

方法 (Method)

@M了个J
李明杰

<https://github.com/CoderMJLee>

<http://cnblogs.com/mjios>



实力IT教育 www.520it.com

码拉松



方法 (Method)

■ Java 中的方法，其实就是其他编程语言中的函数 (Function)

■ 方法的书写格式

```
修饰符 返回值类型 方法名(参数列表) {  
    方法体  
}
```

可变参数 (Variable Argument)

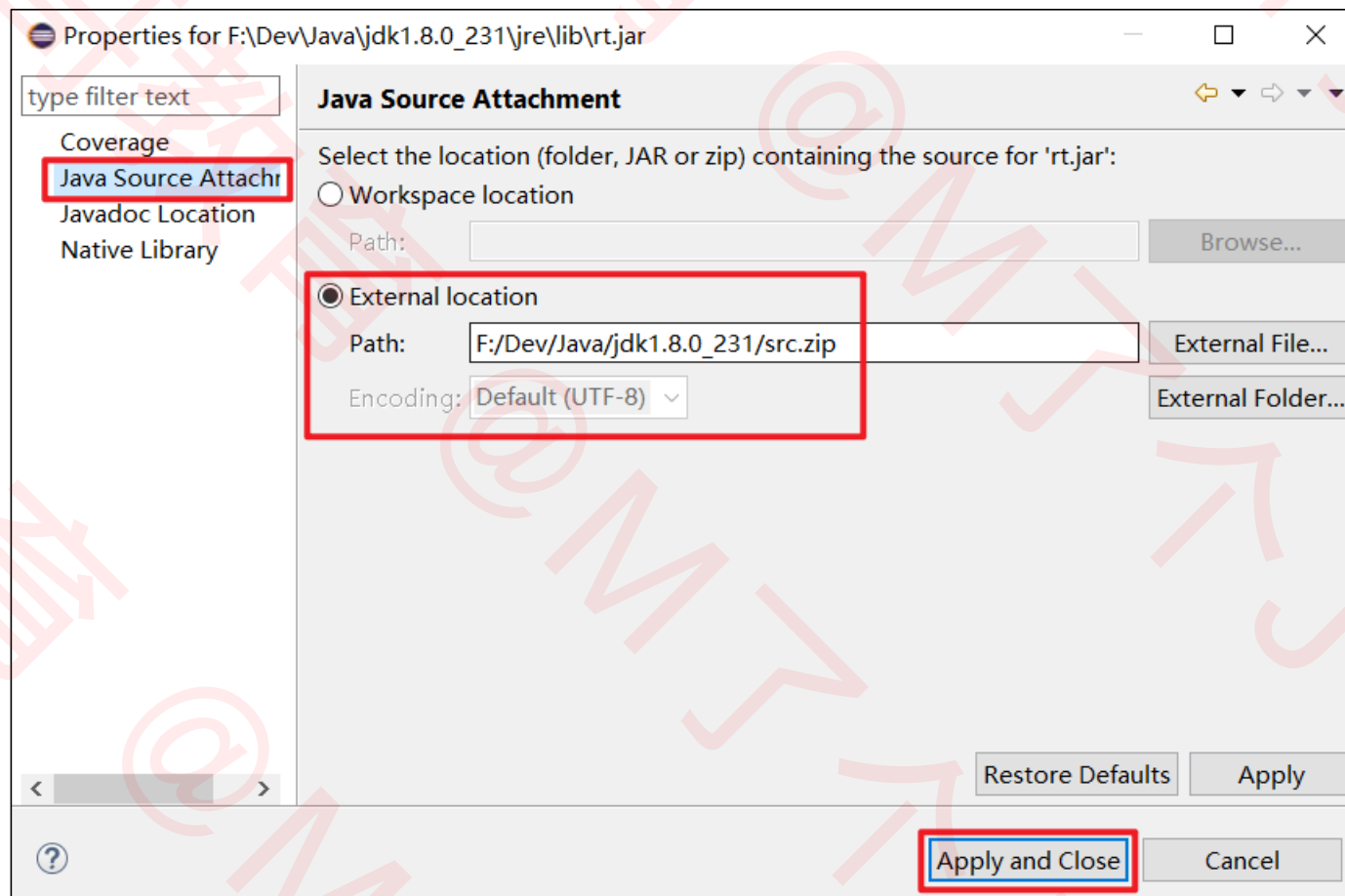
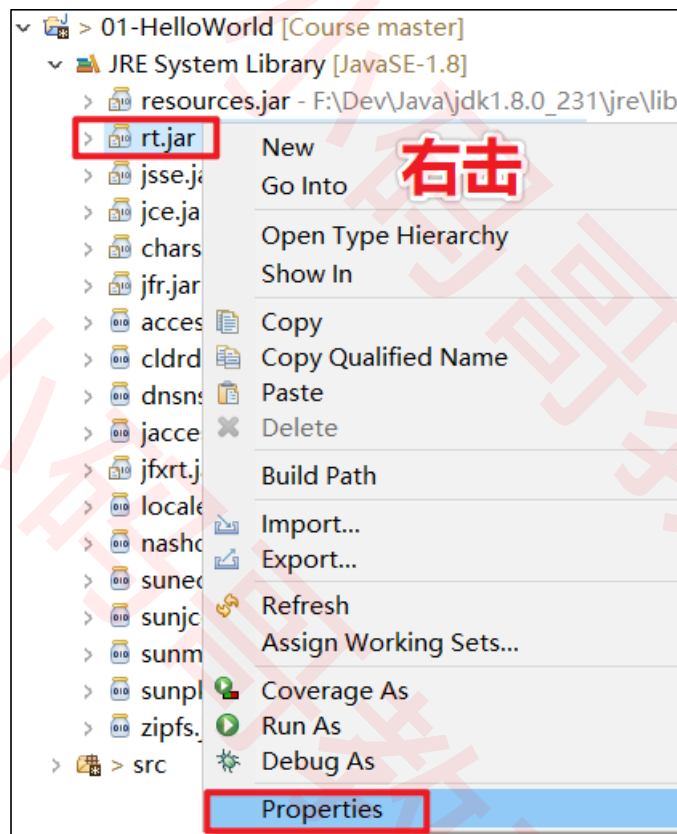
```
public static void main(String[] args) {  
    sum(10, 20, 30);  
}  
  
public static int sum(int... numbers) {  
    int result = 0;  
    for (int i : numbers) {  
        result += i;  
    }  
    return result;  
}
```

- 可变参数必须是方法中的最后一个参数
- 举例：JDK自带的 `System.out.printf` 方法使用了可变参数
- 格式字符串参考API文档的 `java.util.Formatter` 类

```
public PrintStream printf(String format, Object ... args) {  
    return format(format, args);  
}
```

```
String name = "Jack";  
int age = 20;  
String phone = "13866668888";  
System.out.printf("name is %s, age is %d, phone is %s\n", name, age, phone);
```

Eclipse中关联JDK源码



参数传递

- 基本类型作为参数是值传递
- 基本类型作为返回值，返回的是值
- 引用类型作为参数是引用传递（地址传递）
- 引用类型作为返回值，返回的是引用（地址）

方法签名 (Method Signature)

- 方法签名由 2 部分组成：方法名、参数类型

- 下面方法的方法签名是：sum(int, long, double)

```
public static double sum(int i, long l, double d) {  
    return i + l + d;  
}
```

- 在同一个类中，不能定义 2 个方法签名一样的方法

方法的重载 (Overload)

- Java 的方法支持重载：方法名相同，方法签名不同

- 参数个数不同

- 参数类型不同

- 重载与返回值类型、参数名称无关

栈帧 (Frame)

- 栈帧随着方法的调用而创建，随着方法结束而销毁，存储了方法的局部变量信息

```
public static void main(String[] args) {  
    test1(10);  
    test2(20);  
}  
public static void test1(int v) {}  
public static void test2(int v) {  
    test3(30);  
}  
public static void test3(int v) {}
```

栈空间

test3
v = 30

test2
v = 20

main
args

递归调用

```
public static void main(String[] args) {  
    sum(4);  
}  
public static int sum(int n) {  
    if (n <= 1) return n;  
    return n + sum(n - 1);  
}
```

- 如果递归调用没有终止，将会一直消耗栈空间
- 最终导致栈内存溢出 (Stack Overflow)
- 所以必需要有一个明确的结束递归的条件
- 也叫作边界条件、递归基

栈空间

