

Java基础内容

- 特点
- 基本程序结构
- 变量和数据类型
- 流程控制
- 面向对象

Java特点

Java是一门面向对象编程语言，不仅吸收了C++语言的各种优点，还摒弃了C++里难以理解的多继承、指针等概念。

Java优点

- 功能强大和简单易用
- 应用极其广泛：桌面应用程序、Web应用程序、分布式系统和嵌入式系统应用程序
- 开发者被大量需求
- 引入Java语言虚拟机后，Java语言在不同平台上运行时不需要重新编译。Java语言使用Java虚拟机屏蔽了与具体平台相关的信息，使得Java语言编译程序只需生成在Java虚拟机上运行的目标代码（字节码），就可以在多种平台上不加修改地运行。

Java缺点

- 大量使用内存:Java相比C++,需要使更多的内存
- 学习曲线:Java学起来不是很简单的语言
- 漫长的启动时间

实例操作

在Android Studio中开发普通Java

1. 在已有的Android项目app右键，选择New > Module
2. 选择Module Type 为 Java or Kotlin Library

Java程序结构

```
/**
 * 可用来自动创建文档的注释
 *  Java语法和C++几乎一样
 */
public class Hello {
    public static void main(String[] args) {
        // 向屏幕输出文本:
        System.out.println("Hello, world!");
        /* 多行注释开始
        注释内容
        注释结束 */
    }
} // class定义结束
```

class命名习惯

- 类名必须以英文字母开头，后接字母，数字和下划线的组合
- 习惯以大写字母开头


方法命名习惯

- 类名必须以英文字母开头，后接字母，数字和下划线的组合
- 习惯以小写字母开头

数据类型分类

- 基本数据类型
- 引用数据类型

基本数据类型

byte 

short 

int 

long 

float 

double 

char 

整型取值范围

byte: -128 ~ 127

short: -32768 ~ 32767

int: -2147483648 ~ 2147483647

long: -9223372036854775808 ~ 9223372036854775807

布尔类型

- 布尔类型boolean只有true和false两个值。
- Java语言对布尔类型的存储并没有做规定，因为理论上存储布尔类型只需要1 bit，但是通常JVM内部会把boolean表示为4字节整数。

浮点数

- 浮点数常常无法精确表示

比较两个浮点数是否相等常常会出现错误的结果

- 浮点数比较时用差的绝对值
- 参与运算的两个数其中一个是整型，那么整型可以自动提升到浮点型
- 整数运算在除数为0时会报错，而浮点数运算在除数为0时，不会报错

三元运算

语法: `cond ? x : y`

问题: 分数大于90属于A级, 60-80分属于B级, 低于60分属于C级。用三元运算如何实现?

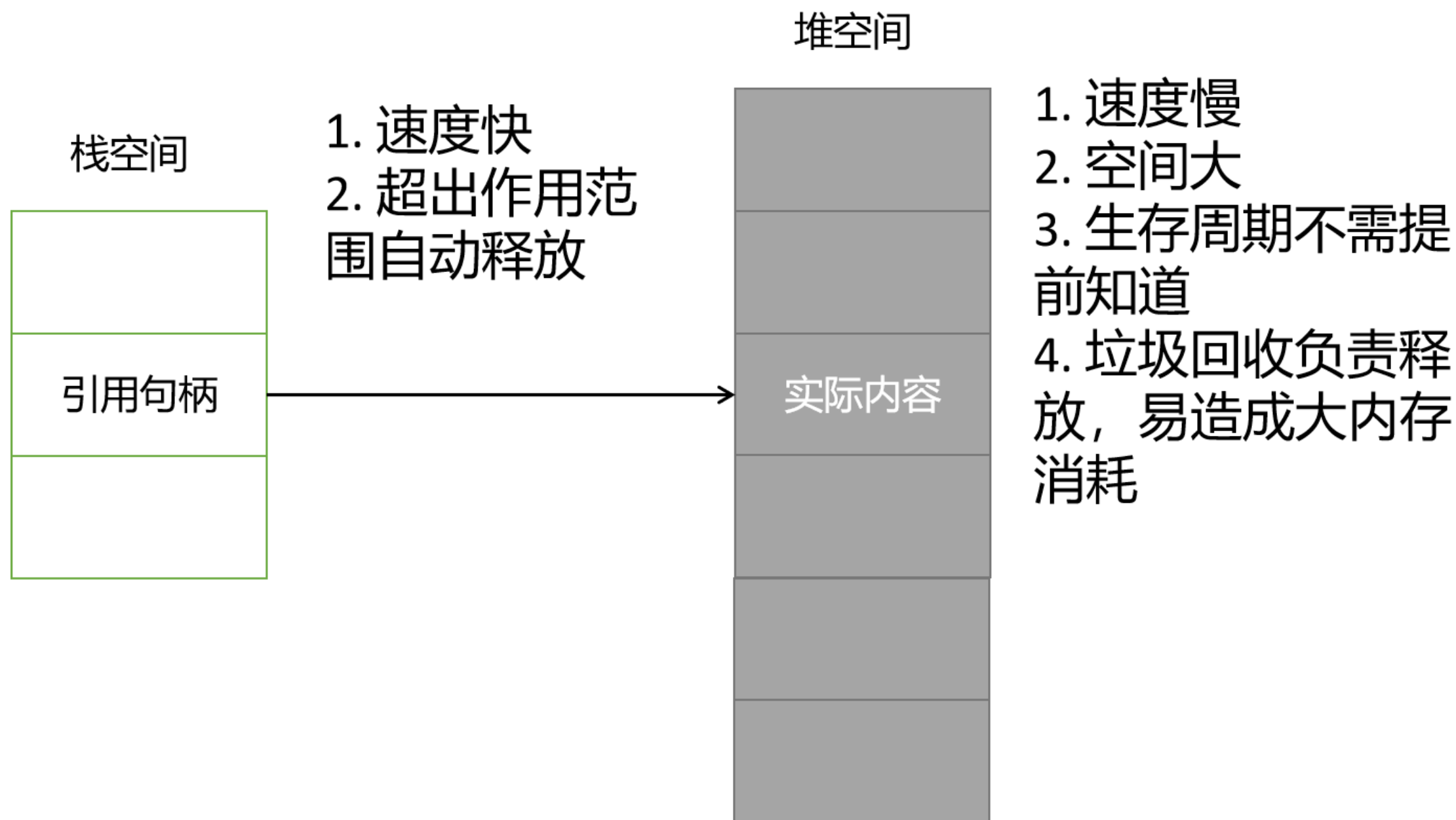
引用类型

引用类型的变量类似于C语言的指针，它内部存储一个“地址”，指向某个对象在内存的位置。

除了上述基本类型的变量，剩下的都是引用类型。

String/数组/类

堆栈内存



String

String是引用类型，用双引号"..."表示.

- 转义符
- 字符串连接
- null和""
- 不可变特性

数组

- 数组是同一数据类型的集合，数组一旦创建后，大小就不可变
- 数组元素可以是值类型（如int）或引用类型（如String），但数组本身是引用类型
- 可以通过索引访问数组元素，但索引超出范围将报错

输出

- print 不换行
- println 换行
- printf 格式化

输入

- `Scanner`和`System.in`
- `next()` 与 `nextLine()`
- `int/float`
- `hasNextxxx` 有效性检查

循环

- while
- do while
- for

和c++几乎一样

switch多重选择

重点注意：**穿透性**和break

使用switch时，如果遗漏了break，就会造成严重的逻辑错误，而且不易在源代码中发现错误