Java程 序 设 计

第 8 讲：经典排序算法与 ArrayList 动态列表

厦门大学嘉庚学院

XIAMEN UNIVERSITY TAN KAH KEE COLLEGE

CONTENTS

冒泡排序(Bubble Sort)原理与优化

【知识要点】相邻比较，两两交换，大数下沉

每轮循环将未排序部分最大值浮至末端；引入 swapped 标志位，当轮无交换则提前退出，最优复杂度达 $O(n)$。

```java
public static void bubbleSort(int[] arr) {
    for (int i = 0; i < arr.length - 1; i++) {
        boolean swapped = false;
        for (int j = 0; j < arr.length - 1 - i; j++) {
            if (arr[j] > arr[j + 1]) {
                int temp = arr[j]; arr[j] = arr[j + 1]; arr[j + 1] = temp;
                swapped = true;
            }
        }
        if (!swapped) break; // 已有序，提前终止
    }
}
```

经典排序算法综合性能对比表

算法名称	最好时间复杂度	平均时间复杂度	空间复杂度	稳定性
冒泡排序	$O(n)$ (优化版)	$O(n^2)$	$O(1)$	稳定
选择排序	$O(n^2)$	$O(n^2)$	$O(1)$	不稳定
插入排序	$O(n)$	$O(n^2)$	$O(1)$	稳定
快速排序	$O(n \log n)$	$O(n \log n)$	$O(\log n)$	不稳定
Arrays.sort()	$O(n \log n)$	$O(n \log n)$ (Dual-Pivot)	$O(n)$	工业级首选

定长数组的局限与 ArrayList 引入

原生数组的痛点

- 长度固定，无法在运行时动态扩容；
- 插入/删除元素需要手动搬移后续元素。

ArrayList 动态列表

- 底层自动扩容数组（默认扩容 1.5 倍）；
- 支持泛型 `ArrayList<T>`；
- 提供丰富的增删改查 API。

ArrayList 核心常用 API 全解

API 方法	方法功能说明	时间复杂度
`list.add(E e)`	向列表末尾追加元素	$O(1)$ 分摊
`list.add(int idx, E e)`	在指定索引位置插入元素	$O(n)$ 需后移元素
`list.get(int idx)`	读取指定索引处元素	$O(1)$ 极速访问
`list.set(int idx, E e)`	替换修改指定索引处元素	$O(1)$
`list.remove(int idx)`	删除指定索引处元素	$O(n)$ 需前移元素
`list.size()`	返回当前列表存储的元素总个数	$O(1)$

【代码演练】基本类型与包装类无缝转换

```java
// 自动装箱：int 自动包装为 Integer 对象
Integer obj = 100; // 相当于 Integer.valueOf(100);

// 自动拆箱：Integer 自动提取内部 int 数值
int val = obj;     // 相当于 obj.intValue();

// 集合中自动装箱
ArrayList<Double> scores = new ArrayList<>();
scores.add(95.5); // 自动装箱为 Double 对象
```

随堂概念抢答 1：ArrayList 泛型与包装类

【随堂抢答·概念检测】问题：以下代码中，哪一行会发生编译报错？

```
1 // 行 1:
2 ArrayList<int> list1 = new ArrayList<>();
3 // 行 2:
4 ArrayList<Integer> list2 = new ArrayList<>();
5 // 行 3:
6 list2.add(100);
7 // 行 4:
8 int val = list2.get(0);
```

A. 行 1

B. 行 2

C. 行 3

D. 行 4

随堂概念抢答 1：答案与深度解析

【知识要点】正确答案：【A 】

原理解析：

- Java 集合泛型参数**只能是引用数据类型，不能是基本数据类型**！因此 `ArrayList<int>` 是非法语法，必须使用包装类 `ArrayList<Integer>`；
- 行 3 的 `list2.add(100)` 会触发**自动装箱（Autoboxing）**，自动将 `int` 包装为 `Integer`；
- 行 4 的 `list2.get(0)` 会触发**自动拆箱（Unboxing）**，自动将 `Integer` 转换为 `int`。

随堂找茬 1:遍历中删除元素的并发修改异常

【随堂找茬·避坑挑错】看看以下代码为什么会在运行时抛出 ConcurrentModificationException?

```
1 ArrayList<String> names = new ArrayList<>(List.of("A", "B", "B", "C"));
2
3 // 致命错误: 在 foreach 遍历中直接调用 list.remove()!
4 for (String s : names) {
5     if ("B".equals(s)) names.remove(s);
6 }
```

【知识要点】正确解法

使用迭代器的 `iterator.remove()` 或现代 Java 8+ 的 `names.removeIf(s -> "B".equals(s));`!

【代码演练】集合升序与降序排序

```java
ArrayList<Integer> nums = new ArrayList<>(List.of(30, 10, 50, 20));

// 升序排序: [10, 20, 30, 50]
Collections.sort(nums);

// 降序排序: [50, 30, 20, 10]
Collections.sort(nums, Collections.reverseOrder());
```

随堂实操练笔 1：冒泡排序优化版实战

【随堂实操·动手练笔】任务 1：编写 BubbleSortDemo.java

编写 `BubbleSortDemo.java`，实现：

1. 生成一个包含 10 个随机整数的数组；
2. 使用优化版冒泡排序进行升序排列，并统计实际执行的比较次数与交换次数；
3. 输出排序前后的数组内容。

随堂实操练笔 2：学生成绩排行榜系统

【随堂实操·动手练笔】任务 2：编写 StudentRosterManager.java

编写 `StudentRosterManager.java`，实现：

1. 动态接收多个学生成绩存入 `ArrayList<Double>`；
2. 调用 `Collections.sort(list, Collections.reverseOrder())` 降序排列；
3. 格式化打印班级第 1 名至第 N 名的榜单。

【工程视野】权衡思维与工程决策

无放之四海皆准的最优算法，唯有约束下的权衡（时间 vs 空间）。工程决策需立足业务场景与资源边界。

【实战技巧】代码审查与辅助开发

实战建议：借助 AI 绘制快速排序分治递归树或 Mermaid 过程图，辅助复杂算法推演理解。

【随堂实操与课后思考】课后思考与巩固

- **练习**：实现插入排序，对比近乎有序数组下的性能。
- **思考**：`ArrayList` 与原生数组在连续性与缓存命中率上有何区别？

选择排序 (Selection Sort) 算法思想

【知识要点】寻找未排序区间极值，交换至首位

每轮外层循环在未排序序列中找到最小（或最大）元素，存放到排序序列的起始位置，直到全部待排序的数据元素排完。

```java
public static void selectionSort(int[] arr) {
    for (int i = 0; i < arr.length - 1; i++) {
        int minIdx = i;
        for (int j = i + 1; j < arr.length; j++) {
            if (arr[j] < arr[minIdx]) minIdx = j;
        }
        int temp = arr[minIdx]; arr[minIdx] = arr[i]; arr[i] = temp;
    }
}
```

插入排序 (Insertion Sort) 算法思想

【知识要点】构建有序序列，逐个向前插入

对于未排序数据，在已排序序列中从后向前扫描，找到相应位置并插入。在数据近乎有序时效率极高，接近 $O(n)$！

ArrayList 动态扩容机制底层原理

【知识要点】默认容量 10，1.5 倍扩容法则

- 初始容量默认为 10；
- 当添加元素导致容量不足时，JVM 自动计算新容量：`newCapacity = oldCapacity + (oldCapacity >> 1)`（即 1.5 倍）；
- 随后调用 `Arrays.copyOf` 分配新数组并搬移数据。

随堂抢答 3：ArrayList 动态扩容倍数

【随堂抢答·概念检测】问题：ArrayList 在容量不足时，默认扩容为原来容量的多少倍？

A. 1.2 倍

B. 1.5 倍

C. 2.0 倍

D. 10 倍

随堂抢答 3：答案与深度解析

【知识要点】正确答案：【B】

原理解析：

- `ArrayList` 源码中使用位运算 `oldCapacity + (oldCapacity >> 1)` 进行扩容；
- 即扩容为原容量的 1.5 倍，兼顾了内存分配效率与空间利用率！

【代码演练】Iterator 安全遍历与删除

```java
ArrayList<String> list = new ArrayList<>(List.of("A", "B", "C"));
Iterator<String> it = list.iterator();
while (it.hasNext()) {
    String item = it.next();
    if ("B".equals(item)) {
        it.remove(); // 安全删除，绝不抛出 ConcurrentModificationException!
    }
}
```

随堂抢答 4：List.of 创建不可变集合

【随堂抢答·概念检测】问题：使用 List.of(”A”, ”B”) 创建的列表，调用 list.add(”C”) 会发生什么？

A. 成功添加

B. 抛出 UnsupportedOperationException (不可修改集合)

C. 忽略添加操作

D. 编译报错

课程互动与讲师主页

在线主页与课程资源

- **个人主页**：
  `https://matnoble.top`
- **课程专栏**：
  `https://matnoble.top/courses/`
- **GitHub 仓库**：
  `https://github.com/MatNoble`

思考：Questions & Answers

“学贵善疑，逻辑筑基”

感谢大家的专注聆听！

课件最新版本与补充资料将持续同步至课程主页。

现在进入自由提问与交流时间。