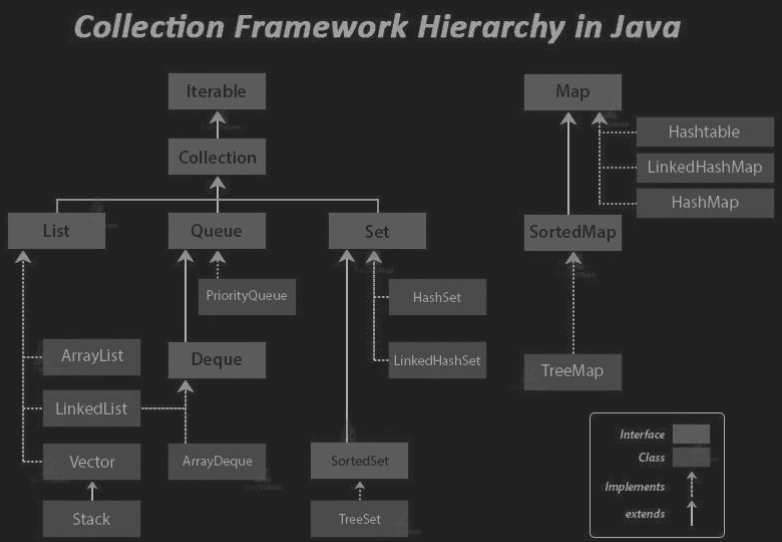
Java Collection I

Java Collection Framework



Java Collection Framework предоставляет архитектуру для хранения и манипулирования группой объектов.

Он включает в себя различные интерфейсы, реализации и алгоритмы для работы с коллекциями.

Еще одна статья покороче: ссылка

Collection, Iterable, Iterator

```java
package l34.slides.ex1;

import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;

public class Main {
   public static void main(String[] args) {
       // Создание списка ArrayList для строк
       List<String> list = new ArrayList<>();
       // Добавление элемента "Apple" в список
       list.add("Apple");
       list.add("Banana");
       list.add("Cherry");

       // Получение итератора для списка
       Iterator<String> iterator = list.iterator();
       // Пока в списке есть элементы...
       while (iterator.hasNext()) {
           // Получение следующего элемента списка
           String fruit = iterator.next();
           // Вывод текущего элемента (фрукта) на экран
           System.out.println(fruit);
       }

   }
}
```

Collection - это корневой интерфейс в иерархии коллекций Java. Он предоставляет основные методы для работы с группами объектов, такие как add(), remove(), size(), isEmpty() и другие.

Iterable - это интерфейс, позволяющий объекту возвращать итератор, который может перебирать элементы. Все реализации Collection наследуют Iterable, что позволяет использовать цикл for-each для перебора коллекций.

Iterator - это интерфейс, предоставляющий методы для итерации по элементам коллекции. Основные методы: hasNext() (проверяет, есть ли следующий элемент) и next() (возвращает следующий элемент).

List, LinkedList, ArrayList

```java
package l34.slides.ex2;

import java.util.ArrayList;
import java.util.LinkedList;
import java.util.List;

public class Main {
    public static void main(String[] args) {
        // Создание списка ArrayList
        List<String> arrayList = new ArrayList<>();
        // Добавление "Apple" в ArrayList
        arrayList.add("Apple");
        arrayList.add("Banana");

        // Создание списка LinkedList
        List<String> linkedList = new LinkedList<>();
        // Добавление "Cherry" в LinkedList
        linkedList.add("Cherry");
        linkedList.add("Date");

        System.out.println(arrayList);
        System.out.println(linkedList);
    }
}
```

List - это упорядоченная коллекция, которая может содержать дубликаты. Она предоставляет методы для вставки, удаления и доступа к элементам по индексу.

LinkedList реализует интерфейс List и использует двусвязный список для хранения элементов. Это обеспечивает эффективное добавление и удаление элементов.

ArrayList также реализует интерфейс List, но использует динамический массив для хранения элементов. Это обеспечивает быстрый доступ к элементам по индексу.

Хеширование

```java
package l34.slides.ex3;

import java.util.HashSet;
import java.util.Set;

public class Main {
    public static void main(String[] args) {
        // Создание хеш-сета для строк
        Set<String> hashSet = new HashSet<>();
        // Добавление "Apple" в хеш-сет
        hashSet.add("Apple");
        hashSet.add("Banana");
        hashSet.add("Apple");

        System.out.println(hashSet);

    }
}
```

Хеширование - это процесс преобразования большого количества информации в маленькое целое число (хеш-код), которое используется для идентификации элементов в хеш-таблицах, таких как HashSet или HashMap.

В этом примере, несмотря на попытку добавить "Apple" дважды, в HashSet он будет присутствовать только один раз из-за уникальности элементов в хеш-таблице.

Контрольная точка

- Понятно ли?

Если все ясно, ставим плюсы, иначе - задаем вопросы.

Домашнее задание

Задачи минимум:

-

The end