**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

**ЛАБОРАТОРНАЯ РАБОТА №1**

по дисциплине: «Технологии разработки программного обеспечения»

на тему: «Развитие представлений о разработке программ. Объектно-ориентированное программирование в Java»

Выполнил: студент гр. ИТП-21

Качура И. И.

Принял: преподаватель

Малиновский И.Л.

Гомель 2024

**Задание:**

1. Разработать UML-диаграмму иерархии классов, согласно варианта (таблица 1).

2. При наименовании компонентов руководствоваться соглашением о наименовании (https://www.oracle.com/technetwork/java/codeconventions-150003.pdf).

3. При описании иерархии использовать наследование и композицию.

4. На основе UML-диаграммы разработать иерархию классов на языке Java.

5. Весь код должен быть снабжен элементами документирования

(https://www.jetbrains.com/help/idea/working-with-code-documentation.html).

6. Разработанную иерархию поместить в .jar файл для дальнейшего использования

в качестве библиотечных классов.

7. Создать консольное приложение для демонстрации работы созданных классов.

8. Составить отчет о проделанной работе.

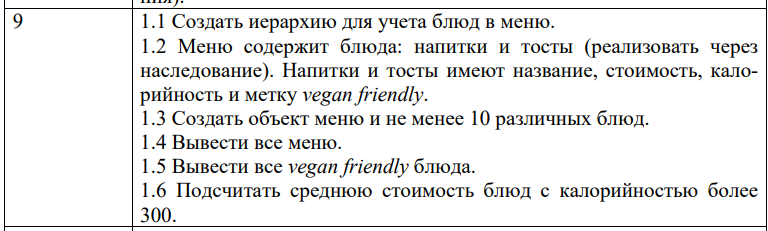


Рисунок 1. Задание для Варианта 9

Работа программы представлена на рисунке 2

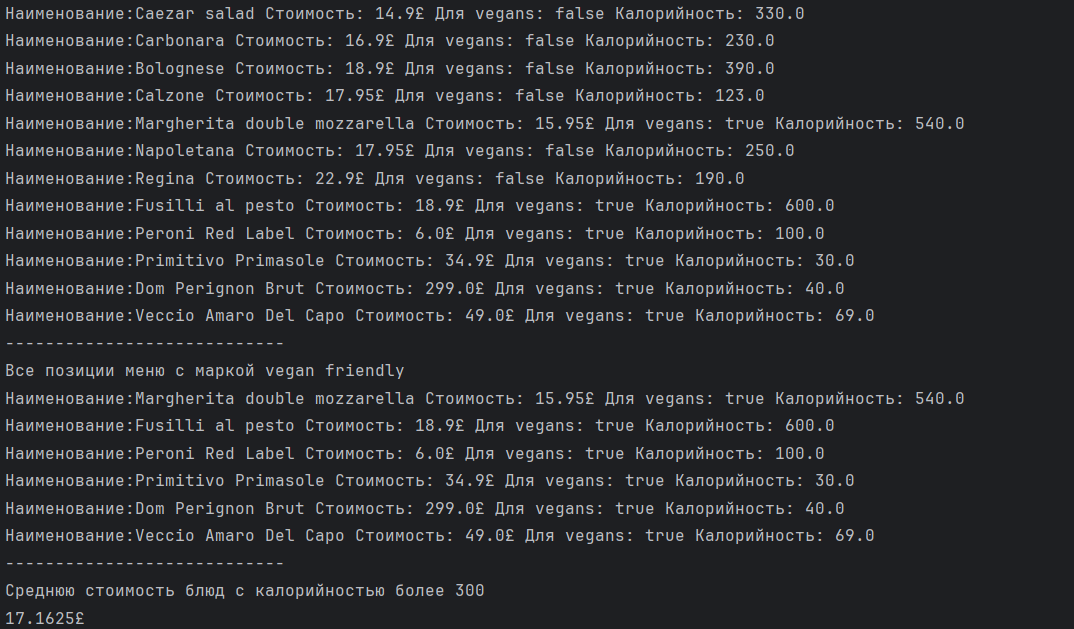


Рисунок 2. Работа программы

Диаграмма иерархии классов представлена на рисунке 3.

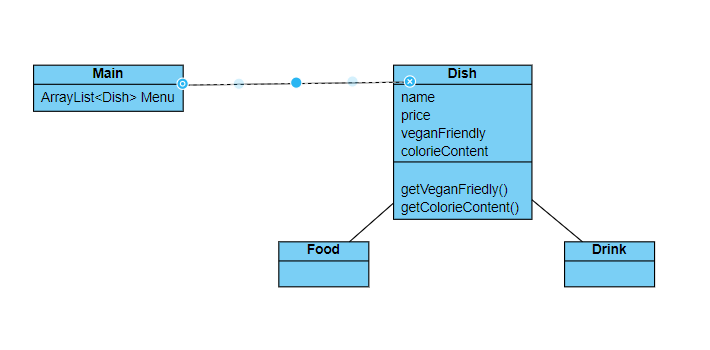


Рисунок 3. UML-диаграмма

Листинг программы представлен в приложениях А-Г.

**Вывод:** в ходе лабораторной работы изучили методы разработки библиотек классов на языке “*Java*”.

.

**ПРИЛОЖЕНИЕ А**

**Листинг класса «*main»***

import java.io.Console;

import java.util.ArrayList;

import java.util.List;

//TIP To <b>Run</b> code, press <shortcut actionId="Run"/> or

// click the <icon src="AllIcons.Actions.Execute"/> icon in the gutter.

public class Main {

public static void main(String[] args) {

//TIP Press <shortcut actionId="ShowIntentionActions"/> with your caret at the highlighted text

// to see how IntelliJ IDEA suggests fixing it.

final List<Dish> Menu = new ArrayList<>();

AddInMenu:

{

Food CaezarSalad = new Food("Caezar salad", 14.90, false, 330);

Food Carbonara = new Food("Carbonara", 16.90, false, 230);

Food Bologneze = new Food("Bolognese", 18.90, false, 390);

Food Calzone = new Food("Calzone", 17.95, false, 123);

Food Margherita = new Food("Margherita double mozzarella", 15.95, true, 540);

Food Napoletana = new Food("Napoletana", 17.95, false, 250);

Food Regina = new Food("Regina", 22.90, false, 190);

Food Fusilli = new Food("Fusilli al pesto", 18.90, true, 600);

Drink Beer = new Drink("Peroni Red Label", 6, true, 100);

Drink RedWine = new Drink("Primitivo Primasole", 34.90, true, 30);

Drink WhiteWine = new Drink("Dom Perignon Brut", 299.0, true, 40);

Drink Digestif = new Drink("Veccio Amaro Del Capo", 49, true, 69);

Menu.add(CaezarSalad);

Menu.add(Carbonara);

Menu.add(Bologneze);

Menu.add(Calzone);

Menu.add(Margherita);

Menu.add(Napoletana);

Menu.add(Regina);

Menu.add(Fusilli);

Menu.add(Beer);

Menu.add(RedWine);

Menu.add(WhiteWine);

Menu.add(Digestif);

}

//1.4 Вывести все меню.

System.out.println("----------------------------");

System.out.println("Все позиции меню");

for(int i =0;i<Menu.size();i++)

{

System.out.println(Menu.get(i).toString());

}

//1.5 Вывести все vegan friendly блюда.

System.out.println("----------------------------");

System.out.println("Все позиции меню c маркой vegan friendly");

for(int i =0;i<Menu.size();i++)

{

if(Menu.get(i).getVeganFriedly() == true)

System.out.println(Menu.get(i).toString());

}

//1.6 Подсчитать среднюю стоимость блюд с калорийностью более 300

int index = 0;

double sum = 0;

System.out.println("----------------------------");

System.out.println("Cреднюю стоимость блюд с калорийностью более 300");

for(int i =0;i<Menu.size();i++)

{

if(Menu.get(i).getColorieContent() >= 300)

{

sum+=Menu.get(i).getPrice();

index++;

}

}

System.out.println(sum/index + "£");

}

}

**ПРИЛОЖЕНИЕ Б**

**Листинг класса Dish**

public abstract class Dish

{

//Меню содержит блюда: напитки и тосты (реализовать через

//наследование). Напитки и тосты имеют название, стоимость, калорийность и метку vegan friendly

private String name;

private double price;

private boolean veganFriedly;

private double colorieContent;

public Dish(String name,Double price,Boolean veganFriedly,Double colorieContent )

{

this.name = name;

this.price = price;

this.veganFriedly = veganFriedly;

this.colorieContent = colorieContent;

}

@Override

public String toString()

{

return("Наименование:" + name.toString() + " Стоимость: " + price + "£" + " Для vegans: " + veganFriedly+ " Калорийность: " + colorieContent);

}

public boolean getVeganFriedly()

{

return veganFriedly;

}

public double getColorieContent()

{

return colorieContent;

}

public double getPrice()

{

return price;

}

}

**ПРИЛОЖЕНИЕ В**

**Листинг класса “Food ”**

import java.text.DecimalFormat;

public class Food extends Dish

{

public Food(String name, double price, Boolean veganFriedly,double calorieContent) {

super(name,price,veganFriedly,calorieContent);

}

}

**ПРИЛОЖЕНИЕ Г**

**Листинг класса «** **Drink »**

public class Drink extends Dish

{

public Drink(String name, double price, Boolean veganFriedly,double calorieContent) {

super(name,price,veganFriedly,calorieContent);

}

}