

Task1.单文件代码结构

```
package com.Example;

/*-----*/
import com.Example.tool.Print;
/*-----*/

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

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

1.包 (package) 声明：

- 表示当前类文件属于 `com.Example` 包
- 包的作用：
 1. 用于组织和管理类文件，避免类名冲突
 2. 提高代码的可维护性和可读性
 3. 可以控制访问权限（配合 `public`、`protected`、`default`、`private`）

2.导入语句：

- 告诉编译器我们要使用 `com.Example.tool` 包中的 `Print` 类
- 作用：不用写完整语句以使用其他包中的类

3.类文件声明与main函数：

- 声明了该类文件为public的访问权限，类名为HelloWorld
- main函数提供了程序的主入口

4.辅助类：

- 定义了一个静态方法以便main函数引用

`args[]`参数为

Edit Configuration Settings

Name: ☐ Store as project file

Run on: 🏠 Local machine ▼ [Manage targets...](#)

Run configurations may be executed locally or on a target: for example in a Docker Container or on a remote host using SSH.

Build and run Modify options ▼ ⌘M

java 17 SDK of 'Glimmer' ▼

CLI arguments to your application. ⌘R

Working directory:

Environment variables:

Separate variables with semicolon: VAR=value; VAR1=value1

Open run/debug tool window when started ×

Code Coverage Modify ▼

Packages and classes to include in coverage data

—

+

?

Cancel

Apply

Run

Task2.多文件Java项目

详见uestc.glimmerjava2.com.Example