

Міністерство освіти та науки України
Національний університет «Львівська політехніка»
Кафедра «Електронних обчислювальних машин»



ЗВІТ

з лабораторної роботи № 4

з дисципліни: «Кросплатформенні засоби програмування»

на тему: «СПАДКУВАННЯ ТА ІНТЕРФЕЙСИ»

Виконав: ст. гр. КІ-36

Зелений Е.А.

Прийняв:

доцент кафедри ЕОМ

Іванов Ю.С

Львів-2022

Мета роботи: ознайомитися з спадкуванням та інтерфейсами у мові Java.

ЗАВДАННЯ

1. Написати та налагодити програму на мові Java, що розширює клас, що реалізований у лабораторній роботі №3, для реалізації предметної області заданої варіантом. Суперклас, що реалізований у лабораторній роботі №3, зробити абстрактним. Розроблений підклас має забезпечувати механізми свого коректного функціонування та реалізовувати мінімум один інтерфейс. Програма має розміщуватися в пакеті Група.Прізвище.Lab4 та володіти коментарями, які дозволять автоматично згенерувати документацію до розробленого пакету.
2. Автоматично згенерувати документацію до розробленого пакету.
3. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації.
4. Дати відповідь на контрольні запитання.

Варіант 2:

2. Багаторазовий космічний корабель

Текст програми:

Лістинг Moveable:

```
package lab4;
public interface Moveable {
    void Move(int cycles);
}
```

Лістинг Conqueror

```
package lab4;

import java.io.FileNotFoundException;

public class Conqueror extends Spaceship{
    @Override
    public void Move(int km) {
        System.out.println("Conqueror is moving by:" + km);
    }

    public Conqueror(int fuel, int ammo, boolean isAim, int maxAmmo) throws
FileNotFoundException {
        super (fuel, ammo, maxAmmo,isAim);
        this.fuel = fuel;
        this.ammo = ammo;
        this.maxAmmo = maxAmmo;
        this.isAim = isAim;
    }

    void doubleShoot() {
        ammo -= 2;

        System.out.println("You've done 2 hits");
        fout.print("You've done 2 hits\n");
    }
}
```

```

public void printInfo() {
    System.out.println("\n");
    System.out.println("Ammo " + ammo);
    System.out.println("MaxAmmo " + maxAmmo);

    fout.print("\n");
    fout.print("Ammo" + ammo + "\n");
    fout.print("MaxAmmo" + maxAmmo + "\n");
}

public void deleteInfo() {
    this.ammo = 0;
    this.maxAmmo = 0;
    this.isAim = false;
    System.out.println("Information deleted");
    fout.print("Information deleted");
}

void fewShots(int ammo) {
    int AfterShots = this.ammo - ammo;
    System.out.println("You've done " + ammo + "hits, your current ammo is" +
AfterShots);
    fout.print("You've done " + ammo + "hits, your current ammo is" + AfterShots
    );
}
}

```

Лістинг Spaceship:

```

package lab4;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.PrintWriter;

public abstract class Spaceship implements Moveable{
    protected int fuel;
    protected int ammo;
    protected boolean isAim;

    protected PrintWriter fout;
    protected int maxAmmo;
    public Spaceship() {
        this.fuel = 0;
        this.ammo = 0;
    }

    public Spaceship(int fuel, int ammo, int maxAmmo, boolean isAim) throws
FileNotFoundException {
        this.fuel = fuel;
        this.ammo = ammo;
        this.maxAmmo = maxAmmo;
        this.isAim = false;
        fout = new PrintWriter(new File("Log.txt"));
    }

    public int GetFuel() {
        return this.fuel;
    }

    public int GetAmmo() {
        return this.ammo;
    }

    public void LoadAmmo(int res) {
        res = ammo;
    }
}

```

```

    }

    public void setAmmo(int ammo) {
        this.ammo = ammo;
    }

    public boolean isAim() {
        return isAim;
    }

    public void setAim(boolean aim) {
        isAim = aim;
    }

    public int getMaxAmmo() {
        return maxAmmo;
    }

    public void setMaxAmmo(int maxAmmo) {
        this.maxAmmo = maxAmmo;
    }

    public void shoot() {
        if(isAim) {
            if(ammo == 0){
                System.out.println("No ammo");
                fout.println("No ammo");
            }
            else {
                System.out.println("Boom");
                fout.print("Boom\n");
                ammo -= 1;
                isAim = false;
            }
        }
        else {System.out.println("First you need to aim");}

    }

    public void reload() {
        ammo = maxAmmo;
        System.out.println("Reloading..");
        fout.print("Reloading..\n");
    }

    public void aim() {
        if (isAim) {
            System.out.println("Your target is already aimed");
            fout.print("Your target is already aimed\n");
        } else {
            System.out.println("Aimed.");
            fout.print("Aimed\n");
            isAim = true;
        }
    }

    public void dispose()
    {
        fout.flush();
        fout.close();
    }

}

```

Лістинг SpaceshipDriver:

```
package lab4;
import java.io.*;

public class SpaceshipDriver {
    public static void main(String[] args) throws FileNotFoundException {
        Conqueror ship = new Conqueror(10, 10, false, 50);
        ship.printInfo();
        ship.aim();
        ship.shoot();
        ship.shoot();
        ship.aim();
        ship.shoot();
        ship.reload();
        ship.aim();
        ship.shoot();
        ship.printInfo();
        ship.Move(3);
        ship.doubleShoot();
        ship.fewShots(3);
        ship.deleteInfo();
        ship.printInfo();
        ship.dispose();
    }
}
```

Результат роботи:

```
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "  
  
Ammo 10  
MaxAmmo 50  
Aimed.  
Boom  
First you need to aim  
Aimed.  
Boom  
Reloading..  
Aimed.  
Boom  
  
Ammo 49  
MaxAmmo 50  
Conqueror is moving by:3  
You've done 2 hits  
You've done 3hits, your current ammo is44  
Information deleted  
  
Ammo 0  
MaxAmmo 0  
  
Process finished with exit code 0
```

Рис.1 Результат роботи програми

Відповіді на контрольні запитання:

1. Синтаксис реалізації спадкування.

Спадкування в ООП призначене для розширення функціональності існуючих класів шляхом утворення нових класів на базі вже існуючих. У Java реалізована однокоренева архітектура класів згідно якої всі класи мають єдиного спільного

предка (кореневий клас в ієрархії класів) – клас Object. Решта класів мови Java утворюються шляхом успадковування даного класу. Будь-яке спадкування у мові Java є відкритим, при цьому аналогів захищеному і приватному спадкуванню мови C++ не існує. На відміну від C++ у Java можливе спадкування лише одного базового класу (множинне спадкування відсутнє). Спадкування реалізується шляхом вказування ключового слова `class` після якого вказується назва підкласу, ключове слово `extends` та назва суперкласу, що розширюється у новому підкласі. Синтаксис реалізації спадкування:

```
class Підклас extends Суперклас
{
    Додаткові поля і методи
}
```

2. Що таке суперклас та підклас?

В термінах мови Java базовий клас найчастіше називається суперкласом, а похідний клас – підкласом. Дана термінологія запозичена з теорії множин, де підмножина міститься у супермножині.

Висновок:

Під час виконання роботи я ознайомився з спадкуванням та інтерфейсами у мові Java.