Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра «Електронних обчислювальних машин»



Звіт

з лабораторної роботи № 2

з дисципліни: «Кросплатформенні засоби програмування»

на тему: «Класи та пакети»

**Виконав:**

студент групи КІ-307

Трофименко Є.Е.

**Прийняв:**

доцент кафедри ЕОМ

Іванов Ю. С.

Львів – 2023

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

**Завдання (варіант № 22)**

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.Lab2;

• 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 розробленої програми.

3. Завантажити код на GitHub згідно методичних вказівок по роботі з GitHub.

4. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації та завантажити його у ВНС.

5. Дати відповідь на контрольні запитання.



**Вихідний код програми**

*KI307\_Trofymenko\_LAB\_02.java*

/\*\*

\* Програма, що реалізує у вигляді класу предметну область автомату.

\*/

package KI307\_Trofymenko\_LAB\_02;

/\*\*

\* Клас для створення предметної області автомату...

\* @version 1.0

\*/

public class KI307\_Trofymenko\_LAB\_02{

/\*\*

\* Конструктор за замовченням

\*/

public KI307\_Trofymenko\_LAB\_02() {

}

/\*\*

\* Вхідний метод програми

\* @param args визначення методу main

\*/

public static void main(String[] args) {

Rifle rifle = new Rifle (new MagazineCartridge(),new Fuse(),new Trigger());

rifle.Fire();

rifle.Reload();

rifle.TurnOnTheFuse();

rifle.Fire();

rifle.TurnOffTheFuse();

rifle.Fire();

rifle.Fire();

rifle.Fire();

rifle.Fire();

rifle.CheckMagazineCartridge();

rifle.ClosePrintWriter();

}

}

*Rifle.java*

package KI307\_Trofymenko\_LAB\_02;

import java.io.FileNotFoundException;

import java.io.PrintWriter;

/\*\*

\* Головний клас програми Автомату

\*/

public class Rifle {

/\*\*

\* new MagazineCartridge

\*/

MagazineCartridge magazineCartridge;

/\*\*

\* new Fuse

\*/

Fuse fuse;

/\*\*

\* new Trigger

\*/

Trigger trigger;

/\*\*

\* new PrintWriter

\*/

PrintWriter printWriter;

/\*\*

\* Constructor

\*/

public Rifle() {

}

/\*\*

\* Constructor

\* @param magazineCartridge Ємність магазину

\* @param trigger Курок

\*/

public Rifle(MagazineCartridge magazineCartridge, Trigger trigger) {

this.magazineCartridge = magazineCartridge;

this.trigger = trigger;

}

/\*\*

\* Constructor

\* @param magazineCartridge Ємність магазину

\* @param fuse Запобіжник

\* @param trigger Курок

\*/

public Rifle(MagazineCartridge magazineCartridge, Fuse fuse, Trigger trigger){

this.magazineCartridge = new MagazineCartridge();

this.fuse = new Fuse();

this.trigger = new Trigger();

try {

this.printWriter = new PrintWriter("Log.txt");

} catch (FileNotFoundException e) {

throw new RuntimeException(e);

}

}

/\*\*

\* Method that firing and returns boolean

\* @return boolean

\*/

public boolean Fire(){

if(!magazineCartridge.isEmpty() && !fuse.isWorking()) {

UseStock();

TakeAim();

trigger.setOn(true);

System.out.println("Firing");

printWriter.println("Firing");

printWriter.flush();

magazineCartridge.setCapacity(magazineCartridge.getCapacity() - 1);

return true;

}

return false;

}

/\*\*

\* Method of reloading that returns void

\*/

public void Reload(){

magazineCartridge.setEmpty(false);

magazineCartridge.setCapacity(30);

System.out.println("Reloading");

printWriter.println("Reloading");

printWriter.flush();

}

/\*\*

\* Method that setting fuse true

\*/

public void TurnOnTheFuse(){

fuse.setWorking(true);

System.out.println("Putting on the Fuse");

printWriter.println("Putting on the Fuse");

printWriter.flush();

}

/\*\*

\* Method that setting fuse false

\*/

public void TurnOffTheFuse(){

fuse.setWorking(false);

System.out.println("Putting off the Fuse");

printWriter.println("Putting off the Fuse");

printWriter.flush();

}

/\*\*

\* Method that check Magazine Capacity

\*/

public void CheckMagazineCartridge(){

magazineCartridge.getCapacity();

System.out.println("Checking amount of bullets: " + magazineCartridge.getCapacity());

printWriter.println("Checking amount of bullets: " + magazineCartridge.getCapacity());

printWriter.flush();

}

/\*\*

\* Method that cleaning rifle

\*/

public void CleanRifle(){

System.out.println("Cleaning rifle");

printWriter.println("Cleaning rifle");

printWriter.flush();

}

/\*\*

\* Method that taking aim

\*/

public void TakeAim(){

System.out.println("Taking aim");

printWriter.println("Taking aim");

printWriter.flush();

}

/\*\*

\* Method that using stock

\*/

public void UseStock(){

System.out.println("Using stock");

printWriter.println("Using stock");

printWriter.flush();

}

/\*\*

\* Method that wearing belt

\*/

public void UseBelt(){

System.out.println("Wearing belt");

printWriter.println("Wearing belt");

printWriter.flush();

}

/\*\*

\* Method that regulating resistance of trigger

\* @param resistance Опір

\*/

public void RegulateResistanceOfTrigger(int resistance){

trigger.setResistance(resistance);

System.out.println("Setting Resistance Of Trigger");

printWriter.println("Setting Resistance Of Trigger");

printWriter.flush();

}

/\*\*

\* Method to close file

\*/

public void ClosePrintWriter(){

printWriter.println("Closing file");

printWriter.flush();

printWriter.close();

}

}

**Результат виконання програми**

Изображение выглядит как текст, снимок экрана, число

Автоматически созданное описание

**Фрагмент згенерованої документації**

Изображение выглядит как текст, программное обеспечение, веб-страница, число

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, линия, белый

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, линия, чек

Автоматически созданное описание

**Відповіді на контрольні запитання**

1. Синтаксис визначення класу.

*Синтаксис оголошення простого класу в мові Java має наступний вигляд:*

*[public] class НазваКласу*

*{*

*[конструктори]*

*[методи]*

*[поля]*

*}*

1. Синтаксис визначення методу.

*Синтаксис оголошення методу наступний:*

*[СпецифікаторДоступу] [static] [final] Тип назваМетоду([параметри]) [throws класи]*

*{*

*[Тіло методу]*

*[return [значення]];*

*}*

1. Синтаксис оголошення поля.

*Синтаксис оголошення поля наступний:*

*[СпецифікаторДоступу] [static] [final] Тип НазваПоля [= ПочатковеЗначення];*

1. Як оголосити та ініціалізувати константне поле?

*Приклад оголошення константного поля: private final int i;*

1. Які є способи ініціалізації полів?

*Ініціалізацію полів при створенні об’єкту можна здійснювати трьома способами:*

*• у конструкторі;*

*• явно при оголошені поля;*

*• у блоці ініціалізації (виконується перед виконанням конструктора).*

1. Синтаксис визначення конструктора.

*Синтаксис оголошення*

*конструктора:*

*[СпецифікаторДоступу] НазваКласу([параметри])*

*{*

*Тіло конструктора*

*}*

1. Синтаксис оголошення пакету.

*Синтаксис оператора package: package НазваПакету{.НазваПідпакету};*

1. Як підключити до програми класи, що визначені в зовнішніх пакетах?

*Щоб підключити до програми класи, що визначені в зовнішніх пакетах, потрібно використовувати оператор import.*

1. В чому суть статичного імпорту пакетів?

*Статичний імпорт дозволяє не вживати явно назву класу при звертанні до статичного поля або методу класу.*

1. Які вимоги ставляться до файлів і каталогів при використанні пакетів?

*Використання пакетів вимагає, щоб файли і каталоги проекту та їх ієрархія були строго структурованими. Так назви пакету і його підпакетів мають співпадати з назвами каталогів, де вони розміщуються. Назви загальнодоступних класів мають співпадати з назвами файлів, де вони розміщуються. Ієрархія каталогів і файлів проекту має співпадати з ієрархією пакетів.*

**Висновок**

На данній лабораторній роботі ознайомився з процесом розробки класів та пакетів мовою Java. Також ознайомився з конструкторами, методами полями. Створив програму, згідно індивідуального завдання та автоматично згенерував документацію Javadoc до свого проекту.