

1. Тема роботи

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 ВИМОГИ

1.1 Розробник: Мокрик Ярослав Любомирович; КН-108; варіант 19 (завдання 19).

1.2

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 No2. 20 варіантів.

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

1.3 Задача

Екскурсії: дата, час початку і тривалість; місце проведення; опис; учасники (кількість не обмежена).

2 ОПИС ПРОГРАМИ

2.1 Засоби ООП

Я застосував 5 класів, кожен з яких містить статичні і звичайні методи. Таким чином, я можу створювати об'єкти класу Tour (який я сам написав). Це - клас типу JavaBean, тобто він містить поля і методи (getters & setters). Також, я створив клас GenericLinkedList з приватними підкласами LinkedListNode і InnerIterator. Клас GenericLinkedList - це параметризований клас, який ітерується, серіалізується і є імплементацією зв'язних списків. в Класі TourManager також є методи для серіалізації та десеріалізації даних за стандартним протоколом, а ще методи для збереження і завантаження даних в форматі XML файлу.

2.2 Ієрархія та структура класів

Клас Main - головний клас. Керує роботою всієї програми і викликає методи інших класів.

Клас Menu - клас меню. Керує показом головного меню, приймає ввід користувача і викликає відповідні методи.

Клас TourManager - клас менеджера екскурсій. Керує вводом, виведенням, збереженням, завантаженням, десеріалізацією і серіалізацією даних.

Клас Tour - JavaBean клас, який містить всі необхідні властивості та методи.

Клас `GenericLinkedList` - містить імплементацію зв'язних списків, які ітеруються, параметризуються і серіалізуються. Також містить підкласи `InnerIterator` (ітератор) і `LinkedListNode` (частинка зв'язного списку).

2.3 Важливі фрагменти програми

Це - клас `GenericLinkedList`:

```
package ua.lpnuai.oop.mokryk04;

import java.io.Serializable;
import java.util.Iterator;

public class GenericLinkedList<T> implements Iterable<T>, Serializable {

    private class LinkedListNode<T> implements Serializable { . . . }

    private LinkedListNode<T> head = null, tail = null;
    private int size = 0;

    public void insert(T t) {
        LinkedListNode<T> node = new LinkedListNode<T>(t);
        if (head == null) {
            head = node;
            tail = node;
        }
        else {
            tail.next = node;
            tail = node;
        }
        size ++;
    }

    public int indexOf(T t) {
        LinkedListNode<T> node = head;
        int i = 0;
        while (node != null && node.value != t) {
            node = node.next;
            i ++;
        }
        return i;
    }

    public void remove(int index){
        size --;
        LinkedListNode<T> node = head;
        for (int i = 0; i < index-1; i ++){
            node = node.next;
        }
    }
}
```

```

        if (node.next == null) {
            head = null;
            tail = null;
        }
        if (node == head)
            head = head.next;
        if (node.next == tail) {
            tail = node;
            tail.next = null;
        }
        else if (node != head && node.next != tail)
            node.next = node.next.next;
    }

    public int getSize() {
        return size;
    }

    public T[] toArray() {
        T arr[] = (T[]) new Object[size];
        LinkedListNode<T> node = head;
        for (int i = 0; i < size; i++) {
            arr[i] = node.value;
            node = node.next;
        }
        return arr;
    }

    public String toString() {
        StringBuilder str = new StringBuilder();
        LinkedListNode<T> node = head;
        for (int i = 0; i < size; i++) {
            str.append(node.value);
            node = node.next;
        }
        return str.toString();
    }

    private class InnerIterator implements Iterator<T> { ... }

    @Override
    public Iterator<T> iterator() {
        return new InnerIterator();
    }
}

```

3 ВАРІАНТИ ВИКОРИСТАННЯ

Головне меню:

```
a. Add tour
b. Delete tour
c. Delete all
d. Display all tours
e. Serialize
f. Deserialize
g. Save to XML
h. Load from XML
i. Quit
Your option: |
```

Додавання даних:

```
Date: 13.04.2019
Time: 16:00
Length (min): 40
Place: Lviv
Description: Lviv Tour
Add person? (y/n): y
Person: Yaroslav
Add person? (y/n): n|
```

Виведення даних:

```
1
    Date: 13.04.2019
    Time: 16:00
    Length: 40 min
    Place: Lviv
    Description: Lviv Tour
    People:
        Yaroslav
2
    Date: 14.04.2019
    Time: 09:00
    Length: 30 min
    Place: Lviv
    Description: Second Lviv Tour
    People:
        Andriy
        Danylo
```

Автоматичний режим виконання програми:

```
Automatic mode. Loading file.txt
Object read successfully.
1
    Date: 13.04.2019
    Time: 16:00
    Length: 40 min
    Place: Lviv
    Description: Lviv Tour
    People:
        Yaroslav
2
    Date: 14.04.2019
    Time: 09:00
    Length: 30 min
    Place: Lviv
    Description: Second Lviv Tour
    People:
        Andriy
        Danylo
```

ВИСНОВКИ Я ознайомився з принципами використання регулярних виразів для перевірки рядка на відповідність шаблону. Ознайомився з принципами використання регулярних виразів для обробки тексту.