

1. Тема роботи
Параметризація в Java.
Обробка параметризованих контейнерів

Мета:

- Вивчення принципів параметризації в Java.
- Розробка параметризованих класів та методів.
- Розширення функціональності параметризованих класів.

1 ВИМОГИ

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

1.2

1. Створити власний клас-контейнер, що параметризується (Generic Type), на основі зв'язних списків для реалізації колекції domain-об'єктів з лабораторної роботи No10 (Прикладні задачі. Список No2. 20 варіантів)
2. Для розроблених класів-контейнерів забезпечити можливість використання їх об'єктів у циклі foreach в якості джерела даних.
3. Забезпечити можливість збереження та відновлення колекції об'єктів: 1) за допомогою стандартної серіалізації; 2) не використовуючи протокол серіалізації.
4. Продемонструвати розроблену функціональність: створення контейнера, додавання елементів, видалення елементів, очищення контейнера, перетворення у масив, перетворення у рядок, перевірку на наявність елементів.
5. Забороняється використання контейнерів (колекцій) з Java Collections Framework.
6. Розробити параметризовані методи (Generic Methods) для обробки колекцій об'єктів згідно (Прикладні задачі. Список No2. 20 варіантів).
7. Продемонструвати розроблену функціональність (створення, управління та обробку власних контейнерів) в діалоговому та автоматичному режимах.
 - a. Автоматичний режим виконання програми задається параметром командного рядка -auto. Наприклад, java ClassName -auto.
 - b. В автоматичному режимі діалог з користувачем відсутній, необхідні данні генеруються, або зчитуються з файлу.

1.3 Задача

Експерсії: 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 TourMangager - 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 вивчив принципи параметризації в Java, навчився розробляти параметризовані класи та методи і розширювати функціональності параметризованих класів.