Звіт

Автор: Пумня О., КІТ118Б

Дата: 10.02.2020

Лабораторна робота №12

РЕГУЛЯРНІ ВИРАЗИ. ОБРОБКА ТЕКСТУ

**Мета.** Ознайомлення з принципами використання регулярних виразів для обробки тексту.

**Вимоги:**

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.

ЗАВДАННЯ ДО РОБОТИ

Планувальник. Знайти всі конференції, що пройшли за останні три роки в Харкові та області з тривалістю не менше доби.

ОПИС ПРОГРАМИ

**Ієрархія та структура класів**

Pumnya12 – точка входу в програму

EventSearcher – клас для пошуку заходів, що відповідають умові завдання роботи

ТЕКСТ ПРОГРАМИ

**Текст файлу Pumnya12.java**

package labs.pumnya12;

import java.io.IOException;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

import labs.pumnya07.SchedulerEvent;

import labs.pumnya09.GenericList;

import labs.pumnya10.Dialog;

public class Pumnya12 {

private Pumnya12(){

}

public static void main(String[] args) throws IOException, ClassNotFoundException {

boolean isExit = false;

List<String> list = Arrays.asList(args);

boolean isAuto = list.contains("-auto");

if (isAuto) {

GenericList<SchedulerEvent> events = new GenericList<>();

events.pushFront(SchedulerEvent.generate(false));

SchedulerEvent event = new SchedulerEvent();

event.setDate("20.05.2019");

event.setDescription("Конференция.");

event.setDuration(26);

event.setTime("20:20");

event.setVenue("Харьков.");

event.setParticipants(new ArrayList<>(Arrays.asList("Алекс", "Другие")));

events.pushFront(event);

System.out.print("Данные:");

System.out.println(events.toString());

System.out.println("Поиск:");

System.out.print(EventSearcher.searchEvents(events));

} else {

while (!isExit) {

isExit = Dialog.run(false);

}

}

}

}

**Текст файлу EventSearcher.java**

package labs.pumnya12;

import labs.pumnya07.SchedulerEvent;

import labs.pumnya09.GenericList;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class EventSearcher {

private EventSearcher(){

}

public static String searchEvents(final GenericList<SchedulerEvent> list) {

SchedulerEvent[] events = dayOrMore(list);

StringBuilder found = new StringBuilder();

for (SchedulerEvent event : events) {

if (event != null) {

found.append(event.toString()).append("\n");

}

}

return found.toString();

}

private static SchedulerEvent[] dayOrMore(final GenericList<SchedulerEvent> list) {

SchedulerEvent[] inKharkov = inKharkov(list);

SchedulerEvent[] events = new SchedulerEvent[inKharkov.length];

for (int i = 0, j = 0; i < inKharkov.length; i++) {

if (inKharkov[i] != null) {

events[j++] = inKharkov[i];

}

}

for (int i = 0; i < events.length; i++) {

if (events[i] != null) {

if (events[i].getDuration() < 24) {

events[i] = null;

}

}

}

return events;

}

private static SchedulerEvent[] inKharkov(final GenericList<SchedulerEvent> list) {

SchedulerEvent[] lastYears = lastYears(list);

SchedulerEvent[] events = new SchedulerEvent[lastYears.length];

for (int i = 0, j = 0; i < lastYears.length; i++) {

if (lastYears[i] != null) {

events[j++] = lastYears[i];

}

}

Pattern pattern;

Matcher matcher;

final String REGEX = "(^Х)арьков(ская область)?(\\.|$)";

pattern = Pattern.compile(REGEX);

for (int i = 0; i < events.length; i++) {

if (events[i] != null) {

matcher = pattern.matcher(events[i].getVenue());

if (!matcher.matches()) {

events[i] = null;

}

}

}

return events;

}

private static SchedulerEvent[] lastYears(final GenericList<SchedulerEvent> list) {

Object[] arr = list.toArray();

SchedulerEvent[] events = new SchedulerEvent[arr.length];

for (int i = 0, j = 0; i < arr.length; i++) {

if (arr[i] != null) {

events[j++] = (SchedulerEvent) arr[i];

}

}

Pattern pattern;

Matcher matcher;

final String REGEX\_BOUNDS = "^[201]{3}[7-9]";

pattern = Pattern.compile(REGEX\_BOUNDS);

for (int i = 0; i < events.length; i++) {

matcher = pattern.matcher(events[i].getDate().substring(6, 10));

if (!matcher.matches()) {

events[i] = null;

}

}

return events;

}

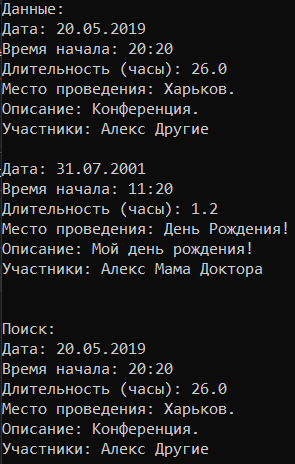
}

ВАРІАНТИ ВИКОРИСТАННЯ

Продемонструємо роботу пошуку (рис. 1).

|  |  |
| --- | --- |
|  |  |
| Дані заходів | Результат пошуку |

Рисунок 1 – Робота пошуку

  
Рисунок 2 – Виконання в авто-режимі

ВИСНОВКИ

При виконанні лабораторної роботи набуто практичних навичок розробки системи пошуку, яка застосовує регулярні вирази як засіб пошуку даних, які відповідають заданій умові.