

# JAVA基本语法

## 标识符和关键字

- 标识符 (自己创建)
  - 标识符就是一个名字
- 关键字 (Java创建)
  - 赋予了其特殊的含义, 只能用于特定的地方

## CONTINUE、BREAK 和 RETURN 的区别

- continue
  - 跳出当前的这一次循环, 继续下一次循环
- break
  - 跳出整个循环体, 继续执行循环下面的语句
- return
  - 用于跳出当前方法, 结束该方法的运行

## 成员变量与局部变量的区别

- 语法形式
    - 成员变量属于类的
      - 成员变量可以被 public,private,static 等修饰符所修饰
    - 局部变量是在代码块或方法中定义的变量或是方法的参数
      - 局部变量不能被访问控制修饰符及 static 所修饰
  - 存储方式 (内存中的位置不同)
    - 成员变量: 在堆中 (方法区中的静态区)
    - 局部变量: 在栈中
  - 生存时间 (生命周期不同)
    - 成员变量: 随着对象的创建而存在, 随着对象的消失而消失
    - 局部变量: 随着方法的调用或者代码块的执行而存在, 随着方法的调用完毕后消失或者代码块的执行完毕而消失 (伴随方法的调用或代码的执行而存亡)
  - 默认值 (初始值)
    - 成员变量: 有默认初始值
    - 局部变量: 没有默认初始值, 使用之前需要赋值, 否则编译器会报错 (The local variable xxx may not have been initialized)
- 都能被 final 所修饰

## 静态变量作用

- 被类的所有实例共享。无论一个类创建了多少个对象, 它们都共享同一份静态变量
- 通常情况下, 静态变量会被 final 关键字修饰成为常量

## 字符型常量和字符串常量的区别

- 形式
  - 字符常量
    - 单引号引起的一个字符 (char 在 Java 中占两个字节)
  - 字符串常量
    - 双引号引起的 0 个或若干个字符
- 含义
  - 字符常量
    - 相当于一个整型值 (ASCII 值), 可以参加表达式运算
  - 字符串常量
    - 代表一个地址值 (该字符串在内存中存放位置)
- 占内存大小
  - 字符常量只占 2 个字节
  - 字符串常量占若干个字节

## 静态方法为什么不能调用非静态成员

- 静态方法
    - 属于类的, 在类加载的时候就会分配内存, 可以通过类名直接访问
  - 非静态成员
    - 属于实例对象, 只有在对象实例化之后才存在, 需要通过类的实例对象去访问
- 类加载时静态方法就存在了, 非静态方法可能未实例化, 调用不存在的对象是非法操作

## 静态方法和实例方法区别

- 调用方式
    - 静态方法
      - 类名.方法名 & 对象.方法名 (不建议)
    - 实例方法
      - 仅支持: 对象.方法名
  - 访问类成员限制
    - 静态方法
      - 静态方法在访问本类的成员时, 只允许访问静态成员 (即静态成员变量和静态方法), 不允许访问实例成员 (即实例成员变量和实例方法)
    - 实例方法
      - 无限制
- 调用静态方法可以无需创建对象

## 重载和重写有什么区别

- 重载
    - 同样的一个方法能够根据输入数据的不同, 做出不同的处理
    - 方法名必须相同, 参数类型不同、个数不同、顺序不同, 方法返回值和访问修饰符可以不同
  - 重写
    - 当子类继承自父类的相同方法, 输入数据一样, 但要做出有别于父类的响应时, 需要覆盖父类方法
    - 重写发生在运行期, 是子类对父类的允许访问的方法的实现过程进行重新编写
- 重载就是同一个类中多个同名方法根据不同的传参来执行不同的逻辑处理
- 方法名、参数列表必须相同, 子类方法返回值类型应比父类方法返回值类型更小或相等, 抛出的异常范围小于等于父类, 访问修饰符范围大于等于父类。