15,05,2018 Коллекции Интерфейс Set и SortedList 7

В отличии от интерфейса **java.util.List** эти интерфейс **java.util.Set** не гарантирует вам порядок и не позволяет вставить одинаковые элементы в коллекцию. Наиболее популярными классами на мой взгляд являются следующие:

* **HashSet** — коллекция позволяет дотаточно быстро получить доступ к объекту по хеш-коду
* **TreeSet** — это наследник интерфейса **SortedSet**. И этот интерфейс сортирует все элементы и накладывает на них ограничение — они должны реализовать интерфейс **Comparable** — мы уже с ним сталкивались, когда говорили о сортировке
* **LinkedHashSet** — достаточно любопытный класс, который СОХРАНЯЕТ порядок вставленных элементов и НЕ МЕНЯЕТ их порядок, если элемент вставляется повторно.

package edu.javacourse.collection;

import java.util.LinkedHashSet;

import java.util.Set;

public class ListExample

{

    public static void main(String[] args) {

        Set<String> test = new LinkedHashSet<>();

        // Заполняем от 5 до 1

        for (int i = 5; i > 0; i--) {

            test.add("Строка " + i);

        }

        for (String s : test) {

            System.out.println(s);

        }

        System.out.println();

        // Заполняем (заменяем) от 1 до 5

        for (int i = 1; i < 6; i++) {

            test.add("Строка " + i);

        }

        for (String s : test) {

            System.out.println(s);

        }

        System.out.println();

    }

}

Как видите, мы сначала заполняем элементы от «Строка 5» до «Строка 1», а потом делаем повторную вставку, но в другом порядке. Порядок элементов в самой коллекции при этом не меняется.

run:

Строка 5

Строка 4

Строка 3

Строка 2

Строка 1

Строка 5

Строка 4

Строка 3

Строка 2

Строка 1