**Міністерство Освіти І НАУКИ України**

**Національний університет "Львівська політехніка"**

Інститут **ІКНІ**

Кафедра **ПЗ**

### ЗВІТ

До лабораторної роботи № 2

**З дисципліни:** *“Кросплатформне програмування”*

**На тему:** *“* Робота з класами. Наслідування, поліморфізм. Створення класу-обгортки для колекції екземплярів. Сортування та пошук.*”*

**Лектор:**

доц. каф. ПЗ

Дяконюк Л.М.

**Виконав:**

ст. гр. ПЗ-36

Матвіїв Т.Т.

**Прийняв:**

асист. каф. ПЗ

Шкраб Р. Р.

« \_\_\_\_ » \_\_\_\_\_\_\_\_ 2021 р.

∑= \_\_\_\_\_

Львів – 2021

**Тема:** Робота з класами. Наслідування, поліморфізм. Створення класу-обгортки для колекції екземплярів. Сортування та пошук.

**Завдання**

Потрібно створити класи, що описують сутності запропоновані в завданні. Відобразити UML-діаграму класів. Створити програму та продемонструвати її роботу.

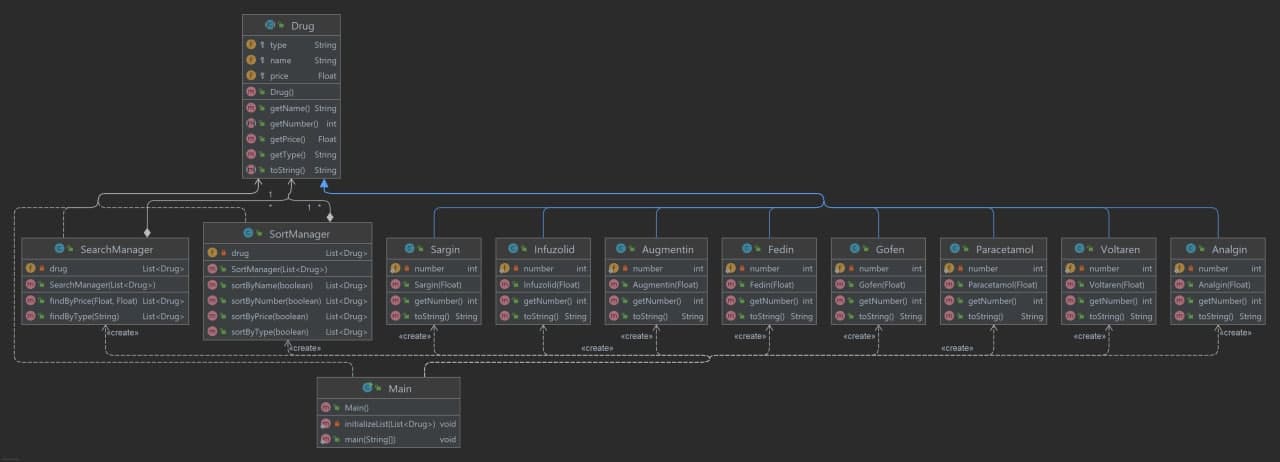
**Індивідуальне завдання**

Аптека. Реалізувати ієрархію товарів, присутніх в аптеці. Реалізувати пошук товарів залежно від категорії (наприклад протизапальні) та діапазону ціни.

Реалізувати можливість  сортування знайдених товарів за двома типами параметрів (на вибір, реалізовано як два окремі методи)

Реалізація сортування має передбачати можливість сортувати як за спаданням, так і за зростанням

**UML-діаграма**



**Код програми**

Analgin.java

package com.company;  
  
public class Analgin extends Drug{  
 private static int *number* = 0;  
  
 public Analgin(Float price) {  
 super.type = "Anti-inflammatory";  
 super.name = "Analgin";  
 super.price = price;  
 *number*++;  
 }  
  
  
 @Override  
 public String toString() {  
 return "Drug{" +  
 "type='" + type + '\'' +  
 ", name='" + name + '\'' +  
 ", price='" + price + '\'' +  
 ", number=" + *number* +  
 '}';  
 }  
  
 @Override  
 public int getNumber() {  
 return *number*;  
 }  
}

Augmentin.java

package com.company;  
  
public class Augmentin extends Drug{  
  
 private static int *number* = 0;  
  
 public Augmentin(Float price) {  
 super.type = "Antibiotics";  
 super.name = "Augmentin";  
 super.price = price;  
 *number*++;  
 }  
  
  
 @Override  
 public String toString() {  
 return "Drug{" +  
 "type='" + type + '\'' +  
 ", name='" + name + '\'' +  
 ", price='" + price + '\'' +  
 ", number=" + *number* +  
 '}';  
 }  
  
 @Override  
 public int getNumber() {  
 return *number*;  
 }  
}

Drug.java

package com.company;  
  
public abstract class Drug {  
 protected String type;  
 protected String name;  
 protected Float price;  
  
 public String getType() {  
 return type;  
 }  
  
 public String getName(){  
 return name;  
 }  
  
 public Float getPrice() {  
 return price;  
 }  
  
 public abstract String toString();  
  
 public abstract int getNumber();  
}

Fedin.java

package com.company;  
  
public class Fedin extends Drug{  
 private static int *number* = 0;  
  
 public Fedin(Float price) {  
 super.type = "Anti-inflammatory";  
 super.name = "Fedin";  
 super.price = price;  
 *number*++;  
 }  
  
  
 @Override  
 public String toString() {  
 return "Drug{" +  
 "type='" + type + '\'' +  
 ", name='" + name + '\'' +  
 ", price='" + price + '\'' +  
 ", number=" + *number* +  
 '}';  
 }  
  
 @Override  
 public int getNumber() {  
 return *number*;  
 }  
}

Gofen.java

package com.company;  
  
public class Gofen extends Drug{  
 private static int *number* = 0;  
  
 public Gofen(Float price) {  
 super.type = "Antipyretic";  
 super.name = "Gofen";  
 super.price = price;  
 *number*++;  
 }  
  
  
 @Override  
 public String toString() {  
 return "Drug{" +  
 "type='" + type + '\'' +  
 ", name='" + name + '\'' +  
 ", price='" + price + '\'' +  
 ", number=" + *number* +  
 '}';  
 }  
  
 @Override  
 public int getNumber() {  
 return *number*;  
 }  
}

Infuzolid.java

package com.company;  
  
public class Infuzolid extends Drug{  
 private static int *number* = 0;  
  
 public Infuzolid(Float price) {  
 super.type = "Antibiotics";  
 super.name = "Infuzolid";  
 super.price = price;  
 *number*++;  
 }  
  
 @Override  
 public String toString() {  
 return "Drug{" +  
 "type='" + type + '\'' +  
 ", name='" + name + '\'' +  
 ", price='" + price + '\'' +  
 ", number=" + *number* +  
 '}';  
 }  
  
 @Override  
 public int getNumber() {  
 return *number*;  
 }  
}

Main.java

package com.company;  
  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Scanner;  
  
public class Main {  
  
 public static void main(String[] args) {  
  
 List<Drug> drugList = new ArrayList<>();  
 *initializeList*(drugList);  
  
 System.*out*.println("Search by: 1 - Type; 2 - Price");  
 Scanner scanner = new Scanner(System.*in*);  
 String search = scanner.next();  
 switch (search) {  
 case "1":  
 System.*out*.println("1 - Antibiotics; 2 - Anti-inflammatory; 3 - Antipyretic");  
 String choise = scanner.next();  
 SearchManager searchManager = new SearchManager(drugList);  
  
 switch (choise) {  
 case "1":  
 drugList = searchManager.findByType("Antibiotics");  
 break;  
 case "2":  
 drugList = searchManager.findByType("Anti-inflammatory");  
 break;  
 case "3":  
 drugList = searchManager.findByType("Antipyretic");  
 break;  
 default:  
 return;  
 }  
  
 SortManager sortManager = new SortManager(drugList);  
  
 List<Drug> tempResult = sortManager.sortByNumber(true);  
 for (Drug r : tempResult) {  
 System.*out*.println(r);  
 }  
 System.*out*.println();  
 tempResult = sortManager.sortByPrice(false);  
 for (Drug r : tempResult) {  
 System.*out*.println(r);  
 }  
 System.*out*.println();  
 tempResult = sortManager.sortByType(false);  
 for (Drug r : tempResult) {  
 System.*out*.println(r);  
 }  
 System.*out*.println();  
 tempResult = sortManager.sortByName(false);  
 for (Drug r : tempResult) {  
 System.*out*.println(r);  
 }  
 scanner.close();  
 break;  
 case "2":  
 System.*out*.println("Enter min price:");  
 Float min = Float.*valueOf*(scanner.next());  
 System.*out*.println("Enter max price:");  
 Float max = Float.*valueOf*(scanner.next());  
  
 SearchManager searchManager2 = new SearchManager(drugList);  
  
 drugList = searchManager2.findByPrice(min,max);  
 for (Drug r : drugList) {  
 System.*out*.println(r);  
 }  
  
 System.*out*.println();  
 SortManager sortManager2 = new SortManager(drugList);  
 tempResult = sortManager2.sortByType(false);  
 for (Drug r : tempResult) {  
 System.*out*.println(r);  
 }  
 break;  
 default:  
 return;  
 }  
 }  
 private static void initializeList (List < Drug > drug) {  
 for (int i = 0; i < 3; i++) {  
 drug.add(new Infuzolid(202.8f));  
 drug.add(new Sargin(90.25f));  
 drug.add(new Analgin(40.00f));  
 }  
  
 for (int i = 0; i < 2; i++) {  
 drug.add(new Voltaren(267.80f));  
 drug.add(new Augmentin(157.60f));  
 drug.add(new Paracetamol(28.90f));  
 }  
  
 drug.add(new Fedin(67.50f));  
 drug.add(new Gofen(49.50f));  
 }  
}

Paracetamol.java

package com.company;  
  
public class Paracetamol extends Drug{  
 private static int *number* = 0;  
  
 public Paracetamol(Float price) {  
 super.type = "Antipyretic";  
 super.name = "Paracetamol";  
 super.price = price;  
 *number*++;  
 }  
  
  
 @Override  
 public String toString() {  
 return "Drug{" +  
 "type='" + type + '\'' +  
 ", name='" + name + '\'' +  
 ", price='" + price + '\'' +  
 ", number=" + *number* +  
 '}';  
 }  
  
 @Override  
 public int getNumber() {  
 return *number*;  
 }  
  
}

Sargin.java

package com.company;  
  
public class Sargin extends Drug{  
 private static int *number*;  
  
 public Sargin(Float price) {  
 super.type = "Antibiotics";  
 super.name = "Sargin";  
 super.price = price;  
 *number*++;  
 }  
  
 @Override  
 public String toString() {  
 return "Drug{" +  
 "type='" + type + '\'' +  
 ", name='" + name + '\'' +  
 ", price='" + price + '\'' +  
 ", number=" + *number* +  
 '}';  
 }  
  
 @Override  
 public int getNumber() {  
 return *number*;  
 }  
}

SearchManager.java

package com.company;  
  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Objects;  
  
public class SearchManager {  
  
 private List<Drug> drug;  
  
 public SearchManager(List<Drug> drug){  
 this.drug = drug;  
 }  
  
 public List<Drug> findByType(String type){  
 List<Drug> result = new ArrayList<>();  
 for(Drug drug : drug){  
 if (type.equals(drug.getType())){  
 result.add(drug);  
 }  
 }  
 return result;  
 }  
  
 public List<Drug> findByPrice(Float priceMin, Float priceMax){  
 List<Drug> result = new ArrayList<>();  
 for(Drug d : drug){  
 if (priceMin < d.getPrice()&& priceMax > d.getPrice()){  
 result.add(d);  
 }  
 }  
 return result;  
 }  
}

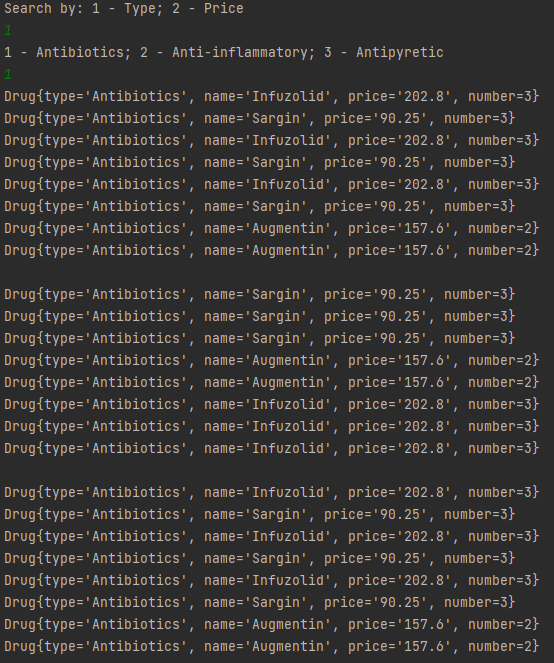
SortManager.java

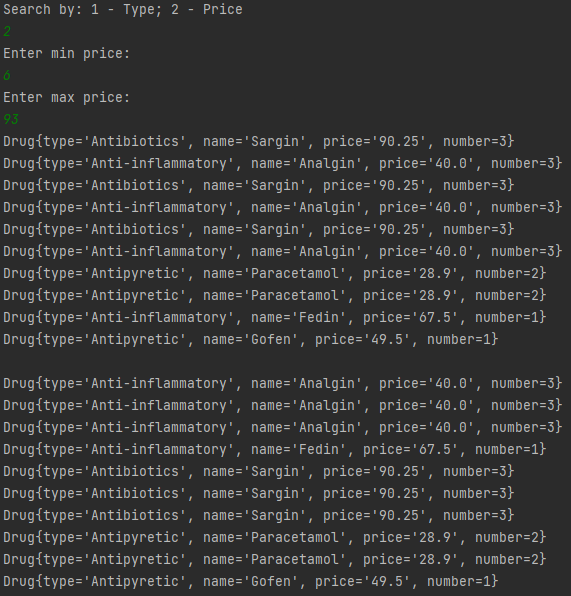
package com.company;  
  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.Comparator;  
import java.util.List;  
  
public class SortManager {  
  
 private List<Drug> drug;  
  
 public SortManager(List<Drug> drug){  
 this.drug = drug;  
 }  
  
  
 public List<Drug> sortByNumber(boolean check){  
  
 List<Drug> result = new ArrayList<>(drug);  
 if(check){  
 NumberComparator comparator = new NumberComparator();  
 result.sort(comparator.reversed());  
 return result;  
 }  
 result.sort(new NumberComparator());  
 return result;  
 }  
  
 public List<Drug> sortByType(boolean check){  
 TypeComparator comparator = new TypeComparator();  
 List<Drug> result = new ArrayList<>(drug);  
 if(check){  
 result.sort(Collections.*reverseOrder*(comparator));  
 return result;  
 }  
 result.sort(comparator);  
 return result;  
 }  
  
 public List<Drug> sortByPrice(boolean check){  
 List<Drug> result = new ArrayList<>(drug);  
  
 Comparator<Drug> priceComparator = new Comparator<Drug>() {  
 @Override  
 public int compare(Drug o1, Drug o2) {  
 int res = o1.getPrice().compareTo(o2.getPrice());  
 if(res == 0){  
 res = o1.getName().compareTo(o1.getName());  
 }  
 return res;  
 }  
 };  
  
 if (check){  
 result.sort(Collections.*reverseOrder*(priceComparator));  
 return result;  
 }  
  
 result.sort(priceComparator);  
 return result;  
  
 }  
  
 public List<Drug> sortByName(boolean check){  
 List<Drug> result = new ArrayList<>(drug);  
  
 Comparator<Drug> nameComparator = (o1, o2) -> o1.getName().compareTo(o2.getName());  
  
 if (check){  
 result.sort(Collections.*reverseOrder*(nameComparator));  
 return result;  
 }  
  
 result.sort(nameComparator);  
 return result;  
 }  
  
 private static class NumberComparator implements Comparator<Drug>{  
  
 @Override  
 public int compare(Drug o1, Drug o2) {  
 int res = Integer.*compare*(o1.getNumber(), o2.getNumber());  
 return res;  
 }  
 }  
  
 private class TypeComparator implements Comparator<Drug>{  
  
 @Override  
 public int compare(Drug o1, Drug o2) {  
 return o1.getType().compareTo(o2.getType());  
 }  
  
 }  
  
}

Voltaren.java

package com.company;  
  
public class Voltaren extends Drug{  
  
 private static int *number* = 0;  
  
 public Voltaren(Float price) {  
 super.type = "Anti-inflammatory";  
 super.name = "Voltaren";  
 super.price = price;  
 *number*++;  
 }  
  
 @Override  
 public String toString() {  
 return "Drug{" +  
 "type='" + type + '\'' +  
 ", name='" + name + '\'' +  
 ", price='" + price + '\'' +  
 ", number=" + *number* +  
 '}';  
 }  
  
 @Override  
 public int getNumber() {  
 return *number*;  
 }  
}

**Результат виконання**





Посилання на репозиторій:

**Висновок**

Виконуючи дану лабораторну роботу, я навчився працювати з класами в Java, зрозумів як працює інкапсуляція, поліморфізм і наслідування в Java. Також я вивчив колекції, які є на цій мові, зрозумів та реалізував різні способи сортування, такі як за допомогою компаратора з використанням static inner class, inner class, anonymous inner class та за допомогою лямбда-виразів. На основі отриманих знань я реалізував програму “аптека”, де я і використав все вивчене.