1.Тема роботи

Параметризація в Java. Обробка параметризованих контейнерів

Мета

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

Розробник: Бердник Д.І КН-108

1-Варіант

2.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 №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.**

Завдання для варіанту

Кадрове агентство. Дані про претендента: реєстраційний номер; досвід роботи - набір значень “спеціальність, стаж”; освіта; дата звільнення; вимоги до майбутньої роботи - набір необов’язкових властивостей у **вигляді “спеціальність, умови праці, мінімальна зарплата”.**

Опис програми

Засоби ООП

Я створив 7 класів: Applicant (мій domain об’єкт), ApplicantFunct (операції з Applicant), Experience (підклас для досвіду Applicant) , Main (головний клас), Menu (клас що контролює меню і викликання операцій через меню) , MyLinkedList(custom ArrayList що базується на зв’язному списку), Node (для MyLinkedList)

Важливі фрагменти програми

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

private class InnerIterator implements Iterator<T>{

    private int count;

    {
        count = 0;
    }

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

    @Override
    public T next() {

        return get(count++);
    }

}
```

Варіанти використання програми