Міністерство освіти та науки України

Національний університет «Львівська політехніка»

Кафедра «Електронних обчислювальних машин»



**ЗВІТ**

**з лабораторної роботи № 3**

**з дисципліни:** «Кросплатформенні засоби програмування»

**на тему:** «КЛАСИ ТА ПАКЕТИ»

Виконав: ст. гр. КІ-35

Флик Є.В.

Прийняв:

доцент кафедри ЕОМ

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 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.

1. Автоматично згенерувати документацію до розробленого пакету.
2. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації.
3. Дати відповідь на контрольні запитання.

**Варіант 22:**



**Текст програми:**

|  |
| --- |
| ***Лістинг AutomaticGun***  */\*\*  \* lab 3 package  \*/* package KI35.Flyk.Lab3;  import java.io.\*; import java.util.SortedMap;  */\*\*  \* Class <code>AutomaticGun</code> implements automatic gun  \** ***@author*** *Flyk Yevhenii  \** ***@version*** *1.0  \*/* public class AutomaticGun {  public static int *count*;  private Magazine m\_magazine;  private Fuse m\_fuse;  private Barrel m\_barrel;    private PrintWriter fout;  */\*\*  \* Constructor  \** ***@throws*** *FileNotFoundException  \*/* public AutomaticGun()throws FileNotFoundException {  m\_magazine =new Magazine();  m\_fuse = new Fuse();  m\_barrel = new Barrel();   fout = new PrintWriter(new File("Log.txt"));  }  */\*\*  \* Constructor  \** ***@param*** *\_length Length of barrel  \** ***@param*** *\_caliber Caliber of barrel  \** ***@throws*** *FileNotFoundException  \*/* public AutomaticGun(int \_length,float \_caliber)throws FileNotFoundException{  m\_magazine =new Magazine();  m\_fuse= new Fuse();  m\_barrel =new Barrel(\_length,\_caliber);   fout = new PrintWriter(new File("Log.txt"));  *count*++;  }  */\*\*  \* Method implements get of count  \*/* public int getCount(){  return *count*;  }  */\*\*  \* Method implements reload ofmagazine  \*/* public void ReloadAutomaticGun(){  m\_magazine.load();  m\_magazine.setAmmunition(30);  fout.println("New state of magazine: " + m\_magazine.isInserted());  fout.println("New numbers of ammunition in magazine: " + m\_magazine.getAmmunition());  System.*out*.println("RELOAD!!!");  }  */\*\*  \* Method implements single firing  \*/* public void SingleFiring(){  if(m\_magazine.getAmmunition()>0 && !m\_fuse.isState()){  m\_magazine.reduction();   fout.println("New numbers of ammunition in magazine: " + m\_magazine.getAmmunition());  System.*out*.println("SHOOT!!!");  }else  {  System.*out*.println("Shooting is impossible!");  if(m\_magazine.getAmmunition()==0){  System.*out*.println("Reload the AutomaticGun");  }else  {  System.*out*.println("Turn off the fuse");  }  }  }  */\*\*  \* Method implements burst firing  \*/* public void BurstFiring(){  if(m\_magazine.getAmmunition()>0 && !m\_fuse.isState()){  for(int i=0;i<3;++i) {  m\_magazine.reduction();   fout.println("New numbers of ammunition in magazine: " + m\_magazine.getAmmunition());  System.*out*.println("SHOOT!!!");  }  }else  {  System.*out*.println("Shooting is impossible!");  if(m\_magazine.getAmmunition()==0){  System.*out*.println("Reload the AutomaticGun");  }else  {  System.*out*.println("Turn off the fuse");  }  }  }  */\*\*  \* Method implements switch on fuse  \*/* public void SwitchOnFuse(){  if(!m\_fuse.isState()){  m\_fuse.setState(true);   fout.println("New state of fuse: " + m\_magazine.isInserted());  System.*out*.println("Fuse switching!!!");  }  else{  System.*out*.println("The fuse is already on!");  }  }  */\*\*  \* Method implements switch off fuse  \*/* public void SwitchOffFuse(){   if(m\_fuse.isState()){  m\_fuse.setState(false);   fout.println("New state of fuse: " + m\_magazine.isInserted());  System.*out*.println("Fuse switching!!!");  }  else{  System.*out*.println("The fuse is already off!");  }  }   */\*\*  \* Method implements set information about barrel  \** ***@param*** *\_length Length of barrel  \** ***@param*** *\_caliber Caliber of barrel  \*/* public void SetBarrel(int \_length,float \_caliber){  m\_barrel.setLength(\_length);  m\_barrel.setCaliber(\_caliber);   fout.println("New length of barrel: " + m\_barrel.getLength());  fout.println("New caliber of barrel: " + m\_barrel.getCaliber());  }  */\*\*  \* Method implements printing information about gun  \*/* public void PrintGun(){  System.*out*.println("Amount of ammunition: " + m\_magazine.getAmmunition());  System.*out*.println("Current state of magazine: "+m\_magazine.isInserted());  System.*out*.println("Current state of fuze: "+m\_fuse.isState());  System.*out*.println("Length of barrel: "+m\_barrel.getLength());  System.*out*.println("Caliber of barrel: "+m\_barrel.getCaliber());  }  */\*\*  \* Method implements set of magazine  \** ***@param*** *\_ammunition  \** ***@param*** *\_state  \*/* public void SetMagazine(int \_ammunition,boolean \_state){  m\_magazine.setAmmunition(\_ammunition);  m\_magazine.setInserted(\_state);   fout.println("New state of magazine: " + m\_magazine.isInserted());  fout.println("New numbers of ammunition in magazine: " + m\_magazine.getAmmunition());    }  */\*\*  \* Method implements set of fuse  \** ***@param*** *\_state  \*/* public void SetFuse(boolean \_state){  m\_fuse.setState(\_state);   fout.println("New state of fuse: " + m\_fuse.isState());  }    public void dispose()  {  fout.close();  } } */\*\*  \* Class <code>Magazine</code> implements magazine  \** ***@author*** *Flyk Yevhenii  \** ***@version*** *1.0  \*/* class Magazine{  //number of ammunition  private int ammunition;  //current state of magazine  private boolean inserted;   */\*\*  \* Constructor  \*/* public Magazine() {  this.ammunition = 30;  this.inserted =true;  }  */\*\*  \* Method returns ammunition  \** ***@return*** *Number of ammunition  \*/* public int getAmmunition() {  return ammunition;  }  */\*\*  \* Method returns current state magazine  \** ***@return*** *Current state magazine  \*/* public boolean isInserted() {  return inserted;  }  */\*\*  \* Method set current state magazine  \** ***@param*** *inserted New current state magazine  \*/* public void setInserted(boolean inserted) {  this.inserted = inserted;  }  */\*\*  \* Method implements a reduction in ammunition  \*/* public void reduction(){  --(this.ammunition);  }  */\*\*  \* Method implements store removal  \*/* public void unLoad(){  this.inserted=false;  }  */\*\*  \* Method implements magazine insertion  \*/* public void load(){  this.inserted=true;  }  */\*\*  \* Method set current state magazine  \** ***@param*** *ammunition New current state magazine  \*/* public void setAmmunition(int ammunition) {  this.ammunition = ammunition;  } } */\*\*  \* Class <code>Fuse</code> implements fuse  \** ***@author*** *Flyk Yevhenii  \** ***@version*** *1.0  \*/* class Fuse {  //State fuse  private boolean state;  */\*\*  \* Constructor  \*/* public Fuse(){  this.state = false;  }  */\*\*  \* Method returns current state  \** ***@return*** *Returns current state of Fuse  \*/* public boolean isState() {  return state;  }  */\*\*  \* Method set state of fuze  \** ***@param*** *state Set state of fuze  \*/* public void setState(boolean state) {  this.state = state;  } } */\*\*  \* Class <code>Barrel</code> implements barrel  \** ***@author*** *Flyk Yevhenii  \** ***@version*** *1.0  \*/* class Barrel{  //length of barrel(mm)  int length;  //caliber of barrel  float caliber;   */\*\*  \* Constructor  \*/* public Barrel(){  this.length = 415;//standard in Ukraine  this.caliber = (float) 5.45;//standard in Ukraine  }  */\*\*  \* Constructor with parameters  \*/* public Barrel(int \_length , float \_caliber ){  this.length = \_length;  this.caliber = \_caliber;  }  */\*\*  \* Method length of barrel  \** ***@return*** *Returns length  \*/* public int getLength() {  return length;  }  */\*\*  \* Method set length of barrel  \** ***@param*** *length of barrel  \*/* public void setLength(int length) {  this.length = length;  }  */\*\*  \* Method caliber of barrel  \** ***@return*** *Returns caliber  \*/* public float getCaliber() {  return caliber;  }  */\*\*  \* Method set caliber of barrel  \** ***@param*** *caliber of barrel  \*/* public void setCaliber(float caliber) {  this.caliber = caliber;  } }  ***Лістинг AutomaticGunApp***  */\*\*  \* lab 3 package  \*/* package KI35.Flyk.Lab3;  import static java.lang.System.*out*; import java.io.\*; */\*\*  \* Automatic Gun Application class implements main method for AutomaticGun  class possibilities demonstration  \** ***@author*** *Flyk Yevhenii  \** ***@version*** *1.0  \*/* public class AutomaticGunApp {  */\*\*  \** ***@param*** *args  \** ***@throws*** *FileNotFoundException  \*/* public static void main(String[] args) throws FileNotFoundException{  AutomaticGun rifle = new AutomaticGun();  rifle.SwitchOnFuse();   rifle.SingleFiring();   rifle.ReloadAutomaticGun();  rifle.SwitchOffFuse();  rifle.SingleFiring();  *out*.println("\n");  rifle.PrintGun();  *out*.println("\n");  rifle.BurstFiring();  *out*.println("\n");  rifle.PrintGun();   rifle.SwitchOnFuse();  *out*.println("\n");  rifle.PrintGun();  rifle.SwitchOffFuse();  *out*.println("\n");  rifle.PrintGun();   rifle.dispose();  } } |

**Результат виконання:**

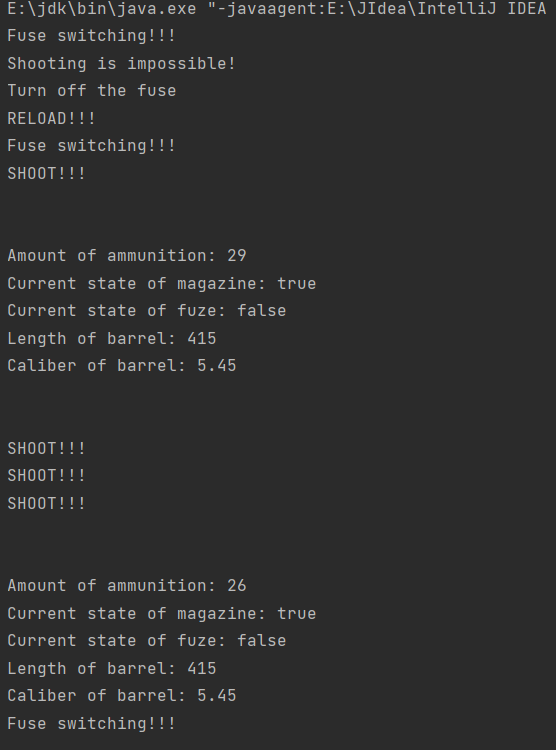
****

Рис.1.1 Результат виконання програми

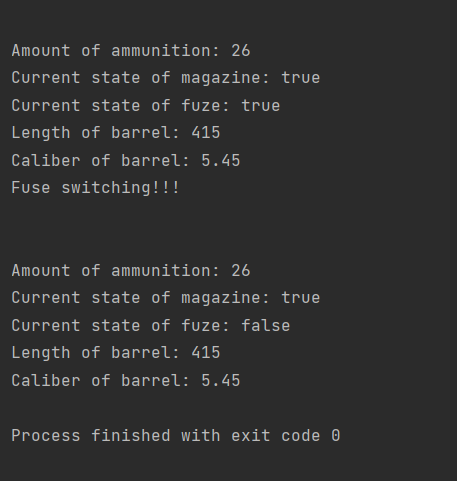
****

Рис.1.2 Результат виконання програми

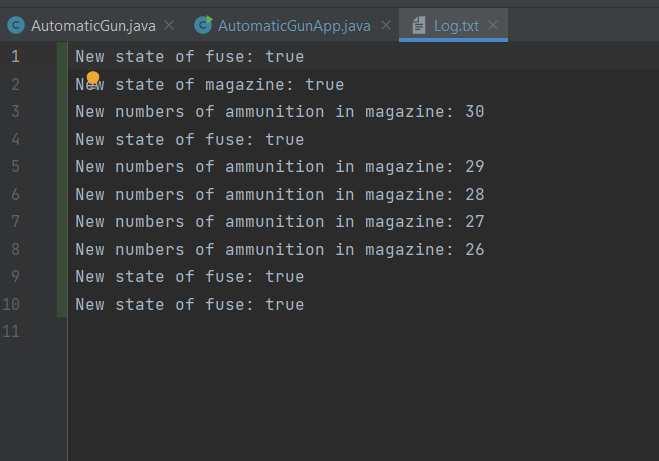
****

Рис.2 Результат запису в Log-file

**Фрагмент згенерованої документації**

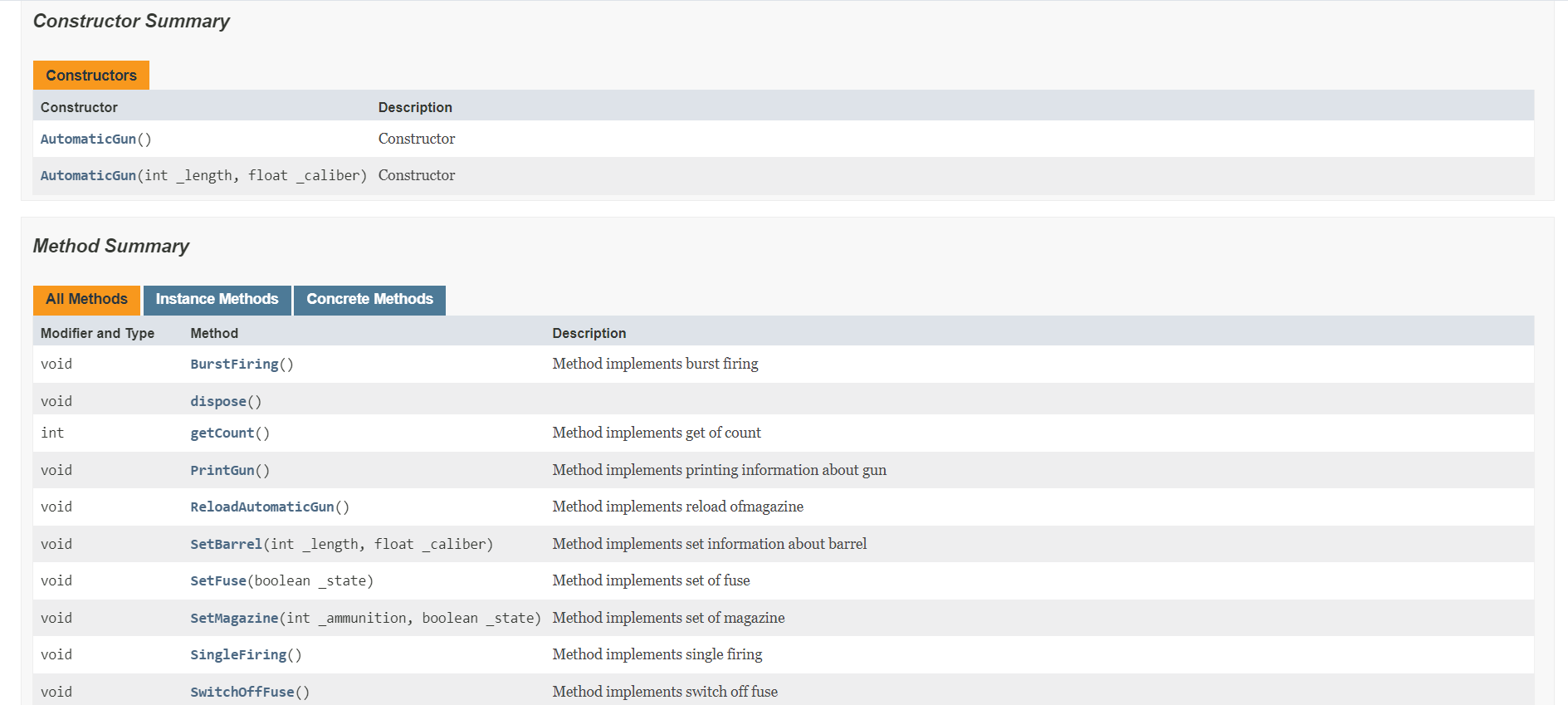


Рис.3.1 Фрагмент документації файлу AutomaticGun

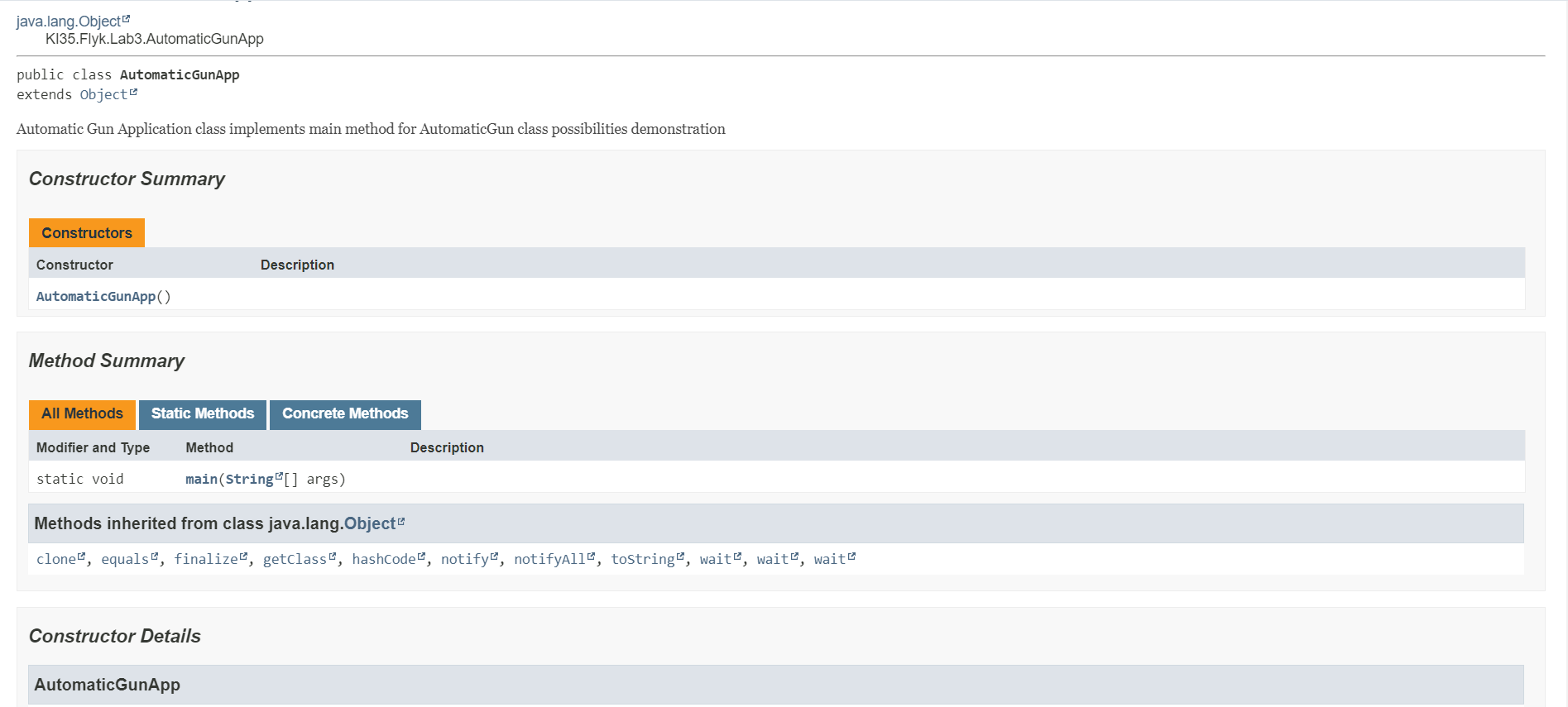
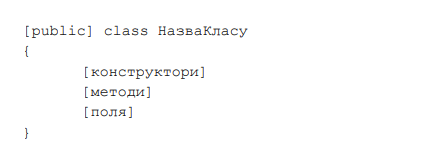


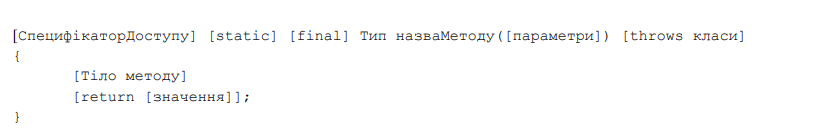
Рис.3.2 Фрагмент документації файлу AutomaticGunApp

**Відповіді на контрольні запитання:**

1. Синтаксис визначення класу.



1. Синтаксис визначення методу.



**Висновок:**

Під час виконання роботи я ознайомився з процесом розробки класів та пакетів мовою Java.