Міністерство освіти і науки України

Національний університет „Львівська політехніка”

Кафедра ЕОМ



**Звіт**

з лабораторної роботи №3

з дисципліни: “Кросплатформні засоби програмування”

на тему: “Класи та пакети”

Виконала: ст. гр. КІ-305

Скоропад А.І.

Прийняв:

Іванов Ю.С.

Львів – 2023

**Мета:** ознайомитися з процесом розробки класів та пакетів мовоюJava.

**Індивідуальне завдання:** написати та налагодити програму на мові Java, що реалізує у вигляді класу предметну область згідно варіанту:

4**. Кіт**

**Хід роботи:**

1. Запустила середовище Intellij Idea та написала програму згідно індивідуального завдання:

package ki.hordon.lab3;  
  
  
  
import ki.hordon.lab3.Cat;  
import ki.hordon.lab3.Toy;  
import ki.hordon.lab3.Species;  
import ki.hordon.lab3.Vaccine;  
  
import java.io.FileNotFoundException;  
import java.util.Scanner;  
  
  
  
public class CatApp {  
 */\*\*  
 \*  
 \** ***@param*** *args command line  
 \*/* public static void main(String[] args) {  
 Cat cat = new Cat();  
 Species species = new Species();  
 Toy toy = new Toy();  
 Vaccine vaccine = new Vaccine();  
  
 Scanner scanner = new Scanner(System.*in*);  
 String command = new String();  
 boolean running = true;  
  
 System.*out*.println("Command:\nadd toy - add toy list to cat\nadd vaccine - add vaccine list to cat\nadd species - turn species into the cat\nprint - print cat info to the file\nstop - stops the program\n\n");  
  
 while (running)  
 {  
 System.*out*.print("Enter command:\n--> ");  
 command = scanner.nextLine();  
  
 switch (command) {  
 case "add toy":  
 System.*out*.println("Enter list of toys");  
 while(true)  
 {  
 System.*out*.print("--> ");  
 command = scanner.nextLine();  
 toy.addToy(command);  
 if (command.equals(""))  
 break;  
 }  
 cat.addToy(toy);  
 break;  
 case "add vaccine":  
 System.*out*.println("Enter list of vaccines");  
 while(true)  
 {  
 System.*out*.print("--> ");  
 command = scanner.nextLine();  
 vaccine.addVaccines(command);  
 if (command.equals(""))  
 break;  
 }  
 cat.addVaccine(vaccine);;  
 break;  
 case "add species":  
 System.*out*.println("Enter species:\n1 - SHORTHAIR\n2 - CURL\n3 - WIREHAIR\n4 - UNKNOWN");  
 System.*out*.print("--> ");  
 command = scanner.nextLine();  
 switch (command)  
 {  
 case "1":  
 species.setSpecies(FilletType.*SHORTHAIR*);  
 cat.setSpecies(species);  
 break;  
 case "2":  
 species.setSpecies(FilletType.*CURL*);  
 cat.setSpecies(species);  
 break;  
 case "3":  
 species.setSpecies(FilletType.*WIREHAIR*);  
 cat.setSpecies(species);  
 break;  
 case "4":  
 species.setSpecies(FilletType.*UNKNOWN*);  
 cat.setSpecies(species);  
 break;  
 default:  
 System.*out*.println("Wrong species type!");  
 break;  
 }  
 break;  
 case "print":  
 System.*out*.println("Enter file path:");  
 System.*out*.print("--> ");  
 command = scanner.nextLine();  
 try {  
 cat.openFile(command);  
 cat.printToFile();  
 cat.closeFile();  
 } catch (FileNotFoundException e) {  
 // *TODO Auto-generated catch block* e.printStackTrace();  
 }  
 break;  
 case "stop":  
 running = false;  
 System.*out*.println("Program stoped.");  
 break;  
 default:  
 System.*out*.println("Wrong command! Try again.");  
 break;  
 }  
 }  
 }  
}

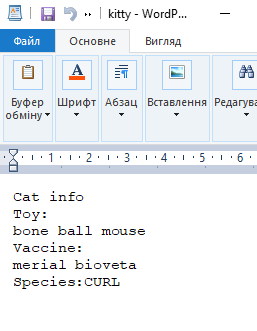
package ki.hordon.lab3;  
  
import java.util.ArrayList;  
import java.util.Arrays;  
  
  
  
public class Vaccine {  
  
 private ArrayList<String> vaccines;  
  
 */\*\*  
 \* Constructor  
 \*/* public Vaccine()  
 {  
 this.vaccines = new ArrayList<>();  
 }  
 */\*\*  
 \* Constructor  
 \** ***@param*** *str new vaccine  
 \*/* public Vaccine(String str)  
 {  
 this();  
 addVaccines(str);  
 }  
 */\*\*  
 \* Constructr  
 \** ***@param*** *args range of vaccines  
 \*/* public Vaccine(String[] args)  
 {  
 this();  
 addVaccines(args);  
 }  
 */\*\*  
 \* Add new vaccine  
 \** ***@param*** *vaccines new vaccine  
 \*/* public void addVaccines(String vaccines)  
 {  
 this.vaccines.add(vaccines);  
 }  
 */\*\*  
 \* Add range of vaccines  
 \** ***@param*** *vaccines new vaccines  
 \*/* public void addVaccines(String[] vaccines)  
 {  
 this.vaccines.addAll(Arrays.*asList*(vaccines));  
 }  
 */\*\*  
 \* Get list of vaccines  
 \** ***@return*** *list of vaccines  
 \*/* public ArrayList<String> getList()  
 {  
 return this.vaccines;  
 }  
 */\*\*  
 \* Transform vaccines to string list  
 \*/* public String toString()  
 {  
 String string = new String();  
 for (int i = 0; i < this.vaccines.size(); i++)  
 string += vaccines.get(i) + " ";  
 return string;  
 }  
}

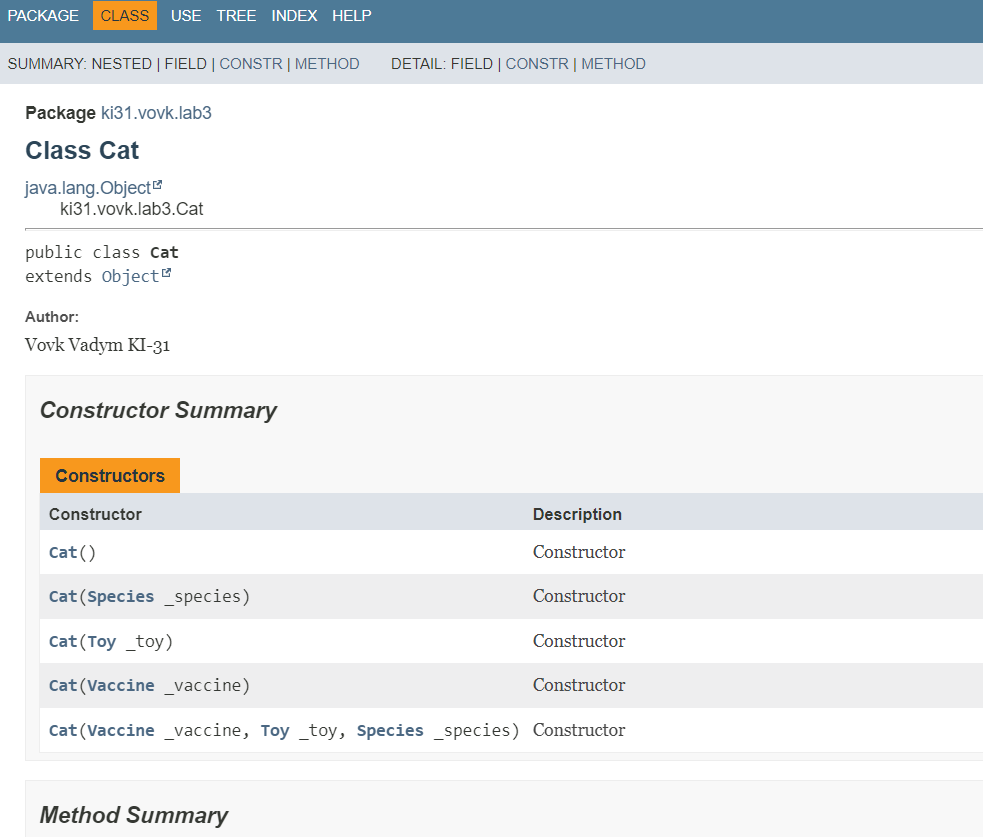
package ki.hordon.lab3;  
  
import java.util.ArrayList;  
import java.util.Arrays;  
  
  
  
public class Toy {  
  
 private ArrayList<String> toys;  
  
 */\*\*  
 \* Constructor  
 \*/* public Toy()  
 {  
 this.toys = new ArrayList<>();  
 }  
 */\*\*  
 \* Constructor  
 \** ***@param*** *toy new toy  
 \*/* public Toy(String toy)  
 {  
 this();  
 addToy(toy);  
 }  
 */\*\*  
 \* Constructor  
 \** ***@param*** *toy new toys  
 \*/* public Toy(String[] toy)  
 {  
 this();  
 addToys(toys.toArray(new String[0]));  
 }  
 */\*\*  
 \* Add toy to the object  
 \** ***@param*** *toy new toy  
 \*/* public void addToy(String toy)  
 {  
 this.toys.add(toy);  
 }  
 */\*\*  
 \* Add range of toys to the object  
 \** ***@param*** *toys range of toys  
 \*/* public void addToys(String[] toys)  
 {  
 this.toys.addAll(Arrays.*asList*(toys));  
 }  
 */\*\*  
 \* Get all toys  
 \** ***@return*** *list of toys  
 \*/* public ArrayList<String> getList()  
 {  
 return this.toys;  
 }  
  
 public String toString()  
 {  
 String string = new String();  
 for (int i = 0; i < this.toys.size(); i++)  
 string += this.toys.get(i) + " ";  
 return string;  
 }  
}

package ki.hordon.lab3;  
  
enum FilletType  
{  
 *SHORTHAIR*,  
 *CURL*,  
 *WIREHAIR*,  
 *UNKNOWN*,  
};  
  
public class Species {  
  
 private FilletType type;  
  
 */\*\*  
 \* Constructor  
 \*/* public Species()  
 {  
 this.type = FilletType.*UNKNOWN*;  
 }  
 */\*\*  
 \* Constructor  
 \** ***@param*** *filter new filler  
 \*/* public Species(FilletType filter)  
 {  
 this.type = filter;  
 }  
 */\*\*  
 \* Set new Filler  
 \** ***@param*** *type new FillerType  
 \*/* public void setSpecies(FilletType type)  
 {  
 this.type = type;  
 }  
 */\*\*  
 \* Get FillerType  
 \** ***@return*** *FillerType  
 \*/* public FilletType getSpecies()  
 {  
 return this.type;  
 }  
}

package ki.hordon.lab3;  
  
import java.io.File;  
import java.io.FileNotFoundException;  
import java.io.PrintWriter;  
  
  
  
public class Cat {  
  
 private Toy toy;  
 private Vaccine vaccine;  
 private Species species;  
 private PrintWriter printWriter;  
 */\*\*  
 \* Constructor  
 \*/* public Cat()  
 {  
 toy = new Toy();  
 vaccine = new Vaccine();  
 species = new Species();  
 species.setSpecies(FilletType.*UNKNOWN*);  
 }  
 */\*\*  
 \* Constructor  
 \** ***@param*** *\_vaccine Vaccine object  
 \*/* public Cat(Vaccine \_vaccine)  
 {  
 vaccine = \_vaccine;  
 species = new Species();  
 toy = new Toy();  
 species.setSpecies(FilletType.*UNKNOWN*);  
 }  
 */\*\*  
 \* Constructor  
 \** ***@param*** *\_toy Toy object  
 \*/* public Cat(Toy \_toy)  
 {  
 toy = \_toy;  
 vaccine = new Vaccine();  
 species = new Species();  
 species.setSpecies(FilletType.*UNKNOWN*);  
 }  
 */\*\*  
 \* Constructor  
 \** ***@param*** *\_species Species object  
 \*/* public Cat(Species \_species)  
 {  
 toy = new Toy();  
 vaccine = new Vaccine();  
 species = \_species;  
 }  
 */\*\*  
 \* Constructor  
 \** ***@param*** *\_vaccine Vaccine object  
 \** ***@param*** *\_toy Toy object  
 \** ***@param*** *\_species Species object  
 \*/* public Cat(Vaccine \_vaccine, Toy \_toy, Species \_species)  
 {  
 toy = \_toy;  
 vaccine = \_vaccine;  
 species = \_species;  
 }  
 */\*\*  
 \* Set new Toy to Cat  
 \** ***@param*** *\_toy new toy object  
 \*/* public void setToy(Toy \_toy)  
 {  
 this.toy = \_toy;  
 }  
 */\*\*  
 \* Add toy range to cat  
 \** ***@param*** *\_toy toy range  
 \*/* public void addToy(Toy \_toy)  
 {  
 String[] liStrings = new String[\_toy.getList().size()];  
 for (int i = 0; i < liStrings.length; i++)  
 liStrings[i] = \_toy.getList().get(i);  
 this.toy.addToys(liStrings);  
 }  
 */\*\*  
 \* Set new Vaccine to Cat  
 \** ***@param*** *\_vaccine new vaccine object  
 \*/* public void setVaccine(Vaccine \_vaccine)  
 {  
 this.vaccine = \_vaccine;  
 }  
 */\*\*  
 \* Add vaccine range to Cat  
 \** ***@param*** *\_vaccine new vaccine  
 \*/* public void addVaccine(Vaccine \_vaccine)  
 {  
 String[] liStrings = new String[\_vaccine.getList().size()];  
 for (int i = 0; i < liStrings.length; i++)  
 liStrings[i] = \_vaccine.getList().get(i);  
 this.vaccine.addVaccines(liStrings);  
 }  
 */\*\*  
 \* Set new Species to cat  
 \** ***@param*** *\_species new species  
 \*/* public void setSpecies(Species \_species)  
 {  
 this.species = \_species;  
 }  
 */\*\*  
 \* Return Cat Vaccine  
 \** ***@return*** *cat Vaccine  
 \*/* public Toy getToy()  
 {  
 return this.toy;  
 }  
 */\*\*  
 \* Return Cat Vaccine  
 \** ***@return*** *Cat Vaccine  
 \*/* public Vaccine getVaccine()  
 {  
 return this.vaccine;  
 }  
 */\*\*  
 \* Return Cat Species  
 \** ***@return*** *Cat Species  
 \*/* public Species getSpecies()  
 {  
 return this.species;  
 }  
 */\*\*  
 \* Open file by the path  
 \** ***@param*** *path way to the file  
 \** ***@throws*** *FileNotFoundException file exception  
 \*/* public void openFile(String path) throws FileNotFoundException  
 {  
 this.printWriter = new PrintWriter(new File(path));  
 }  
 */\*\*  
 \* Print Cat info to the file  
 \*/* public void printToFile()  
 {  
 if (this.printWriter != null)  
 {  
 this.printWriter.write("Cat info\n");  
 this.printWriter.write("Toy:\n");  
 this.printWriter.write(this.toy.toString());  
 this.printWriter.write("\nVaccine:\n");  
 this.printWriter.write(this.vaccine.toString());  
 this.printWriter.write("\nSpecies:");  
 switch (this.species.getSpecies()) {  
 case *SHORTHAIR*:  
 this.printWriter.write("SHORTHAIR");  
 break;  
 case *CURL*:  
 this.printWriter.write("CURL");  
 break;  
 case *WIREHAIR*:  
 this.printWriter.write("WIREHAIR");  
 break;  
 default:  
 this.printWriter.write("UNKNOWN");  
 break;  
 }  
 this.printWriter.write("\n");  
 }  
 }  
 */\*\*  
 \* Close file.  
 \*/* public void closeFile()  
 {  
 if (this.printWriter != null)  
 {  
 this.printWriter.flush();  
 this.printWriter.close();  
 }  
 }  
}

1. Після виконання програми переглянула створений файл kitty.txt:



1. Згенерувала документацію 

*Рис.3. Згенерований html-файл*

**Висновок:**

На даній лабораторній роботі ознайомилася з процесом розробки пакетів та класів мовою Java