

**АМІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**



ЗВІТ

До лабораторної роботи №4

З дисципліни: «Кросплатформні засоби програмування»

На тему: «СПАДКУВАННЯ ТА ІНТЕРФЕЙСИ»

Варіант 2

Виконав:
ст. гр. КІ-34
Галій О.М.
Перевірив:
доцент
Іванов Ю.С.

Львів 2022

Мета роботи: ознайомитися з процесом розробки класів та пакетів мовою 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.

2. Багаторазовий космічний корабель

Код програми:

```
package lab4;

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

//оголошуємо інтерфейс Moveable
interface Moveable
{
    void move (double x); // прототип методу
}
//оголошуємо інтерфейс Powered, що успадковує інтерфейс Moveable
interface Powered extends Moveable
{
    double milesToFueling (); // прототип методу
    int SPEEDLIMIT = 100; // константа
}

public class ReusableSpaceShip extends Spaceship implements Powered {
    private PrintWriter fout;
    public ReusableSpaceShip (double lFuel, double lMilesPerGalon) throws FileNotFoundException
    {
        fuel = lFuel;
        milesPerGalon = lMilesPerGalon;
        distance = 0.0;
        fout = new PrintWriter(new File("Lab4.txt"));
    }

    public void move (double x)
    {
        distance = distance + x;
        fuel = fuel - distance / milesPerGalon;
```

```

    }
    public double getDistance()
    {
        return distance;
    }
    public double milesToFueling ()
    {
        return fuel*milesPerGalon;
    }
    private double distance;
    private final double milesPerGalon;
    private double fuel;
}

```

```

package lab4;
import java.io.*;

```

```

abstract public class Spaceship {

    private Movement movementDevice;
    private RelativePosition pos;
    private Start st;
    private PrintWriter fout;

    public static int ship = 0;

    public Spaceship() throws FileNotFoundException
    {
        movementDevice = new Movement();
        pos = new RelativePosition();
        st = new Start();
        ship++;

        fout = new PrintWriter(new File("Lab4.txt"));
    }
    public Spaceship(int res) throws FileNotFoundException
    {
        movementDevice = new Movement();
        pos = new RelativePosition();
        st = new Start();
        ship++;

        fout = new PrintWriter(new File("Lab4.txt"));
    }

    public void clickStart()
    {
        st.startButton();

        fout.print("Starting the engines...\n");
        fout.print("Space ship has: " +
            st.getShipFuel() + " fuel \n");
        fout.flush();
    }

    public int getStartShipFuel()
    {
        return st.getShipFuel();
    }
}

```

```

public void MoveUp()
{
    movementDevice.setUpMovement();

    fout.print("Space Ship moved up\n");
    fout.flush();
}

public void MoveDown()
{
    movementDevice.setDownMovement();

    fout.print("Space Ship moved down\n");
    fout.flush();
}

public void MoveRight()
{
    movementDevice.setUpMovement();

    fout.print("Space Ship moved right\n");
    fout.flush();
}

public void MoveLeft()
{
    movementDevice.setDownMovement();

    fout.print("Space Ship moved left\n");
    fout.flush();
}

public void resetMovement()
{
    movementDevice.resetMovement();

    fout.print("Space Ship did not move anywhere \n");
    fout.flush();
}

public Movement.Movements getMovement()
{
    return movementDevice.getMovement();
}

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

public void setShipPosition(int xPos, int yPos)
{
    pos.setXPosition(xPos);
    pos.setYPosition(yPos);
}

public int getShipXPosition()
{
    return pos.getXPosition();
}

public int getShipYPosition()
{
    return pos.getYPosition();
}
}

```

```

class Movement{

    enum Movements {NEUTRAL, UP, DOWN,RIGHT,LEFT};

    private Movements movement;

    public Movement()
    {
        movement = Movements.NEUTRAL;
    }

    public void setNeutralMovement()
    {
        movement = Movements.NEUTRAL;
    }
    public void setUpMovement()
    {
        movement = Movements.UP;
    }

    public void setDownMovement()
    {
        movement = Movements.DOWN;
    }
    public void setRightMovement()
    {
        movement = Movements.RIGHT;
    }
    public void setLeftMovement()
    {
        movement = Movements.LEFT;
    }

    public void resetMovement()
    {
        setNeutralMovement();
    }

    public Movements getMovement()
    {
        return movement;
    }
}

```

```

class Start{
    private int fuel;

    public Start()
    {
        fuel= 1000;
    }
    public Start(int res)
    {
        fuel= res;
    }
    public void startButton()
    {
        fuel = fuel - 10;
    }
    public int getShipFuel(int res)
    {

```

```

        return res;
    }
    public int getShipFuel()
    {
        return fuel;
    }
}

class RelativePosition{

    private int x, y;

    public RelativePosition()
    {
        x = 0;
        y = 0;
    }

    public RelativePosition(int xPos, int yPos)
    {
        x = xPos;
        y = yPos;
    }

    public int getXPosition()
    {
        return x;
    }

    public int getYPosition()
    {
        return y;
    }

    public void getPosition(RelativePosition obj)
    {
        obj.x = x;
        obj.y = y;
    }

    public void setXPosition(int xPos)
    {
        x = xPos;
    }

    public void setYPosition(int yPos)
    {
        y = yPos;
    }
}

```

```

package lab4;
import static java.lang.System.out;
import java.io.FileNotFoundException;

```

```

public class ReusableSpaceShipApp {

```

```

public static void main(String[] args) throws FileNotFoundException
{
    Movement.Movements dir = null;
    ReusableSpaceShip megaship = new ReusableSpaceShip(10, 100);

    megaship.clickStart();
    out.print(megaship.getStartShipFuel() + "\n");

    megaship.move(10);
    out.print(megaship.getDistance() + "\n");
    out.print(megaship.milesToFueling() + "\n");

    megaship.MoveRight();

    dir = megaship.getMovement();
    if (dir == Movement.Movements.UP)
        out.print ("Down" + "\n");
    else if (dir == Movement.Movements.DOWN)
        out.print ("Up" + "\n");
    else if (dir == Movement.Movements.RIGHT)
        out.print ("Right" + "\n");
    else if (dir == Movement.Movements.LEFT)
        out.print ("Left" + "\n");
    else
        out.print ("Neutral" + "\n");

    megaship.dispose();
}
}

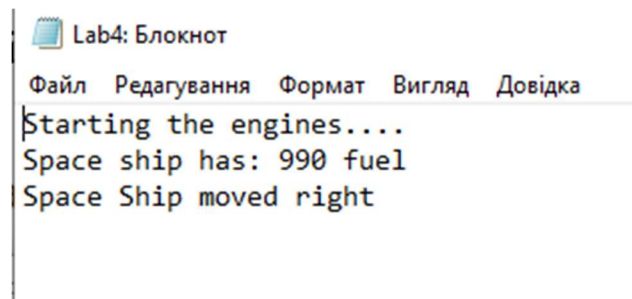
```

Результат виконання програми:

```

990
10.0
990.0
Down

```



Lab4: Блокнот

Файл Редагування Формат Вигляд Довідка

```

Starting the engines....
Space ship has: 990 fuel
Space Ship moved right

```

Висновок: під час лабораторної роботи я ознайомився з процесом розробки класів та пакетів мовою Java.