МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”
Кафедра систем штучного інтелекту

Лабораторна робота № 11

з дисципліни
«Об’єктно-орієнтоване програмування»

Виконала:
студентка групи КН-109
Пелещак Ю. М.
Викладач:
Гасько Р.Т.

Львів – 2018 р.

Лабораторна робота №11.
Параметризація в Java. Обробка параметризованих контейнерів

Мета

● Вивчення принципів параметризації в Java.
● Розробка параметризованих класів та методів.
● Розширення функціональності параметризованих класів.

Вимоги:

1. Створити власний клас-контейнер, що параметризується (Generic Type), (docs.oracle.com/javase/tutorial/java/generics/types.html) на основі зв'язних списків для реалізації колекції domain-об'єктів з лабораторної роботи №10 (Прикладні задачі. Список №2. 20 варіантів)

2. Для розроблених класів-контейнерів забезпечити можливість використання їх об'єктів у циклі foreach в якості джерела даних.

3. Забезпечити можливість збереження та відновлення колекції об'єктів:
1) за допомогою стандартної серіалізації;
2) не використовуючи протокол серіалізації.

4. Продемонструвати розроблену функціональність: створення контейнера, додавання елементів, видалення елементів, очищення контейнера, перетворення у масив, перетворення у рядок, перевірку на наявність елементів.

5. Забороняється використання контейнерів (колекцій) з Java Collections Framework - docs.oracle.com/javase/8/docs/technotes/guides/collections/

6. Розробити параметризовані методи (Generic Methods - docs.oracle.com/javase/tutorial/java/generics/methods.html) для обробки колекцій об'єктів згідно (Прикладні задачі. Список №2. 20 варіантів).

7. Продемонструвати розроблену функціональність (створення, управління та обробку власних контейнерів) в діалоговому та автоматичному режимах.
a. Автоматичний режим виконання програми задається параметром командного рядка -auto. Наприклад, java ClassName -auto.
b. В автоматичному режимі діалог з користувачем відсутній, необхідні данні генеруються, або зчитуються з файлу.

Частина програми,що відповідає за виконання поставленої задачі:

```
package ua.lpnuai.oop.peleshchak11;

public class LinkedList<T1, T2, T3, T4, T5> {

    public T1 getT1() {
        return t1;
    }

    public void setT1(T1 t1) {
        this.t1 = t1;
    }

    public T2 getT2() {
        return t2;
    }

    public void setT2(T2 t2) {
        this.t2 = t2;
    }

    public T3 getT3() {
        return t3;
    }

    public void setT3(T3 t3) {
        this.t3 = t3;
    }

    public T4 getT4() {
        return t4;
    }

    public void setT4(T4 t4) {
        this.t4 = t4;
    }

    public T5 getT5() {
        return t5;
    }

    public void setT5(T5 t5) {
        this.t5 = t5;
    }

    private T1 t1;
    private T2 t2;
    private T3 t3;
```

```
    private T4 t4;
    private T5 t5;
    Node head;

    public LinkedList() {

    }

    /* Додати елемент в кінець списку */
    public void add(LinkedList <T1, T2, T3, T4, T5> data) {

        if (head == null) {
            head = new Node();
            head.setData(data);
        } else {
            Node node = new Node();
            node.setData(data);
            Node temp = head;
            while (temp.getNext() != null) {
                temp = temp.getNext();
            }
            temp.setNext(node);
        }
    }

    /* Отримати елемент по індексу, повертає null якщо такий елемент
недоступний */
    public LinkedList<T1, T2, T3, T4, T5> get(int index) {
        if (index == 0) {
            return head.getData();
        } else if (index > 0) {
            if (head.getNext() != null) {
                Node temp = head;
                int position = 1;
                while (temp.getNext() != null) {
                    if (position == index) {
                        return temp.getNext().getData();
                    } else {
                        temp = temp.getNext();
                        position++;
                    }
                }
            }
        }
        return null;
    }
```

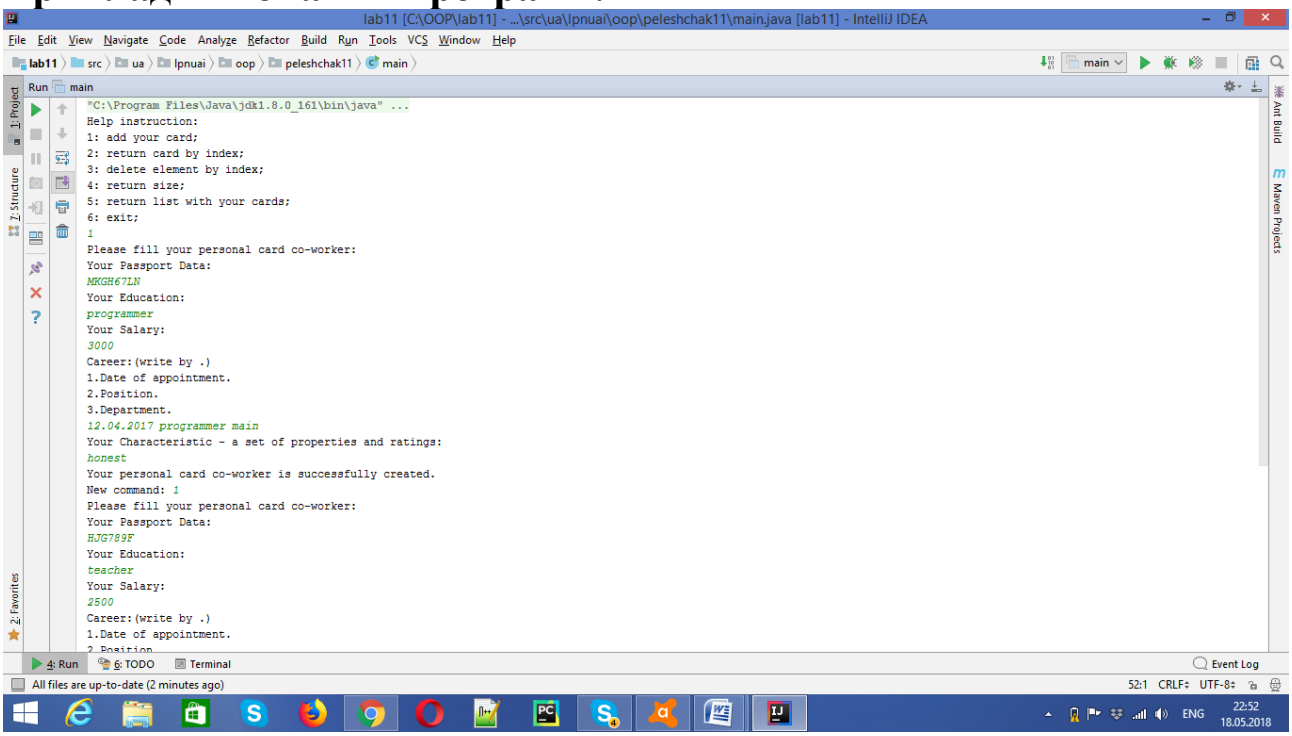
```
    /* Вилучення елементу за індексом, повертає true у разі успіху або false в іншому випадку */
    public boolean delete(int index) {
        if (index == 0) {
            if (head == null) {
                return false;
            } else {
                if (head.getNext() == null) {
                    head = null;
                    return true;
                } else {
                    head = head.getNext();
                    return true;
                }
            }
        }

        if (index > 0) {
            if (head.getNext() != null) {
                Node temp = head;
                int position = 1;
                while (temp.getNext() != null) {
                    if (position == index) {
                        temp.setNext(temp.getNext().getNext());
                        return true;
                    } else {
                        temp = temp.getNext();
                        position++;
                    }
                }
            }
        }
        return false;
    }

    /*Поверта розмір списку: якщо елементів в списку нема то повертає 0 (нуль)*/
    public int size() {
        int size = 0;
        if (head != null) {
            if (head.getNext() != null) {
                Node temp = head;
                size = 1;
                while (temp.getNext() != null) {
                    temp = temp.getNext();
                    size++;
                }
```

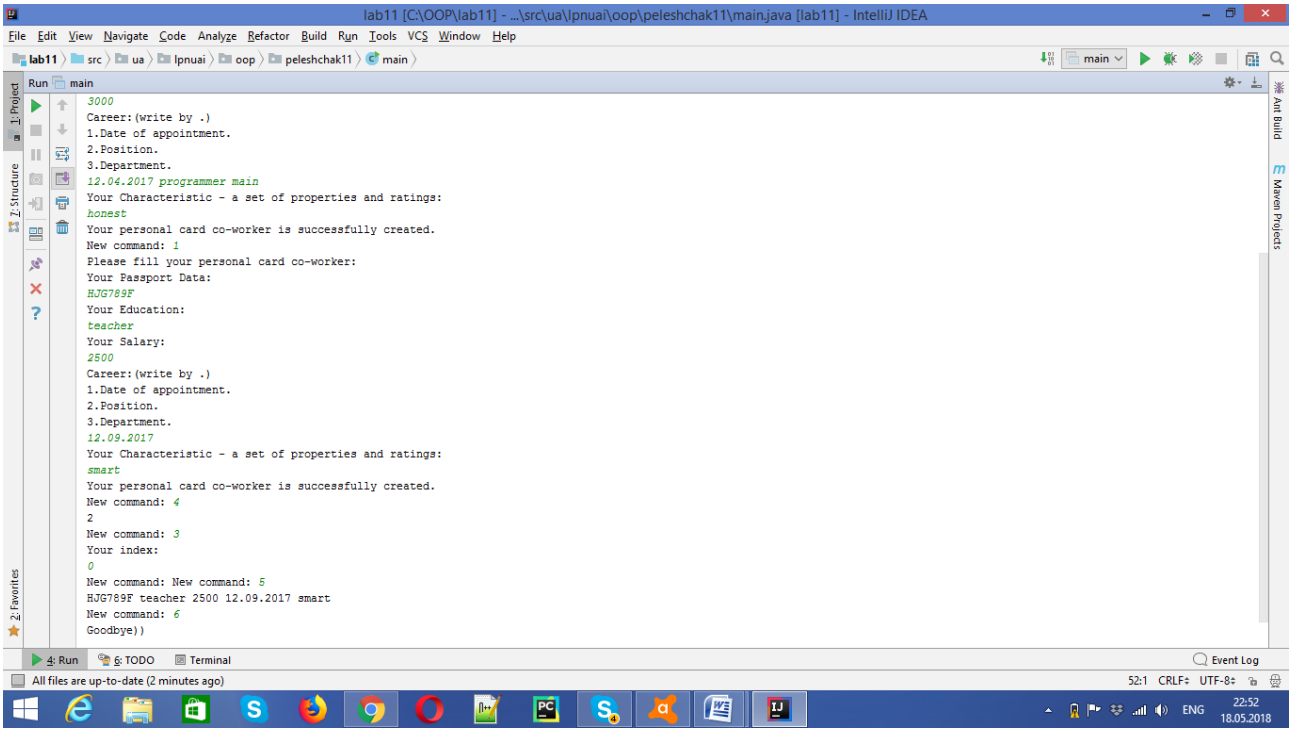
```
            } else {
                size = 1;
            }
        }
        return size;
    }

    @Override
    public String toString() {
        StringBuffer str = new StringBuffer();

        Node toGet = head;
        while (toGet != null){
            str.append(toGet.getData().getT1().toString()).append(" ");
            str.append(toGet.getData().getT2().toString()).append(" ");
            str.append(toGet.getData().getT3().toString()).append(" ");
            str.append(toGet.getData().getT4().toString()).append(" ");
            str.append(toGet.getData().getT5().toString()).append(" ");
            toGet = toGet.getNext();
        }

        return str.toString();
    }

    public class Node {
        private Node next;
        private LinkedList <T1, T2, T3, T4, T5> data;

        public Node() {
        }

        public Node getNext() {
            return next;
        }
        public void setNext(Node next) {
            this.next = next;
        }
        LinkedList <T1, T2, T3, T4, T5> getData() {
            return data;
        }
        public void setData(LinkedList <T1, T2, T3, T4, T5> data) {
            this.data = data;
        }

    }
}
```

```
package ua.lpnuai.oop.peleshchak11;

import java.io.FileWriter;
import java.util.Scanner;

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

      LinkedList LinkedList = new LinkedList();
      LinkedList LinkedList1 = new LinkedList();
      Scanner scan = new Scanner(System.in);
      String line = scan.nextLine();

      while(!("6".equals(line))) {
         //LinkedList LinkedList1 = new LinkedList();
         if ("1".equals(line)) {
            System.out.println("Please fill your personal card co-worker:");
            System.out.println("Your Passport Data: ");
            String data = new String();
            data = scan.nextLine();
            System.out.println("Your Education: ");
            String education = new String();
            education = scan.nextLine();
            System.out.println("Your Salary: ");
            String salary = new String();
            salary = scan.nextLine();
            System.out.println("Career:(write by .) ");
            System.out.println("1.Date of appointment. ");
            System.out.println("2.Position. ");
            System.out.println("3.Department. ");
            String career = new String();
            career = scan.nextLine();
            System.out.println("Your Characteristic - a set of properties and ratings: ");
            String characteristic = new String();
            characteristic = scan.nextLine();
            System.out.println("Your personal card co-worker is successfully created.");

            LinkedList.setT1(data);
            LinkedList.setT2(education);
            LinkedList.setT3(salary);
            LinkedList.setT4(career);
            LinkedList.setT5(characteristic);
            LinkedList1.add(LinkedList);


         }
         else if("2".equals(line)){
            System.out.println("Your index:");
            int b = scan.nextInt();
            System.out.println(LinkedList1.get(b));
         }

         else if("3".equals(line)){
```

```
            System.out.println("Your index:");
            int a = scan.nextInt();
            LinkedList1.delete(a);
        }
        else if("4".equals(line)){
            System.out.println(LinkedList1.size());

        }
        else if("5".equals(line)){
            System.out.println(LinkedList1.toString());
        }
        System.out.print("New command: ");
        line = scan.nextLine();
    }
    System.out.println("Goodbye))");



  }
  public static void show(){
    System.out.println("Help instruction:\n" +
            "1: add your card;\n" +
            "2: return card by index;\n" +
            "3: delete element by index;\n" +
            "4: return size;\n" +
            "5: return list with your cards;\n"+
            "6: exit;");
  }
}
```

Приклад виконання програми:





Висновок: під час виконання лабораторної роботи №11 я створила програму відповідно до поставленої задачі.