

## **Лабораторна робота №3.**

### **Мета:**

- 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. Використовуючи об'єктно-орієнтований аналіз, реалізувати класи для представлення сутностей відповідно списку прикладних задач - domain-об'єктів (Прикладні задачі. Список №2. 20 варіантів)
2. Забезпечити та продемонструвати коректне введення та відображення кирилиці.
3. Продемонструвати можливість управління масивом domain-об'єктів.
4. Забезпечити можливість збереження і відновлення масива об'єктів рішення завдання з Прикладні задачі. Список №2. 20 варіантів.
5. Забороняється використання стандартного протокола серіалізації .
6. Продемонструвати використання моделі Long Term Persistence .
7. 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.

### **1.1 Розробник**

Левицький Богдан, КН-108

### **1.2 Задача**

Захід: дата, час початку і тривалість; місце проведення; опис; учасники (кількість не обмежена)

### **2 Опис програми**

Програма містить інформацію про заходи. Можна додавати,

чистити, виписувати, серіалізувати та десеріалізувати ваші заходи у файл вибраний користувачем

## 2.1 Засоби ООП

Були використані різні класи та методи, структури даних та модифікатори доступу.

## 2.2 Ієрархія та структура класів

1. Клас `Main`, який викликає всі методи та класи створенні користувачем
2. Клас `Event` - domain-об'єкт
3. Клас `PathMover` повертає шлях збереження файлу та демонструє вміст
4. Клас `Date` під-клас класу
5. Клас `Time` під-клас класу
6. Клас `Members` під-клас класу

## 2.3 Важливі фрагменти програми.

### PathMover

```
1 import java.io.File;
2
3
4 public class PathMover
5 {
6     StringBuilder path;
7
8     public PathMover()
9     {
10         path = new StringBuilder(System.getProperty("user.dir"));
11         path.append("\\");
12     }
13
14     private int index()
15     {
16         for(int i = path.length()-2; i >= 0; i--)
17         {
18             if(path.charAt(i) == '\\')
19                 return i;
20         }
21         return path.length()-1;
22     }
23 }
```

```

private void dir()
{
    File files = new File(path.toString());

    for(File list: files.listFiles())
    {
        if(list.isFile())
        {
            System.out.println("File: " + list.getName());
        }
        else
        {
            System.out.println("Directory: " + list.getName());
        }
    }
}

private void cd()
{
    Scanner in = new Scanner(System.in);
    String input;
    while(true)
    {
        System.out.println("$ " + path + "\t\n");
        dir();
        input = in.nextLine();
        if(input.substring(0,3).equals("cd "))
        {
            if(input.substring(3,5).equals(".."))
            {
                path = new StringBuilder(path.substring(0, index()+1));
            }
            else
            {
                System.out.println(input.substring(3,input.length()));
                path.append(input.substring(3,input.length()));
                path.append("\\");
            }
        }
        else
        {
            path.append(input);
            break;
        }
        System.out.println("$ " + path + "\t\n");
    }
}

public String directory()
{
    cd();
    return path.toString();
}

```

## 7. Варіанти використання

Можна використовувати для зберігання важливих заходів

## **ВИСНОВКИ**

У ході роботи ми навчились використовувати об'єктно-орієнтований підхід для розробки об'єкта предметної (прикладної) галузі

Оволоділи навичками управління введенням/виведенням даних з використанням класів Java SE.