

Лабораторна робота №8

Основи введення/виведення Java SE

Мета: оволодіння навичками управління введення/виведення даних з використанням класів платформи Java SE.

1 ВИМОГИ

- 1) Забезпечити можливість збереження і відновлення масива об'єктів рішення завдання лабораторної роботи №7.
- 2) Забороняється використання стандартного протокола серіалізації.
- 3) Продемонструвати використання моделі Long Term Persistence.
- 4) Забезпечити діалог з користувачем у вигляді простого текстового меню.
- 5) При збереженні та відновленні даних забезпечити діалоговий режим вибору директорії з відображенням вмісту і можливістю переміщення по підкаталогах.

1.1 Розробник

- Абдулаєв Ібрагім Заурбекович
- Група КІТ-119В
- Варіант 1

2 ОПИС ПРОГРАМИ

2.1 Опис класів

Modifier and Type	Method	Description
static void	<code>appMenu()</code>	Головне меню програми
static void	<code>clearConsole()</code>	Очищення вікна консолі
static java.util.ArrayList<java.io.File>	<code>listDirectoryFiles(java.io.File folder)</code>	Отримати список файлів у директорії
static void	<code>main(java.lang.String[] args)</code>	
static java.lang.String	<code>pickFileMenu()</code>	Рекурсивне меню директорії.

Рис. 2.1 – Опис класу **Main**

2.2 Тест програми

Main.java

```
package ua.khpi.oop.abdulaev08;

import ua.khpi.oop.abdulaev07.Challenger;
import ua.khpi.oop.abdulaev07.RecruitingAgency;

import java.io.File;
import java.util.*;

public class Main {

    /**
```

```

    * Очищення вікна консолі
    */
    public static void clearConsole() {
        System.out.println("\033[H\033[2J");
        System.out.flush();
    }

    public static void main(String[] args) {
        appMenu();
    }

    /**
     * Отримати список файлів у директорії
     * @param folder директорія
     * @return список файлів
     */
    public static ArrayList<File> listDirectoryFiles(final File
folder) {
        return new
ArrayList<>(Arrays.asList(folder.listFiles()));
    }

    /**
     * Рекурсивне меню директорії.
     * Якщо обрано файл який не є директорію, повертає
абсолютний шлях до цього файлу
     * @return шлях до обраного файлу
     */
    public static String pickFileMenu() {
        Scanner sc = new Scanner(System.in);
        File root = new File("c:");
        ArrayList<File> dir;
        int menuOption;

        while (true) {
            clearConsole();
            System.out.print("0. Назад\n-----\n");
            System.out.printf("Вміст директорії %s\n",
root.getPath());

            dir = listDirectoryFiles(root);

            for (int i = 0; i < dir.size(); i++) {
                System.out.printf("%d) %s\n", i + 1,
dir.get(i).getName());
            }

            System.out.print("----> ");

            menuOption = sc.nextInt();

            if (menuOption == 0) {
                if (root.getPath().equals("c:")) {

```

```

        break;
    } else {
        root = root.getParentFile();
        continue;
    }
}

if (menuOption > 0 && menuOption <= dir.size()) {
    File subFile = new File(root.getAbsolutePath(),
dir.get(menuOption - 1).getName());

    if (subFile.isDirectory()) {
        root = subFile;
        dir = listDirectoryFiles(root);
    } else {
        return subFile.getAbsolutePath();
    }
}

return null;
}

/**
 * Головне меню програми
 */
public static void appMenu() {
    RecruitingAgency agency = new RecruitingAgency();
    Scanner sc = new Scanner(System.in);
    Challenger challenger;
    int menuOption;
    String filePath;

    while (true) {
        clearConsole();
        System.out.print(
            "0. Вихід\n"
            + "1. Вивести список претендентів\n"
            + "2. Додати претендента\n"
            + "3. Видалити претендента\n"
            + "4. Показати претендента за
номером в списку\n"
            + "5. Зчитати список претендентів з
файлу\n"
            + "6. Записати список претендентів у
файл\n"
            + "-----\n"
            + "----> "
        );
        menuOption = sc.nextInt();

        switch (menuOption) {

```

```

        case 0: return;
        case 1:
            clearConsole();
            agency.print();
            sc.nextLine();
            sc.nextLine();
            break;
        case 2:
            sc.nextLine();
            System.out.print("Досвід: ");
            String[] _exp = sc.nextLine().split(";");
            System.out.print("Освіта: ");
            String _edu = sc.nextLine();
            System.out.print("Дата звільнення: ");
            String _dofd = sc.nextLine();
            System.out.print("Вимоги до роботи: ");
            String[] _rfw = sc.nextLine().split(";");

            challenger = new Challenger();

            Map<String, Integer> _exp_map = new
HashMap<>();
            ArrayList<ArrayList<String>> _req_for_work =
new ArrayList<>();

            for (String kv : _exp) {
                String[] kvArr = kv.split(":");
                _exp_map.put(kvArr[0],
Integer.valueOf(kvArr[1]));
            }

            for (String vls : _rfw) {
                _req_for_work.add(new
ArrayList<>(Arrays.asList(vls.split(","))));
            }

            String[][] req_arr = new
String[_req_for_work.size()][];

            for (int i = 0; i < _req_for_work.size();
i++) {
                ArrayList<String> row =
_req_for_work.get(i);
                req_arr[i] = row.toArray(new
String[row.size()]);
            }

            challenger.setExperience(_exp_map);
            challenger.setEducation(_edu);
            challenger.setDateOfDismissal(_dofd);
            challenger.setRequirementsForWork(req_arr);

            agency.add(challenger);

```

```

        break;
    case 3:
        clearConsole();
        System.out.print("Введіть індекс претендента
для видалення: ");
        int rmi = sc.nextInt();
        if (rmi < 0 || rmi >= agency.size()) {
            System.out.println("Неможливо знайти
запис");
        } else {
            agency.remove(rmi);
        }
        break;
    case 4:
        clearConsole();
        System.out.println("Введіть індекс для
пошуку претендента: ");
        int gi = sc.nextInt();
        if (gi < 0 || gi >= agency.size()) {
            System.out.println("Неможливо знайти
запис");
        } else {
            System.out.println(agency.get(gi).toString());
        }
        sc.nextLine();
        sc.nextLine();
        break;
    case 5:
        filePath = pickFileMenu();
        if (filePath != null)
            agency.read(filePath);
        break;
    case 6:
        filePath = pickFileMenu();
        if (filePath != null)
            agency.write(filePath);
        break;
    }
}
}
}

```

3 ВИСНОВКИ

```
0. Вихід
1. Вивести список претендентів
2. Додати претендента
3. Видалити претендента
4. Показати претендента за номером в списку
5. Зчитати список претендентів з файлу
6. Записати список претендентів у файл
```

Рис. 3.1 – Програмне меню

```
0. Вихід
1. Вивести список претендентів
2. Додати претендента
3. Видалити претендента
4. Показати претендента за номером в списку
5. Зчитати список претендентів з файлу
6. Записати список претендентів у файл
-----
----> 2
Досвід: rabota1:2;
Освіта: vishee;
Дата звільнення: 2020-10-10
Вимоги до роботи: req1,req2;
```

Рис. 3.2 – Додати претендента

```
0. Вихід
1. Вивести список претендентів
2. Додати претендента
3. Видалити претендента
4. Показати претендента за номером в списку
5. Зчитати список претендентів з файлу
6. Записати список претендентів у файл
-----
----> 1
1. Challenger[experience={rabota1=2}, education='vishee;', dateOfDismissal='2020-10-10', requirementsForWork=[[req1, req2]]]
```

Рис. 3.3 – Виведення списку претендентів

```
0. Вихід
1. Вивести список претендентів
2. Додати претендента
3. Видалити претендента
4. Показати претендента за номером в списку
5. Зчитати список претендентів з файлу
6. Записати список претендентів у файл
-----
----> 4

Введіть індекс для пошуку претендента:
0
Challenger[experience={rabota1=2}, education='vishee;', dateOfDismissal='2020-10-10', requirementsForWork=[[req1, req2]]]
```

Рис. 3.4 – Пошук елемента

На лабораторній роботі оволоділи навичками управління введення/виведення даних з використанням класів платформи Java SE.