

# Java-02

## 一、Task1.

### A.代码解析

1.

```
package com.Example;
```

重点解释 **package**

要明白什么是 **package (包)** 就需要知道java程序的**组成**

java程序从大单位到小由**项目**，**包**和**类**组成 (类以下不在此处作过多阐述)

**含义** 这也就意味着 **包 (package)** :

- 是一种用于组织集合**相关类** 的命名空间
- 扮演着**文件夹**的作用

**作用** : 所以此语句声明了 **包** 的名称, 避免 **默认包** 导致的命名冲突

2.

```
import com.Example.tool.Print;
```

重点解释 **import**

需要使用其他包中的 **print** 方法, 原本应当使用完整的 **全限定名** com.Example.tool.Print

为了简化书写, 便在main函数前使用 **import** 关键词声明从而使得后续只需使用 **print** 便可调用

3.

```
public class HelloWorld {  
    public static void main(String[] args){  
        Test.test();  
    }  
}
```

**public//static//void//string//args**

此部分

- 声明了 **类** 的名称 **HelloWorld**
- 建立了公开可访问的 **main** 函数 (**main**函数是Java程序的入口起点)
- 在 **main** 函数中调用了 **Test** 类中的 **test** 方法

**public** 使该 **类** 和 **main** 函数可访问

**static** 规定该 **方法** 为 **静态** 与之相对的是 **非静态** 或者 **实例** 方法 (需要有实例才能调用)

**void** 是不需要返回值的 **返回类型**

**string** 是 **字符** 参数类型

**String[] args** 是一个字符串数组, 用于接收命令行传递的参数

4.

```
class Test{
    public static void test(){
        Print.print("Hello World");
    }
}
```

定义了Test 类中的 test 方法为调用 Print 类中的 print 方法并向其输入("Hello World")

## B.单文件Java程序基本结构

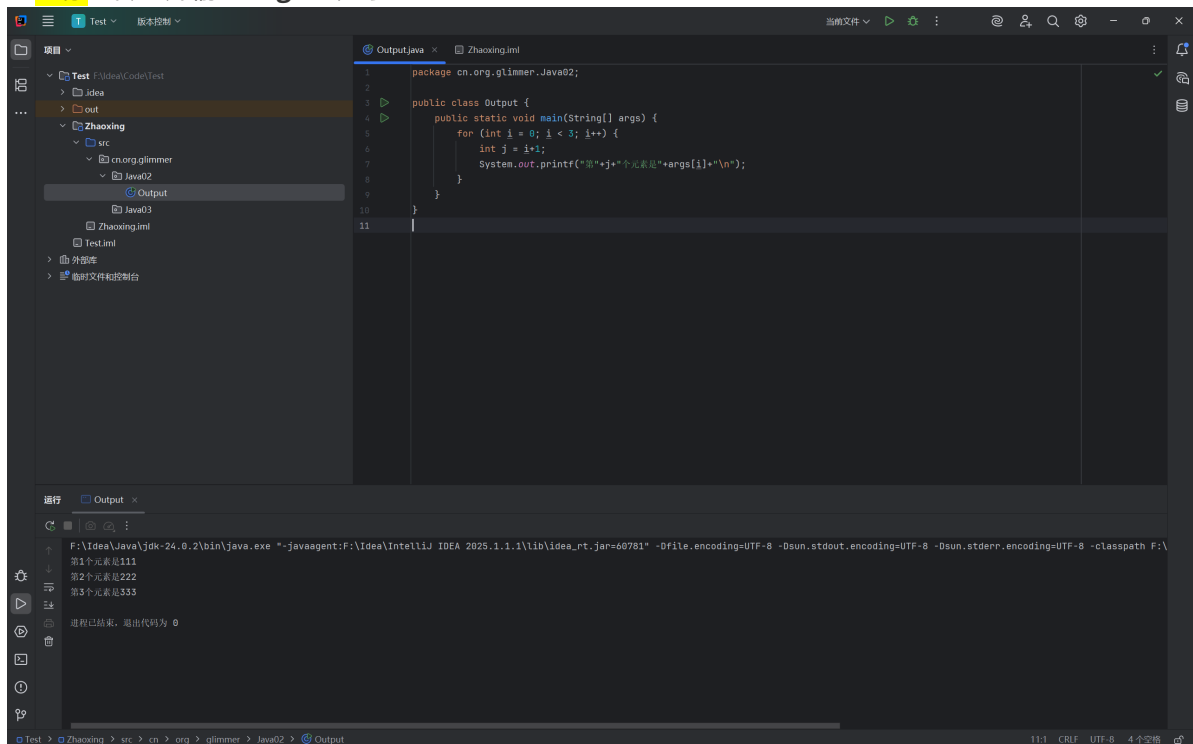
```
package packagename;//包声明
import packagename.classname.name;//引用方法声明
public class classname//类声明
{
    public static void main(String[] args)
    //建立main函数，返回类型为不需要返回值
    {

    }

}
```

## C.运行演示

**目标：** 传入并输出 args 的元素



我选择了通过修改程序实参的方式为数组 args 输入了变量，详细步骤如下

**Alt+4 打开 运行 → 修改运行配置 → 程序实参 → 应用 → 确定**

详细代码如下

```
package cn.org.glimmer.Java02;

public class Output {
    public static void main(String[] args) {
        for (int i = 0; i < 3; i++) {
            int j = i+1;
            System.out.printf("第"+j+"个元素是"+args[i]+"\\n");
        }
    }
}
```

## 二、Task2.完成Print.class的编写

已上传至仓库中 ([Print.class](#)) 内容如下

```
package com.Example.tool;

public class Print {
    public static void print(String str){

        System.out.println(str);

    }
}
```