

Тема роботи

Параметризовані контейнери

Мета

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

Загальне завдання

Вимоги

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.
 - 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.

Важливі фрагменти програми

```
private GenericLinkedList<Point> points = new GenericLinkedList<Point>();
```

```
public class GenericLinkedList<T> implements Iterable<T>, Serializable {  
  
    private class L_Node<T> implements Serializable {  
        private T value;  
        protected L_Node<T> next;  
  
        public L_Node(T value) { this.value = value; }  
    }  
}
```

```
private class InnerIterator implements Iterator<T> {  
    private L_Node<T> node = head;  
  
    @Override  
    public boolean hasNext() { return node != null; }  
  
    @Override  
    public T next() {  
        if (hasNext()) {  
            T nxt = node.value;  
            node = node.next;  
            return nxt;  
        }  
        return null;  
    }  
  
    @Override  
    public void remove() { throw new UnsupportedOperationException(); }  
}
```

Приклад використання програми

```
Name: As As
Adress: asd
Specialization: fff
Worktime: days and hours: asd : 2
telephones: , 123123123, 12312213, 233232334444, asd, 2, , 123123123, 12312213, 233232334444
```

```
enter name of the point:
As As
enter adress of the point:
asd
enter days:
asd
enter hours:
2
enter number of telephones
you're going to add:
3
123123123
12312213
233232334444
enter spec:
fff
sale point succesfully added!!!
```