

## **Лабораторна робота №5.**

### **Регулярні вирази. Перевірка даних. Обробка тексту.**

#### **Мета:**

- Ознайомлення з принципами використання регулярних виразів для перевірки рядка на відповідність шаблону.
- Ознайомлення з принципами використання регулярних виразів для обробки тексту.

#### **1.1 Розробник**

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

#### **1.2 Загальна завдання**

Продемонструвати ефективне(оптимальне) використання регулярних виразів для перевірки коректності(валідації) даних, що вводяться, перед записом в domain-об'єкти відповідно до призначення кожного поля для заповнення розробленого контейнера в попередній роботі:

- при зчитуванні даних з текстового файла в автоматичному режимі;
- при введенні даних користувачем в діалоговому режимі.

Використовуючи програми рішень попередніх задач, продемонструвати ефективне(оптимальне) використання регулярних виразів при вирішенні практичного завдання з Прикладні задачі. Список №2. 20 варіантів.

- Передбачити можливість незначної зміни умов пошуку.
- Продемонструвати розроблену функціональність в діалоговому та автоматичному режимах.

#### **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 є покращеною версією попередньої лабораторної роботи: додана валідація даних(коректне введення імен та дат).

## 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.

Г) Клас Matcher1 для валідації даних.

## 2.3 Важливі фрагменти програми.

```
public class Matcher1 {  
    public static boolean Data(String x){  
        Pattern p = Pattern.compile("(^([0-9]|[1-2][0-9]|3[0-1])\\.(0[1-9]|1[0-2])\\.(1[0-9][0-9]|20[0-3][0-9])$");  
        Matcher m = p.matcher(x);  
        return m.matches();  
    }  
    public static boolean Name(String x){  
        Pattern p = Pattern.compile("^[A-Z][a-z]+\\s[A-Z][a-z]+\\s[A-Z][a-z]+$");  
        Matcher m = p.matcher(x);  
        return m.matches();  
    }  
}
```

```
public void setBirth_day(String string){  
    if(Matcher1.Data(string))  
        Birth_day = string;  
    else  
    {  
        while (Matcher1.Data(string)==false)  
        {  
            System.out.print("Неправильне введення даних!\n "  
                + "(Введіть коректну дату - ДД.ММ.РРРР)\n "  
                + "Спробуйте: ");  
            string = write.nextLine();  
        }  
        Birth_day = string;  
    }  
}
```

## 3. Варіанти використання

Дана програма все так само може використовувати поліцейськими – небезпеки ввести неіснуючу дату чи неправильне ім'я немає.

## ВИСНОВКИ

Протягом виконання лабораторної роботи я ознайомився з принципами використання регулярних виразів для перевірки рядка на відповідність шаблону; реалізував свій власний клас Matcher.