Тема роботи

Розробка власних утилітарних класів. Набуття навичок вирішення прикладних задач з використанням масивів і рядків.Реалізація діалогового режиму роботи з користувачем в консольних програмах мовою Java.

ВИМОГИ

Розробник:

Подоба Арсен Мирославович
КН-108

Загальне 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 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 №2. 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.

ОПИС ПРОГРАМИ

Засоби ООП

Декомпозиція для розділення завдання між класами.

Важливі фрагменти програми

```
public void add(T
element)
                                {
                                    Node<T> temp = lastNode;
                                    temp.currentElement = element;
                                    lastNode = new Node<>(null, null);
                                    temp.nextElement = lastNode;
                                    size++;
                                }


                                public T get(int index)
                                {
                                    Node<T> target = firstNode.nextElement;


                                    for (int i = 0; i < index; i++)
                                        target = getNextElement(target);
```

```
        return target.currentElement;
    }

    public boolean set(int index, T elemetn)
    {
        if(index >= size)
            return false;
        else
        {
            Node<T> target = firstNode.nextElement;

            for (int i = 0; i < index; i++)
                target = getNextElement(target);

            target.currentElement = elemetn;
            return true;
        }
    }

    public boolean remove (T element)
    {
        Node<T> targer = firstNode;

        for(int i = 0; i < size; i++)
        {
if(targer.nextElement.currentElement.equals(element))
            {
                targer.nextElement =
targer.nextElement.nextElement;
```

```
                size--;
                return true;
            }
            else
                targer = getNextElement(targer);
        }
        return false;
    }
```

ВИСНОВКИ

В ході лабораторної роботи, я навчився розробляти параметризовані контейнера і методи.