МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

## Факультет информационных технологий и робототехники

Кафедра программного обеспечения информационных систем и технологий

**Отчет по лабораторной работе № 2**

по дисциплине: ”Конструирование программного обеспечения”

на тему: ***””***

Вариант 7

Выполнил**:** студенты группы 10701221 Стальмакова А.С.

Василенко В.Д.

Принял**:** ст.пр. Станкевич С.Н.

Минск 2023

**Цель работы**:

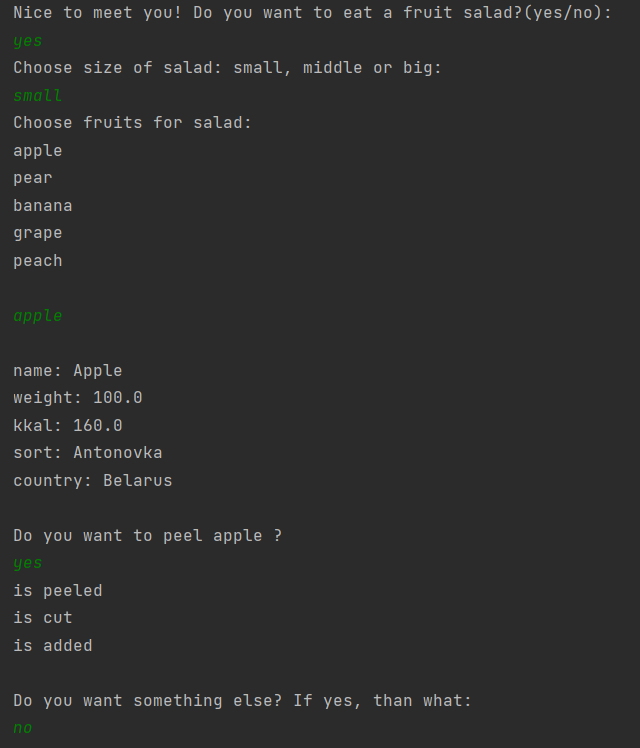
Научиться грамотно анализировать предметную область и с помощью аб-  
стракции выделять существенные детали, на базе которых в дальнейшем про-  
ектируются классы и объекты будущей программной системы согласно мето-  
дологии ООП, а также практически закрепить данные навыки при решении  
соответствующих задач (бизнес проблем).

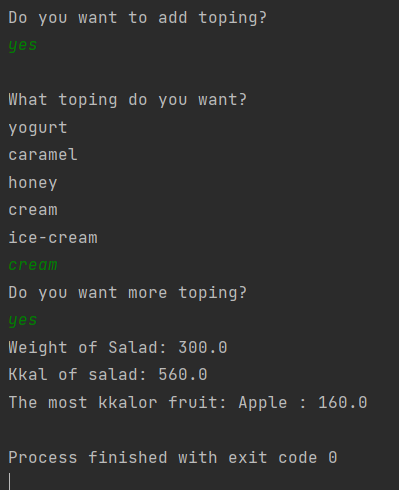
**Основное задание :**

Необходимо решить задачу с использованием методологии ООП. Для чего  
необходимо подобрать самостоятельно соответствующую проблемную  
(предметную/доменную) область, которая базируется на объектах и событиях  
реального мира (примеры соответствующих предметных областей приведены  
ниже). Спроектировать классы (собственные пользовательские типы данных)  
в языке Java для программного представления данных объектов и основной  
логики будущей программной системы.  
Система должна решать, как минимум, два полезных действия и иметь допол-  
нительно следующие опции:  
− не менее 3 разнообразных классов предметной области;  
− не менее 5 атрибутов (состояния) и методов (поведения) в классе-сущно-  
сти;  
− не менее 3 методов, которые реализуют бизнес-логику программы, в со-  
ответствующих функциональных классах;  
− хранить глобальные характеристики системы или характеристики уровня  
отдельных классов.  
На базе спроектированной программной системы реализовать программу и  
продемонстрировать её работоспособность.

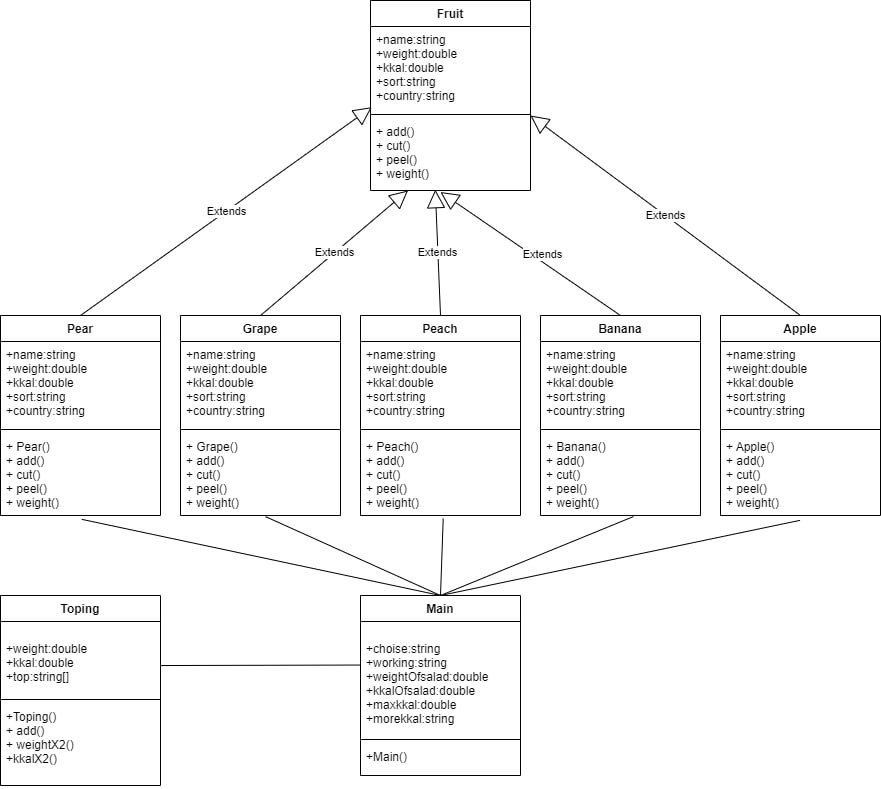
Шеф-повар (Chef). Необходимо приготовить фруктовый салат и определить  
вес салата и его калорийность, а также найти самый калорийных фрукт в са-  
лате.

**Результат работы программы:**

****

****

**UML-диаграмма программы:**



**Листинг программы:**

**Class Apple:**

public class Apple extends Fruits{  
 public Apple()  
 {  
 name = "Apple";  
 weight = 100;  
 kkal = 160;  
 sort = "Antonovka";  
 country = "Belarus";  
 }  
}

**Class Pear:**

public class Pear extends Fruits{  
 public Pear()  
 {  
 name = "Pear";  
 weight = 100;  
 kkal = 120;  
 sort = "Decertnaya";  
 country = "Krym";  
 }  
}

**Class Grape:**

public class Grape extends Fruits{  
  
 public Grape()  
 {  
 name = "Grape";  
 weight = 100;  
 kkal = 60;  
 sort = "Isabella";  
 country = "China";  
 }  
}

**Class Banana:**

public class Banana extends Fruits{  
 public Banana()  
 {  
 name = "Banana";  
 weight = 100;  
 kkal = 200;  
 sort = "Povan";  
 country = "India";  
 }  
}

**Class Peach:**

public class Peach extends Fruits{  
 public Peach()  
 {  
 name = "Peach";  
 weight = 100;  
 kkal = 95;  
 sort = "Gloria";  
 country = "Moldavia ";  
 }  
}

**Class Toping:**

import java.util.List;  
  
public class Toping {  
 String [] top= new String[]{"yogurt", "caramel", "honey", "ice-cream", "cream"};  
 double weight = 100;  
 double kkal = 200;  
 public Toping()  
 {  
  
 }  
 public void add()  
 {  
 System.*out*.println("is added");  
 }  
 public double weightX2()  
 {  
 return weight \*= 2;  
 }  
  
 public double kkalX2()  
 {  
 return kkal \*= 2;  
 }  
  
}

**Class Main:**

import java.util.Scanner;  
  
public class Main {  
 public static void main(String[] args) {  
  
 String working;  
 System.*out*.println("Nice to meet you! Do you want to eat a fruit salad?(yes/no):");  
 Scanner scanner = new Scanner(System.*in*);  
 working = scanner.nextLine();  
 String choiseOfSize;  
 double weightOfSalad = 0, kkalOfSalad = 0;  
 double maxKkal = 0;  
 String moreKkal = null;  
 if(working.equals("yes"))  
 {  
 System.*out*.println("Choose size of salad: small, middle or big: ");  
 choiseOfSize = scanner.nextLine();  
 System.*out*.println("Choose fruits for salad:\napple\npear\nbanana\ngrape\npeach\n");  
 while (working.equals("yes"))  
 {  
 String choiseFruit = scanner.nextLine();  
 String PeelYesOrNo;  
 switch (choiseFruit)  
 {  
 case "apple":  
 Apple apple = new Apple();  
 System.*out*.println("\nname: " + apple.name + "\nweight: " + apple.weight + "\nkkal: " + apple.kkal + "\nsort: " + apple.sort + "\ncountry: " + apple.country);  
 switch(choiseOfSize)  
 {  
 case "small":  
 weightOfSalad += apple.weight;  
 kkalOfSalad += apple.kkal;  
 break;  
 case "middle":  
 weightOfSalad += apple.weightX2();  
 kkalOfSalad += apple.kkalX2();  
 break;  
 case "big":  
 weightOfSalad += apple.weightX3();  
 kkalOfSalad += apple.kkalX3();  
 default: break;  
 }  
 System.*out*.println("\nDo you want to peel " + choiseFruit +" ?");  
 PeelYesOrNo = scanner.nextLine();  
 if(PeelYesOrNo.equals("yes"))  
 {  
 apple.peel();  
 }  
 apple.cut();  
 apple.add();  
 if(maxKkal < apple.kkal)  
 {  
 maxKkal = apple.kkal;  
 moreKkal = apple.name;  
 }  
 break;  
 case "pear":  
 Pear pear = new Pear();  
 System.*out*.println("\nname: " + pear.name + "\nweight: " + pear.weight + "\nkkal: " + pear.kkal + "\nsort: " + pear.sort + "\ncountry: " + pear.country);  
 switch(choiseOfSize)  
 {  
 case "small":  
 weightOfSalad += pear.weight;  
 kkalOfSalad += pear.kkal;  
 break;  
 case "middle":  
 weightOfSalad += pear.weightX2();  
 kkalOfSalad += pear.kkalX2();  
 break;  
 case "big":  
 weightOfSalad += pear.weightX3();  
 kkalOfSalad += pear.kkalX3();  
 default: break;  
 }  
 System.*out*.println("\nDo you want to peel " + choiseFruit +" ?");  
 PeelYesOrNo = scanner.nextLine();  
 if(PeelYesOrNo.equals("yes"))  
 {  
 pear.peel();  
 }  
 pear.cut();  
 pear.add();  
 if(maxKkal < pear.kkal)  
 {  
 maxKkal = pear.kkal;  
 moreKkal = pear.name;  
 }  
 break;  
 case "banana":  
 Banana banana = new Banana();  
 System.*out*.println("\nname: " + banana.name + "\nweight: " + banana.weight + "\nkkal: " + banana.kkal + "\nsort: " + banana.sort + "\ncountry: " + banana.country);  
 switch(choiseOfSize)  
 {  
 case "small":  
 weightOfSalad += banana.weight;  
 kkalOfSalad += banana.kkal;  
 break;  
 case "middle":  
 weightOfSalad += banana.weightX2();  
 kkalOfSalad += banana.kkalX2();  
 break;  
 case "big":  
 weightOfSalad += banana.weightX3();  
 kkalOfSalad += banana.kkalX3();  
 default: break;  
 }  
 banana.cut();  
 banana.add();  
 if(maxKkal < banana.kkal)  
 {  
 maxKkal = banana.kkal;  
 moreKkal = banana.name;  
 }  
 break;  
 case "grape":  
 Grape grape = new Grape();  
 System.*out*.println("\nname: " + grape.name + "\nweight: " + grape.weight + "\nkkal: " + grape.kkal + "\nsort: " + grape.sort + "\ncountry: " + grape.country);  
 switch(choiseOfSize)  
 {  
 case "small":  
 weightOfSalad += grape.weight;  
 kkalOfSalad += grape.kkal;  
 break;  
 case "middle":  
 weightOfSalad += grape.weightX2();  
 kkalOfSalad += grape.kkalX2();  
 break;  
 case "big":  
 weightOfSalad += grape.weightX3();  
 kkalOfSalad += grape.kkalX3();  
 default: break;  
 }  
 grape.add();  
 if(maxKkal < grape.kkal)  
 {  
 maxKkal = grape.kkal;  
 moreKkal = grape.name;  
 }  
 break;  
 case "peach":  
 Peach peach = new Peach();  
 System.*out*.println("\nname: " + peach.name + "\nweight: " + peach.weight + "\nkkal: " + peach.kkal + "\nsort: " + peach.sort + "\ncountry: " + peach.country);  
 switch(choiseOfSize)  
 {  
 case "small":  
 weightOfSalad += peach.weight;  
 kkalOfSalad += peach.kkal;  
 break;  
 case "middle":  
 weightOfSalad += peach.weightX2();  
 kkalOfSalad += peach.kkalX2();  
 break;  
 case "big":  
 weightOfSalad += peach.weightX3();  
 kkalOfSalad += peach.kkalX3();  
 default: break;  
 }  
 System.*out*.println("\nDo you want to peel " + choiseFruit +" ?");  
 PeelYesOrNo = scanner.nextLine();  
 if(PeelYesOrNo.equals("yes"))  
 {  
 peach.peel();  
 }  
 peach.cut();  
 peach.add();  
 if(maxKkal < peach.kkal)  
 {  
 maxKkal = peach.kkal;  
 moreKkal = peach.name;  
 }  
 break;  
 default:  
 System.*out*.println("\nWe have not such fruit!");  
 break;  
  
 }  
 System.*out*.println("\nDo you want something else? If yes, than what: ");  
 working = scanner.nextLine();  
 }  
 }  
 System.*out*.println("\nDo you want to add toping?");  
 String topYesOrNo = scanner.nextLine();  
 if(topYesOrNo.equals("yes"))  
 {  
 Toping toping = new Toping();  
 System.*out*.println("\nWhat toping do you want?\nyogurt\ncaramel\nhoney\ncream\nice-cream");  
 String choiseToping = scanner.nextLine();  
 for (int i = 0; i < toping.top.length; i++)  
 {  
 if(choiseToping.equals(toping.top[i])) {  
 System.*out*.println("Do you want more toping?");  
 String moreTopping = scanner.nextLine();  
 if (moreTopping.equals("yes")) {  
 weightOfSalad += toping.weightX2();  
 kkalOfSalad += toping.kkalX2();  
 } else {  
 weightOfSalad += toping.weight;  
 kkalOfSalad += toping.kkal;  
 }  
 }  
 }  
 }  
 System.*out*.println("Weight of Salad: " + weightOfSalad);  
 System.*out*.println("Kkal of salad: " + kkalOfSalad);  
 System.*out*.println("The most kkalor fruit: " + moreKkal + " : " + maxKkal);  
 }  
}

**Class Fruit:**

public class Fruits {  
 String name;  
 double weight;  
 double kkal;  
 String sort;  
 String country;  
  
 public Fruits()  
 {  
  
 }  
  
 public void add()  
 {  
 System.*out*.println("is added");  
 }  
  
 public void cut()  
 {  
 System.*out*.println("is cut");  
 }  
  
 public void peel()  
 {  
 System.*out*.println("is peeled");  
 }  
  
 public double weightX2()  
 {  
 return weight \*= 2;  
 }  
  
 public double kkalX2()  
 {  
 return kkal \*= 2;  
 }  
  
 public double weightX3()  
 {  
 return weight \*= 3;  
 }  
  
 public double kkalX3()  
 {  
 return kkal \*= 3;  
 }  
  
  
}