

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ  
УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

# **Звіт**

## **Лабораторна робота №3**

З дисципліни:

Об’єктно орієнтоване програмування

**Виконав**

студент групи КН-111

Жигайло Ярослав

**Викладач:**

Грабовська Н. Р.

# Тема роботи

Використання об'єктно-орієнтованого підходу для розробки об'єкта предметної (прикладної) галузі. Оволодіння навичками управління введенням/виведенням даних з використанням класів Java SE.

## 1.Вимоги

### 1.1 Розробник

Жигайло Ярослав Олегович

КН-111

7 варіант

### 1.2 Загальне завдання

Використовуючи об'єктно-орієнтований аналіз, реалізувати класи для представлення сутностей відповідно списку прикладних задач - domain-об'єктів

### 1.3 Завдання

Забезпечити та продемонструвати коректне введення та відображення кирилиці.

Продемонструвати можливість управління масивом domain-об'єктів.

Забезпечити можливість збереження і відновлення масива об'єктів рішення завдання

Забороняється використання стандартного протокола серіалізації.

Продемонструвати використання моделі Long Term Persistence.

Забезпечити діалог з користувачем у вигляді текстового меню. 8. При збереженні та відновленні даних забезпечити діалоговий режим вибору директорії з відображенням вмісту і можливістю переміщення по підкаталогах.

Бюро знайомств. Запис про клієнта: стать; реєстраційний номер; дата реєстрації; відомості про себе (довільний набір властивостей: ім'я, зріст, колір очей, дата народження, хобі тощо); вимоги до партнера (довільний набір властивостей).

## 2.Опис програми

### 2.1 Засоби ООП

Декомпозиція для розділення завдання між методами і класами. Використання XML для серіалізації і десеріалізації.

### 2.2 Ієрархія і структура класів

Для завдання написано клас-domain DomainClient , головний клас Main , клас Menu , клас PathGetter.

### 2.3 Важливі частини коду

#### Вивід меню

```
System.out.println("Choose what you want to do:");
System.out.println("1. Create new user");
System.out.println("2. Change gender of the user");
System.out.println("3. Change number of the user");
System.out.println("4. Change registration date");
System.out.println("5. Change name");
System.out.println("6. Change weight");
System.out.println("7. get gender");
System.out.println("8. get number");
System.out.println("9. get registration date");
System.out.println("10. get name");
System.out.println("11. get weight");
System.out.println("12. Print to the file");
System.out.println("13. Get from the file");
System.out.println("14. Change the user");
System.out.println("15 . Close the program");
k = in.nextInt();

return k;
```

## Серіалізація/Десеріалізація

```
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<DomainClient>) decoder.readObject();
    decoder.close();
    break;
```

## Вибір директорії

```
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;
            }
        }
    }
}
```

## **Висновки**

На цій лабораторній роботі, я розробив консольну програму на java для реалізації діалогового режиму роботи з користувачем. Вирішив прикладну задачу з використанням масивів , рядків , класів , ітераторів і серіалізаторів.