**Міністерство освіти і науки України**

**Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"**

**Факультет інформатики та обчислювальної техніки**

**Звіт**

з лабораторної роботи № 1 з дисципліни

«Основи розробки програмного

забезпечення на платформі Java»

„**Дослідження роботи з класами у мові Java**”

**Виконала**

*Колбун Ірина Олексіївна, ІП-92*

**Варіант 26**

Київ 2021

Зміст

[Мета роботи та завдання 3](#_Toc65105277)

[Код програми 3](#_Toc65105278)

[Результати виконання програми 8](#_Toc65105279)

[Висновок 9](#_Toc65105280)

# Мета роботи та завдання

**Мета роботи:** вивчити семантику опису класу, створення об'єктів, реалізацію принципу ООП – інкапсуляція.

**Завдання:** напишіть програму на Java, яка включає наступні дії:

− опис класу згідно з варіантом завдання (таблиця);

− створення набору даних (масиву) з об’єктів, описаного класу (не менш 10 елементів);

− оброблює масив об’єктів згідно з варіантом завдання та виводить в консоль результат.

|  |  |  |
| --- | --- | --- |
| 26. | **Сантехніка**: Ідентифікаційний номер, Вид, Фірма, Назва, Кількість, Ціна, Модель; Конструктор, Методи доступу, Метод *toString()* | 1. Отримати список сантехніки заданої фірми. 2. Отримати список сантехніки заданого виду і менше вказаної ціни. |

# Код програми

Файл “Main.java”

public class Main {  
 public static void main(String[] args) {  
 PlumbingController controller = new PlumbingController();  
 controller.run();  
 }  
}

Файл “Plumbing.java”

class Plumbing { // Model  
 private int idNum;  
 private String kind;  
 private String manufacturer;  
 private String name;  
 private int quantity;  
 private double price;  
 private String model;  
  
 Plumbing(int idNum, String kind, String manufacturer,  
 String name, int quantity, double price, String model) {  
 this.idNum = idNum;  
 this.kind = kind;  
 this.manufacturer = manufacturer;  
 this.name = name;  
 this.quantity = quantity;  
 this.price = price;  
 this.model = model;  
 }  
  
 public int getId() {  
 return idNum;  
 }  
  
 public String getKind() {  
 return kind;  
 }  
  
 public String getManufacturer() {  
 return manufacturer;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public int getQuantity() {  
 return quantity;  
 }  
  
 public double getPrice() {  
 return price;  
 }  
  
 public String getModel() {  
 return model;  
 }  
  
}

Файл “PlumbingController.java”

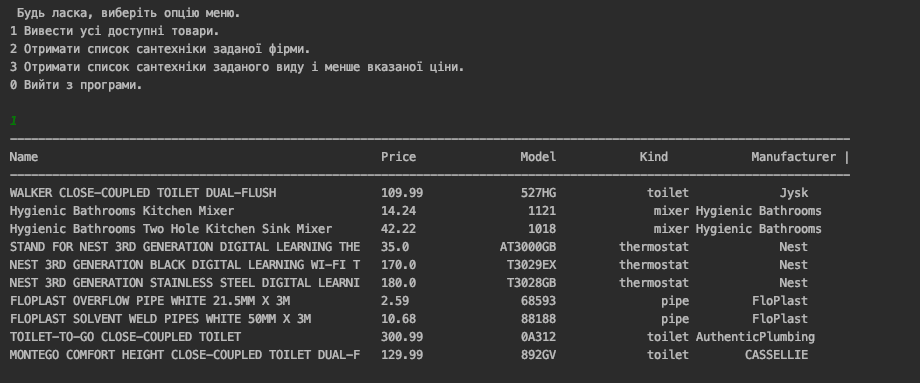
import java.util.Arrays;  
  
class PlumbingController { // Controller  
 private Plumbing[] models;  
 private PlumbingView view;  
  
 public PlumbingController(){  
 this.models = getModelsFromDatabase();  
 this.view = new PlumbingView();  
 }  
  
 public Plumbing[] getPlumbings() {  
 return models;  
 }  
  
 public void run()  
 {  
 int option;  
 do {  
 do {  
 option = view.menuPrompt();  
 } while(option < 0 || option > 3); // This will make the menu repeat if option is higher than 6 or lowen than 0.  
  
 switch(option) {  
 case 1:  
 view.printPlumbings(getPlumbings());  
 break;  
 case 2:  
 String manufacturer = view.manufacturerPrompt();  
 view.printPlumbings(*getByManufacturer*(manufacturer, getPlumbings()));  
 break;  
 case 3:  
 String kind = view.kindPrompt();  
 double price = view.maxPricePrompt();  
 view.printPlumbings(getByKindCostLessThan(kind, price, getPlumbings()));  
 break;  
 default:  
 break; //I always use this break, even when not needed.  
 }  
 } while (option != 0);  
 }  
  
 public static Plumbing[] getByManufacturer(String manufacturer, Plumbing[] plumbings) {  
 Plumbing[] filtered\_plumbings = new Plumbing[0];  
 for (Plumbing plumbing : plumbings) {  
 if (plumbing.getManufacturer().equalsIgnoreCase(manufacturer))  
 {  
 filtered\_plumbings = Arrays.copyOf(filtered\_plumbings, filtered\_plumbings.length + 1);  
 filtered\_plumbings[filtered\_plumbings.length - 1] = plumbing;  
 }  
 }  
 return filtered\_plumbings;  
 }  
  
 public static Plumbing[] getByKindCostLessThan(String kind, double max\_price, Plumbing[] plumbings) {  
 Plumbing[] filtered\_plumbings = new Plumbing[0];  
 for (Plumbing plumbing : plumbings) {  
 if (plumbing.getKind().equalsIgnoreCase(kind) && plumbing.getPrice() < max\_price)  
 {  
 filtered\_plumbings = Arrays.copyOf(filtered\_plumbings, filtered\_plumbings.length + 1);  
 filtered\_plumbings[filtered\_plumbings.length - 1] = plumbing;  
 }  
 }  
 return filtered\_plumbings;  
 }  
  
 private Plumbing[] getModelsFromDatabase()  
 {  
 Plumbing[] plumbings = new Plumbing[10];  
 plumbings[0] = new Plumbing(0, "toilet", "Jysk", "WALKER CLOSE-COUPLED TOILET DUAL-FLUSH", 0, 109.99, "527HG");  
 plumbings[1] = new Plumbing(1, "mixer", "Hygienic Bathrooms", "Hygienic Bathrooms Kitchen Mixer", 30, 14.24, "1121");  
 plumbings[2] = new Plumbing(2, "mixer", "Hygienic Bathrooms", "Hygienic Bathrooms Two Hole Kitchen Sink Mixer", 80, 42.22, "1018");  
 plumbings[3] = new Plumbing(3, "thermostat", "Nest", "STAND FOR NEST 3RD GENERATION DIGITAL LEARNING THERMOSTAT", 60, 35.00, "AT3000GB");  
 plumbings[4] = new Plumbing(4, "thermostat", "Nest", "NEST 3RD GENERATION BLACK DIGITAL LEARNING WI-FI THERMOSTAT", 2, 170.00, "T3029EX");  
 plumbings[5] = new Plumbing(5, "thermostat", "Nest", "NEST 3RD GENERATION STAINLESS STEEL DIGITAL LEARNING WI-FI THERMOSTAT", 5, 180.00, "T3028GB");  
 plumbings[6] = new Plumbing(6, "pipe", "FloPlast", "FLOPLAST OVERFLOW PIPE WHITE 21.5MM X 3M", 30, 2.59, "68593");  
 plumbings[7] = new Plumbing(7, "pipe", "FloPlast", "FLOPLAST SOLVENT WELD PIPES WHITE 50MM X 3M", 100, 10.68, "88188");  
 plumbings[8] = new Plumbing(8, "toilet", "AuthenticPlumbing", "TOILET-TO-GO CLOSE-COUPLED TOILET", 0, 300.99, "0A312");  
 plumbings[9] = new Plumbing(9, "toilet", "CASSELLIE", "MONTEGO COMFORT HEIGHT CLOSE-COUPLED TOILET DUAL-FLUSH", 0, 129.99, "892GV");  
 return plumbings;  
 }  
  
}

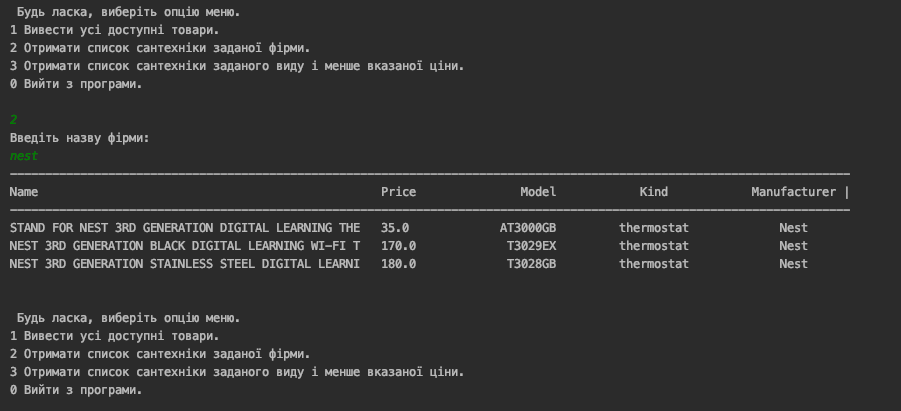
Файл “PlumbingView.java”

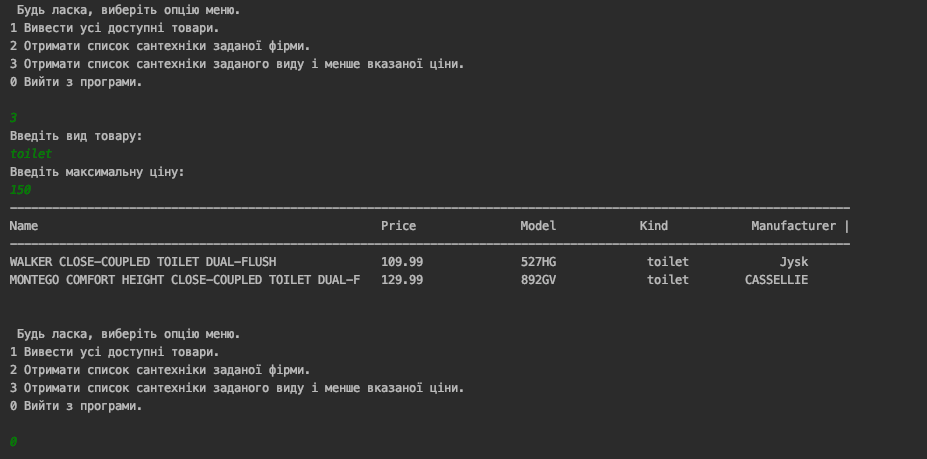
import java.util.Scanner;  
  
class PlumbingView { // View  
 Scanner in;  
 String menu = "\n Будь ласка, виберіть опцію меню."  
 + "\n1 Вивести усі доступні товари."  
 + "\n2 Отримати список сантехніки заданої фірми."  
 + "\n3 Отримати список сантехніки заданого виду і менше вказаної ціни."  
 + "\n0 Вийти з програми. \n";  
  
 PlumbingView() {  
 in = new Scanner(System.*in*);  
 }  
  
 void scannerSkip()  
 {  
 in.skip("(\r\n|[\n\r\u2028\u2029\u0085])?");  
 }  
  
 int menuPrompt() {  
 System.*out*.println(menu);  
 int res = in.nextInt();  
 scannerSkip();  
 return res;  
 }  
  
 String manufacturerPrompt() {  
 System.*out*.println("Введіть назву фірми: ");  
 return in.nextLine();  
 }  
  
 String kindPrompt() {  
 System.*out*.println("Введіть вид товару: ");  
 return in.nextLine();  
 }  
  
 double maxPricePrompt() {  
 System.*out*.println("Введіть максимальну ціну: ");  
 double res = in.nextDouble();  
 scannerSkip();  
 return res;  
 }  
  
 public void printPlumbing(Plumbing plumbing){  
 System.*out*.printf("%-50s %-7s %15s %18s %16s %n",  
 plumbing.getName().substring(0, Math.*min*(plumbing.getName().length(), 50)),  
 plumbing.getPrice(), plumbing.getModel(), plumbing.getKind(), plumbing.getManufacturer());  
 }  
  
 public void printPlumbings(Plumbing[] plumbings) {  
 PlumbingView view = new PlumbingView();  
 System.*out*.println("-".repeat(119) + "-");  
 System.*out*.println("Name" + " ".repeat(48) + " Price " + " ".repeat(13) + " Model " + " ".repeat(10) + " Kind " + " ".repeat(10) + " Manufacturer |");  
 System.*out*.println("-".repeat(119) + "-");  
 for (Plumbing plumbing : plumbings) {  
 printPlumbing(plumbing);  
 }  
 System.*out*.println();  
 }  
}

# Результати виконання програми

Нижче представлені результати виконання програми у консолі.







# Висновок

Під час виконання даної лабораторної роботи я повторила використання синтаксису мови програмування Java. А саме те, як описувати класи, створювати об’єкти певних класів та працювати із ними. Я вивчила та застосувала паттерн MVC (model, view, controller) для проектування своєї програми.

Також, повторила принципи ООП, застосувала інкапсуляцію – об’єднання даних та методів роботи з ними у класі, при тому що деталі реалізації недоступні користувачу.