

Лабораторна робота №3.

Об'єктно-орієнтована декомпозиція.

Основи введення/виведення Java SE.

Мета:

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

1.1 Розробник

Гірник Юрій, КН-108, номер варіанту індивідуального завдання – 6.

1.2 Загальна завдання

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.

1.3 Задача

Поліцейська картотека. Дані про злочинця: П.І.Б.; дата народження; дати судимостей (кількість не обмежена); дата останнього позбавлення волі; дата останнього звільнення.

2 Опис програми

Дана програма дозволяє керувати Поліцейською картотекою: додавати до неї певних злочинців з унікальними полями(П.І.Б, дата народження, дати судимостей, дата останнього позбавлення волі, дата останнього звільнення), коректувати та змінювати окремо дані потрібного злочинця.

Також дана програма дозволяє вільно пересувати по директоріям комп'ютера та зберігати/зчитувати дані в XML файли.

2.1 Засоби ООП

У лабораторній роботі використано різні класи з багатьма методами та різними приватними полями. Використано інкапсуляцію даних.

2.2 Ієрархія та структура класів

А) Клас Main з консольною обробкою даних.

Б) Клас Bandit для злочинців.

В) Клас ListFiles, де реалізовано перехід та вибір потрібної директорії.

2.3 Важливі фрагменти програми.

```
public class Bandit {
    private String PIB;
    private String Birth_day;
    private ArrayList <String> Murders = new ArrayList<>();
    private String last_badjob;
    private String last_chanse;

    public void setPIB(String string){ PIB = string; }
    public void setBirth_day(String string){ Birth_day = string; }
    public void setMurders(String string){ Murders.add(string); }
    public void setLast_badjob(String string){ last_badjob = string; }
    public void setLast_chanse(String string){ last_chanse = string; }
    public String getPIB(){ return PIB; }
    public String getBirth_day(){ return Birth_day; }
    public String getMurders(){ return Murders.toString(); }
    public String getLast_badjob(){ return last_badjob; }
    public String getLast_chanse(){ return last_chanse; }
}
```

```

case "4":
    println("\nВиконуємо запис даних картотеки в XML файл. Виберіть файл: ");
    ListFiles SearchDirectory = new ListFiles();
    XMLEncoder encoder = new XMLEncoder(
        new BufferedOutputStream(
            new FileOutputStream(SearchDirectory.search())));

    encoder.writeObject(bandits.size());
    for(Bandit i: bandits){
        encoder.writeObject(i);
    }
    encoder.close();
    println("\nПроцес пройшов успішно!");
    break;
case "5":
    println("\nВиконуємо зчитування даних картотеки з XML файл. Виберіть файл: ");
    ListFiles SearchDirectory1 = new ListFiles();
    l:
    try{
        XMLDecoder decoder = new XMLDecoder(
            new BufferedInputStream(
                new FileInputStream(SearchDirectory1.search())));

        int size = (int) decoder.readObject();
        for(int i = 0; i < size; i++){
            bandits.add((Bandit) decoder.readObject());
        }
        decoder.close();
        println("\nПроцес пройшов успішно!");
        break;
    }
    catch(FileNotFoundException ex){
        System.out.println("Схоже, що такого файлу не існує. Спробуйте ще раз!");
        continue;
    }
}

```

3. Варіанти використання

Дана програма може використовувати поліцейськими для зберігання інформації про різних злочинців.

ВИСНОВКИ

У даній лабораторній роботі розвинулись навички управління введенням/виведенням даних з використанням класів Java SE(ArrayList). Також, крім стандартної серіалізації, було вивчено збереження даних за допомогою XML файлів.