15,05,2018 Интерфейс List 6

В реальных проектах коллекции типа **List** используются очень часто, так что постарайтесь как можно быстрее к ним «привыкнуть» — эти классы должны стать для вас рабочими лошадками, которые надо уметь использовать.  
Продемонстрируем на примере те функциональные возможности, которые предлагает интерфейс **List**

package edu.javacourse.collection;

import java.util.ArrayList;

import java.util.Iterator;

import java.util.List;

import java.util.ListIterator;

public class ListExample

{

    public static void main(String[] args) {

        // Создаем список для примера

        List<String> test = new ArrayList<>();

        for (int i = 1; i < 6; i++) {

            test.add("Строка " + i);

        }

        // Стандартный проход по всем элементам коллекции

        System.out.println();

        System.out.println("Печать через цикл foreach");

        for(String s : test) {

            System.out.println(s);

        }

        // Использование итератора

        System.out.println();

        System.out.println("Печать через итератор");

        for(Iterator<String> it = test.iterator(); it.hasNext(); ) {

            String s = it.next();

            System.out.println(s);

        }

        // Используя ListIterator можем двигаться в обоих направлениях

        System.out.println();

        System.out.println("Печать через итератор списка от конца к началу");

        for( ListIterator<String> li = test.listIterator(test.size()); li.hasPrevious(); ) {

            String s = li.previous();

            System.out.println(s);

        }

        System.out.println("Печать через итератор списка от начала к концу");

        for( ListIterator<String> li = test.listIterator(); li.hasNext(); ) {

            String s = li.next();

            System.out.println(s);

        }

        // Обращение к элементу через индекс (позицию)

        System.out.println();

        System.out.println("Печать через индекс элемента");

        for(int i=0; i<test.size(); i++) {

            String s = test.get(i);

            System.out.println(s);

        }

    }

}

* Однозначный порядок элементов — в каком порядке вставляли, в таком они и будут
* Возможность добавлять одинаковые объекты
* Доступ к элементу на определенной позиции
* Возможность перемещаться по списку как от головы к хвосту, так и от хвоста к голове

run:

Печать через цикл foreach

Строка 1

Строка 2

Строка 3

Строка 4

Строка 5

Печать через итератор

Строка 1

Строка 2

Строка 3

Строка 4

Строка 5

Печать через итератор списка от конца к началу

Строка 5

Строка 4

Строка 3

Строка 2

Строка 1

Печать через итератор списка от начала к концу

Строка 1

Строка 2

Строка 3

Строка 4

Строка 5

Печать через индекс элемента

Строка 1

Строка 2

Строка 3

Строка 4

Строка 5

СБОРКА УСПЕШНО ЗАВЕРШЕНА (общее время: 0 секунд)