

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТУ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Інститут комп’ютерних наук та інформаційних технологій
Кафедра систем штучного інтелекту

Лабораторна робота №4

з дисципліни

«Об’єктно-орієнтоване програмування»

Виконала:

студентка групи КН-108

Семич Тамара

Прийняла:

Грабовська Н.Р.

Львів – 2019 р.

Зміст

1. Тема лабораторної роботи.
2. Мета роботи.
3. Вимоги.
4. Висновок.

Тема : Параметризація в Java. Обробка параметризованих контейнерів.

Мета:

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

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.

Розробник: Семич Тамара, КН-108, номер варіанту індивідуального завдання- 4.

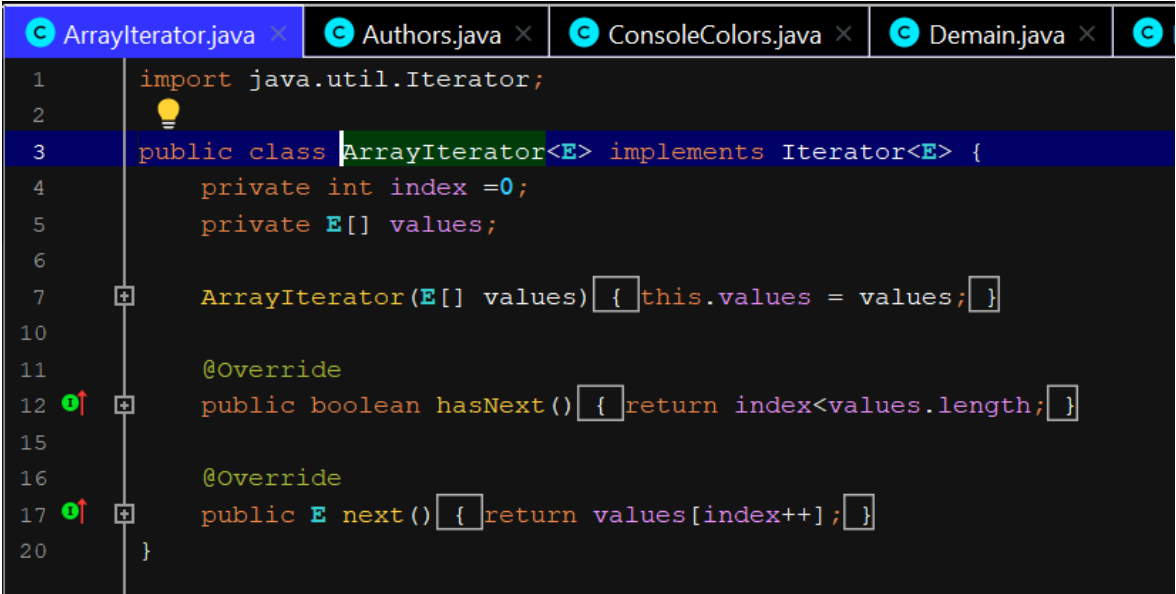
Завдання: Бібліотека. Дані про книгу: ISBN ; назва; автори (кількість не обмежена); видавництво; жанр; дата видання.

Ієрархія та структура об'єктів:

1. Клас Main, який містить функцію main.
2. Клас Filego, який містить функції, doFile та make_info.
3. Клас Demain, який містить поля numberISBN, title, genre, edition, date та їхні гетери / сетери.
4. Клас Authors, який містить поля author, surname.
5. Клас FoXML, який містить функції WriteParamXML, read.
6. Клас ConsoleColors,
7. Клас SimpleArray – контейнер.
8. Інтерфейс Simple
9. Клас ArrayIterator, який містить функції hasNext, next.

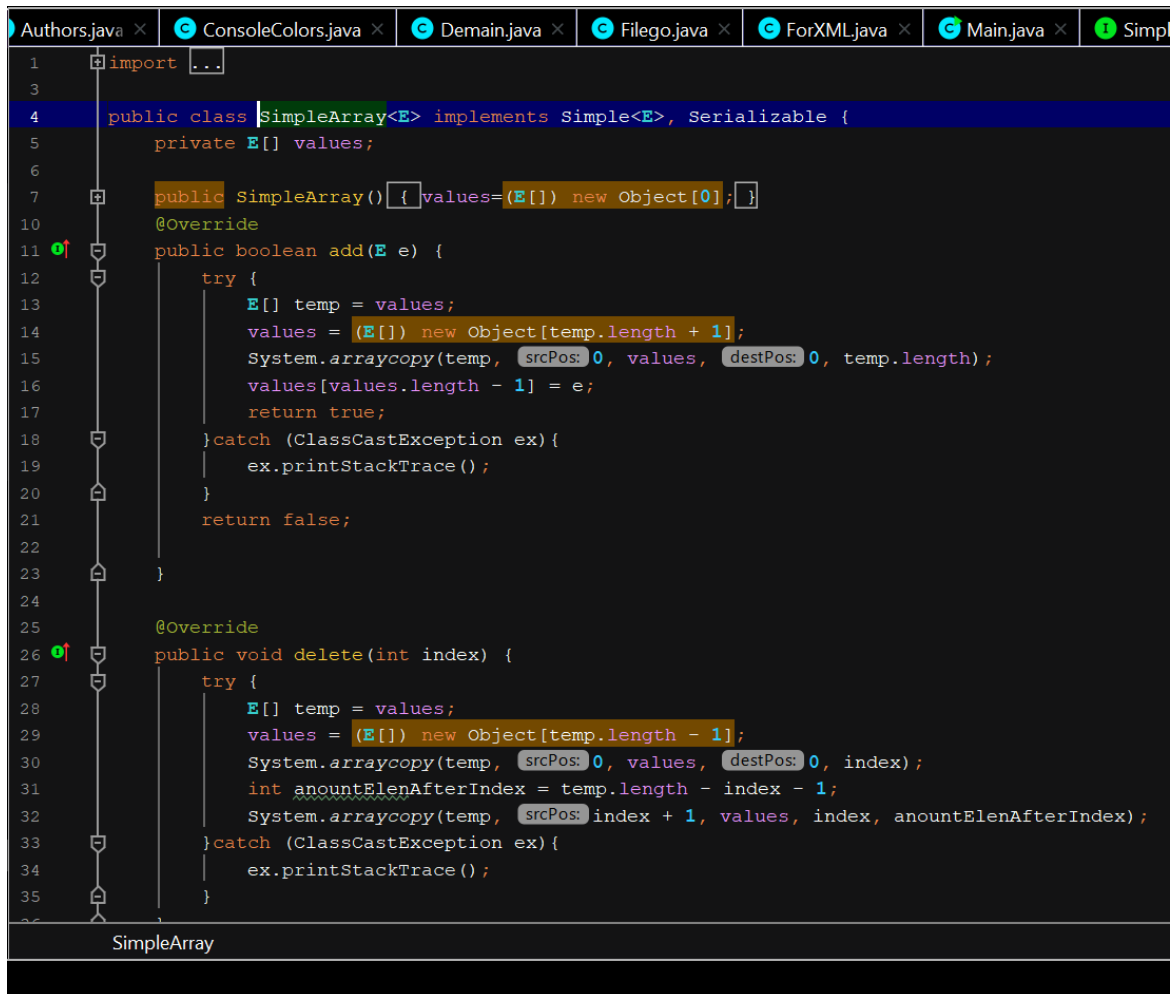
Важливі фрагменти коду:

Клас ArrayIterator:



```
1 import java.util.Iterator;
2
3 public class ArrayIterator<E> implements Iterator<E> {
4     private int index =0;
5     private E[] values;
6
7     ArrayIterator(E[] values) {this.values = values;}
8
9
10
11     @Override
12     public boolean hasNext() {return index<values.length;}
13
14
15
16     @Override
17     public E next() {return values[index++];}
18
19
20 }
```

Клас SimpleArray:



```
1  import ...
2
3
4  public class SimpleArray<E> implements Simple<E>, Serializable {
5      private E[] values;
6
7      public SimpleArray() { values = (E[]) new Object[0]; }
8
9      @Override
10
11     public boolean add(E e) {
12         try {
13             E[] temp = values;
14             values = (E[]) new Object[temp.length + 1];
15             System.arraycopy(temp, srcPos: 0, values, destPos: 0, temp.length);
16             values[values.length - 1] = e;
17             return true;
18         } catch (ClassCastException ex) {
19             ex.printStackTrace();
20         }
21         return false;
22     }
23
24
25     @Override
26     public void delete(int index) {
27         try {
28             E[] temp = values;
29             values = (E[]) new Object[temp.length - 1];
30             System.arraycopy(temp, srcPos: 0, values, destPos: 0, index);
31             int amountElenAfterIndex = temp.length - index - 1;
32             System.arraycopy(temp, srcPos: index + 1, values, index, amountElenAfterIndex);
33         } catch (ClassCastException ex) {
34             ex.printStackTrace();
35         }
36     }
37 }
```

SimpleArray

Висновок: на цій лабораторній роботі я навчилася розробляти власний контейнер, інтерфейс та зберігати дані 2 способами.