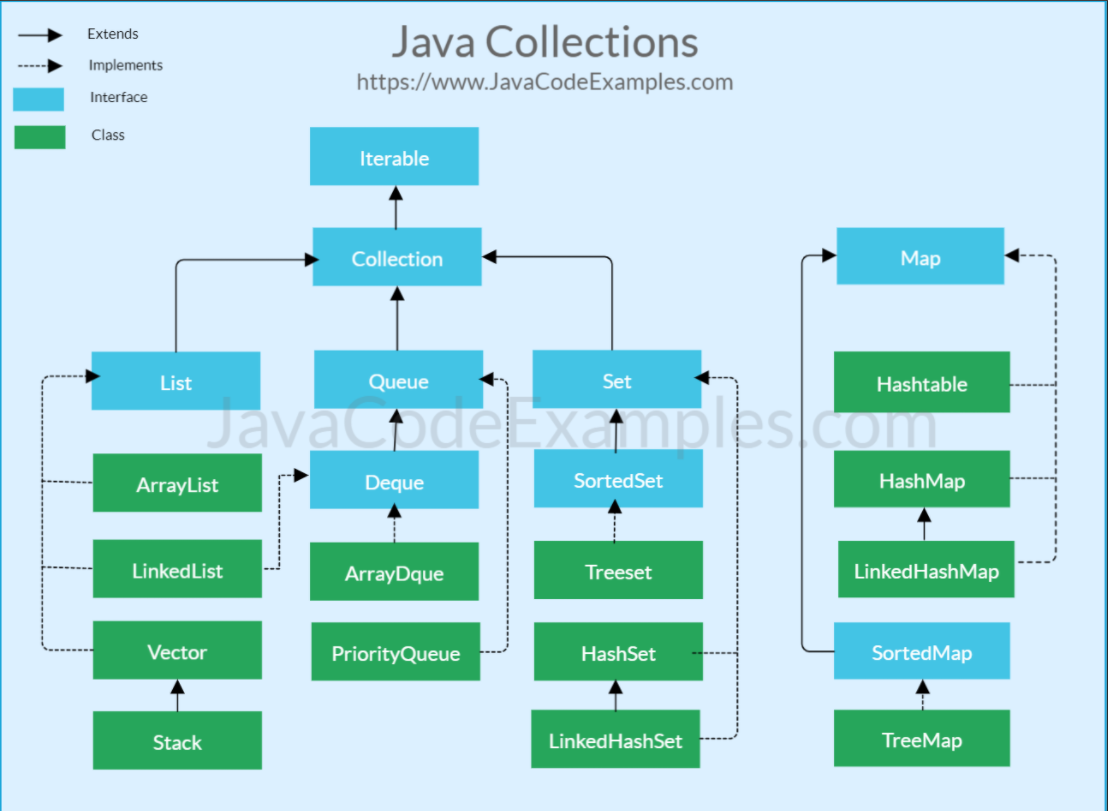
Java Collections Framework



Выше представлен общий вид Java Collections.

1. Все классы коллекции хранятся в пакете (java.util.ArrayList)
2. Все коллекции параметризированы
3. Параметром можно передать только ссылочный тип (примитивы нельзя)
4. При удалении элемента удаление происходит за O(n), так как ArrayList реализован как массив, но удаляя элемент мы сдвигаем все остальные элементы влево, коме случая конца, так как там происходить за O(1)
5. При нехватки ArrayList места, он увеличивается в два раза (плюс 1 к конечному результату), остальные элементы переносятся в новый массив, что делает добавление в массив не за O(1), а примерно за O(1.5)
6. По конвенции при создании коллекции мы ссылаем объекты на интерфейсы:

List<Integer> list1 = new ArrayList<Integer>();

6.1 Ссылаться на класс тоже можно, но тогда в ходе выполнения программы, нельзя будет сменить реализацию интерфейса:

List<Integer> list1 = new ArrayList<Integer>();  
list1 = new LinkedList<Integer>();

Код методов ниже

import java.util.ArrayList;  
import java.util.LinkedList;  
import java.util.List;  
  
public class Start {  
 public static void main(String[] args) {  
  
 List<Integer> list1 = new ArrayList<Integer>();  
 list1 = new LinkedList<Integer>();  
  
 for (int i = 0; i < 100; i++) {  
 list1.add(i);  
 }

// Так можно вывести все элементы ArrayList без цикла

System.*out*.println(list1);  
  
 // Получение элемента по индексу  
 System.*out*.println(list1.get(0));  
 System.*out*.println(list1.get(99));  
  
 // Пройтись по всем элементам ArrayList(не рассматриваем lambda function)  
 for (int i = 0; i < list1.size(); i++) {  
 System.*out*.print(list1.get(i) + " ");  
 }  
  
 System.*out*.println();  
  
 for(Integer elem : list1){  
 System.*out*.print(elem + " ");  
 }  
  
 // Удаление элемента  
 list1.remove(6);  
 }  
}