Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Кафедра МКС



**Звіт**

З лабораторної роботи №7

З дисципліни: «Кросплатформні засоби програмування»

На тему:

«Параметризоване програмування»

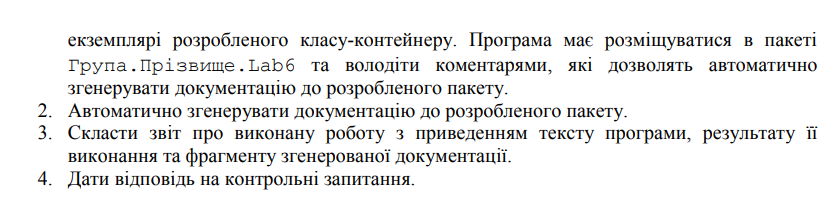
Виконав: ст.гр. КІ-35

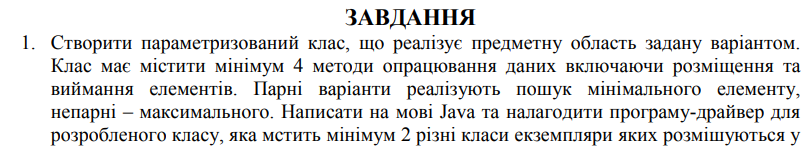
Куденчук Владислав

Прийняв:

Іванов Ю. С.

**Львів 2022**

Мета: оволодіти навиками параметризованого програмування мовою Java.

Виконання:

Код:

Class BasketDriver

package KI\_35.Kudenchuk.Lab7;  
  
public class BasketDriver {  
 public static void main(String[] args)  
 {  
 Basket <? super Fruit> basket = new Basket<>();  
 basket.AddData(new Pears("Green", 9));  
 basket.AddData(new Apples("Red" , 10));  
 basket.AddData(new Apples("Green" , 8));  
 basket.AddData(new Pears("Yellow" ,10));  
 Fruit res = basket.find\_by\_color("Red");  
 System.*out*.println();  
 System.*out*.print("The amount of red objects: \n");  
 System.*out*.print(res.getAmount());  
// res.print();  
 }  
}

Class Basket

package KI\_35.Kudenchuk.Lab7;  
  
import java.util.ArrayList;  
  
class Basket <T extends Fruit>  
{  
 private final ArrayList<T> arr;  
 public Basket()  
 {  
 arr = new ArrayList<>();  
 }  
 public T findMax()  
 {  
 if (!arr.isEmpty())  
 {  
 T max = arr.get(0);  
 for (int i=1; i< arr.size(); i++)  
 {  
 if ( arr.get(i).compareTo(max) > 0 )  
 max = arr.get(i);  
 }  
 return max;  
 }  
 return null;  
 }  
  
 public T find\_by\_color(String color)  
 {  
 if (!arr.isEmpty())  
 {  
 T fruit;  
 for (int i = 0; i< arr.size(); i++)  
 {  
 fruit = arr.get(i);  
 if (fruit.getColor().equals(color))  
 {  
 return fruit;  
 }  
 }  
 }  
 return null;  
 }  
 public void AddData(T fruit)  
 {  
 arr.add(fruit);  
 System.*out*.print("Fruit added: ");  
 fruit.print();  
 }  
 public void DeleteData(int i)  
 {  
 arr.remove(i);  
 }  
}

Class Apples

package KI\_35.Kudenchuk.Lab7;  
  
class Apples implements Fruit  
{  
 private String appleColor;  
 private int appleAmount;  
  
 public Apples(String pColor, int pAmount)  
 {  
 appleColor = pColor;  
 appleAmount = pAmount;  
 }  
  
 public String getColor()  
 {  
 return appleColor;  
 }  
  
 public int getAmount()  
 {  
 return appleAmount;  
 }  
  
 public void setAppleColor(String color)  
 {  
 appleColor = color;  
 }  
  
 public void SetAmount(int n)  
 {  
 appleAmount = n;  
 }  
  
 public int compareTo(Fruit p)  
 {  
 Integer s = appleAmount;  
 return s.compareTo(p.getAmount());  
 }  
 public void print()  
 {  
 System.*out*.print("Apples has color: " + appleColor + ", Amount of apples: " + appleAmount + ";\n");  
 }  
}

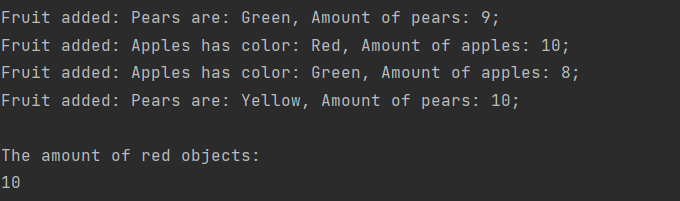
Class Pears

package KI\_35.Kudenchuk.Lab7;  
  
  
class Pears implements Fruit  
{  
 private String pearsColor;  
 private final int pearsAmount;  
  
  
 public Pears(String pColor, int pAmount)  
 {  
 pearsColor = pColor;  
 pearsAmount = pAmount;  
 }  
  
 public String getColor()  
 {  
 return pearsColor;  
 }  
  
 public void setPearsColor(String color)  
 {  
 pearsColor = color;  
 }  
  
 public int getAmount()  
 {  
 return pearsAmount;  
 }  
  
 public int compareTo(Fruit p)  
 {  
 Integer s = pearsAmount;  
 return s.compareTo(p.getAmount());  
 }  
  
 public void print()  
 {  
 System.*out*.print("Pears are: " + pearsColor + ", Amount of pears: " + pearsAmount + ";\n");  
 }  
}

Interface Fruit

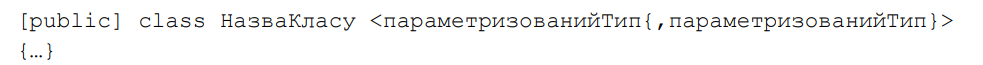
package KI\_35.Kudenchuk.Lab7;  
  
public interface Fruit extends Comparable<Fruit>  
{  
 int getAmount();  
 String getColor();  
 void print();  
}

Console:

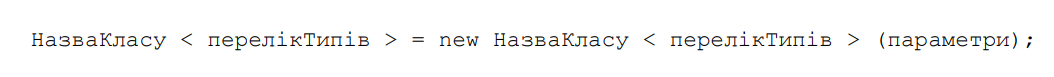


**Відповіді на контрольні запитання**

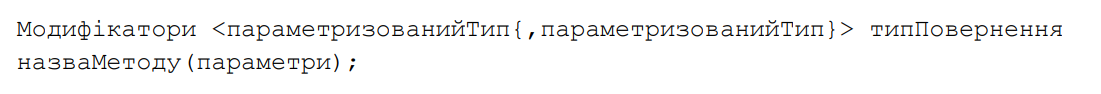
1. **Розкрийте синтаксис визначення простого параметризованого класу.**



1. **Розкрийте синтаксис створення об’єкту параметризованого класу.**



1. **Розкрийте синтаксис визначення параметризованого методу.**



**Висновок:**  оволодів навиками параметризованого програмування мовою Java.