

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Криворізький національний університет
Кафедра моделювання та програмного забезпечення

ЛАБОРАТОРНА РОБОТА №3
З дисципліни «Програмування на основі Java технологій»
Тема: «Вивчення основних принципів ООП у мові Java»

Виконав студент групи ІПЗ-21-2
Губарєв Р.В.

Перевірив викладач
Котов І.А.
Гриценко А.М.
Карабут Н.О.

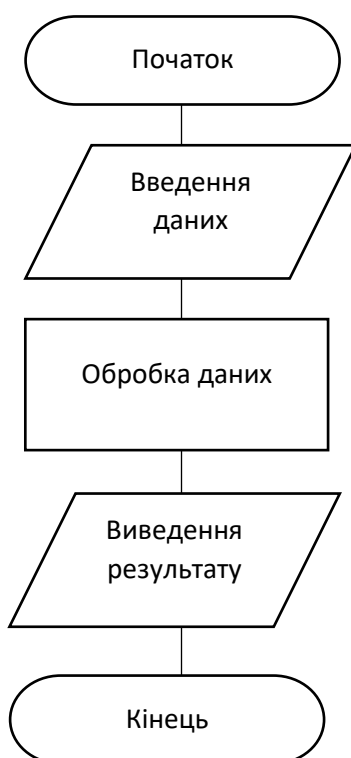
1. Визначення основних положень ООП: інкапсуляція, наслідування, поліморфізм

Інкапсуляція — один з трьох основних механізмів об'єктно-орієнтованого програмування. Йдеться про те, що об'єкт вміщує не тільки дані, але і правила їх обробки, оформлені в вигляді виконуваних фрагментів (методів). А також про те, що доступ до стану об'єкта напряду заборонено, і ззовні з ним можна взаємодіяти виключно через заданий інтерфейс (відкриті поля та методи), що дозволяє знизити зв'язність. Таким чином контролюються звернення до полів класів та їхня правильна ініціалізація, усуваються можливі помилки пов'язані з неправильним викликом методу. Оскільки користувачі працюють лише через відкриті елементи класів, то розробники класу можуть як завгодно змінювати всі закриті елементи і, навіть, перейменовувати та видаляти їх, не турбуючись, що десь хтось їх використовує у своїх програмах.

Наслідування — це один з принципів об'єктно-орієнтованого програмування, який дає класу можливість використовувати програмний код іншого (базового) класу, доповнюючи його своїми власними деталями реалізації. Іншими словами, під час успадкування відбувається отримання нового (похідного) класу, який містить програмний код базового класу з зазначенням власних особливостей використання.

Поліморфізм — концепція в програмуванні та теорії типів, в основі якої лежить використання єдиного інтерфейсу для різнотипних сутностей або у використанні однакового символу для маніпуляцій над даними різного типу

2. Схематичне (графічне) представлення структури найпростішого проекту мовою Java



3. Блок-схеми алгоритмів роботи методів класів



4. Скріншот екрану програми з результатом роботи програми

```
C:\Users\Home\.jdk\openjdk-19.0.2\bin\java.exe "-javaagent:H:\AP
-----
Ім'я: Олег
Вік: 24
-----
Студент
Ім'я: Іван
Вік: 19
Група: ІПЗ-20-1
Спеціальність: Інженерія програмного забезпечення
-----
Process finished with exit code 0
```

5. Текст вихідних кодів програми

class3.java

```
package class3;

public class class3 {
    public static class TLine{
        char Symbol;
        public TLine(char symbol){
            Symbol = symbol;
        };
        public void show(){
            for (int i = 0; i < 10; i++){
                System.out.print(Symbol);
            }
            System.out.println();
        }
    }

    public static void main(String[] args) {
        TLine line = new TLine('-');

        person Oleg = new person("Олег", 24);
        line.show();
        Oleg.show();
        line.show();

        student Ivan = new student("Іван", 19, "ІПЗ-20-1", "Інженерія
програминого забезпечення");
        Ivan.show();
        line.show();
    }
}
```

student.java

```
package class3;

public class student extends person{
    String Group;
    String Speciality;
    public student(String name, int age, String group, String
speciality){
        super(name, age);
        Group = group;
        Speciality = speciality;
    }
    public void show(){
        System.out.println("Студент \nІм'я: " + Name + "\nВік: " + Age +
"\nГрупа: " + Group + "\nСпеціальність: " + Speciality);
    }
}
```

person.java

```
package class3;

public class person {
    public String Name;
    public int Age;

    public person(String name, int age) {
        Name = name;
        Age = age;
    }

    public void show() {
        System.out.println("Ім'я: " + Name + "\nВік: " + Age);
    }
}
```

6. Висновки

В цій лабораторній роботі я навчився створювати класи, класи-спадкоємці, вкладені класи, навчився створювати конструктор класу і оголошувати об'єкти класів в головному класі.

7. Список використаних джерел

-