні методи бізнес-логіки, а також конструктори, сетери та/або гетери, методи equals() та toString(). Продемонструвати використання:
 - this;
 - super;
 - перевантаження (overloading) та заміщення(overriding) методів;
 - перевантаження (overloading) конструкторів.
4. Відповісти на контрольні запитання

```
import javax.print.Doc;

public class Main {

    public static void main(String[] args) {

        Document document = new Document();
        System.out.println(document);

        Document document1 = new Document( name: "Lab", size: 2048);
        System.out.println(document1);
        document1.setName("LAB");
        System.out.println("Переіменовано на "+document1.getName());

        Text_Doc textDocument = new Text_Doc();
        System.out.println( textDocument);
        Text_Doc textDocument1 = new Text_Doc( name: "Lab", info: "Моє завдання",
        text: "Конструктор в Java - це спеціальний метод, який використовується для ініціалізації об'єктів");
        System.out.println(textDocument1);

    }

}
```

```
4 usages
public class Text_Doc extends Document{

    2 usages
    String info;

    2 usages
    String text;

    1 usage
    public Text_Doc(String name, String info, String text ){

        super.name = name;
        this.size = text.toCharArray().length *2;
        this.text = text;
        this.info = info;

    }

    1 usage
    public Text_Doc(){

    }

    @Override
    public String toString(){

        return "Ім'я: " + name + ", Розмір: " + size + ", Зміст: " + info + ", Текст: " + text;

    }

}
```

```

5 usages 1 inheritor
public class Document {
    7 usages
    String name;
    4 usages
    int size;
    1 override
    @Override
    public String toString() { return "Ім'я: " + name + ", Розмір: " + size; }
    3 usages
    public Document() { name = "Новий файл"; }
    1 usage
    public String getName() { return name; }
    1 usage
    public void setName(String name) { this.name = name; }
    1 usage
    public Document(String name, int size){
        this.name = name;
        this.size = size;
    }
}

```

```

C:\Users\kovalskii\jdk\openjdk-19.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2022.2.3\lib\idea_rt.jar=64004:C:\Program
Ім'я: Новий файл, Розмір: 0
Ім'я: Lab, Розмір: 2048
Перейменовано на LAB
Ім'я: Новий файл, Розмір: 0, Зміст: null, Текст: null
Ім'я: Lab, Розмір: 182, Зміст: Мое завдання, Текст: Конструктор в Java - це спеціальний метод, який використовується для ініціалізації об'єктів

```

Висновок: При виконанні лабораторної роботи ми розробили базовий клас, клас-нащадок та допоміжний клас Main. Також спробували на практиці використання перенавантажень(overloading) та заміщення(overriding) методів, а також перенавантаження(overloading) конструкторів.