Java 入门- 预科班内容正课会重讲

内参保密不得外传

达内 Java 培优方向

陈子枢 主讲

今日头条极速版

2020

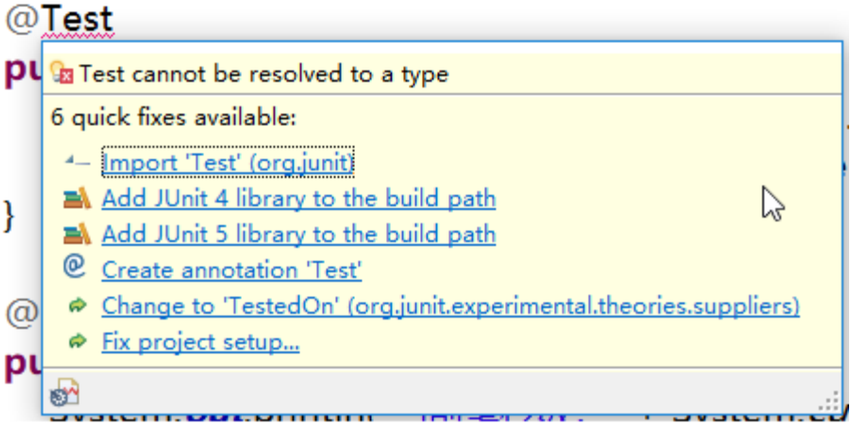
1 第二天：基础语法：JUnit+关键字+拼串+运算符

1.1 扩展：系统环境

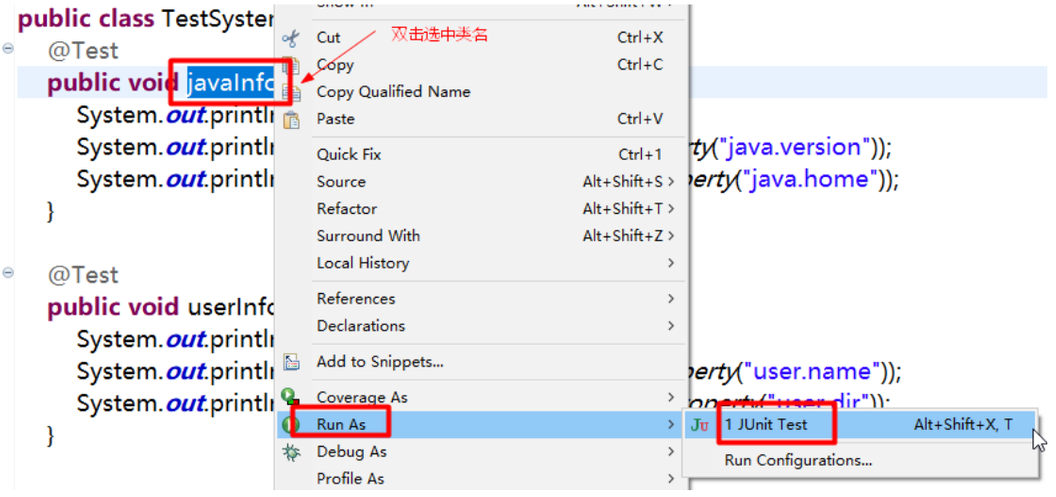
1.1.1 TestHello.java

```
package hello;

import org.junit.jupiter.api.Test;

public class TestHello {
    public static void main(String[] args) {
        System.out.println("Hello main method.");
    }

    @Test
    public void junit() {
        System.out.println("Hello junit method");
    }
}
```

1.1.2 导入 JUnit 支持



1.1.3 运行单元测试



1.1.4 单元测试特点

- 单元测试类独立，只用于测试阶段，其它无用
- @Test 注解修饰的方法可以独立执行，双击选中类名，右键 RunAs“JUnit Test”
- 空白处右键 RunAs“JUnit Test”，如果一个方法直接执行，如果多个方法乱序执行。因为单元测试方法之间是没有关系的，没有先后顺序的
- jUnit 包 eclipse 都支持，无需准备，直接导入即可

1.1.5 单元测试和 main 有什么区别

main 函数是程序的唯一入口，一个类只能有一个 main 函数，开发时没问题，唯一就不会引起歧义争抢，但对于测试就不好用，我有多个独立方法需要执行，这时 main 就显得无能为力，jUnit 单元测试就有了永无之地，注意也只用于测试哦。

简单的说：main 只能运行一个方法，jUnit 可以支持多个方法。

1.2 基础概念

1.2.1 关键字

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World.");
    }
}
```

上面的紫色字什么意思呢？java 中把它称为关键字。

在 java 语言中已经被赋予特定意义的一些单词。关键字不能被用作标识符。例如：古代贫民的名字不能和皇帝的名讳相同，相同就要杀头的哦。

【自熟练】下面的关键字很多，对于初学者这么多怎么记啊，根本不用记，后面这些内容，我们要写数百遍，数千遍，甚至有些我都写了数万遍了。你还用死记吗？

关键字的定义和特点				
定义：被Java语言赋予了特殊含义的单词 特点：关键字中所有字母都为小写				50+
用于定义数据类型的关键字				
class	interface	byte	short	int
long	float	double	char	boolean
void				
用于定义数据类型值的关键字				
true	false	null		
用于定义流程控制的关键字				
if	else	switch	case	default
while	do	for	break	continue
return				

用于定义访问权限修饰符的关键字				
private	protected	public		
用于定义类，函数，变量修饰符的关键字				
abstract	final	static	synchronized	
用于定义类与类之间关系的关键字				
extends	implements			
用于定义建立实例及引用实例，判断实例的关键字				
new	this	super	instanceof	
用于异常处理的关键字				
try	catch	finally	throw	throws
用于包的关键字				
package	import			
其他修饰符关键字				
native	strictfp	transient	volatile	assert

特殊关键字:

- 别忘了 java 虽没有强调其是关键字，但也不许使用
- Test.java 不能使用，否则无法使用 JUnit 的 Test
- native 即 JNI，Java 平台有个用户和本地 C 代码进行互操作的 API，称为 Java Native Interface（Java 本地接口）
- strictfp，即 strict float point（精确浮点）
- transient 当串行化某个对象时，如果该对象的某个变量是 transient，那么这个变量不会被串行化进去。
- volatile，多线程并发，变量可见
- assert，断言是为了方便调试程序，并不是发布程序的组成部分。默认情况下，JVM 是关闭断言的

1.2.2 标识符

关键字是 java 定好的名字，这些名字我们不能重复的，它们是公用的。那既然可以公用，那有没有私用的呢？当然有，很多时候我们要自己起名字，比如我们之前创建 HelloWorld.java 文件时类名是不是就是我们自己起的啊。

那这个名字，java 把它称为什么呢？在 Java 中，我们需要标识代码的很多元素，包括类名、方法、属性、变量、包名等。我们起的名称就称为标识符。

标识符不是随便写的，它必须遵循以下规则：

- 标识符可以由字母、数字、下划线（_）、美元符（$）组成，但不能包含 @、%、空格等其它特殊字符
- 不能以数字开头
- 标识符是严格区分大小写的
- 标识符的命名最好能反映出其作用，做到见名知意

1.2.3 变量

在 JAVA 中，有的数据值是不固定的，总在变，我们还需要记录这些值，我们可以把这些值理解为变量。

格式：变量类型 变量名 = 变量值;

```
String name = "tony";
int count;                //声明 int 类型的变量，int 默认值为 0
count = 18;               //再次设置 cout 的值，这个值变了
```

注意：

- 变量名必须是一个有效的标识符
- 变量名不可以使用 java 关键字
- 局部变量名不能重复

1.2.4 常量

常量可以理解成一种特殊的变量，它的值被设定后，在程序运行过程中不允许被改变。

格式：final 常量名 = 值;

```
final double PI = 3.14;   //定义圆周率，统一引用
final String SYSTEM_NAME = "京淘电商平台";   //一个软件系统一个名称
static final ObjectMapper MAPPER = new ObjectMapper(); //json 转换类
```

常量名习惯使用全大写字符。

程序中使用常量可以提高代码的可维护性。例如，在项目开发时，我们需要指定系统的

名称，此时可以定义一个常量 SYSTEM_NAME，赋值为 "京淘电商平台"，在需要指定系统名称的地方直接调用此常量即可，避免了由于用户的不规范赋值导致程序出错的情况，而且如果程序有几十处引用了，只需改这一处，其他引用的地方自然就被改了。

1.2.5 驼峰规则

命名规则，变量首字母小写，多个单词第二个之后首字母大写，java 中称为驼峰规则。例如：

```
String filaName = "tony.mp4";
String extName = "mp4";
String toUpperCase = "HELLO";
String toLowerCase = "hello";
```

1.2.6 评论

【死记】见名知意：作为一个合格的程序员 java 中起各种名，如类名、包名、变量名、方法名、参数名、返回值名，所有的一切命名中，所起的标识符，都要尽量见名知意，不然会被同行耻笑的哦。

1.3 拼串

1.3.1 +连接符

```
@Test
public void testString() {
    int count = 100000;
    String s = "";
    long start = System.currentTimeMillis();
    for(int i=0;i<count;i++) {
        s += "a";
    }
    long end = System.currentTimeMillis();

    System.out.println("String 执行时间：" + (end-start));
    System.out.println(s.length());
}
```

1.3.2 Concat

```
@Test
public void testConcatString() {
    int count = 100000;
    String s = "";
```

```
        long start = System.currentTimeMillis();
        for(int i=0;i<count;i++) {
            s.concat("a");
        }
        long end = System.currentTimeMillis();

        System.out.println("String 执行时间: " + (end-start));
        System.out.println(s.length());
    }
```

1.3.3 StringBuffer

```
    @Test
    public void testStringBuffer() {
        int count = 100000;
        StringBuffer sb = new StringBuffer();
        long start = System.currentTimeMillis();
        for(int i=0; i<count; i++) {
            sb.append("a");
        }
        long end = System.currentTimeMillis();

        System.out.println("StringBuffer 执行时间: " + (end-start));
        System.out.println(sb.length());
    }
```

1.3.4 StringBuilder

```
    @Test
    public void testStringBuilder() {
        int count = 100000;
        StringBuilder sb = new StringBuilder();
        long start = System.currentTimeMillis();
        for(int i=0; i<count; i++) {
            sb.append("a");
        }
        long end = System.currentTimeMillis();

        System.out.println("StringBuilder 执行时间: " + (end-start));
        System.out.println(sb.length());
    }
```

1.3.5 评论

【熟练】拼串少时使用连接符最简洁,最易懂,性能最低;concat 用的少;StringBuffer 线程安全性能适中; StringBuilder 线程非安全性能最高。

当你不懂线程安全为何物时，请使用 StringBuffer。

1.4 运算符

1.4.1 常见运算符

算术运算符	+ - * /	基本运算
	%	取余数，求模，算整除
	++ --	自增 自减
比较运算符	==	相等比较
	！=	不等比较
连接运算符	+	字符串连接 “abc”+”xyz”=”abcxyz”
逻辑运算符	&&　　&	逻辑与(短路与)，两边同为真结果才为真
	‖　　|	逻辑或(短路或)，两边只要有一个真结果就是真
	!	非，非真是假，非假是真
三元运算符	c ? x : y	三项运算 c?x:y c 是真取返回 x，是假返回 y
赋值运算符	=	赋值运算
	+=、-=、*=、/=	复合的赋值运算 a+=2; 等价于 a=a+2 写法简洁
优先执行	( )小括号	小括号中的先执行 x+y+z = 按顺序相加 x+y+z x+(y+z)= 优先括号内的先执行 y+z+x

1.4.2 取余

取余操作%很多地方都会使用，特别 redis 分布式缓存中，就使用 hash 一致性算法，而其算法核心就是取余。

```
public static void main(String[] args) {
    int num = 10;
    if(num%2 == 0) {
        System.out.println("这是偶数: " + num + " 余数是: "+ num%2);
    }else {
        System.out.println("这是奇数: " + num + " 余数是: "+ num%2);
    }
}
```

1.4.3 自增自减

口诀：位置前后，代码执行顺序就不同。

```
    public static void main(String[] args) {
        int count = 0;
        System.out.println(count++); //先打后加 0
        System.out.println(++count); //先加后打 2
        System.out.println(count--); //先打后减 2
        System.out.println(count);   //减过的值 1
    }
```

1.4.4 三元运算符

也称为三目运算符，是 java 世界里唯一的一个，宝贝哦，不用多重 if 判断，代码简洁，但实际用的不多。

```
    public static void main(String[] args) {
        int x = 100;
        int y = 200;

        int z = x>y?x:y;
        System.out.print("条件为: "+(x>y));
        System.out.println(z);

        int w = x<y?x:y;
        System.out.print("条件为: "+(x<y));
        System.out.println(w);
    }
```

1.4.5 三个数求最大值

```
package cn.tedu.hello;

//三目运算符
public class Ternary {
    public static void main(String[] args) {
        // 求三个值最大值
        int a = 100;
        int b = 200;
        int c = 300;

        // 结构更好，结构清晰，代码稍多（推荐）
        int max = (a > b) ? a : b;
        max = (c > max) ? c : max;
        System.out.println("最大值是: " + max);

        // 简洁，可读性差
        int m = (a > ((b > c) ? b : c)) ? a : ((b > c) ? b : c);
```

```
        System.out.println("最大值是: " + m);
    }
}
```

变成方法调用:

```
package cn.tedu.hello;

//利用三目表达式来实现求最大值
public class TestMax {
    public static void main(String[] args) {
        int x = 100;
        int y = 20;

        //调用方法，创建对象，通过对象来调用方法
        //类 对象名= new 类();
        TestMax obj = new TestMax();
        int max = obj.max(x, y);     //实参
        System.out.println("最大值是: "+max);
    }

    //如何写一个方法
    public int max(int x, int y) {       //形参
        int r = (x>y) ? x : y;
        return r;                 //写代码要简单易读
    }
}
```

1.4.6 评论

【了解】初学者常问一个问题，我数学不好，能不能学计算机啊？告诉你们一个秘密，实际开发中你只要会+、-、*、/、% 就基本够用。