

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

Звіт

Лабораторна робота №7

З дисципліни:

Об’єктно орієнтоване програмування

Виконав

студент групи КН-111

Бойко Андрій

Викладач:

Грабовська Н. Р.

Тема роботи

Ознайомлення з бібліотекою колекцій Java SE

- Використання колекцій для розміщення об'єктів розроблених класів.

Вимоги

1.1 Розробник

Бойко Андрій Віталійович

КН-111

3 варіант

1.2 Загальне завдання

Ознайомлення з бібліотекою колекцій Java SE

- Використання колекцій для розміщення об'єктів розроблених класів.

Завдання

1. Розробити консольну програму для реалізації завдання обробки даних згідно Прикладні задачі. Список №2. 20 варіантів. 2. Для розміщення та обробки даних використовувати контейнери (колекції) і алгоритми з Java Collections Framework . 3. Забезпечити обробку колекції об'єктів: додавання, видалення, пошук, сортування згідно списку з Прикладні задачі. Список №2. 20 варіантів). 4. Передбачити можливість довготривалого зберігання даних: 1) за допомогою стандартної серіалізації; 2) не використовуючи протокол серіалізації. 5. Продемонструвати розроблену функціональність в діалоговому та автоматичному режимах за результатом обробки параметрів командного рядка

1.Опис програми

Main

```
package lab_7;
```

```
import java.beans.XMLDecoder;  
import java.beans.XMLEncoder;  
import java.io.BufferedInputStream;  
import java.io.BufferedOutputStream;  
import java.io.FileInputStream;  
import java.io.FileNotFoundException;  
import java.io.FileOutputStream;  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.Comparator;  
import java.util.Scanner;
```

```

class Menu
{
    int k;
    public int execute(Scanner in)
    {
        System.out.println("Choose what you want to do:");
        System.out.println("1. Create new person");
        System.out.println("2. Change name of the person");
        System.out.println("3. Change date of the person");
        System.out.println("4. Change numbers");
        System.out.println("5. Change adress");
        System.out.println("6. Change regdate");
        System.out.println("7. name");
        System.out.println("8. date");
        System.out.println("9. number");
        System.out.println("10. adress");
        System.out.println("11. regdate");
        System.out.println("12. Print to the file");
        System.out.println("13. Get from the file");
        System.out.println("14. Choose the person");
        System.out.println("15 . Finish the program");
        k = in.nextInt();

        return k;
    }
}

public class Main {

    public static void main(String[] args) throws FileNotFoundException {
        boolean d = false;
        if(args.length > 0)
            d = comLineCheck(args);

        ArrayList<AdressBook> list = new ArrayList<AdressBook>();
        ArrayList<Integer> temp=new ArrayList<>();
        ArrayList<Integer> temp1=new ArrayList<>();
        int user = 0;

        Scanner in = new Scanner(System.in);
        Scanner on = new Scanner(System.in);
        Menu main = new Menu();
        if(d) {
            list.add(new AdressBook());

            String name= "Harry";
            list.get(user).setName(name);

            String date="21.04.1954";
            list.get(user).setDate(date);
        }
    }
}

```

```

        ArrayList<Integer> temp2=new ArrayList<>();
        temp2.add(402415454);
        temp2.add(474446464);
        list.get(user).setNumbers(temp2);

        String adress="RedStreet";
        list.get(user).setAdress(adress);

        String regdate="20.02.2017";
        list.get(user).setRegdate(regdate);}

while(true)
{
int k = main.execute(in);

switch (k)
{
case 1 :
        list.add(new AdressBook());
        break;

case 2 :
        System.out.println("Enter the name : ");
        list.get(user).setName(on.nextLine());
        break;

case 3 :
        System.out.println("Enter the date : ");
        list.get(user).setDate(on.nextLine());
        break;

case 4 :
        System.out.println("Enter phone number and its place : ");
int t;
        while(true) {
            t=in.nextInt();
            if(t==0)
            {
                break;
            }
            else {
                temp.add(t);
            }
        }
        list.get(user).setNumbers(temp);

        break;

case 5 :
        System.out.println("Enter adress : ");

```

```

        list.get(user).setAdress(on.nextLine());
break;

case 6 :
    System.out.println("Enter regdate : ");
    list.get(user).setRegdate(on.nextLine());
break;

case 7:
    System.out.println(list.get(user).getname());
break;

case 8 :
    System.out.println(list.get(user).getDate());
break;

case 9 :
    temp1=list.get(user).getNumbers();
    System.out.println(temp1);

break;

case 10 :
    System.out.println(list.get(user).getAdress());
break;

case 11 :
    System.out.println(list.get(user).getRegdate());
break;
case 12:
    XMLEncoder encoder;
    try {
        encoder = new XMLEncoder(
            new BufferedOutputStream(
                new FileOutputStream(PathGetter.getPath(in).toString())));
        encoder.writeObject(list);
        encoder.close();
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }
break;
case 13 :
    XMLDecoder decoder = new XMLDecoder(
        new BufferedInputStream(
            new FileInputStream(PathGetter.getPath(in).toString())));

    list =(ArrayList<AdressBook>) decoder.readObject();
    decoder.close();

    break;

```

```

        case 14 :
            System.out.println("Choose the person from 0 to "+(list.size()-1));
            user = in.nextInt();
        break;
        case 15:
            System.out.println("Choose the person you want to delete from 0 to
"+(list.size()-1));
            list.remove(in.nextInt());
            break;

        case 17 :
            in.close();
            on.close();
            return;
        case 16:
            Comparator<AdressBook> comp = new compByName();
            list.sort(comp);
            System.out.println("Done!");

    }}}
    private static boolean comLineCheck(String[] args) {
        if(args[0].compareTo("-auto") == 0) return true;
        else return false;
    }
}

```

PathGetter

```

package lab_7;

import java.io.IOException;
import java.nio.file.*;
import java.util.ArrayList;
import java.util.Scanner;

public class PathGetter {

    public static Path getPath(Scanner input) {
        Path p = Paths.get(System.getProperty("user.home"));
        while (true) {
            System.out.println(p);
            try (DirectoryStream<Path> stream = Files.newDirectoryStream(p)) {
                int i = 1;
                ArrayList<String> foldersList = new ArrayList<>();
                for (Path file: stream) {
                    String check = file.getFileName().toString();
                    if (check.contains(".xml") || !check.contains(".")) {
                        System.out.printf("[%d] %s%n", i++, check);
                        foldersList.add(check);
                    }
                }
            }
        }
    }
}

```

```

        System.out.printf("~[%d] return%n", i++);
        System.out.printf("~[%d] select XML file or create a new one%n", i);
        System.out.println("chose option or directory: ");
        int option = input.nextInt();
        if (option < 1 || option > i)
            System.err.println("error");
        else if (option < i-1) {
            if (Files.isDirectory(p.resolve(foldersList.get(option-1))))
                p = p.resolve(foldersList.get(option-1));
            else System.err.println("it isn't a directory");
        } else if (option == i-1)
            p = p.getRoot().resolve(p.getParent());
        else {
            System.out.println("enter filename in format name.xml:");
            input.nextLine();
            String filename = input.nextLine();
            if (filename.contains(".xml")) {
                return p.resolve(filename);
            } else System.err.println("wrong file format");
        }
    } catch (IOException | DirectoryIteratorException x) {
        System.err.println("error!");
        return null;
    }
}
}
}
}
}

```

Comparator

```

package lab_7;

import java.util.Comparator;

public class Comparators { }

    class compByName implements Comparator<AdressBook>{
        public int compare(AdressBook a, AdressBook b){
            return a.getname().compareTo(b.getname());
        }
    }
}

```

Висновки

На цій лабораторній роботі, я ознайомився з бібліотекою колекцій java SE