

Міністерство освіти і науки, молоді та спорту України

Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №5

З дисципліни «Об'єктно-орієнтоване програмування»

Тема: «Наслідування та поліморфізм»

Виконав:

студент групи ІВ-71

Мазан Ян Владиславович

Номер залікової книжки:

7109

Перевірив:

Подрубайло О. О.

Київ 2018

1. Варіант завдання.

Номер залікової книжки — 7109

$C_{17} = 3$

C_{17}	Завдання
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.

2. Код програми

Файл Main.java:

```
public class Main {  
    public static void main(String[] args) {  
        //ініціалізація  
        Vegetable tomato = new Vegetable("Помідор", 18);  
        Vegetable cucumber = new Vegetable("Огірок", 16);  
        Vegetable onion = new Vegetable("Цибуля", 40);  
        Vegetable cabbage = new Vegetable("Капуста", 25);  
        Vegetable carrot = new Vegetable("Морква", 41);  
        Vegetable potato = new Vegetable("Картопля", 77);  
  
        Salad salad = new Salad(tomato, tomato, cucumber, onion,  
                                cabbage, carrot, potato);  
        Salad_methods operations = new Salad_methods();  
  
        //операції над салатом  
        salad.print();  
        operations.by_nutrition_sort(salad);  
        operations.total_nutrition(salad);  
        operations.find_by_nutrition(salad,20,45);  
    }  
}
```

Файл Salad.java:

```
public class Salad {  
    Vegetable[] salad_elements;
```

```
public Salad(Vegetable... vegetable) {  
    salad_elements = vegetable;  
}
```

```
public void print() {  
    System.out.println("Салат: ");  
    System.out.printf("%s %41s", "Овоч:", "Калорійність овоча:\n");  
    for (Vegetable i: salad_elements) {  
        i.print();  
    }  
}  
}
```

Файл Vegetable.java:

```
public class Vegetable extends Salad {  
    public String name = "";  
    public int nutrition = 0;  
  
    public Vegetable(String name, int calories_value) {  
        try {  
            this.name = name;  
            this.nutrition = calories_value;  
        }  
        catch (Exception exc) {  
            System.out.println("Ви ввели неправильні дані для описання овоча");  
        }  
    }  
  
    @Override  
    public void print() {  
        System.out.printf("%10s; %20d кал\n", name, nutrition);  
    }  
  
    public void find_by_nutrition() {  
  
    }  
}
```

Файл Salad_methods.java

```

public class Salad_methods extends Salad {

    public void find_by_nutrition(Salad salad, int min_nutr, int max_nutr) {
        Vegetable []salad_elements = salad.salad_elements;
        /*Скільки елементів знайду*/
        int return_numb = 0;
        for (Vegetable i : salad_elements) {
            if (i.nutrition >= min_nutr && i.nutrition <= max_nutr) {
                return_numb++;
            }
        }
        /*Знаходжу овочі та додаю в масив*/
        Vegetable[] found_vegetables = new Vegetable[return_numb];
        int k = 0;
        for (Vegetable i : salad_elements) {
            if (i.nutrition >= min_nutr && i.nutrition <= max_nutr) {
                found_vegetables[k] = i;
                k++;
            }
        }
        System.out.println("Знайдено овочі за калорійністю в діапазоні від " + min_nutr + " до " + max_nutr + "\n");
        new Salad(found_vegetables).print();
    }
}

```

```

public void by_nutrition_sort(Salad salad) {
    /*Виконаю сортування вибором для калорійності овочів салату*/
    Vegetable []salad_elements = salad.salad_elements;
    int counter = 0;
    int min_index;
    for (int i = 0; i < salad_elements.length; i++) {
        min_index=counter;
        for (int j = counter; j < salad_elements.length; j++) {
            if (salad_elements[j].nutrition < salad_elements[min_index].nutrition) {
                min_index = j;
            }
        }
        Vegetable temporary = salad_elements[counter];
        salad_elements[counter] = salad_elements[min_index];
        salad_elements[min_index] = temporary;
        counter++;
    }
}

```

```

    }
    System.out.println("\nВідсортований салат: ");
    new Salad(salad_elements).print();
}

```

```

public void total_nutrition(Salad salad) {
    Vegetable []salad_elements = salad.salad_elements;
    int total_n = 0;
    for (Vegetable i: salad_elements) {
        total_n += i.nutrition;
    }
    System.out.println("Загальна калорійність" + total_n);
}
}

```

3. UML-діаграма класів

