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»

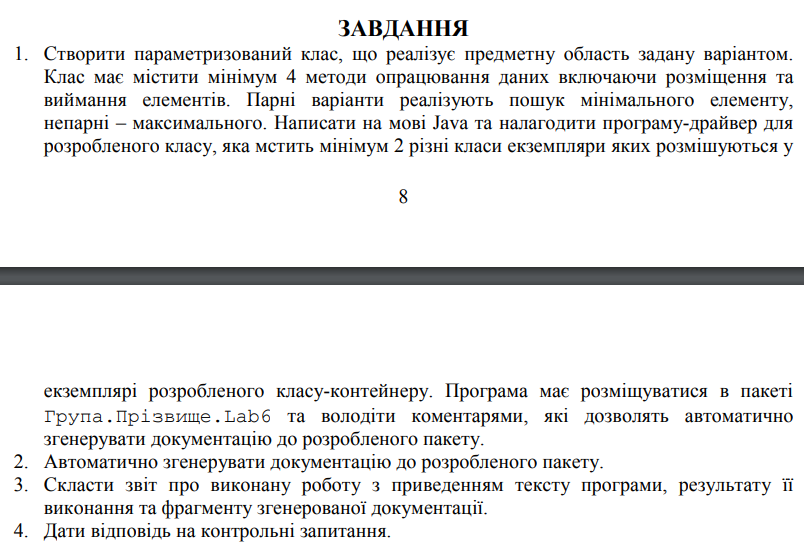
Виконав: ст.гр. КІ-35

Качур В.В.

Прийняв:

Іванов Ю. С.

**Львів 2022**

Мета: оволодіти навиками параметризованого програмування мовою Java. 



Виконання:

Код:

**Class Main**

**package** KI\_35\_Kachur.lab7;

/\*\*

\* **@author** Kachur Vladyslav

\* **@version** 1.0

\*/

**public** **class** main {

/\*\*

\* Methods using

\* **@param** args string arguments

\*/

**public** **static** **void** main(String[] args) {

Vagon <?**super** Instrument> box = **new** Vagon<>();

box.addInstrument(**new** SpareWheel(50,15,"Bosch"));

box.addInstrument(**new** SpareWheel(100,12,"Ukrzaliznytsia"));

box.addInstrument(**new** ConnectingRod(250,1,"Bosch"));

box.addInstrument(**new** ConnectingRod(400,0.5,"Ukrzaliznytsia"));

Instrument max = box.findMax();

max.output();

box.printAll();

}

}

**Interface Instrument**

**package** KI\_35\_Kachur.lab7;

/\*\*

\* **@author** Kachur Vladyslav

\* **@version** 1.0

\*/

**public** **interface** Instrument **extends** Comparable <Instrument>{

/\*\*

\* Getter price

\* **@return** double

\*/

**double** getPrice();

**void** output();

}

**Class Vagon**

**package** KI\_35\_Kachur.lab7;

**import** java.util.ArrayList;

/\*\*

\* **@author** Kachur Vladyslav

\* **@version** 1.0

\* **@param** <T>

\*/

**public** **class** Vagon <T **extends** Instrument>{

**private** **final** ArrayList<T> arr;

/\*\*

\* Constructor

\*/

**public** Vagon() {

arr = **new** ArrayList<T>();

}

/\*\*

\* Method to find max element of an array

\* **@return** T

\*/

**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**;

}

/\*\*

\* Method to add an instrument to box

\* **@param** instrument instrument

\*/

**public** **void** addInstrument (T instrument){

arr.add(instrument);

}

/\*\*

\* Method to remove an instrument from box

\* **@param** instrument instrument

\*/

**public** **void** removeInstrument (T instrument){

arr.remove(instrument);

}

/\*\*

\* Print method

\*/

**public** **void** printAll(){

**for**(T instrument : arr){

instrument.output();

}

}

}

**Class SpareWheel**

**package** KI\_35\_Kachur.lab7;

/\*\*

\* **@author** Kachur Vladyslav

\* **@version** 1.0

\*/

**public** **class** SpareWheel **implements** Instrument{

**double** price;

**double** radius;

String firm;

/\*\*

\* Constructor

\* **@param** price price

\* **@param** radius radius

\* **@param** firm firm

\*/

**public** SpareWheel(**double** price, **double** radius, String firm) {

**this**.price = price;

**this**.radius = radius;

**this**.firm = firm;

}

/\*\*

\* Setter price

\* **@param** price price

\*/

**public** **void** setPrice(**double** price) {

**this**.price = price;

}

/\*\*

\* Getter radius

\* **@return** double

\*/

**public** **double** getRadius() {

**return** radius;

}

/\*\*

\* Setter radius

\* **@param** radius radius

\*/

**public** **void** setRadius(**double** radius) {

**this**.radius = radius;

}

/\*\*

\* Getter Firm

\* **@return** String

\*/

**public** String getFirm() {

**return** firm;

}

/\*\*

\* Setter firm

\* **@param** firm firm

\*/

**public** **void** setFirm(String firm) {

**this**.firm = firm;

}

/\*\*

\* Getter price

\* **@return** double

\*/

@Override

**public** **double** getPrice() {

**return** price;

}

/\*\*

\*

\* **@return** details about object as a string

\*/

@Override

**public** String toString() {

**return** "SpareWheel{" + "price=" + price + ", radius=" + radius + ", firm='" + firm + '\'' + '}';

}

/\*\*

\* **@return** details about object as a string

\*/

@Override

**public** **void** output() {

System.***out***.println( "ConnectingRod{" + "price=" + price + ", radius=" + radius + ", firm='" + firm + '\'' + '}');

}

/\*\*

\* Compare method

\* **@param** o the object to be compared.

\* **@return** comparing value

\*/

@Override

**public** **int** compareTo(Instrument o) {

Double pr = price;

**return** pr.compareTo(o.getPrice());

}

}

**Class ConnectingRod**

**package** KI\_35\_Kachur.lab7;

/\*\*

\* **@author** Kachur Vladyslav

\* **@version** 1.0

\*/

**public** **class** ConnectingRod **implements** Instrument{

**double** price;

**double** size;

String firm;

/\*\*

\* Constructor

\* **@param** price price

\* **@param** size size

\* **@param** firm firm

\*/

**public** ConnectingRod(**double** price, **double** size, String firm) {

**this**.price = price;

**this**.size = size;

**this**.firm = firm;

}

/\*\*

\* Setter price

\* **@param** price price

\*/

**public** **void** setPrice(**double** price) {

**this**.price = price;

}

/\*\*

\* Getter Size

\* **@return** double

\*/

**public** **double** getSize() {

**return** size;

}

/\*\*

\* Setter size

\* **@param** size size

\*/

**public** **void** setSize(**double** size) {

**this**.size = size;

}

/\*\*

\* Getter Firm

\* **@return** String

\*/

**public** String getFirm() {

**return** firm;

}

/\*\*

\* Setter firm

\* **@param** firm firm

\*/

**public** **void** setFirm(String firm) {

**this**.firm = firm;

}

/\*\*

\* Getter price

\* **@return** double

\*/

@Override

**public** **double** getPrice() {

**return** price;

}

/\*\*

\*

\* **@return** details about object as a string

\*/

@Override

**public** **void** output() {

System.***out***.println( "ConnectingRod{" + "price=" + price + ", size=" + size + ", firm='" + firm + '\'' + '}');

}

/\*\*

\* **@return** details about object as a string

\*/

@Override

**public** **int** compareTo(Instrument o) {

Double pr = price;

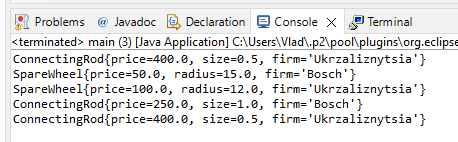
**return** pr.compareTo(o.getPrice());

}

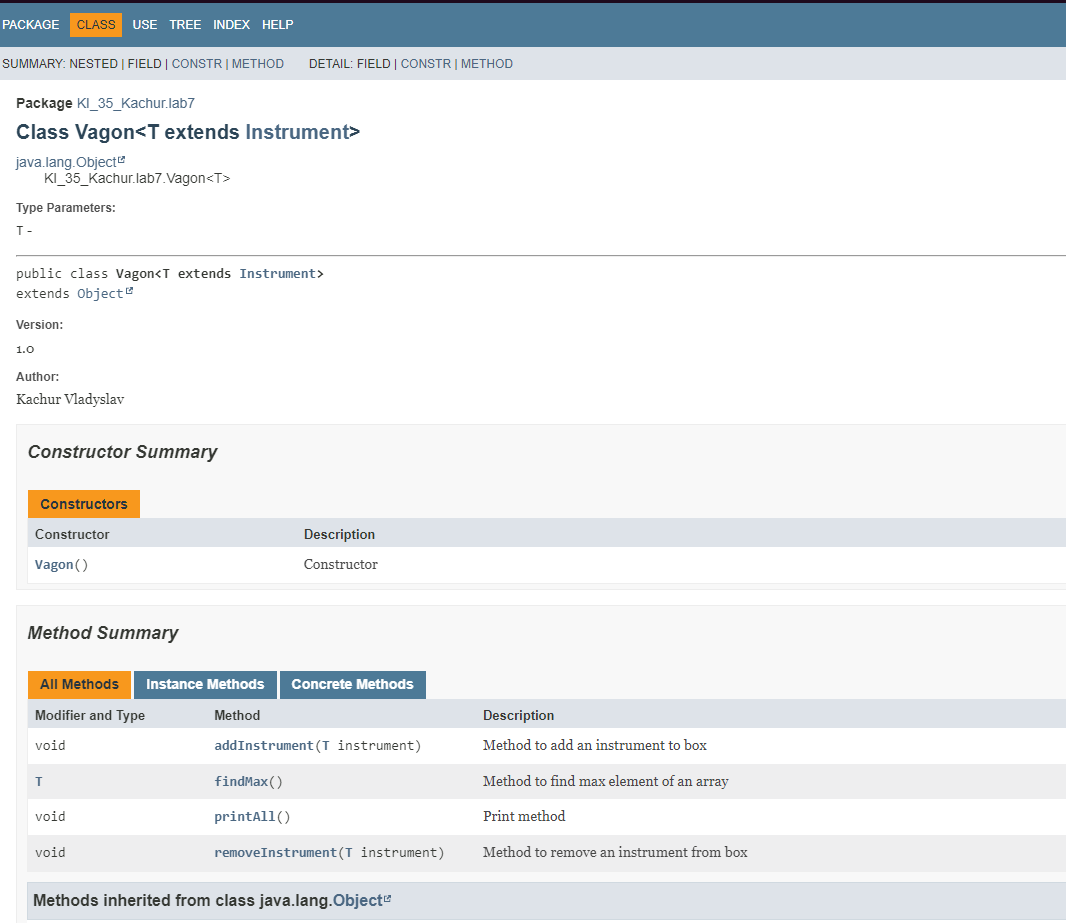
}

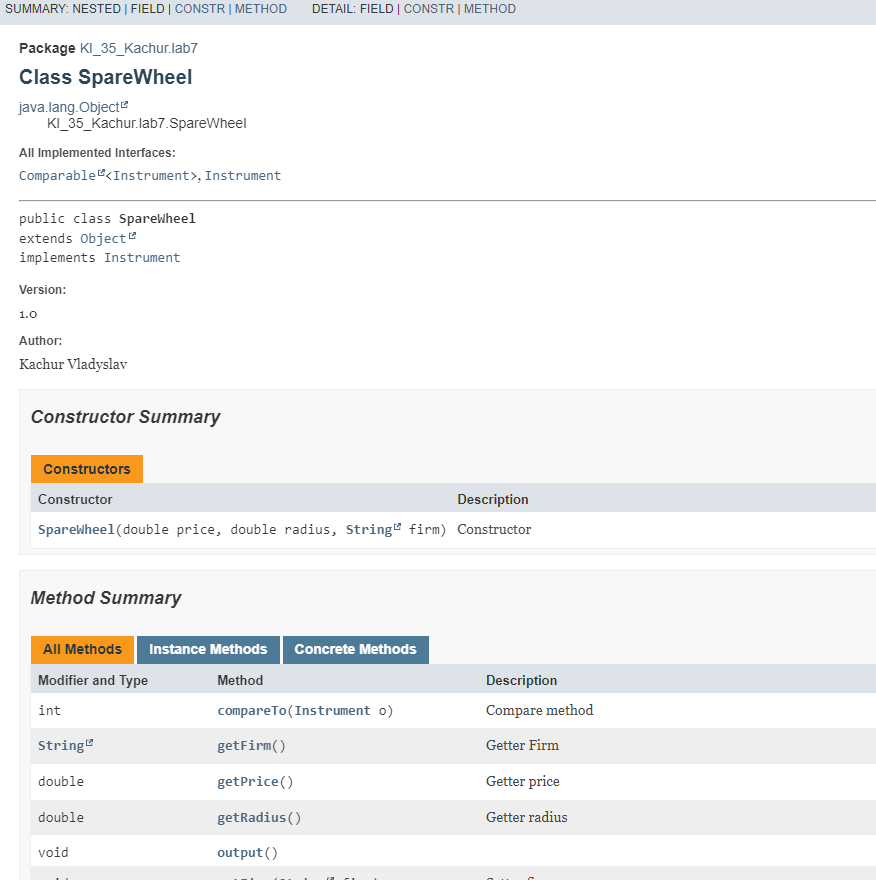
Результати:

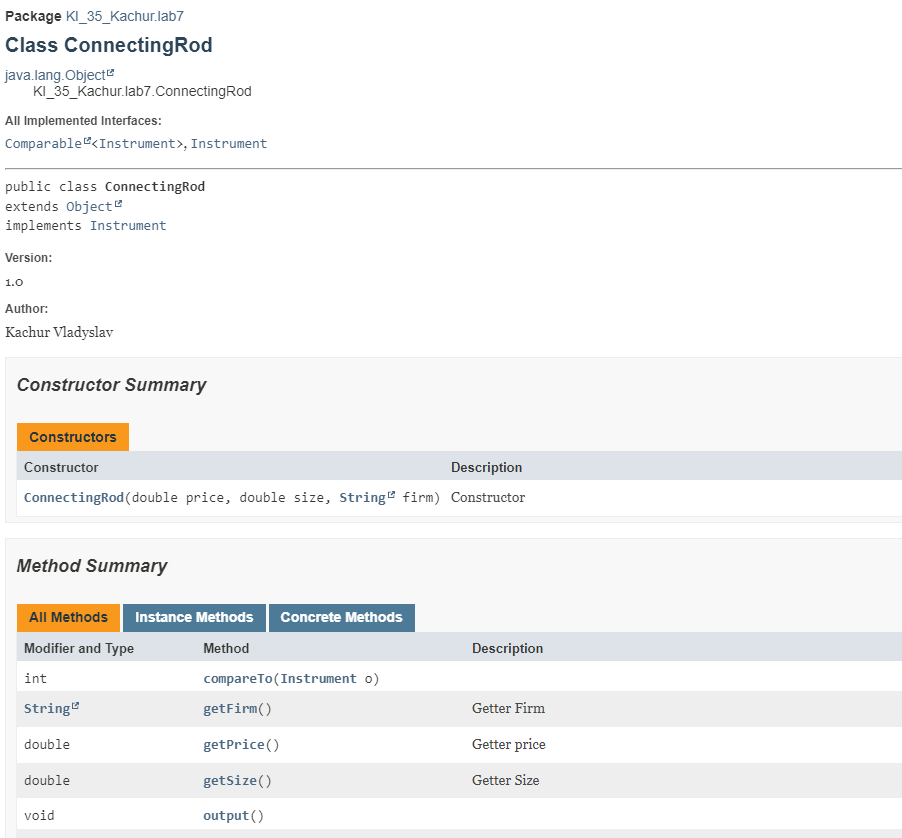
Console:



JAVADOC







Висновок: оволодів навиками параметризованого програмування мовою Java.