НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

**Лабораторна робота №5**

з дисципліни **«**Основи технологій програмування**»**

Варіант: «6412»

Виконав:

студент 2 курсу

група ІП-64

Лейзьо Сергій Іванович

Перевірив:

Подрубайло Олександр Олександрович

Київ – 2018

**Завдання**

**Тема:** Наслідування та поліморфізм.

**Мета:** Ознайомлення з механізмом наслідування та принципом поліморфізму.

Використання механізму наслідування та принципу поліморфізму в мові

Java. Здобуття навичок у використанні механізму наслідування та принципу

поліморфізму.

**6412%17=3**

3) Визначити ієрархію овочів. Зробити салат. Порахувати калорійність салату.

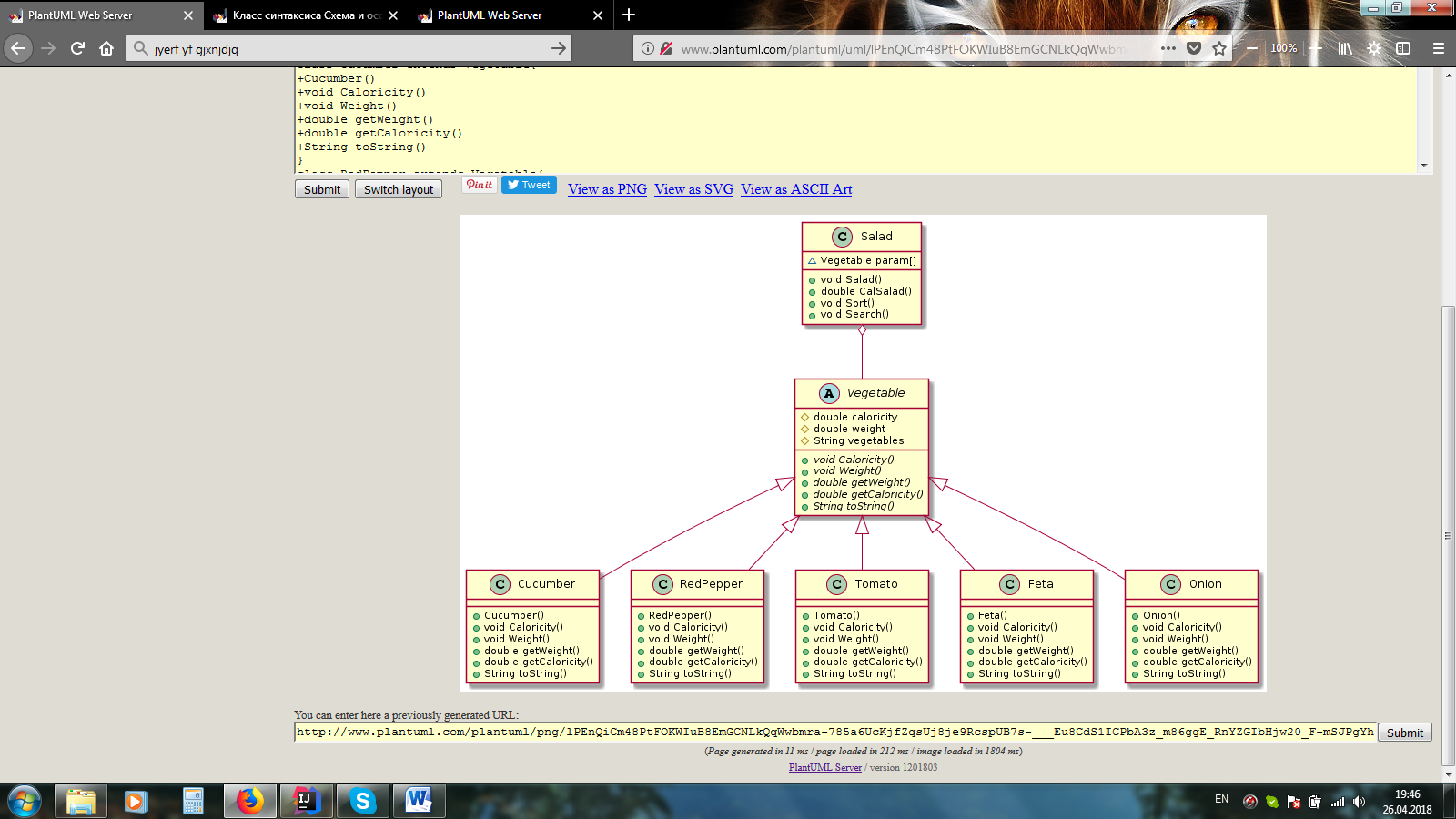
Провести сортування овочів для салату на основі одного з параметрів. Знайти

овоч у салаті, , що відповідає заданому діапазону калорійності.

**Програмний код**

package com.company;  
import java.util.Scanner;  
import java.util.Comparator;  
import java.util.Arrays;  
  
public class Main {  
 public static void main(String[] args) {  
 Salad salad=new Salad();  
 salad.Salad();  
 System.*out*.println("\nKaлорийность салата: " + salad.CalSalad());  
 System.*out*.println("Как вы хотите отсортировать овощи? 1-по калориям 2-по массе");  
 Scanner sort=new Scanner(System.*in*);  
 int number=sort.nextInt();  
 salad.Sort(number);  
 System.*out*.println("Введите диапазон калорийности от a до b:");  
 System.*out*.print("a:");  
 Scanner from=new Scanner(System.*in*);  
 int a=from.nextInt();  
 System.*out*.print("b:");  
 Scanner to=new Scanner(System.*in*);  
 int b=to.nextInt();  
 salad.Search(a,b);  
 }  
}  
class Salad{  
 Vegetable param[]={new Cucumber(16,100,"Cucumber"),new RedPepper(20,110,"RedPepper"),new Tomato(180,101,"Tomato"),new Feta(264,200,"Feta"),new Onion(35,90,"Onion")};  
 public void Salad(){  
 for(int i=0;i<param.length;i++){  
 this.param[i].Weight();  
 this.param[i].Caloricity();  
 }  
 }  
 public double CalSalad(){  
 double cal=0;  
 for(int i=0;i<param.length;i++){  
 cal=cal+param[i].caloricity;  
 }  
 return cal;  
 }  
 public void Sort(int number)  
 {  
 switch (number){  
 case 1: Arrays.*sort*(param, Comparator.*comparingDouble*(o -> o.getCaloricity()));  
 System.*out*.println("Отсортировано по калориям: ");  
 for(int i=0;i<param.length;i++){  
 System.*out*.print(param[i].getCaloricity()+" ");  
 System.*out*.println(param[i].toString());  
 }  
 break;  
 case 2: Arrays.*sort*(param, Comparator.*comparingDouble*(o -> o.getWeight()));  
 System.*out*.println("Отсортировано по весу: ");  
 for(int i=0;i<param.length;i++){  
 System.*out*.print(param[i].getWeight()+" ");  
 System.*out*.println(param[i].toString());  
 }  
 break;  
 default: System.*out*.println("Вы ввели не то число!");  
 }  
 }  
 public void Search(int a,int b){  
 System.*out*.println("В этот диапазон входят: ");  
 for(int i=0;i<param.length;i++){  
 if((param[i].getCaloricity()>=a)&&(param[i].getCaloricity()<=b)){  
 System.*out*.println(param[i].toString());  
 }  
 }  
 }  
}  
abstract class Vegetable{  
 protected double caloricity;  
 protected double weight;  
 protected String vegetables;  
 abstract public void Caloricity();  
 abstract public void Weight();  
 abstract public double getWeight();  
 abstract public double getCaloricity();  
 abstract public String toString();  
}  
class Cucumber extends Vegetable{  
 public Cucumber(double caloricity, double weight, String vegetables){  
 super.caloricity=caloricity;  
 super.weight=weight;  
 super.vegetables=vegetables;  
 }  
 @Override  
 public void Caloricity(){  
 System.*out*.println(" Калорийность огурца: "+ super.caloricity);  
 }  
 @Override  
 public void Weight(){  
 System.*out*.print("Вес огурца: "+super.weight);  
 }  
 @Override  
 public double getWeight(){  
 return super.weight;  
 }  
 @Override  
 public double getCaloricity(){  
 return super.caloricity;  
 }  
 @Override  
 public String toString() {  
 return super.vegetables;  
 }  
}  
class RedPepper extends Vegetable{  
 public RedPepper(double caloricity, double weight, String vegetables){  
 super.caloricity=caloricity;  
 super.weight=weight;  
 super.vegetables=vegetables;  
 }  
 @Override  
 public void Caloricity(){  
 System.*out*.println(" Калорийность красного перца: "+ super.caloricity);  
 }  
 @Override  
 public void Weight(){  
 System.*out*.print("Вес красного перца: "+super.weight);  
 }  
 @Override  
 public double getWeight(){  
 return super.weight;  
 }  
 @Override  
 public double getCaloricity(){  
 return super.caloricity;  
 }  
 @Override  
 public String toString() {  
 return super.vegetables;  
 }  
}  
class Tomato extends Vegetable{  
 public Tomato(double caloricity, double weight, String vegetables){  
 super.caloricity=caloricity;  
 super.weight=weight;  
 super.vegetables=vegetables;  
 }  
 @Override  
 public void Caloricity(){  
 System.*out*.println(" Калорийность помидора: "+ super.caloricity);  
 }  
 @Override  
 public void Weight(){  
 System.*out*.print("Вес помидора: "+super.weight);  
 }  
 @Override  
 public double getWeight(){  
 return super.weight;  
 }  
 @Override  
 public double getCaloricity(){  
 return super.caloricity;  
 }  
 @Override  
 public String toString() {  
 return super.vegetables;  
 }  
}  
class Feta extends Vegetable{  
 public Feta(double caloricity, double weight, String vegetables){  
 super.caloricity=caloricity;  
 super.weight=weight;  
 super.vegetables=vegetables;  
 }  
 @Override  
 public void Caloricity(){  
 System.*out*.println(" Калорийность феты: "+ super.caloricity);  
 }  
 @Override  
 public void Weight(){  
 System.*out*.print("Вес феты: "+super.weight);  
 }  
 @Override  
 public double getWeight(){  
 return super.weight;  
 }  
 @Override  
 public double getCaloricity(){  
 return super.caloricity;  
 }  
 @Override  
 public String toString() {  
 return super.vegetables;  
 }  
}  
class Onion extends Vegetable{  
 public Onion(double caloricity, double weight, String vegetables){  
 super.caloricity=caloricity;  
 super.weight=weight;  
 super.vegetables=vegetables;  
 }  
 @Override  
 public void Caloricity(){  
 System.*out*.println(" Калорийность лука: "+ super.caloricity);  
 }  
 @Override  
 public void Weight(){  
 System.*out*.print("Вес лука: "+super.weight);  
 }  
 @Override  
 public double getWeight(){  
 return super.weight;  
 }  
 @Override  
 public double getCaloricity(){  
 return super.caloricity;  
 }  
 @Override  
 public String toString() {  
 return super.vegetables;  
 }  
}

**Дiаграма класiв**



**Висновок**

Під час виконання завдання було розроблено клас та 5 нащадків. Також, було створено абстрактний клас овочів, де прописані абстрактні методи, щоб в кожному класі-нащадку задавати свої параметри для цих методів. Умова задачі виконана, в програмі є 2 види сортування (за калоріями, за вагою). При введенні діапазону виводяться продукти, що належать цьому діапазону калорійності.