



# Java 核心技术(高阶)

第一章 Java语法糖

第三节 语法糖(2)

华东师范大学 陈良育



# 不定项参数(1)

- 普通函数的形参列表是固定个数/类型/顺序
- JDK 5提供了不定项参数(可变参数)功能
  - 类型后面加3个点, 如int.../double.../String.../
  - 可变参数, 本质上是一个数组

```
print();  
print("aaa");  
print("aaa", "bbb");  
print("aaa", "bbb", "ccc");
```

```
public static void print(String... args) {  
    System.out.println(args.length);  
    for (String arg : args) {  
        System.out.println(arg);  
    }  
}
```



# 不定项参数(2)

- 不定项参数(可变参数)

- 一个方法只能有一个不定项参数，且必须位于参数列表的最后
- 重载的优先级规则1：固定参数的方法，比可变参数优先级更高
- 重载的优先级规则2：调用语句，同时与两个带可变参数的方法匹配，则报错





# 静态导入(1)

- import 导入程序所需要的类
- import static 导入一个类的静态方法和静态变量(JDK5引入)

```
import static java.lang.Math.pow;
import static java.lang.Math.sqrt;
import static java.lang.System.*;
import static java.lang.System.out;

public class ImportStaticTest {
    public static void importMath() {
        int a=3, b=4, c=0;
        c = (int) sqrt(pow(a,2)+pow(b,2));
        c = (int) Math.sqrt(Math.pow(a,2)+Math.pow(b,2));
        out.println("c is " + c);
        System.out.println("c is " + c);
    }
}
```



# 静态导入(2)

- import static 导入一个类的静态方法和静态变量
  - 少使用\*通配符，不滥用，最好具体到静态变量或方法
  - 静态方法名具有明确特征，如有重名，需要补充类名

```
import static java.lang.Double.*;
import static java.lang.Math.*;
import static java.lang.Integer.*;
import static java.text.NumberFormat.*;

import java.text.NumberFormat;
```

```
public class BadImportStaticTest {

    public static void main(String[] args) {
        double s = PI * pow(parseDouble(args[0]),2);
        NumberFormat nf = getInstance();
        nf.setMaximumFractionDigits(parseInt(args[1]));
        System.out.println("圆面积是: " + nf.format(s));
        //System.out.println("最大值: " + MAX_VALUE); //error
        System.out.println("最大值: " + Integer.MAX_VALUE);
    }
}
```

# 总结



- 不定项参数(5)：注意方法重载的优先级，建议少用
- 静态导入(5)：有大量静态方法调用时使用，注意“\*”不滥用



谢谢!