

Лабораторна робота №4.

Параметризація в Java.

Обробка параметризованих контейнерів

Мета:

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

1.1 Розробник

Гірник Юрій, КН-108, номер варіанту індивідуального завдання – 6.

1.2 Загальна завдання

1. Створити власний клас-контейнер, що параметризується (Generic Type), на основі зв'язних списків для реалізації колекції domain-об'єктів з лабораторної роботи №10 (Прикладні задачі. Список №2. 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 наявності 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 №2. 20 варіантів).

7. Продемонструвати розроблену функціональність(створення, управління та обробку власних контейнерів) в діалоговому та автоматичному

режимах.

а. Автоматичний режим виконання програми задається параметром командного рядка -auto. Наприклад, `java ClassName -auto`.

б. В автоматичному режимі діалог з користувачем відсутній, необхідні данні генеруються, або зчитуються з файлу.

1.3 Задача

Поліцейська картотека. Дані про злочинця: П.І.Б.; дата народження; дати судимостей (кількість не обмежена); дата останнього позбавлення волі; дата останнього звільнення.

2 Опис програми

Дана програма реалізовує власний паратрезивані класи з вкладеними параметризованими методами(власний `ArrayList` на основі зв'язних списків). Весь інший функціонал залишився з минулої програми лаб.№2.

2.1 Засоби ООП

У лабораторній роботі використано різні класи з багатьма методами та різними приватними полями. Використано інкапсуляцію даних, параметризовані типи даних.

2.2 Ієрархія та структура класів

А) Клас `Main` з консольною обробкою даних.

Б) Клас `Bandit` для злочинців.

В) Клас `ListFiles`, де реалізовано перехід та вибір потрібної директорії.

Г) Клас `MyList` – параметризований клас на основі зв'язних списків.

2.3 Важливі фрагменти програми.

```
public class MyList<T> implements Iterable<T> {  
  
    private class Node {  
        private T data;  
        private Node next;  
  
        public T getData(){  
            return data;  
        }  
  
        public void setData(T data){  
            this.data = data;  
        }  
  
        public Node getNext(){  
            return next;  
        }  
  
        public void setNext(Node next){  
            this.next = next;  
        }  
    }  
  
    private Node first;  
  
    private int size = 0;  
  
    public int size(){  
        return size;  
    }  
}
```

```
    public void add(T data){  
        Node init = new Node();  
        init.setData(data);  
        if (first != null){  
            Node temp = first;  
            while ((temp.getNext()) != null){  
                temp = temp.getNext();  
            }  
            temp.setNext(init);  
        }  
        else {  
            first = init;  
        }  
        size++;  
    }  
  
    public T get(int index){  
        if (index > size || index < 0)  
            return null;  
        Node temp = first;  
        int count = 0;  
        while(temp!=null){  
            if(count++==index){  
                return temp.getData();  
            }  
            temp = temp.getNext();  
        }  
        return null;  
    }  
}
```

```
@Override
public Iterator<T> iterator() {
    Iterator<T> l;
    l = new Iterator<T>(){
        int counter = 0;

        @Override
        public boolean hasNext(){
            return counter < size();
        }

        @Override
        public T next(){
            return get(counter++);
        }
    };
    return l;
}
```

3. Варіанти використання

Дана програма може використовуватись поліцією в якості поліцейської картотеки. Однак, не є дуже зручною через діалог з користувачем безпосередньо через консольні команди.

ВИСНОВКИ

У даній лабораторній роботі було вивчено всі принципи параметризації Java; розвинуто навички у написанні та обробці параметризованих класів-контейнерів та методів.