ArrayList

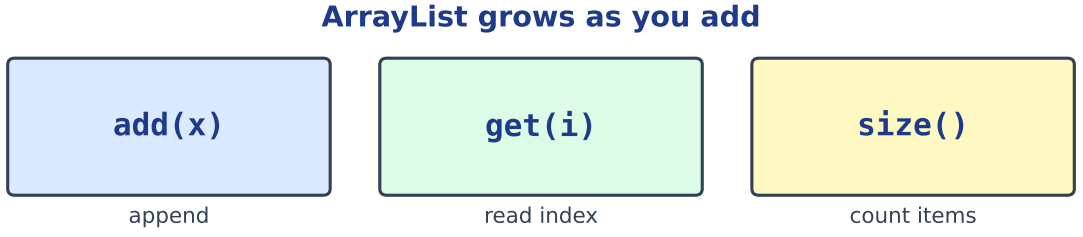
Java Reference

ArrayList 基础

`ArrayList` 是一个可变大小 (resizable) 的列表 — 你添加或删除元素时它会自动增大或缩小。它存储对象, 所以要用包装类 (wrapper) 类型, 例如 `Integer`(不能用 `int`)。`<Integer>` 这部分是泛型 (generic) 类型。常用方法:`.add(x)`、`.get(i)`、`.set(i, x)`、`.size()`、`.remove(i)`。

```
import java.util.ArrayList;

public class Main {
    public static void main(String[] args) {
        ArrayList<Integer> nums = new ArrayList<Integer>();
        nums.add(10);
        nums.add(20);
        nums.add(30);
        System.out.println(nums.size());   // 3
        System.out.println(nums.get(1));   // 20
        nums.set(0, 99);
        System.out.println(nums);          // [99, 20, 30]
    }
}
```



Exam tip: ArrayList<Type> needs a type; indices start at 0 like arrays

ArrayList: add, get, size on a resizable list

ArrayList 算法与删除陷阱

`.remove(i)` 会把后面的每个元素向左移动 (shift) 一位。如果你一边让 `i` **递增**一边删除, 就会跳过下一个元素。解决办法: **倒着**循环, 或删除后不要让 `i` 递增。

```
import java.util.ArrayList;

public class Main {
    public static void main(String[] args) {
        ArrayList<Integer> nums = new ArrayList<Integer>();
        for (int n : new int[]{4, 7, 4, 9, 4}) nums.add(n);
        // 删除所有的 4 -- 倒着循环，这样删除时不会跳过元素。
        for (int i = nums.size() - 1; i >= 0; i--) {
            if (nums.get(i) == 4) nums.remove(i);
        }
        System.out.println(nums);   // [7, 9]
    }
}
```

常见错误

- `ArrayList` 用 `.size()`、`.get(i)`、`.add(...)`— 不是数组那种 `[]`。
- 用下标正向遍历时删除元素会漏掉下一个 (remove bug)。要么倒着遍历, 要么用迭代器。
- 存的是对象而非基本类型: 用 `ArrayList<Integer>`,Java 会自动把 `int` 装箱。