

**АМІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**



ЗВІТ

До лабораторної роботи №7

З дисципліни: «Кросплатформні засоби програмування»

На тему: «ПАРАМЕТРИЗОВАНЕ ПРОГРАМУВАННЯ»

Варіант 2

Виконав:
ст. гр. КІ-34
Галій О.М.
Перевірів:
доцент
Іванов Ю.С.

Львів 2022

Мета роботи: оволодіти навиками параметризованого програмування мовою Java.

ТЕОРЕТИЧНІ ВІДОМОСТІ

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

Користувачів параметризованого програмування можна поділити на 3 рівні кваліфікації:

1. ті, що користуються готовими класами;
2. ті, що користуються готовими класами і вміють виправляти помилки, що виникають при їх використанні;
3. ті, що пишуть власні параметризовані класи.

Для успішного застосування параметризованого програмування слід навчитися розуміти помилки, що генерує середовище при компіляції програми, що можуть стосуватися, наприклад, неоднозначності визначення спільного суперкласу для всіх переданих об'єктів. З іншої сторони необхідно передбачити захист від присвоєння об'єктів параметризованого класу, що містять об'єкти підкласу об'єктам параметризованого класу, що містять об'єкти суперкласу і дозволити зворотні присвоєння. Для вирішення цієї проблеми у мові Java введено так звані підстановочні типи. Це далеко не всі «підводні камені», що виникають при застосуванні параметризованого програмування.

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 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.

2. Однозв'язний список

Код програми:

```
package lab7;

import java.util.ArrayList;

public class LinkedList {
    public static void main(String[] args) {

        LinkedList <? super Data>newList = new LinkedList<Data>();

        newList.insertFirst(13, 50);
        newList.insertFirst(40, 345);
        newList.insertFirst(15, 25);
        newList.insertFirst(5, 102);
        newList.insertFirst(54, 12);
        newList.insertFirst(10, 30);

        newList.displayList();

        Link f = newList.find(5);
        System.out.print("The smallest data on LinkedList is: \n");

        f.print();

        System.out.print("Sum = " + newList.sum + "\n");

    }
}

interface Data extends Comparable<Data> {
    public int getKey();
    public void print();
}

class Link implements Data{
    public int iData; // інформація
    public int dData; // інформація
    public Link next; // посилання на наступний елемент списку

    public Link(int iData, int dData) {
        this.iData = iData;
        this.dData = dData;
    }

    public void displayLink() {
        System.out.print(" {" + iData + ", " + dData + "}");
    }
}
```

```

public int compareTo(Data o) {
    Integer s = iData;
    return s.compareTo(o.getKey());
}

```

```

public int getKey() {

    int sum = 0;
    if(dData%2 ==0) {
        //System.out.print("Info: " + dData + "\n");
        sum++;
    }
    System.out.print(sum + "\n");
    return dData;
}

```

```

public void print() {

    System.out.print("Key: " + iData + "\n");

}

```

```

}

```

```

class LinkedList <T extends Data> {
    private Link first;
    private ArrayList<T> arr;
    public static int sum = 0;

    public LinkedList(){
        arr = new ArrayList<T>();
    }

```

```

    public void insertFirst(int iD, int dD) {

```

```

        Link newLink = new Link(iD, dD); //створюємо новий елемент списку
        newLink.next = first;           //вказали на старий "перший" елемент і таким чином зробили
його другим
        first = newLink;                // позначили створений елемент, як перший
        if(dD %2 == 0){
            sum ++;
        }
    }
}

```

```

public Link find(int key) { // пошук елемента с заданием ключем
    Link current = first; // починаємо пошук з першого елемента
    while (current.iData != key) { // якщо, ключ не знайдено,то..

```

```

        if (current.next == null) { //якщо ключа немає в списку взагалі, повертаємо null
            return null;
        } else {
            current = current.next; //то переходимо до наступного елемента і шукаємо ключ
        }
    }
    return current;
}

```

```

public Link delete(int key) { //видалення елемента по заданому ключу
    Link current = first;
    Link previous = first;

    while (current.next != null) { // пошук елемента
        if (current.next == null) {
            return null; // елемент не знайдено
        } else {
            previous = current; // перейти до наступного елемента
            current = current.next;
        }
        // ключ знайдено
        if (current == first) {
            first = first.next; // якщо шуканий елемент - перший
        } else {
            previous.next = current.next; // якщо шуканий елемент лежить всередині списку,
обійдемо його
        }
    }
    return current;
}

public void displayList() {
    Link current = first;
    System.out.print("List (first --> last): ");

    while(current != null) { //доки список не закінчився, відображаємо дані
        current.displayLink(); // відображаємо дані елемента, на якому знаходимося
        current = current.next; // переходимо до наступного елемента
    }
    System.out.println("");
}
}

```

Результат виконання програми:

```
<terminated> LinkedList [Java Application] C:\Program Files\Java\jdk1.6.0_45\bin\javaw.exe (9 лист. 2022 р., 17:59:41 – 17:59:42) [pid: 17156]  
List (first --> last): {10, 30} {54, 12} {5, 102} {15, 25} {40, 345} {13, 50}  
The smallest data on LinkedList is:  
Key: 5  
Sum = 4
```

Висновок: під час лабораторної роботи я оволодів навиками параметризованого програмування мовою Java.