

# ***Лабораторна робота №5.***

## **Мета**

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

## **Вимоги**

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.

**Розробник:** Гринишин Анастасія , КН-108, номер варіанту індивідуального завдання – 9.

## **Задача**

Регулярні вирази. Перевірка даних. Обробка тексту.

## **Ієрархія та структура класів:**

- 1.Клас Main, який містить 1 функцію – main.
- 2.Клас Filego, який містить 2 функції – doFile, make\_info.
- 3.Клас Demain, який містить 3 поля – numberFlight, date, numberFlight і їхні гетери та сетери.
- 4.Клас Station, який містить 4 поля – nameStation, dateArrival, dateDeparture, freeSeat.
- 5.Клас ConstructorsForXML , який містить 2 функції – WriteParamXML, read.
- 6.Клас SimpleArray, який є контейнером.
- 7.Інтерфейс Simple.
- 8.Клас ArrayIterator, який містить 2 функції – hasNext, next.
- 9.Клас ReDex, який містить 5 функції – time, station, seat, date, nameF.

## Фажливий фрагмент коду:

```
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class RedEx {
    public String time(String s){
        Pattern pattern = Pattern.compile("^(((0-1){1}[0-9])|((2){1}[0-9]{1}))(:((0-5){1}[0-9]{1})$");
        Matcher matcher = pattern.matcher(s);
        if (matcher.find()){
            return s;
        }
        else {
            return "Error";
        }
    }

    public String station(String s){
        Pattern pattern = Pattern.compile("^([A-Z]{1})[a-z]{1,20}$");
        Matcher matcher = pattern.matcher(s);
        if (matcher.find()){
            return s;
        }
        else {
            return "Error";
        }
    }

    public String seat(String s){
        Pattern pattern = Pattern.compile("^([0-9]{1,4})$");
        Matcher matcher = pattern.matcher(s);
        if (matcher.find()){
            return s;
        }
        else {
            return "Error";
        }
    }

    public String date(String s){
        Pattern pattern = Pattern.compile("^(((0-2){1}[0-9])|((3){1}[0-1]{1}))(-(((0){1}[1-9]{1})|((1){1}[0-2]{1})).([0]{1}[0]{1}|((1){1}[0-9]{1})|((2){1}[0]{1}))$");
        Matcher matcher = pattern.matcher(s);
        if (matcher.find()){
            return s;
        }
        else {
            return "Error";
        }
    }

    public String nameF(String s){
        Pattern pattern = Pattern.compile("^([a-z]{0-5})$");
        Matcher matcher = pattern.matcher(s);
        if (matcher.matches()){
            return s;
        }
        else {
            return "Error";
        }
    }
}
```

## Висновок

У ході даної роботи навчилася використовувати регулярні вирази, створювати власні і перевіряти внесені дані на правильність в сетерах та гетерах.