

## ЛАБОРАТОРНАЯ РАБОТА № 9

«Исследование возможностей коллекций в языке программирования Java»

**Цель:** Исследование возможностей коллекций в языке программирования Java, сравнение коллекций по времени выполнения операций, знакомство с реализацией коллекций.

### **Учебные вопросы:**

1. Общее описание коллекций в Java;
2. Описание коллекций в соответствии с вариантом;
3. Сравнение коллекций в соответствии с вариантом;
4. Ответы на вопросы.

## Рекомендации к выполнению работы

### Введение

Описание целей работы и задач работы, а также содержимого работы

### 1 Общее описание коллекций в Java

Краткое описание интерфейса Collections и Map, необходимо привести изображения (минимум 3 рисунка), а также две таблицы описывающие основные методы, используемые в интерфейсах Collections и Map.

### 2 Описание коллекций в соответствии с вариантом

Приведите описание реализации конкретной коллекции. (как работает механизм добавления элементов, как работает механизм удаления элементов, как работает механизм поиска элементов и т.д.)

Напишите конкретную вычислительную сложность добавления, удаления и получения элементов в каждой коллекции в соответствии с вариантом;

Каким образом хранятся элементы в коллекции? (упорядочено или хаотично и т.п.)

### 3 Сравнение коллекций в соответствии с вариантом

Необходимо произвести вычисление времени работы нижеперечисленных операций, количество элементов коллекциях и сами коллекции указаны в индивидуальном варианте. После выполнения вычислений необходимо заполнить таблицы и привести скриншоты кода программы и данных вывода в консоль.

1. Сравнение времени выполнения операции добавление элемента в начало коллекции;
2. Сравнение времени выполнения операции добавление элемента в конец коллекции;
3. Сравнение времени выполнения операции добавление элемента в середину коллекции;
4. Сравнение времени выполнения операции удаления элемента в начале коллекции;
5. Сравнение времени выполнения операции удаления элемента в конце коллекции;
6. Сравнение времени выполнения операции удаления элемента в середине коллекции;
7. Сравнение времени выполнения операции получения элемента по индексу;

Листинг 1. Пример кода для вычисления времени добавления n количества элементов в коллекции

```
package com.company;

import java.util.*;

public class Main {

    public static void main(String[] args) {

        List<Integer> arrayList = new ArrayList<>();
        List<Integer> linkedList = new LinkedList<>();

        System.out.println("Время выполнения операции добавления arrayList = " + getRunningTime(arrayList));
        System.out.println("Время выполнения операции добавления arrayList = " + getRunningTime(linkedList));

    }

    private static long getRunningTime(List<Integer> list){

        // точка начала отсчета времени выполнения программы
        long start = System.currentTimeMillis();

        // блок кода в котором выполняется операция добавления
        for (int i = 0; i < 1000000; i++) {
            list.add(i);
        }

        // точка окончания отсчета времени выполнения программы
        long end = System.currentTimeMillis();

        // вывод в консоль времени выполнения блока кода содержащего операцию
        return end - start;

    }

}
```

Пример таблицы 1.

Таблица 1. Сравнение времени выполнения операции добавления

Коллекция	в начало коллекции	в середину коллекции	в конец коллекции
ArrayList			
HashMap			
LinkedList			

Таблица 2. Сравнение времени выполнения операции удаления

Коллекция	в начале коллекции	в середине коллекции	в конце коллекции
ArrayList			
HashMap			
LinkedList			

Таблица 3. Сравнение времени выполнения операции получения элемента по индексу

Коллекция	По индексу для <n> элементов
ArrayList	
HashMap	
LinkedList	

#### **4 Ответы на вопросы в соответствии с вариантом**

Вопросы приведены в описании вариантов

## Варианты

Количество элементов в коллекции:

<номер варианта \* 1 000 000>

Количество элементов для вычисления скорости получения по индексу:

<номер варианта \* 1 000 000 000>

Номер варианта	Коллекции
1	HashMap , LinkedList, Set
2	HashSet , ArrayList, SortedSet
3	HashMap , LinkedHashSet, List
4	HashSet , TreeSet, LinkedHashMap
5	HashMap , TreeMap, SortedSet
6	HashSet , LinkedHashMap, List
7	HashMap , List, Set
8	HashSet , SortedSet, ArrayList
9	HashMap , Set, TreeSet
10	HashSet , Map, LinkedHashMap
11	HashMap , SortedMap, Map
12	HashSet , ArrayList, TreeMap
13	HashMap , LinkedList, TreeSet
14	HashSet , LinkedHashSet, LinkedHashMap
15	HashMap , TreeSet, List
16	HashSet , LinkedHashMap, SortedSet
17	HashMap , TreeMap, ArrayList
18	HashSet , List, LinkedList
19	HashMap , SortedSet, SortedMap
20	HashSet , Map, SortedSet

### Вопросы:

1. В чем разница между fail-safe и fail-fast свойствами?
2. В чем разница между Iterator и Enumeration?
3. В чем разница между Iterator и ListIterator?
4. Дайте определение понятию “итератор”.
5. Как задается порядок следования объектов в коллекции, как отсортировать коллекцию?
6. Как получить коллекцию только для чтения?
7. Как получить не модифицируемую коллекцию?

8. Как получить синхронизированную коллекцию из не синхронизированной?
9. Как реализован цикл `foreach`?
10. Какие есть способы перебора всех элементов `List`?
11. Какие коллекции синхронизированы?
12. Какую функциональность представляет класс `Collections`?
13. Почему `Map` не наследуется от `Collection`?
14. Почему в классе `Iterator` нет метода для получения следующего элемента без передвижения курсора?
15. Почему нет метода `iterator.add()` чтобы добавить элементы в коллекцию?
16. Расскажите о реализации классов `HashSet` и `TreeSet`?
17. Чем отличаются `HashMap` и `TreeMap`? Как они устроены и работают? Что со временем доступа к объектам, какие зависимости?
18. Что будет, если в `Map` положить два значения с одинаковым ключом?
19. Что делать, чтобы не возникло исключение `ConcurrentModificationException`?
20. Что такое `Hashtable`, чем она отличается от `HashMap`? На сегодняшний день она `deprecated`, как все-таки использовать нужную функциональность?