

java基础知识

数据类型

- 基本类型：整型、浮点型、布尔型，强制转换——小范围转换成大范围
- 引用类型：类、接口、数组
- 备注：ASCII值a-z: 97-122, A-Z: 65-90, 0-9: 48-57, (int) a即为a的ASCII值

基本语法

- 运算符：算术、逻辑、三目 `c=a>b ? a:b`
- 流程控制语句
 - 判断语句：if...else if...else `<==>`可以与三目运算符转化
 - 选择语句：switch...case 1 break...case 2 break...case3 break... default....break
 - 循环语句
 - `for(int i = 0 ; i<100 ; i++){语句}`
 - `while(条件){循环体}`
 - `do {循环体}while(条件)`
 - 结束语句
 - break: 跳出循环
 - continue: 跳出当前循环，进行下一次循环
- 方法
 - 定义
 - 无参返回: `public static void name(){参数; 功能; }`
 - 含参返回: `public static name(){参数; 返回值}`
 - 调用：重载，名字可一样，参数要求不一致（标识符 public static不影响）
- 数组
 - 内存管理：main()方法在方法栈，new创建的在堆内存，main方法中数组只保存地址
 - 索引index、遍历for循环、反转
 - 数组作为方法参数、返回值
- 类与对象
 - 定义：类表示抽象的一类的属性和行为的集合，对象表示具体的一个实例
 - 类名：首字母大写
 - 类：`public class name{ 成员属性; 成员方法 public void name(){}}`
 - 调用类创建对象：类名 name= new 类名();例如： `Student stu = new Student();`
 - 全局变量（有默认值），局部变量（无默认值）
 - 类实际上实现了对成员变量和成员方法的封装
 - 理解：类里面还有类，如身体和心脏的关系
 - 调用内部类方法
 - 直接调用：外部类.内部类 `name = new 外部类().内部类()`
 - 间接调用：在成员方法里面创建内部类对象，实现调用内部类方法
 - 同名变量访问
 - 默认：就近原则
 - 内部类：this.name
 - 外部类：Body.this.name
 - 局部内部类
 - 定义：在方法内定义的类，与局部变量同理理解
 - 方法的变量必须不变final，用一次就释放了，只有不变才可不断复制存在
 - 匿名内部类
 - 意义：在接口的Implements文件，只用到了了一次，不符合简便，于是把实现的步骤省去直接在内部类对接口中重写
 - 实现：接口名 name = new 接口名(){@override;
 - 要求：只能用一次场景
- 四大访问权限修饰符
 - public: 权限最大
 - private: 权限最小
 - (default): 默认，相当于没有添加
 - protected
 - 常用
 - 变量：用private修饰去细节
 - 方法：用public修饰方便调用

异常

- 意义：对编程过程可能用到的异常进行处理，用户友好
- 异常之间的关系
 - Object类
 - Throwable子类
 - Exception
 - Error, 跟你没关系，不是你的事
 - RuntimeException
 - InterruptedException
 - IOException
- 常见异常
 - 编译异常
 - 直接报红
 - 必须处理
 - 运行时异常RuntimeException
 - ArraysIndexOutOfBoundsException, 索引超出
 - ClassCastException, 强转类型出错
 - NullPointerException, 空数组去索引它
 - ArithmeticException, 除0
 - NumberFormatException, 非数字型字符串转数字
 - 运行时经常遇到，修改代码，不需要专门处理
- 处理机制
 - 抓、抛
 - 抓：系统捕捉到异常，并生成对应异常的类
 - 抛：异常的对象
 - 系统自动报错抛出
 - 手动抛出throw
 - `throw new RuntimeException("Message")`
 - `throw new Exception,必须进行处理`
 - 处理
 - try-catch-finally, 立马处理
 - try(里面放可能出错的代码)
 - catch (异常类型 变量名){ };机制与case一致，可列多种情况
 - finally{必须执行的代码}, 用于应对垃圾回收机制的不足，比如关闭文件
 - 成员方法
 - 获取错误信息，变量名.getMessage()
 - 打印红错，变量名.printStackTrace()
 - 快捷键，Ctrl+Alt+T
 - throws往上抛，像踢皮球
 - 把错误往上抛，可以链式抛，对于递进的代码结构有好处，中间的变量可以传递运行
 - `public void 方法名() throws ClassCastException { }`
 - 子类往上抛的错误类型不能比父类的大，罩不住哇！
- 自定义异常类
 - 继承父类：Class MyException extends RuntimeException{ }
 - 写上序列号：static final long serialVersionUID = -70397190746939L; (任意)
 - 重载构造方法
 - 无参构造，`public MyException{ }`
 - 带参构造，`public MyException(String message) { super(message); }`
- 问题：IDEA如何给args赋值？
- IDEA查看源码快捷键
 - Ctrl+Shift+i
 - 上面操作之后，Ctrl+Enter，完全打开源码
 - Ctrl+鼠标左键