Java02-程序运行

Task.01 单文件代码结构

```java
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");
    }
}
```

Q1：*四个部分从上至下的理解*：

- **包** (package)：
  包就是 Java 程序中类的"文件夹"，用来对类进行分组和管理。其能**避免类名冲突**（同名类放在不同的包中不会冲突），方便**组织和管理项目结构。**
  上述`package com.Example;`的意思是：这个类在`com/Example`文件夹里。
  package的声明必须放在最前面！！！
- **import**：

  告诉编译器：需要用到某个包里的类，这样在代码里就可以直接用类名，而不用写全路径。
  上述`import com.Example.tool.Print;`表示：以后可以直接用`Print`这个名字，不用写`com.Example.tool.Print`。
  如果不写 import，要用类时必须写完整路径，如`com.Example.tool.Print.print("Hello");`
- **main函数**：
  main函数就是Java程序的入口，程序启动时，JVM 会先执行main函数。其形式必须是：

  ```java
  public static void main(String[] args)
  ```

  其中，`String[] args`是命令行参数。
- **其他类**：

  这是一个辅助类，用来组织逻辑。在一个`.java`文件中可以有多个类，但最多只有一个`public class`，并且文件名必须与这个`public class`的类名相同。

故可以总结Java程序**基本结构**如下：

```java
package 包名;
import 导入的类;

public class 类名 {
    public static void main(String[] args) {

    }
}

class 其他类 {

}
```

Q2：*更改main函数后，如何输入参数并打印出来：*

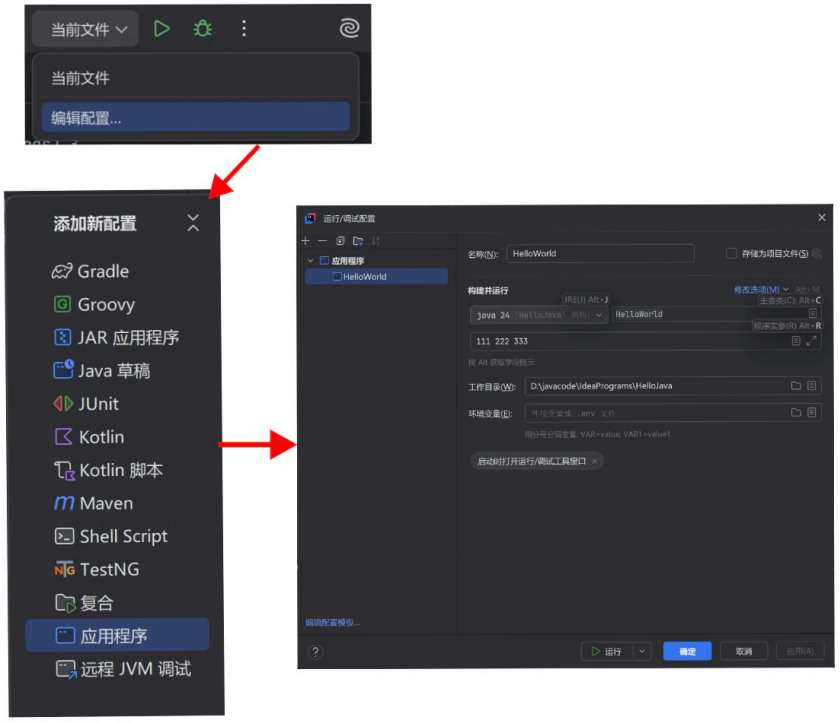
Solution 1:

代码如下：
~~（嗯这一步完完全全地依靠了gpt老师，但是能看明白是用循环和数组）~~

```java
public class HelloWorld {
    public static void main(String[] args) {

        System.out.println("接收到的参数：");
        for (int i = 0; i < args.length; i++) {
            System.out.println("参数 " + i + ": " + args[i]);
        }
    }
}
```

我选择直接从IDEA传入参数而不是用命令行，所以刚开始遇到了一些问题——直接运行的时候貌似没有办法输入参数。所以通过学习~~（仍然询问gpt）~~解决了这一问题。解决过程如下：



最终运行结果如下：

```
接收到的参数：
参数 0: 111
参数 1: 222
参数 2: 333

进程已结束，退出代码为 0
```

Solution 2:

蠢到忘记可以输入……于是自己了解学习了`Scanner`这个类。
以此题为例，Scanner的大概用法如下：

```java
import java.util.Scanner;
//导包--Scanner这个类在哪儿

Scanner sc = new Scanner(System.in);
//创建对象--要开始用Scanner这个类了，其中只有“sc”是变量名可以变动

int i = scanner.nextInt();
//接收数据

scanner.close();
//关闭scanner
```

常用方法：

方法	描述
`next()`	读取下一个单词（以空格分隔）
`nextLine()`	读取下一行文本（*用来清除缓冲区，非常重要！*）
`nextInt()`	读取下一个整数
`nextDouble()`	读取下一个双精度浮点数
`nextBoolean()`	读取下一个布尔值
`hasNext()`	检查是否还有输入
`hasNextInt()`	检查下一个输入是否为整数

代码实现和执行结果如下：

```java
package scan.demo;

import java.util.Scanner;

public class scan {
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        int n1 = sc.nextInt();
```

```
        System.out.print(n1+" ");
        int n2 = sc.nextInt();
        System.out.print(n2+" ");
        int n3 = sc.nextInt();
        System.out.print(n3+" ");
    }
}
```

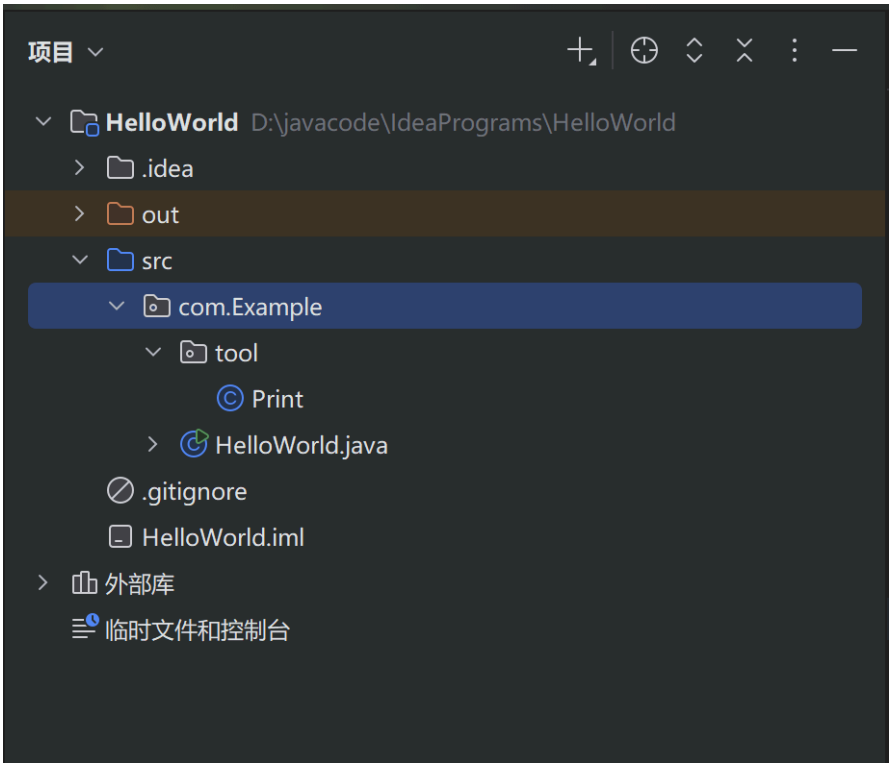
```
111 222 333
111 222 333
进程已结束，退出代码为 0
```

Task.2 多文件Java项目

Q3:

开始疑惑了很久为什么所有package是压缩在一坨的（？），然后从 *外观->压缩空的中间软件包* 大概懂了是怎么创建的。

- 该项目结构如下：



- Print.java文件中的代码，执行结果和解决思路如下：

```
package com.Example.tool;

public class Print {
    public static void print(String a) {
        System.out.println(a);
    }
}
```

```
Hello World

进程已结束，退出代码为 0
```

解决思路：（代码块部分请见注释）

```java
package com.Example;
/*-------------------------------------*/
import com.Example.tool.Print;
/*-------------------------------------*/
public class HelloWorld {
        public static void main(String[] args){
            Test.test();//调用静态方法：类名.方法名()
        }
}
/*-------------------------------------*/
class Test{
    public static void test(){
        Print.print("Hello World");
        //又调用了静态方法，但是这里没有，所以我们需要补写
        //而且题目给出的要求是在另一个package里面写
    }
}
```

补写Print：

```java
package com.Example.tool;//导包

public class Print {//由题意知类名叫Print
    public static void print(String a) {//由题意知方法名叫print，且是静态方法所以要加static
        System.out.println(a);
        //因为是字符串所以用String来定义，println输出即可
    }
}
```

*补充：静态方法的调用（实例方法的调用）

*以下内容完全来自于deepseek

`static` 的意思是“静态的”。它用于修饰类的成员（变量、方法、代码块、内部类），但它**不控制谁可以访问**，它控制的是这个成员的**归属和生命周期**。

特性	静态方法	实例方法
关键字	使用 `static`	不使用 `static`
归属	属于类本身	属于对象实例
调用方式	`类名.方法名()`	`对象名.方法名()`

特性	静态方法	实例方法
内存分配	类加载时分配	对象创建时分配
访问实例变量	❌ 不能直接访问	✅ 可以直接访问
使用 `this`	❌ 不能使用	✅ 可以使用
生命周期	与类相同	与对象相同