—

1.

`package com.Example;`是一个Java包声明，它表示当前Java文件位于com.Example这个包中。 `package`是关键字，作为声明提示，`com.Example`是包的名称，采用反向域名的命名约定，以避免命名冲突。

包是Java提供的一种用于对相关的类与接口进行分组、组织的命名空间机制，类似文件夹。

包的核心作用是组织与分类，使代码结构清晰，易于管理和查找。 同一个目录下不能存在相同文件名称，但实际应用难免重名。 包最大的作用是避免命名冲突。在不同的包中，可以存在相同名称的类而不会发生冲突。 包还有访问控制和数据封装的作用。

2

`import com.Example.tool.Print;`是一个Java的import（导入）语句，作用是告诉Java编译器，在当前类中将要使用`com.Example.tool`包下的`Print`类。 `import`是关键字，用于导入其他包中的类或接口。 `com.Example.tool`是包的完全限定名，表示类的组织路径。 `print`是要导入的具体类的名称。

3

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

这段是JAVA程序，包含一个名为`HelloWorld`的主类，启动后调用另一个名为`Test`的类中的`test()`方法。

`public class HelloWorld` `public`：访问修饰符，表示这个类是公开的，可以被其他任何类访问。 `class`：Java关键字，用于声明一个类。 `HelloWorld`：类的名称，遵循大驼峰命名规范。

`public static void main(String[] args)` 这是Java程序的主方法（入口点），是程序开始执行的地方：`public`：方法公开，可以被JVM调用。 `static`：静态方法，无需创建类实例即可调用。 `void`：方法没有返回值。 `main`：方法名称，固定为main。 `(String[] args)`：方法参数，字符串数组，用于接收命令行参数。

`Test.test();` 这是方法体中的唯一语句，意思是：调用`Test`类的静态方法`test()`。 `Test`：另一个类的类名。 `.`：成员访问运算符。 `test()`：Test类中的一个静态方法。

`{}`：类体、方法体的开始和结束

main函数是Java应用程序的入口点（Entry Point） 和起点。 main的作用有： 程序入口 接收命令行参数 创建对象和启动程序

4

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

这是一个名为Test的Java类，其中包含一个静态方法test(),该方法的功能是调用另一个类Print中的print方法来输出"Hello World"。

class Test{...}是类声明。 class：Java关键字，用于声明一个类。 Test：类的名称，遵循Java命名规范（首字母大写）。

public static void test(){ ... }是方法声明。 public：访问修饰符，表示该方法可以被任何其他类访问。 static：静态方法，属于类本身而不是实例。 void：返回类型，表示该方法不返回任何值。 test()：方法名称，遵循小驼峰命名规范。 ()：参数列表为空，表示该方法不需要参数。

Print.print("Hello World");是方法体,意思是"调用Print类的print方法，并传递字符串'Hello World'作为参数" Print：另一个类的名称。 print：Print类中的一个方法。 ("Hello World")：方法参数，一个字符串字面量。

5总结

基本结构： 1.包声明 2.一个public类（类名与文件名一致） 3.main方法（正确的签名） 4.可执行语句

二

改后代码如下

```
package com.Example;
/*-------------------------------------*/
import com.Example.tool.Print;
/*-------------------------------------*/
public class HelloWorld {
        public static void main(String[] args){
                if (args.length>=3){
                    Print.print(args[0]);
                    Print.print(args[1]);
                    Print.print(args[2]);
                }
Test.test();
}
}
/*-------------------------------------*/
class Test{
    public static void test(){
        Print.print("Hello World");
    }
}
```

因运行需要Print方法，也将Print.java编写如下

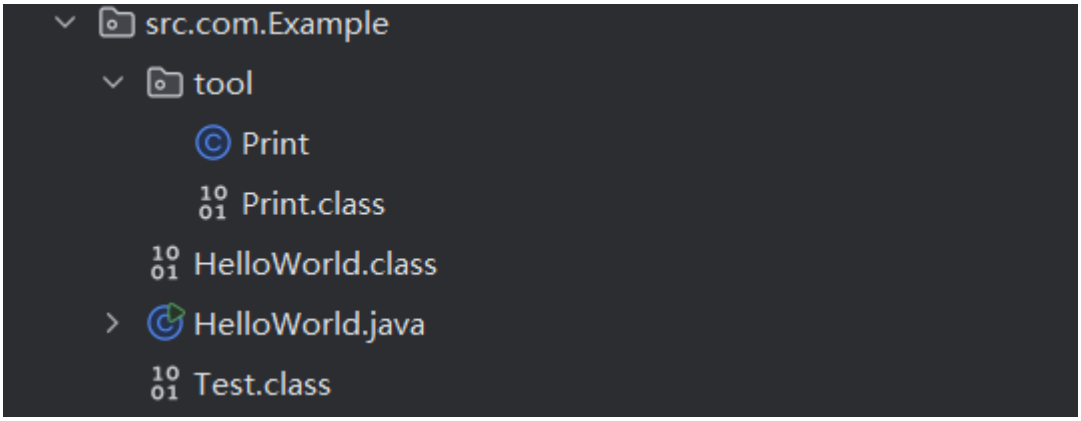
```java
package com.Example.tool;
public class Print {
    public static void print(String S){
        System.out.println(S);
    }
}
```

运行结果如下

```
com\Example\HelloWorld.java:13: 错误: 找不到符号
                Print.success();
                      ^
  符号:   方法 success()
  位置: 类 Print
5 个错误
PS D:\IT\VS\Cuse\src> javac -d . com/Example/HelloWorld.java
PS D:\IT\VS\Cuse\src> java com/Example/HelloWorld.java 111 222 333
111
222
333
Hello World
PS D:\IT\VS\Cuse\src>
```

三

项目在前面已创建并运行



Print.java已在前面写出，项目运行已在前面展示。