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

Гудима О.П.

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 к6оректного 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 2:**

****

**Текст програми:**

|  |
| --- |
| ***Лістинг Spaceship***  *import* java.io.\*;   *public class* Spaceship {  *private static int* spaceshipCount;  *public static int* upCounter;  *public static int* downCounter;  *public static int* leftCounter;  *public static int* rightCounter;  *private* Movement movementDevice;  *private* RelativePosition pos;  *private* Start start;  *private* PrintWriter fout;  *private int* fuel;   *public* Spaceship() *throws* FileNotFoundException  {  start = *new* Start();  movementDevice = *new* Movement();  pos = *new* RelativePosition();  fout = *new* PrintWriter(*new* File("Lab3.txt"));  }   *public* Spaceship(*int* res) *throws* FileNotFoundException  {  spaceshipCount++;  *this*.fuel = res;  }  *public static int* getSpaceshipCountCount() {  *return* spaceshipCount;  }      *public void* clickStart()  {  start.startButton();   fout.print("Engine is on.\n");  fout.print("Space ship tank is: " +  start.getShipFuel() + " fuel \n");  fout.flush();  }   *public int* getStartShipFuel()  {  *return* start.getShipFuel();  }     *public void* MoveUp()  {  movementDevice.setUpMovement();   fout.print("Spaceship -> moved up\n");  fout.flush();  upCounter++;  }   *public void* MoveDown()  {  movementDevice.setDownMovement();   fout.print("Spaceship -> moved down\n");  fout.flush();  downCounter++;  }  *public void* MoveRight()  {  movementDevice.setUpMovement();   fout.print("Spaceship -> moved right\n");  fout.flush();  rightCounter++;  }   *public void* checkifEmpty (Spaceship ship) {  *if* (ship==*null*) {  System.out.println("empty");  }  *else* {System.out.println(ship);}  }   *public void* MoveLeft()  {  movementDevice.setDownMovement();   fout.print("Spaceship -> moved left\n");  fout.flush();  leftCounter++;  }  *public void* resetMovement()  {  movementDevice.resetMovement();   fout.print("Spaceship is stable \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* Fire{  *private int* ammo;  *public* Fire()  {  ammo= 1000;  }  *public void* startButton()  {  ammo = ammo - 30;  }   *public int* getShipAmmo()  {  *return* ammo;  }   *public* Fire(*int* fill)  {  ammo = fill;  }   }      *class* Movement{   *enum* Movements {STABLE, RIGHT, LEFT,DOWN,UP};    *private* Movements movement;    *public* Movement()  {  movement = Movements.STABLE;  }   *public void* setStableMovement()  {  movement = Movements.STABLE;  }  *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()  {  setStableMovement();  }   *public* Movements getMovement()  {  *return* movement;  } }  *class* Start{  *private int* fuel;   *public* Start()  {  fuel= 500;  }    *public* Start(*int* fill)  {  fuel = fill;  }   *public void* startButton()  {  fuel = fuel - 50;  }  *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;  }  }  ***Лістинг SpaceshipDriver***  *import static* java.lang.System.out; *import* java.io.\*;  *public class* SpaceshipDriver {  *public static void* main(String[] args) *throws* FileNotFoundException  {     Spaceship ship1 = *new* Spaceship();  out.println(Spaceship.getSpaceshipCountCount());  ship1.MoveLeft();  ship1.MoveLeft();  ship1.MoveLeft();  ship1.MoveUp();  out.println(ship1.getMovement());  out.println(ship1.leftCounter);  }    } |

**Результат виконання:**

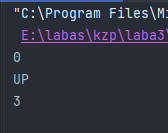
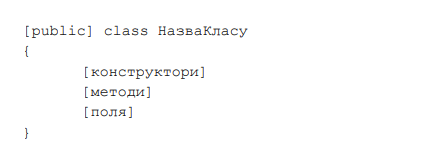
****

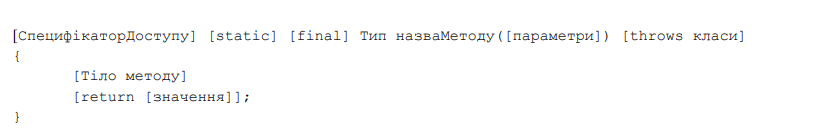
Рис.1.1 Результат виконання програми

**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.