河南中医药大学信息技术学院（智能医疗行业学院）智能医学工程专业《计算机程序设计》课程

第03章：数据类型

黄子杰

河南中医药大学信息技术学院（智能医疗行业学院）与河南方和信息科技有限公司 联合建设
河南中医药大学信息技术学院智能医疗教研室
https://aitcm.hactcm.edu.cn
2025/2/16

1

本章概要

基本数据类型

概述： Java 是一种强类型语言，所有变量在使用前都必须声明数据类型。Java 提供了 8 种基本数据类型，它们分为四类：整数、浮点数、字符和布尔值。这些类型是最基础的数据类型，直接存储数据的值，通常被存储在栈内存中，效率高。

2

3.1 基本数据类型

- 基本数据类型包括6种数字类型（byte、short、int、long、float、double）、1种字符类型（char）、1种布尔类型（boolean）。

下面我们来一一讲解:

3.1 基本数据类型

- 整数类型:
- byte: 占用 1 个字节（8 位）。范围从 -128 到 127。byte 通常用于节省内存空间，特别是在大数组的场景中。
- short: 占用 2 个字节（16 位）。范围从 -32,768 到 32,767。在需要节省内存的地方使用，例如大量的数字计算。
- int: 占用 4 个字节（32 位）。范围从 -2^31 到 2^31-1。这是 Java 中默认的整数类型，最常用。
- long: 占用 8 个字节（64 位）。范围从 -2^63 到 2^63-1。用于需要更大范围的整数值时，比如在大规模计算、计数或时间戳中。

3.1 基本数据类型

- 生活举例：

byte：假设我们要记录一个月中每天的温度变化（范围大约 -20℃ 到 50℃），byte 类型就足够了，因为温度变化的范围很小。

int：如果要记录一个城市的人口数量，通常会使用 int，因为城市人口通常不会超过 21 亿。

long：对于存储像地球到太阳的距离（大约 150,000,000,000 米），long 是更合适的选择。

3.1 基本数据类型

- 浮点类型：
- float: 占用 4 个字节（32 位），单精度，能表示小数部分但精度较低。常用于需要较小范围小数且不太关心精度的情况。
- double: 占用 8 个字节（64 位），双精度，能表示更大范围的小数且精度更高。是 Java 中默认的浮点数类型，适合于大多数数学计算。

生活举例：

- float：在模拟物理实验时，可能需要存储一些大约的值，比如物体的质量，float 就能满足需要。
- double：在金融计算中，比如计算利息和股价的变化，double 能提供更高的精度，避免金钱计算中的误差。

3.1 基本数据类型

- char: 占用 2 个字节（16 位），用于存储单个字符。char 实际上是一个整数，可以用来表示 Unicode 字符，支持各种语言的字符表示。

- 生活举例：
- char：在计算机系统中，每个按键输入的字母或符号都会被转换为一个 char 类型的值。

- 布尔类型：
- boolean: 占用 1 位，但在实际应用中会占用 1 字节（8 位），用于表示 true 或 false。它通常用于条件判断和逻辑操作。

3.1 基本数据类型

- 生活举例：

boolean：在设计闹钟时，是否设定闹钟可以用 boolean 类型表示。boolean isAlarmSet = true; 表示闹钟已经设置。

- 面试题示例：

问：为什么 Java 中的 boolean 类型不直接占用 1 位?

答：在现代计算机中，内存分配的最小单位是字节（8 位）。虽然布尔值理论上只需要 1 位，但为了简化内存管理，通常使用 1 字节来存储 boolean 类型。

3.1 基本数据类型

- 代码示例：

eg3.1.txt

3.2 内置数据类型

- 概述

Java 的内置数据类型（又称为复合数据类型）包括字符串（String）和数组（Array）等。这些类型允许开发者存储和处理更复杂的数据结构。

下面我们一一讲解：

3.2 内置数据类型

- String 类型：
- String 是 Java 中最常用的内置类型之一，表示字符串，即一组字符的序列。
- String 是不可变的（immutable），一旦创建，不能修改它的内容。任何对 String 的修改都会创建一个新的对象。
- 生活举例：
- String：在编写文章时，文章的每一段文字都可以被存储为一个 String 对象。

3.2 内置数据类型

- Array 类型：
- 数组是存储相同类型元素的集合。数组长度在创建时固定，无法动态改变。
- 数组在内存中是连续存储的，访问速度很快，但在插入或删除元素时效率较低。
- 生活举例：
- Array：当我们整理书架时，可能会将同一系列的书放在一起。这些书就像数组中的元素，按顺序排列。
- 面试题示例：
- 问：String 是不可变的，那么当我们进行 String 拼接时，发生了什么？
- 答：每次对 String 对象的操作都会生成一个新的 String 对象。这是因为 String 是不可变的，无法直接修改原对象。

3.2 内置数据类型

- 代码示例：

eg3.2.txt

3.3 字符串的处理

- 概述：

字符串处理在编程中至关重要，Java 提供了多种类和方法来操作字符串。String 类用于存储不可变字符串，StringBuffer 和 StringBuilder 类则用于可变字符串的处理。

3.3.1 字符串的处理

- 概述：

字符串处理在编程中至关重要，Java 提供了多种类和方法来操作字符串。

String类用于存储不可变字符串。

StringBuffer 和 StringBuilder 类则用于可变字符串的处理。

3.3.1 字符串的处理

String 类

- String 类

- 特点：
- String 是 Java 的内置类，用于存储字符序列。
- String 对象是不可变的，每次修改都会创建一个新对象，这在某些场景下可能影响性能，但也提供了线程安全性。

- 生活举例：
- String：把每条短信内容存储在一个 String 对象中，当发送时，这条短信的内容不可更改。

- 常用方法：
- length(): 返回字符串的长度。
- substring(int beginIndex, int endIndex): 提取子字符串。
- toUpperCase(): 将字符串转换为大写。

3.3.1 String 类

- 面试题示例：

问：为什么 String 类是不可变的？

答：String 类的不可变性（immutability）提供了很多好处，比如线程安全、常量池的优化以及防止不必要的副作用。在多线程环境中，不可变对象不会被改变，因此是线程安全的。

代码示例：

eg3.3.1.txt

3.3.2 String 类

- 特点：
- String类具有多个特性，主要包括不可变性、字符串常量池、equals()方法和hashCode()方法重写、字符串连接、字符串长度获取、字符串截取、字符串分割、字符串替换等。
- 不可变性：String对象一旦创建，其值就不能被修改。任何对String对象的操作都会返回一个新的String对象，原始对象保持不变。这种不可变性保证了数据的一致性，并在多线程环境下提供了线程安全性。
- 字符串常量池：Java中的字符串常量池是一个特殊的内存区域，用于存储字符串常量。当使用双引号创建一个字符串时，Java虚拟机会检查字符串常量池中是否存在该字符串。如果存在，则直接返回该字符串的引用；如果不存在，则创建一个新的字符串，并将其放入字符串常量池中。

3.3.2 String 类

- equals()方法和hashCode()方法重写：String类重写了Object类的equals()方法，用于比较两个字符串的内容是否相等。由于String类重写了equals()方法，所以它也必须重写hashCode()方法，以确保当两个字符串的内容相等时，它们的hashCode()方法返回值也相等。
- 字符串连接：Java中可以使用"+"运算符来连接两个字符串。当使用"+"运算符连接两个字符串时，Java编译器会自动调用StringBuilder的append()方法来实现字符串的连接。
- 字符串长度获取：String类提供了一个length()方法，用于获取字符串的长度，即字符串包含的字符数3。
- 字符串截取和分割：String类提供了substring()方法用于截取字符串，以及split()方法用于将字符串按照指定的分隔符进行分割。

3.3.2 StringBuffer 类

- 代码示例：

eg3.3.2.txt

3.4 基本数据类型包装类

- 概述：

Java 提供了包装类来将基本数据类型转换为对象类型。这些包装类不仅可以与 Java 集合框架兼容，还提供了许多有用的方法，比如数值转换、比较等。每个基本类型都有对应的包装类，如 int 对应 Integer，double 对应 Double。

3.4 基本数据类型包装类

- 自动装箱和拆箱：

Java 在 JDK 5 中引入了自动装箱（autoboxing）和拆箱（unboxing）。自动装箱是指自动将基本数据类型转换为对应的包装类对象，而拆箱则是将包装类对象转换为基本数据类型。

生活举例：

包装类：将数字保存为对象时，使用包装类可以对这些数字执行更多的操作，比如格式化输出、转换为字符串等。

常用方法：

valueOf(String s): 将字符串转换为对应的包装类对象。

intValue(): 将包装类对象转换为基本数据类型。

3.4 基本数据类型包装类

- 面试题示例：

问：什么是自动装箱和拆箱？什么时候会发生？

答：自动装箱是指基本数据类型自动转换为其包装类对象的过程，而拆箱是包装类对象自动转换为基本数据类型的过程。通常在需要对象类型或基本类型进行运算时，编译器会自动进行这些转换。

代码示例：

eg3.4.txt

3.5 BigInteger 类

- 概述：

BigInteger 类用于处理超出 long 范围的大整数。它允许进行任意精度的算术运算（包括加、减、乘、除），并且提供了许多有用的数学方法。

3.5 BigInteger 类

- 详细说明：
- 任意精度：BigInteger 不受普通整型范围的限制，可以处理非常大的数字。
- 算术运算：BigInteger 提供了常见的算术运算方法，如 add()、subtract()、multiply() 和 divide()，这些方法可以处理任意大小的整数。
- 生活举例：
- BigInteger：计算天文学中的大数，比如宇宙中的粒子数量或者大规模加密算法中的大整数，BigInteger 就显得非常重要。

3.5 BigInteger 类

- 面试题示例：

问：BigInteger 是如何存储大整数的？它与基本类型的整数有何不同？

答：BigInteger 使用数组来存储大整数的每个部分，这允许它处理任意大小的整数，而不像基本类型那样受限于固定的字节数。

代码示例：

eg3.5.txt

3.6 日期和时间类

- 概述：

Java 提供了多种日期和时间处理类，分别用于不同场景。传统的 Date 和 Calendar 类可以处理大部分需求，而 Java 8 引入的新日期时间 API 提供了更加直观和强大的功能。

3.6.1 Date 类和 SimpleDateFormat 类

- 特点：

- Date 类表示特定的瞬时时间，精确到毫秒。
- SimpleDateFormat 类用于格式化和解析日期字符串。

- 生活举例：
- Date：如果你要存储一个特定的事件发生的时间，比如面试的日期，可以使用 Date 类。
- SimpleDateFormat：要将 Date 对象转换成可读的日期字符串，例如“2024-08-27”，你可以使用 SimpleDateFormat。

3.6.1 Date 类和 SimpleDateFormat 类

- 面试题示例：

问：Date 类有哪些不足之处，为什么 Java 8 引入了新的日期时间 API?
答：Date 类是可变的，这意味着它在多线程环境中不安全。同时，它的设计也不够直观，许多方法已经过时。因此，Java 8 引入了更强大且线程安全的日期时间 API。

代码示例：

eg3.6.1.txt

3.6.2 Calendar 类

- 特点：
- Calendar 类是更为灵活的日期处理类，支持日期的加减操作以及复杂的时间计算。

- 生活举例：
- Calendar：如果你想计算一个日期的 30 天后是什么日期，或者想知道某个月的最后一天，可以使用 Calendar。

- 面试题示例：
- 问：Calendar 与 Date 有什么不同?
- 答：Calendar 类提供了比 Date 更丰富的功能，可以方便地进行日期的加减操作，而 Date 只是一个瞬时的时间点。

3.6.2 Calendar 类

- 代码示例：

eg3.6.2.txt

3.6.3 Java 8 新增的日期和时间类

- 特点：
- Java 8 引入了全新的日期时间 API，如 LocalDate、LocalTime、LocalDateTime，这些类是不可变且线程安全的。
- 提供了更加直观的日期和时间操作方式，并解决了 Date 和 Calendar 类中的诸多问题。

- 生活举例：
- LocalDate：当你需要处理不带时间的日期，比如生日、纪念日或节假日，可以使用 LocalDate。
- LocalTime：当你只关心时间部分，比如每天的会议时间，可以使用 LocalTime。

3.6.3 Java 8 新增的日期和时间类

- 面试题示例：

问：Java 8 的日期时间 API 与 Date 和 Calendar 有何区别？

答：Java 8 的日期时间 API 是不可变且线程安全的，并且提供了更简洁和易用的 API，而 Date 和 Calendar 是可变的，不适合多线程环境，并且设计较为笨拙。

代码示例：

eg3.6.3.txt

本章重点

- 基本数据类型
- 内置数据类型
- String、 StringBuffer、 StringBuilder 类各自的特点及用法
- Date 类和 SimpleDateFormat 类特点及用法
- LocalDate和LocalDateTime类特点及用法

本章作业

- 编写一个Java程序，分别使用String、StringBuffer、StringBuilder来连接100个字符串（例如，将"Hello, "重复100次）。计算并比较每种方法的执行时间，以展示性能差异。
- 编写一个Java程序，使用Date类获取当前日期和时间，并打印出来。
- 编写一个Java程序，允许用户输入一个日期时间字符串（假设格式为“年-月-日 时:分:秒”），使用SimpleDateFormat解析这个字符串，将其转换为Date对象，然后再以另一种格式（如“月/日/年 时:分 上午/下午”）输出。

信创智能医疗系统研发课程体系

河南中医药大学信息技术学院（智能医疗行业学院）

河南中医药大学信息技术学院（智能医疗行业学院）智能医疗教研室

河南中医药大学医疗健康信息工程技术研究所