

JC2002 Java 编程 - 实践 4（第 5 天）

本实践的目标是获得更多关于在 Java 中定义方法和类的经验，以及使用 Maven 和 JUnit 对 Java 程序进行自动测试的经验。

1. 创建文件 **"Calculator.java"**，定义 **Calculator** 类。在该类中定义三个公共方法：**add()**、**subtract()** 和 **multiply()**。每个方法都应接受两个整数参数，并以整数形式返回相应算术运算的结果。记得在代码中添加注释。
2. 创建文件 **"RunApp.java"**，其中包含一个带有 **main()** 方法的类 **RunApp**。使用该方法创建一个计算器实例。使用扫描仪对象接收来自控制台的输入，以使用户执行计算。流程如下

- 用户输入第一个数字（随后输入）。
- 用户输入运算符 "+"、 "-" 或 "*"（后跟回车键）来决定调用计算器对象的哪个方法。提示：可以使用 `scanner.next().charAt(0)`；将用户输入的第一个字符从键盘读入字符变量（扫描仪是系统输入的扫描仪实例）。
- 用户输入第二个数字（随后输入）。
- 程序返回所选操作的结果。

请注意，程序将无限期运行（即用户可以执行多次计算），直到输入操作符 "q" 或 "Q"（表示 "退出"）为止。

3. 将应用程序转化为 Maven 项目。您可以按照幻灯片中的说明创建一个虚拟 Maven 项目。如果尚未安装 Maven，请按照幻灯片中的说明安装。

要创建项目，请输入命令 **"mvn archetype:generate"**。经过一些处理后，Maven 会问一些问题。对于前两个问题，可以使用默认值（按回车键即可）。对于 **groupId**，可以输入 **"jc2002"**；对于 **artifactId**，可以输入 **"calculator"**；对于软件包名称，可以输入 **"mypkg"**（实际上，你可以自己选择软件包名称，但在本例中，我们假设软件包名称为 **"mypkg"**）。

如果项目创建成功，新目录 **"calculator"** 和几个子目录已经创建。将文件 **"Calculator.java"** 和 **"RunApp.java"** 复制到目录 **"calculator/src/main/java/mypkg"**（如果软件包名称不同，请用实际软件包名称替换 **"mypkg"**）。您可以删除同一目录中的 **"App.java"** 文件。确保在 **"Calculator.java"** 和 **"RunApp.java"** 中定义了软件包 **mypkg**。

移至目录 "**calculator**", 打开其中的文件 "**pox.xml**". 编辑 "maven-jar-plugin" 部分
具体如下

<插件

<artifactId>maven-jar-plugin</artifactId>

<版本>3.0.2</版本

<配置

<存档

<管理

<addDefaultImplementationEntries>>true</addDefaultImplementationEntries> 添

加默认实施条目

```
<addDefaultSpecificationEntries>true</addDefaultSpecificationEntries> 添加默认规范条目
<addClasspath>true</addClasspath>
<mainClass>mypkg.RunApp</mainClass> 主类
</manifest>
</归档>
</configuration>
</plugin>
```

然后，使用 `"mvn package"` 命令生成一个包含 **Calculator** 和 **RunApp** 类的 JAR 包。尝试使用 `"java -jar target/calculator-1.0-SNAPSHOT.jar"` 命令在 **"target"** 目录下运行生成的 JAR 文件（确保这是 JAR 文件的名称）。

4. 修改目录 **"src/test/java/myorg"** 中的默认测试文件 **"AppTest.java"**，为 **计算器** 类的 **add()**、**subtract()** 和 **multiply()** 方法创建一些测试。对每个方法至少进行两次测试。包括负值、大值和其他异常输入。输入命令 `"mvn clean test"` 编译代码并运行测试。

当所有测试都通过后，修改 **add()**、**subtract()** 和 **multiply()** 方法，故意给出错误的结果。再次运行测试，确保测试失败。