

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра ЕОМ



Звіт
з лабораторної роботи № 2
з дисципліни «Кросплатформенні засоби програмування»
на тему: «Класи та пакети»

Виконав: ст. гр. КІ-302

Радевич-Винницький Я.А.

Перевірив: викладач

Майдан М.В.

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

Завдання:

1. Написати та налагодити програму на мові Java, що реалізує у вигляді класу предметну область згідно варіанту. Програма має задовольняти наступним вимогам:

- програма має розміщуватися в пакеті Група.Прізвище.Lab3;
- клас має містити мінімум 3 поля, що є об'єктами класів, які описують складові частини предметної області;
- клас має містити кілька конструкторів та мінімум 10 методів;
- для тестування і демонстрації роботи розробленого класу розробити клас-драйвер;
- методи класу мають вести протокол своєї діяльності, що записується у файл;
- розробити механізм коректного завершення роботи з файлом (не надіятися на метод `finalize()`);
- програма має 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: 22. Автомат

Виконання завдання:

Написано програму на мові Java, що реалізує у вигляді класу предметну область автомат.

Код файлу `assaultRifle.java`, що містить реалізацію відповідного класу та його методів:

Лістинг 1

```
/**
 * The assaultRifle class represents a model of an assault rifle.
 */
package KI302.RadevychVynnytskyi.Lab2;
import java.io.IOException;
/**
 * This class defines the properties and behaviors of an assault rifle.
 * It includes attributes like name, cartridge, mass, length, firing range,
 * bullets capacity,
 * bullets currently loaded, price, scope, stock, and muffler.
 *
 * @author RadevychVynnytskyi Yaroslav
 * @version 1.0
 * @since 1.0
 */
public class assaultRifle {
    private String name;
    private double cartridge;
```

```

private double mass;
private double length;
private int firingRange;
private int bulletsCapacity;
private int bulletsCurrently;
private int price;
private boolean scope;
private boolean stock;
private boolean muffler;
/**
 * Constructs an assaultRifle object with default values.
 *
 * @throws IOException if there is an error creating the log file.
 */

public assaultRifle() throws IOException {
    name = "";
    cartridge = 0.0;
    mass = 0.0;
    length = 0.0;
    firingRange = 0;
    bulletsCapacity = 0;
    bulletsCurrently = 0;
    price = 0;
    scope = false;
    stock = false;
    muffler = false;
}

    public assaultRifle(String name, double cartridge, double mass, double
length, int firingRange, int bulletsCapacity, int bulletsCurrently, int price,
boolean scope, boolean stock, boolean muffler) throws IOException {
    this.name = name;
    this.cartridge = cartridge;
    this.mass = mass;
    this.length = length;
    this.firingRange = firingRange;
    this.bulletsCapacity = bulletsCapacity;
    this.bulletsCurrently = bulletsCurrently;
    this.price = price;
    this.scope = scope;
    this.stock = stock;
    this.muffler = muffler;
}

public void getInfo() {
    System.out.println("Name: " + name);
    System.out.println("Cartridge: " + cartridge);
    System.out.println("Mass: " + mass + " kg");
    System.out.println("Length: " + length + " mm");
    System.out.println("FiringRange: " + firingRange + " m");
    System.out.println("Capacity of magazine: " + bulletsCapacity + "
bullets");
    System.out.println("Bullets Balance: " + bulletsCurrently);
    System.out.println("Price: " + price + " $");
    System.out.println("Scope: " + scope);
    System.out.println("Stock: " + stock);
    System.out.println("Muffler: " + muffler);
}

public void fullAutoFire() throws InterruptedException {
    for (int i = getBulletsCurrently(); i >= 0; i--) {
        System.out.println(i + " bullets left");
        Thread.sleep(50);
    }
}

```

```

        setBulletsCurrently(i);
    }
    System.out.println("Run out of bullets!");
}

public void singleFire(int shots) throws InterruptedException {

    if (shots <= getBulletsCurrently()) {
        for (int i = getBulletsCurrently(); i >= getBulletsCurrently() -
shots; i--) {

            System.out.println(i + " bullets left");
            Thread.sleep(700);
        }
        setBulletsCurrently(getBulletsCurrently() - shots);
    } else {
        System.out.println(getBulletsCurrently() + " bullets left. You
can't do more shots than bullets left");
    }
}

public void installMuffler(boolean muffler) {
    if (muffler == false) {
        muffler = true;
        setMuffler(muffler);
        System.out.println("Muffler is installed");
    } else {
        System.out.println("Muffler is already installed");
    }
}

public void unistallMuffler(boolean muffler) {
    if (muffler == true) {
        muffler = false;
        setMuffler(muffler);
        System.out.println("Muffler is uninstalled");
    } else {
        System.out.println("There is no muffler on this gun");
    }
}

public void installScope(boolean scope) {
    if (scope == false) {
        scope = true;
        setScope(scope);
        System.out.println("Scope is installed");
    } else {
        System.out.println("Scope is already installed");
    }
}

public void unistallScope(boolean scope) {
    if (scope == true) {
        scope = false;
        setScope(scope);
        System.out.println("Scope is uninstalled");
    } else {
        System.out.println("There is no scope on this gun");
    }
}

public void reloadGun() {

```

```

        if (getBulletsCurrently() != 30) {
            setBulletsCurrently(getBulletsCapacity());
            System.out.println("The assault rifle is reloaded");
        } else {
            System.out.println("The magazine is full");
        }
    }

    public void compareGuns(assaultRifle a, assaultRifle b) {
        a.getInfo();
        System.out.println();
        b.getInfo();
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setCartridge(double cartridge) {
        this.cartridge = cartridge;
    }

    public void setMass(double mass) {
        this.mass = mass;
    }

    public void setLength(double length) {
        this.length = length;
    }

    public void setFiringRange(int firingRange) {
        this.firingRange = firingRange;
    }

    public void setBulletsCapacity(int bulletsCapacity) {
        this.bulletsCapacity = bulletsCapacity;
    }

    public int getBulletsCapacity() {
        return bulletsCapacity;
    }

    public int getBulletsCurrently() {
        return bulletsCurrently;
    }

    public void setBulletsCurrently(int bulletsCurrently) {
        this.bulletsCurrently = bulletsCurrently;
    }

    public void setPrice(int price) {
        this.price = price;
    }

    public void setStock(boolean stock) {
        this.stock = stock;
    }

    public boolean getMuffler() {

```

```

        return muffler;
    }

    public void setMuffler(boolean muffler) {
        this.muffler = muffler;
    }

    public boolean getScope() {
        return scope;
    }

    public void setScope(boolean scope) {
        this.scope = scope;
    }
}

```

Код файлу main.java, який містить клас-драйвер для тестування і демонстрації роботи класу assaultRifle:

Лістинг 2

```

/**
 * The main class represents the entry point of the program and provides a
 * menu-driven interface
 * to interact with an assault rifle object.
 */
package KI302.RadevychVynnytskyi.Lab2;

import java.io.IOException;
import java.util.Scanner;

/**
 *
 */
public class main {

    /**
     * The main method where the program execution begins.
     *
     * @param args Unused command-line arguments.
     * @throws IOException if there is an error related to file
     * input/output.
     * @throws InterruptedException if a thread is interrupted during
     * execution.
     */
    public static void main(String[] args) throws IOException,
    InterruptedException {
        // TODO Auto-generated method stub
        assaultRifle ak47 = new assaultRifle("AK-47", 7.62, 4.2, 880, 350, 30,
        30, 2700, false, true, false);

        mainMenu(ak47);
    }

    @SuppressWarnings("null")
    public static void mainMenu(assaultRifle gun) throws InterruptedException,
    IOException {
        while(true) {
            System.out.println("Main Menu: \n");

```

```

System.out.println("1. Display Info");
System.out.println("2. Automatic fire");
System.out.println("3. Single fire");
System.out.println("4. Install muffler");
System.out.println("5. Remove muffler");
System.out.println("6. Install scope");
System.out.println("7. Remove scope");
System.out.println("8. Reload the gun");
System.out.println("9. Compare AK-47 with M4");
System.out.println("10. Compare AK-47 with your gun\n");
System.out.println("_____");
System.out.println("Select option: ");
Scanner input = new Scanner(System.in);
int choice = input.nextInt();
int exit = 0;

switch(choice) {
case 1:
    clearConsole();
    gun.getInfo();
    System.out.println("\nPress 0 - Back to Menu");
    exit = input.nextInt();

    if (exit == 0) {
        clearConsole();
        break;
    }
case 2:
    clearConsole();
    gun.fullAutoFire();
    System.out.println("\nPress 0 - Back to Menu");
    exit = input.nextInt();

    if (exit == 0) {
        clearConsole();
        break;
    }
case 3:
    clearConsole();
    System.out.println("How many shots should be done? ");
    int shots = input.nextInt();
    gun.singleFire(shots);
    System.out.println("\nPress 0 - Back to Menu");
    exit = input.nextInt();

    if (exit == 0) {
        clearConsole();
        break;
    }
case 4:
    clearConsole();
    gun.installMuffler(gun.getMuffler());
    System.out.println("\nPress 0 - Back to Menu");
    exit = input.nextInt();

    if (exit == 0) {
        clearConsole();
        break;
    }
case 5:
    clearConsole();
    gun.uninstallMuffler(gun.getMuffler());
    System.out.println("\nPress 0 - Back to Menu");
    exit = input.nextInt();

```

```

        if (exit == 0) {
            clearConsole();
            break;
        }
    case 6:
        clearConsole();
        gun.installScope(gun.getScope());
        System.out.println("\nPress 0 - Back to Menu");
        exit = input.nextInt();

        if (exit == 0) {
            clearConsole();
            break;
        }
    case 7:
        clearConsole();
        gun.uninstallScope(gun.getScope());
        System.out.println("\nPress 0 - Back to Menu");
        exit = input.nextInt();

        if (exit == 0) {
            clearConsole();
            break;
        }
    case 8:
        clearConsole();
        gun.reloadGun();
        System.out.println("\nPress 0 - Back to Menu");
        exit = input.nextInt();

        if (exit == 0) {
            clearConsole();
            break;
        }
    case 9:
        clearConsole();
        assaultRifle ak47 = new assaultRifle("AK-47", 7.62, 4.2, 880,
350, 30, 30, 2700, false, true, false);

        assaultRifle m4 = new assaultRifle("M4", 5.56, 3.39, 840, 600,
30, 30, 3100, true, true, false);

        m4.compareGuns(ak47, m4);

        System.out.println("\nPress 0 - Back to Menu");
        exit = input.nextInt();

        if (exit == 0) {
            clearConsole();
            break;
        }
    case 10:
        clearConsole();
        assaultRifle userGun = new assaultRifle();
        System.out.println("Enter name of your gun: ");
        String name = input.nextLine();
        userGun.setName(name);
        input.nextLine();

        System.out.println("Enter cartridge: ");
        double cart = input.nextDouble();

        userGun.setCartridge(cart);

```



```

        System.out.println("Enter mass: ");
        double mass = input.nextDouble();

        userGun.setMass(mass);

        System.out.println("Enter length: ");
        double length = input.nextDouble();

        userGun.setLength(length);

        System.out.println("Enter firing range: ");
        int firingRange = input.nextInt();

        userGun.setFiringRange(firingRange);

        System.out.println("Enter bullets capacity: ");
        int bulletsCapacity = input.nextInt();

        userGun.setBulletsCapacity(bulletsCapacity);
        userGun.setBulletsCurrently(bulletsCapacity);

        System.out.println("Enter price: ");
        int price = input.nextInt();

        userGun.setPrice(price);

        System.out.println("Enter if there is a scope: ");
        boolean scope = input.nextBoolean();

        userGun.setScope(scope);

        System.out.println("Enter if there is a stock: ");
        boolean stock = input.nextBoolean();

        userGun.setStock(stock);

        System.out.println("Enter if there is a muffler: ");
        boolean muffler = input.nextBoolean();

        userGun.setMuffler(muffler);

        assaultRifle AK47 = new assaultRifle("AK-47", 7.62, 4.2, 880,
350, 30, 30, 2700, false, true, false);

        clearConsole();
        userGun.compareGuns(AK47, userGun);

        System.out.println("\nPress 0 - Back to Menu");
        exit = input.nextInt();

        if (exit == 0) {
            clearConsole();
            break;
        }
        input.close();
    }

}

}

@SuppressWarnings("deprecation")
public static void clearConsole() {

```

```

    try {
        final String os = System.getProperty("os.name");

        if (os.contains("Windows")) {
            // If the OS is Windows, use the "cls" command
            new ProcessBuilder("cmd", "/c",
"cls").inheritIO().start().waitFor();
        } else {
            // If the OS is not Windows (e.g., Linux or macOS), use
"clear"
            Runtime.getRuntime().exec("clear");
        }
    } catch (final Exception e) {
        // Handle exceptions here
        e.printStackTrace();
    }
}
}

```

Результат роботи програми:

```

Main Menu:

1. Display Info
2. Automatic fire
3. Single fire
4. Install muffler
5. Remove muffler
6. Install scope
7. Remove scope
8. Reload the gun
9. Compare AK-47 with M4
10. Compare AK-47 with your gun

-----
Select option:
|

```

Рис. 1 – Головне меню

```

Name: AK-47
Cartridge: 7.62
Mass: 4.2 kg
Length: 880.0 mm
FiringRange: 350 m
Capacity of magazine: 30 bullets
Bullets Balance: 30
Price: 2700 $
Scope: false
Stock: true
Muffler: false

Press 0 - Back to Menu
|

```

Рис. 2 – Display Info

```

How many shots should be done?
7
30 bullets left
29 bullets left
28 bullets left
27 bullets left
26 bullets left
25 bullets left
24 bullets left
23 bullets left

Press 0 - Back to Menu
|

```

Рис. 3 – Single fire

```

Muffler is installed

Press 0 - Back to Menu
|

```

Рис. 4 – Install muffler

```
23 bullets left
22 bullets left
21 bullets left
20 bullets left
19 bullets left
18 bullets left
17 bullets left
16 bullets left
15 bullets left
14 bullets left
13 bullets left
12 bullets left
11 bullets left
10 bullets left
9 bullets left
8 bullets left
7 bullets left
6 bullets left
5 bullets left
4 bullets left
3 bullets left
2 bullets left
1 bullets left
0 bullets left
Run out of bullets!
```

Press 0 - Back to Menu

|

Puc. 5 – Automatic fire

The assault rifle is reloaded

Press 0 - Back to Menu

|

Puc. 6 – Reload the gun

```
Name: AK-47
Cartridge: 7.62
Mass: 4.2 kg
Length: 880.0 mm
FiringRange: 350 m
Capacity of magazine: 30 bullets
Bullets Balance: 30
Price: 2700 $
Scope: false
Stock: true
Muffler: false
```

```
Name: M4
Cartridge: 5.56
Mass: 3.39 kg
Length: 840.0 mm
FiringRange: 600 m
Capacity of magazine: 30 bullets
Bullets Balance: 30
Price: 3100 $
Scope: true
Stock: true
Muffler: false
```

Press 0 - Back to Menu

|

Puc. 7 – Compare AK-47 with M4

```
Enter name of your gun:
Famas
Enter cartridge:
5.5
Enter mass:
4
Enter length:
765
Enter firing range:
300
Enter bullets capacity:
25
Enter price:
2050
Enter if there is a scope:
false
Enter if there is a stock:
true
Enter if there is a muffler:
false|
```

Puc. 8 – Compare AK-47 with your gun

```
Name: AK-47
Cartridge: 7.62
Mass: 4.2 kg
Length: 880.0 mm
FiringRange: 350 m
Capacity of magazine: 30 bullets
Bullets Balance: 30
Price: 2700 $
Scope: false
Stock: true
Muffler: false
```

```
Name:
Cartridge: 5.5
Mass: 4.0 kg
Length: 765.0 mm
FiringRange: 300 m
Capacity of magazine: 25 bullets
Bullets Balance: 25
Price: 2050 $
Scope: false
Stock: true
Muffler: false
```

```
Press 0 - Back to Menu
|
```

Рис. 9 – Compare AK-47 with your gun

```
There is no scope on this gun
```

```
Press 0 - Back to Menu
|
```

Рис. 10 – Remove the scope

Висновок: у ході виконання лабораторної роботи було вивчено процес розробки класів і пакетів мовою Java та реалізовано у вигляді класу предметну область автомат.