Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра «Електронних обчислювальних машин»



Звіт

з лабораторної роботи № 7

з дисципліни: «Кросплатформенні засоби програмування»

на тему: «ПАРАМЕТРИЗОВАНЕ ПРОГРАМУВАННЯ »

**Виконав:**

студент групи КІ-35

Ничай В.Б.

**Прийняв:**

доцент кафедри ЕОМ

Іванов Ю. С.

   
Львів – 2022

Мета: оволодіти навиками параметризованого програмування мовою Java.

Завдання

1. Створити параметризований клас, що реалізує предметну область задану варіантом. Клас має містити мінімум 4 методи опрацювання даних включаючи розміщення та виймання елементів. Парні варіанти реалізують пошук мінімального елементу, непарні – максимального. Написати на мові Java та налагодити програму-драйвер для розробленого класу, яка мстить мінімум 2 різні класи екземпляри яких розмішуються у 9 екземплярі розробленого класу-контейнеру. Програма має розміщуватися в пакеті Група.Прізвище.Lab6 та володіти коментарями, які дозволять автоматично згенерувати документацію до розробленого пакету.

2. Автоматично згенерувати документацію до розробленого пакету.

3. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації.

4. Дати відповідь на контрольні запитання.



Текст програми

Lab7.java

package Nychai.Lab7;  
*/\*\*  
 \* Class <code>Lab7</code> testdrive class  
 \** ***@author*** *Nychai Volodymyr KI-35  
 \** ***@version*** *1.0  
 \*/*public class Lab7 {  
 */\*\*  
 \** ***@param*** *args  
 \*/* public static void main(String [] args){  
 Car<? super Data > car = new Car <Data>();  
 car.AddData(new Coal(10, 89, "Donbas"));  
 car.AddData(new Wheat(40,80,"Summer"));  
 car.AddData(new Wheat(20,40,"Winter"));  
 car.AddData(new Coal(77, 56, "Donbas"));  
 car.AddData(new Coal(74, 8, "Donbas"));  
 car.AddData(new Coal(40, 6, "Chervonograd"));  
  
  
 Data res = car.findMin();  
 System.*out*.print("The greatest price is: \n");  
 res.print();  
 }  
}

Car.java

package Nychai.Lab7;  
  
import java.util.ArrayList;  
  
*/\*\*  
 \* Class <code>Car</code> class describes work and actions with car  
 \** ***@author*** *Nychai Volodymyr KI-35  
 \** ***@version*** *1.0  
 \** ***@param*** <*T*> *everyone data  
 \*/*class Car<T extends Data> {  
 private ArrayList<T> arr;  
 public Car(){  
 arr = new ArrayList<T>();  
 }  
  
 */\*\*  
 \* method find min in arraylist  
 \** ***@return*** *\*/* public T findMin()  
 {  
 if (!arr.isEmpty())  
 {  
 T min = arr.get(0);  
 for (int i=1; i< arr.size(); i++)  
 {  
 if ( arr.get(i).compareTo(min) < 0 )  
 min = arr.get(i);  
 }  
 return min;  
 }  
 return null;  
 }  
  
 */\*\*  
 \* method add elements  
 \** ***@param*** *data  
 \*/* public void AddData(T data)  
 {  
 arr.add(data);  
 System.*out*.print("Element added: ");  
 data.print();  
 }  
  
 */\*\*  
 \* method delete elements  
 \** ***@param*** *i  
 \*/* public void DeleteData(int i)  
 {  
 arr.remove(i);  
 }  
  
}

Coal.java

package Nychai.Lab7;  
*/\*\*  
 \* Class <code>Coal</code> class describes work and actions with coal  
 \** ***@author*** *Nychai Volodymyr KI-35  
 \** ***@version*** *1.0  
 \*/*public class Coal implements Data{  
 private int price;  
 private int weight;  
 private String placeOfMining;  
  
 */\*\*  
 \** ***@param*** *price price of coal  
 \** ***@param*** *weight weight of coal  
 \** ***@param*** *placeOfMining place of mining  
 \*/* Coal(int price,int weight, String placeOfMining){  
 this.price=price;  
 this.weight=weight;  
 this.placeOfMining=placeOfMining;  
 }  
  
 */\*\*  
 \** ***@return*** *weight  
 \*/* public int getWeight() {return weight;}  
  
 */\*\*  
 \* method sets weight  
 \** ***@param*** *weight  
 \*/* public void setWeight(int weight) {this.weight = weight;}  
  
 */\*\*  
 \** ***@return*** *place  
 \*/* public String getPlaceOfMining() {return placeOfMining;}  
  
 */\*\*  
 \* method sets place of mining  
 \** ***@param*** *placeOfMining  
 \*/* public void setPlaceOfMining(String placeOfMining) {this.placeOfMining = placeOfMining;}  
  
 */\*\*  
 \** ***@return*** *price  
 \*/* public int getPrice() {return price;}  
 */\*\*  
 \* method doing some operation with weight  
 \** ***@param*** *p is data  
 \** ***@return*** *\*/* public int compareTo(Data p) {  
 Integer s = weight;  
 return s.compareTo(p.getWeight());  
 }  
 */\*\*  
 \* method print info  
 \*/* public void print(){  
 System.*out*.println("Weight is " + weight + "\tPrice is " + price + "\tPlace of mining "+ placeOfMining);  
 }  
}

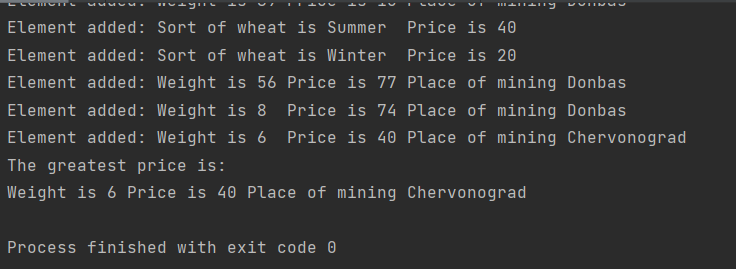
Wheat.java

package Nychai.Lab7;  
  
*/\*\*  
 \* Class <code>Wheat</code> class describes work and actions with wheat  
 \** ***@author*** *Nychai Volodymyr KI-35  
 \** ***@version*** *1.0  
 \*/*public class Wheat implements Data {  
 private String sortOfWheat;  
 private int price;  
 private int weight;  
  
 */\*\*  
 \** ***@param*** *price price of wheat  
 \** ***@param*** *weight weight of wheat  
 \** ***@param*** *sortOfWheat sort of wheat  
 \*/* Wheat(int price,int weight, String sortOfWheat){  
 this.price=price;  
 this.weight=weight;  
 this.sortOfWheat=sortOfWheat;  
 }  
  
 */\*\*  
 \** ***@return*** *sort  
 \*/* public String getSortOfWheat() {return sortOfWheat;}  
  
 */\*\*  
 \* method sets sort  
 \** ***@param*** *sortOfWheat  
 \*/* public void setSortOfWheat(String sortOfWheat) {this.sortOfWheat = sortOfWheat;}  
  
 */\*\*  
 \** ***@return*** *price  
 \*/* public int getPrice() {return 0;}  
  
 */\*\*  
 \* method sets price  
 \** ***@param*** *price  
 \*/* public void setPrice(int price) {this.price = price;}  
  
 */\*\*  
 \** ***@return*** *weight  
 \*/* public int getWeight() {return weight;}  
  
 */\*\*  
 \* method sets weight  
 \** ***@param*** *weight  
 \*/* public void setWeight(int weight) {this.weight = weight;}  
  
 */\*\*  
 \* method doing some operation with weight  
 \** ***@param*** *p is data  
 \** ***@return*** *\*/* public int compareTo(Data p) {  
 Integer s = weight;  
 return s.compareTo(p.getWeight());  
 }  
  
 */\*\*  
 \* method print info  
 \*/* public void print(){  
 System.*out*.println("Sort of wheat is "+ sortOfWheat+"\tPrice is "+price);  
 }  
}

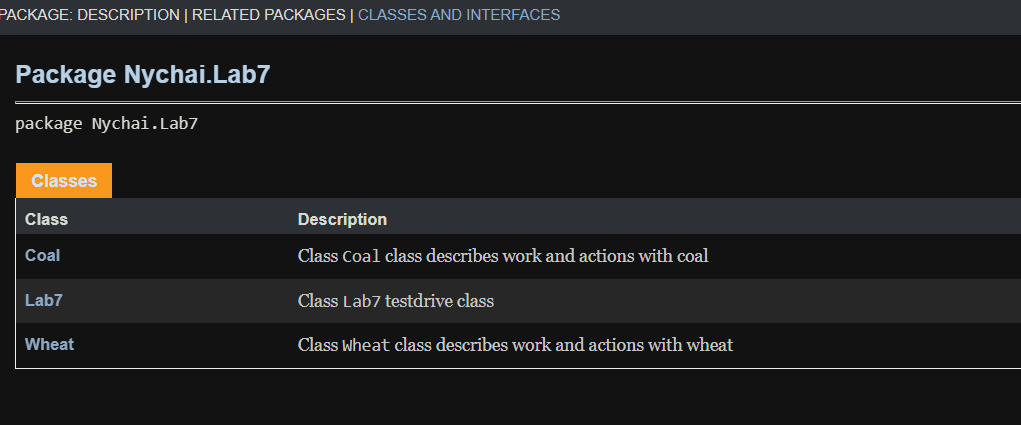
Data.java

package Nychai.Lab7;  
  
*/\*\*  
 \* Interface <code>Wheat</code> interface implemets some method in classes Coal and Wheat  
 \** ***@author*** *Nychai Volodymyr KI-35  
 \** ***@version*** *1.0  
 \*/*interface Data extends Comparable<Data>  
{  
 public int getWeight();  
 public void print();  
}

Результат роботи програми



**Фрагмент згенерованої документації**



Контрольні питання

1. Дайте визначення терміну «параметризоване програмування».

Параметризоване програмування є аналогом шаблонів у С++. Воно полягає у написанні коду, що можна багаторазово застосовувати з об’єктами різних класів

1. Розкрийте синтаксис визначення простого параметризованого класу.

Параметризований клас – це клас з однією або більше змінними типу. Синтаксис оголошення параметризованого класу:

[public] class НазваКласу {…}

Висновок: на даній лабораторній роботі я ознайомився з поняттям параметризованого програмування. Дізнався його основні функції та сфери застосування. Практично оволодів навичками цїєї сфери. Написав та протестував програму з використанням параметризованого програмування.