

Тема 12. Регулярні вирази. Обробка тексту

Мета:

- Ознайомлення з принципами використання регулярних виразів для обробки тексту.

1 ВИМОГИ

1.1 Розробник

Інформація про розробника:

- Гряник Георгій Володимирович
- КІТ-119Д;
- 6 варіант.

1.2 Загальне завдання

1. Використовуючи програми рішень попередніх задач, продемонструвати ефективне (оптимальне) використання регулярних виразів при вирішенні [прикладної задачі](#).
2. Передбачити можливість незначної зміни умов пошуку.
3. Продемонструвати розроблену функціональність в діалоговому та автоматичному режимах.

1.3 Задача

Поліцейська картотека

Поліцейська картотека. Знайти всіх ув'язнених не молодше 20 років з прізвищем, що починається з голосної та містить комбінацію "ко". Наприклад: Архипенко, Ішкова, Єрмаков, Янковський.

2 ОПИС ПРОГРАМИ

2.1 Засоби ООП

Розробити Regex-регулярний вираз(-и) для перевірки та контролю даних при вводі даних чи зчитуванні файлів.

2.2 Ієрархія та структура класів

Клас "PoliceFile" – описує поліцейську картотеку з можливістю додавати та виводити дані класу. Клас описує дані про злочинця відповідно до завданн. Клас «Date» – опису формат часу :день, місяць, рік . Створений для ергономічного запису дат відомостей про злочинця. Клас « Console_program» - клас керування програмою , створений щоб надавати користувачеві можливість керувати програмою. Клас «ContainerList» - клас-контейнер створений для зберігання даних у список. Реалізовано додавання, видалення та інші можливості для керування даними. Клас «Serializator» - клас розроблений для збереження даних контейнеру у файл. При цьому зберігання проходить у звичайний файл та файл типу .xml. При цьому в класі реалізовано методи для відновлення даних як із звичайного файлу так із .xml файлу. Клас Console_File розроблений для роботи із файлами розміщені в директоріях. Цей клас забезпечує можливість користувачеві обирати файл та перемінятися між директоріями. Клас «Helper» - реалізація допоміжних

методів які реалізують допоміжні дії в основній програмі. Клас «Console_program_auto» - клас розроблений для автоматизації програмою. Клас «Regex» - клас реалізація методів перевірок текстів за допомогою регулярних виразів.

2.3 Важливі фрагменти програми

```
/**
 * @author <Георгій>
 *
 */
import java.util.Date;
import java.util.regex.*;

public class Regex {

    public static boolean Name(String name){
        Pattern p = Pattern.compile("^([A-ZА-ЯЁІІ][a-za-яёіі]{1,15}$");
        Matcher m = p.matcher(name);
        return m.matches();
    }

    public static boolean Data(String data){
        Pattern p = Pattern.compile("^(([0-2]\\d|3[0-1])[\\,\\.](0\\d)|(1[0-2]))[\\,\\.](20\\d{2}|19\\d{2})$");
        Matcher m = p.matcher(data);
        return m.matches();
    }

    public static boolean SupperPF(String supper){
        Pattern p = Pattern.compile("^Name:[A-ZА-ЯЁІІ][a-za-яёіі]{1,15}; Surname:[A-ZА-ЯЁІІ][a-za-яёіі]{1,15}; Lastname:[A-ZА-ЯЁІІ][a-za-яёіі]{1,15}"
            + "; ДН:([0-2]\\d|3[0-1])[\\,\\.](0?\\d|1[0-2])[\\,\\.](200[0-1]|19\\d{2})"
            + "; ОПВ:([0-2]\\d|3[0-1])[\\,\\.](0?\\d|1[0-2])[\\,\\.](20\\d{2}|19\\d{2})"
            + "; ОЗ:([0-2]\\d|3[0-1])[\\,\\.](0?\\d|1[0-2])[\\,\\.](20\\d{2}|19\\d{2})"
            + "; ДС:(([0-2]\\d|3[0-1])[\\,\\.])?(0\\d|1[0-2])[\\,\\.](20\\d{2}|19\\d{2})( )?(1,;,$)");
        Matcher m = p.matcher(supper);
        return m.matches();
    }

    public static boolean LitePF(String supper){
        Pattern p = Pattern.compile("^Name:[A-ZА-ЯЁІІa-za-яёіі]{1,15}; Surname:[A-ZА-ЯЁІІa-za-яёіі]{1,15}; Lastname:[A-ZА-ЯЁІІa-za-яёіі]{1,15}"
            + "; ДН:([0-9\\,\\.]){8,10}"
            + "; ОПВ:([0-9\\,\\.]){8,10}"
            + "; ОЗ:([0-9\\,\\.]){8,10}"
            + "; ДС:(([0-9\\,\\.]){8,10}( )?(1,;,$)");
        Matcher m = p.matcher(supper);
        return m.matches();
    }

    public static boolean NotYoung(String supper){
        Date date = new Date();

        Pattern p = Pattern.compile("^(([0-2]\\d|3[0-1])[\\,\\.])?(0\\d|1[0-2])[\\,\\.](200[0-1]|19\\d{2})$");

        Matcher m = p.matcher(supper);
```

```

        return m.matches();
    }

    public static boolean KoName(String KoName){
        Pattern p = Pattern.compile("^.*([KkKк][OoOo]).*$");
        Matcher m = p.matcher(KoName);
        return m.matches();
    }

    public static boolean KoNameNotYoung(String KoName){

        Pattern p =
Pattern.compile("^.*Lastname:[ААΟΥЭЫЕЁЯЮІЇЄ].*([KkKк][OoOo]).*;*ДН:(\\d|[1-2]\\d|3[0-
1])[\\,\\.](0?\\d|1[0-2])[\\,\\.](200[0-1]|19\\d{2});.*$");
        Matcher m = p.matcher(KoName);
        return m.matches();
    }

    public static boolean SupperData(String supper){
        Pattern p = Pattern.compile("^((?:31(\\/|-
|\\.)(?:0?[13578]|1[02]))\\1|((?:29|30)(\\/|-|\\.))"
+ "(?:0?[1,3-9]|1[0-2])\\2)((?:1[6-9]|2-
9)\\d)?\\d{2})$|^((?:29(\\/|-|\\.))0?2"
+ "\\3(?:1[6-9]|2-
9)\\d)?(?:0[48]|[2468][048]|[13579][26])|((?:16|[2468][048]|[3579][26]"
+ ")00)))$|^((?:0?[1-9]|1\\d|2[0-8])(\\/|-|\\.)(?:0?[1-
9])|(?:1[0-2]))\\4(?:1[6-9]|2-9)\\\"
+ "d)?\\d{2})$");
        Matcher m = p.matcher(supper);
        return m.matches();
    }
}
}

```

3 ВАРІАНТИ ВИКОРИСТАННЯ

Регулярні вирази реалізовані таким чином аби можна було перевірити ім'я ,дату на правильність вводу. При вводі даних регулярний вираз контролює введені дані і у разі помилки вводу програма ігнорує введені дані та просто їх не записує до списку.

Оновлено пошук, Пошук виконує прикладне завдання шляхом перебору даних та перевірки цього об'єкта за допомогою регулярного виразу

Ім'я	По-батькові	Прізвище	дата народження	дата ОПВ	дата останнього звільнення	дати судимостей
Оксана	Вікторівна	Анапенко	03.01.1995	14.12.2016	04.05.2019	02.12.2005
Пурпел	Олегович	Чорний	14.12.1993	22.03.2019	12.12.2020	05.03.2009
Павло	Олександрович	Панапенко	12.12.1920	12.12.1980	12.12.1990	02.12.1940 03.12.1950

Рисунок 1. Вивід даних масиву

```
Оберіть команду:
*1 - Вивести поточні записані данні
*2 - Додати данні
*3 - Оновити дані
*4 - пошук
*5 - Кількість злочинців у базі
*6 - зберегти дані(save)
*7 - перетворення у масив, перетворення у рядок
*8 - Сортувати
*9 - Знайти всіх ув'язнених не молодше 20 років з прізвищем, що починається з голосної та містить комбінацію "ко". Наприкл
*10 - Видалити за номером
*11 - очистити дані
*12 (exit)-вийти

ваша команда: 9
Name:Оксана; Surname:Вікторівна; Lastname:Анапенко; ДН:3.1.1995; ОПВ:14.12.2016; ОЗ:4.5.2019; ДС:2.12.2005 ;
```

Рисунок 2. Результат пошуку даних за допомогою перевірки регулярним виразом

ВИСНОВКИ

Під час виконання лабораторної роботи було набуто навички роботи створення регулярних виразів та використання їх при зчитування даних з файлу та пошуку.