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

Хван Д.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.

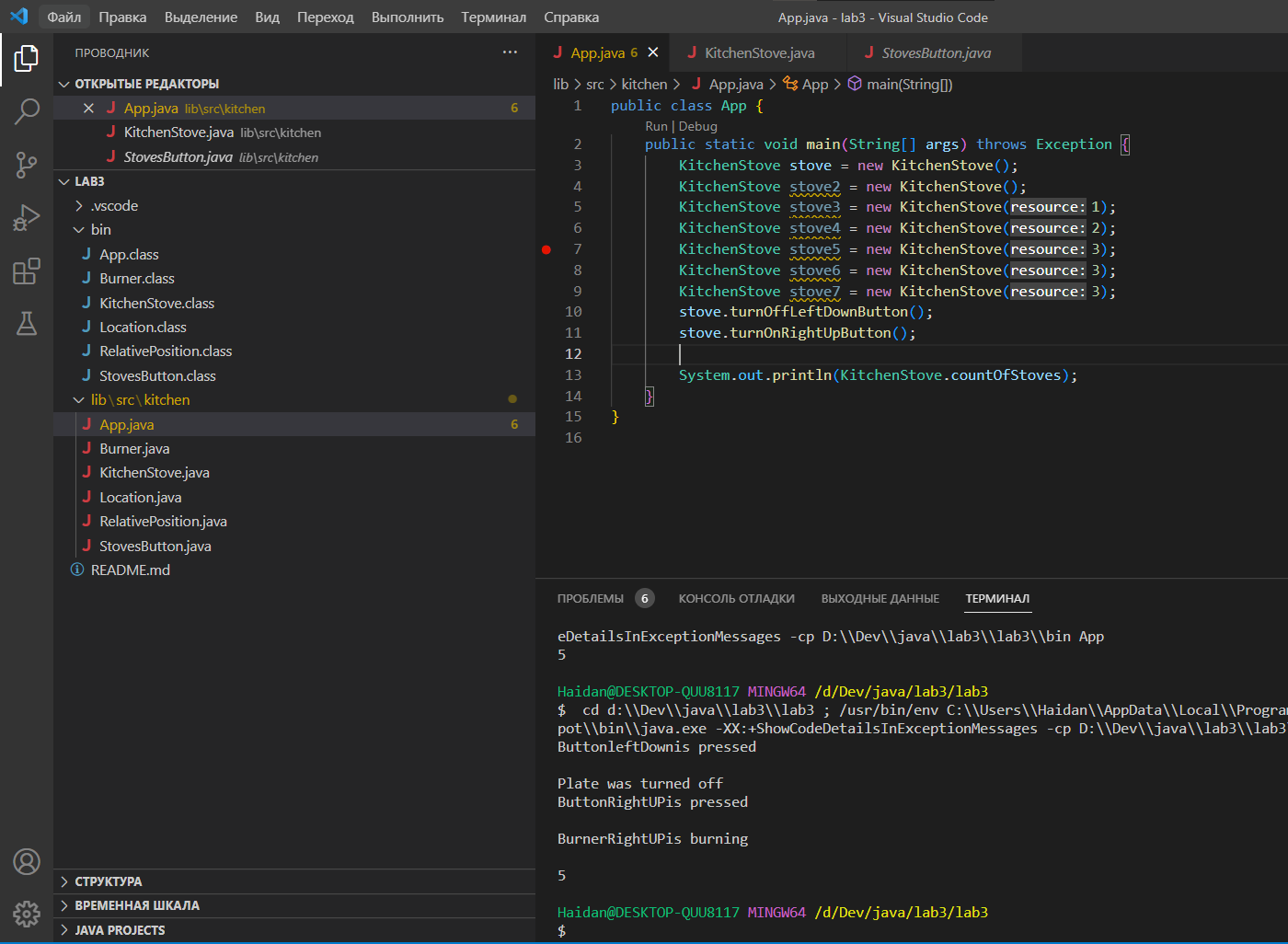
**Варіант 22:**



**Текст програми:**

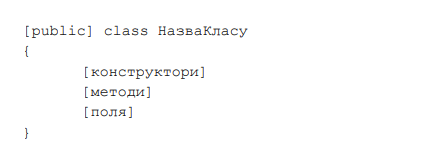
|  |
| --- |
| import java.io.\*;  /\*\*   \* Class <code>KitchenStove</code> implements computer mouse   \*   \* @author EOM Stuff   \* @version 1.0   \*/   class KitchenStove {      private StovesButton leftUpButtonForBurner;      private StovesButton leftDownButtonForBurner;      private StovesButton rightUpButtonForBurner;      private StovesButton rightDownButtonForBurner;      public static int countOfStoves;      /\*\*       \* Constructor       \*       \* @throws FileNotFoundException       \*/      public KitchenStove() throws FileNotFoundException {          leftUpButtonForBurner = new StovesButton(Location.leftUP);          leftDownButtonForBurner = new StovesButton(Location.leftDown);          rightUpButtonForBurner = new StovesButton(Location.RightUP);          rightDownButtonForBurner = new StovesButton(Location.RightDown);      }      /\*\*       \* Constructor       \*       \* @param <code>resource</code> Button clicks resource       \* @throws FileNotFoundException       \*/      public KitchenStove(int resource) throws FileNotFoundException {          countOfStoves++;      }      /\*\*       \* Method implement turn on Left Up plate       \*       \*/      public void turnOnLeftUpButton() {          leftUpButtonForBurner.turnOnButton(5);      }      /\*\*       \* Method implement turn on Left Down plate       \*       \*/      public void turnOnLeftDownButton() {          leftDownButtonForBurner.turnOnButton(4);      }      /\*\*       \* Method implement turn on Right Up plate       \*       \*/      public void turnOnRightUpButton() {          rightUpButtonForBurner.turnOnButton(7);      }      /\*\*       \* Method implement turn on Right Down plate       \*       \*/      public void turnOnRightDownButton() {          rightDownButtonForBurner.turnOnButton(2);      }      /\*\*       \* Method implement turn on Left Up plate       \*       \*/      public void turnOffLeftUpButton() {          leftUpButtonForBurner.turnOffButton();      }      /\*\*       \* Method implement turn on Left Down plate       \*       \*/      public void turnOffLeftDownButton() {          leftDownButtonForBurner.turnOffButton();      }      /\*\*       \* Method implement turn on Right Up plate       \*       \*/      public void turnOffRightUpButton() {          rightUpButtonForBurner.turnOffButton();      }      /\*\*       \* Method implement turn on Right Down plate       \*       \*/      public void turnOffRightDownButton() {          rightDownButtonForBurner.turnOffButton();      }  }  enum Location {      leftUP,      leftDown,      RightUP,      RightDown,  }  /\*\*       \* Class <code>RelativePosition</code> implements relative positioning       \* coordinate system       \*       \* @author EOMStuff       \* @version 1.0       \*/      class RelativePosition {          // coordinates of the mouse position          private int x;          /\*\*           \* Constructor           \*/          public RelativePosition() {              x = 0;          }          /\*\*           \* Constructor           \*           \* @param <code>xPos</code> The X coordinate value           \*/          public RelativePosition(int xPos) {              x = xPos;          }          /\*\*           \* Method returns the X coordinate value           \*           \* @return The X coordinate value           \*/          public int getXPosition() {              return x;          }          /\*\*           \* Method returns coordinates of the position in the <code>obj</code>,           \* that is passed into method through method parameter           \*           \* @param <code>obj</code> The object, where coordinates of the current           \*                         position are set           \*/          public void getPosition(RelativePosition obj) {              obj.x = x;          }          /\*\*           \* Method sets the X coordinate value           \*           \* @param <code>xPos</code> The X coordinate value           \*/          public void setXPosition(int xPos) {              if (xPos < 0) {                  xPos = 0;              }              x = xPos;          }      }  import java.io.FileNotFoundException;  public class StovesButton {      private Location location;      private Burner burner;      private RelativePosition position;      private static boolean turnOn;      public StovesButton(Location location) throws FileNotFoundException {          this.location = location;          this.burner = new Burner();          this.burner.location = this.location;          this.position = new RelativePosition();          turnOn = false;      }      /\*\*       \* Method implements pressing on a button og kitchen stove       \*       \* @param <code>burner</code>       \*/      public void turnOnButton(int xPos) {          position.setXPosition(xPos);          System.out.println("Button" + this.location.toString() + "is pressed\n");          if (xPos > 0) {              burner.burn();              turnOn = true;          }      }      boolean isTurnOn() {          return turnOn;      }      public void turnOffButton() {          position.setXPosition(0);          System.out.println("Button" + this.location.toString() + "is pressed\n");          if(!isTurnOn())          {              System.out.println("Plate was turned off");              return;          }          turnOn = false;          burner.stopBurn();      }  } |

**Результат виконання:**

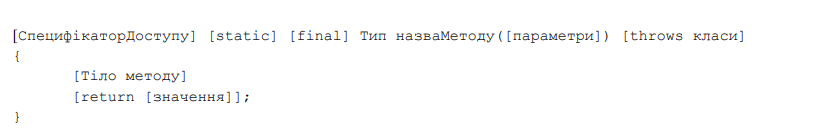
****

**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.