**实 验 报 告**

|  |  |
| --- | --- |
| **课程名称：** | **Java语言程序设计** |
| **学生姓名：** | **葛子午** |
| **学生学号：** | **202330550381** |
| **学生专业：** | **软件工程** |
| **开课学期：** | **2024-2025学年第一学期** |

**软件学院**

**2024年10月**

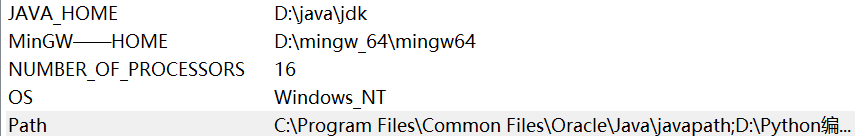
**实验一 熟悉Java 运行环境**

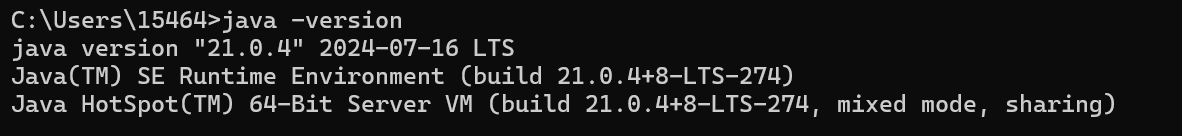
**一、实验目的**

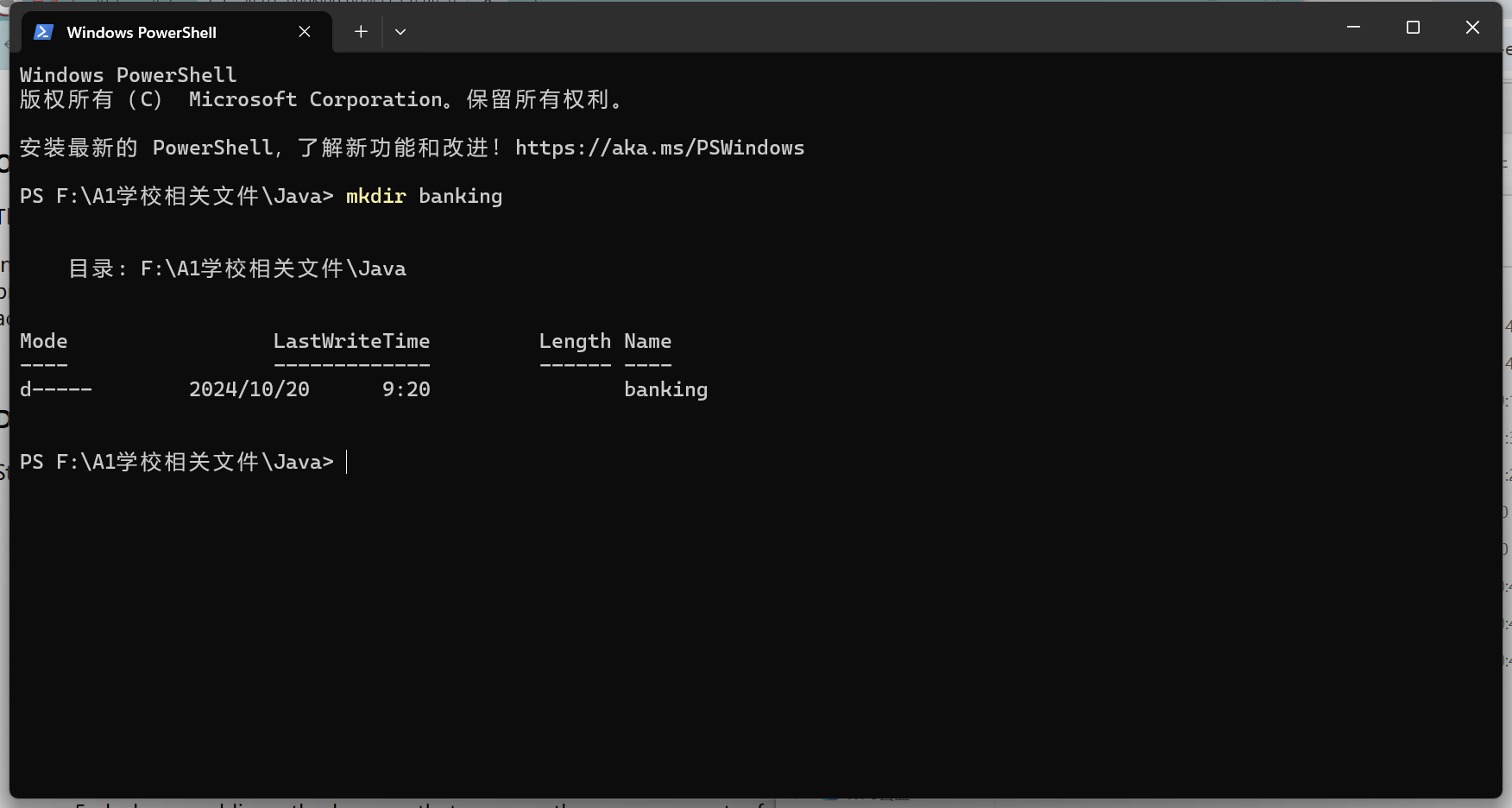
1. 安装与配置JDK环境变量；
2. 掌握Java程序的编辑、编译和运行过程，熟悉Javac和Java命令
3. 掌握Java的基本语法；
4. 使用类来封装对象的属性和方法；熟悉使用package和import语句。
5. **实验内容及结果**

**STEP1**

配置环境、查看版本







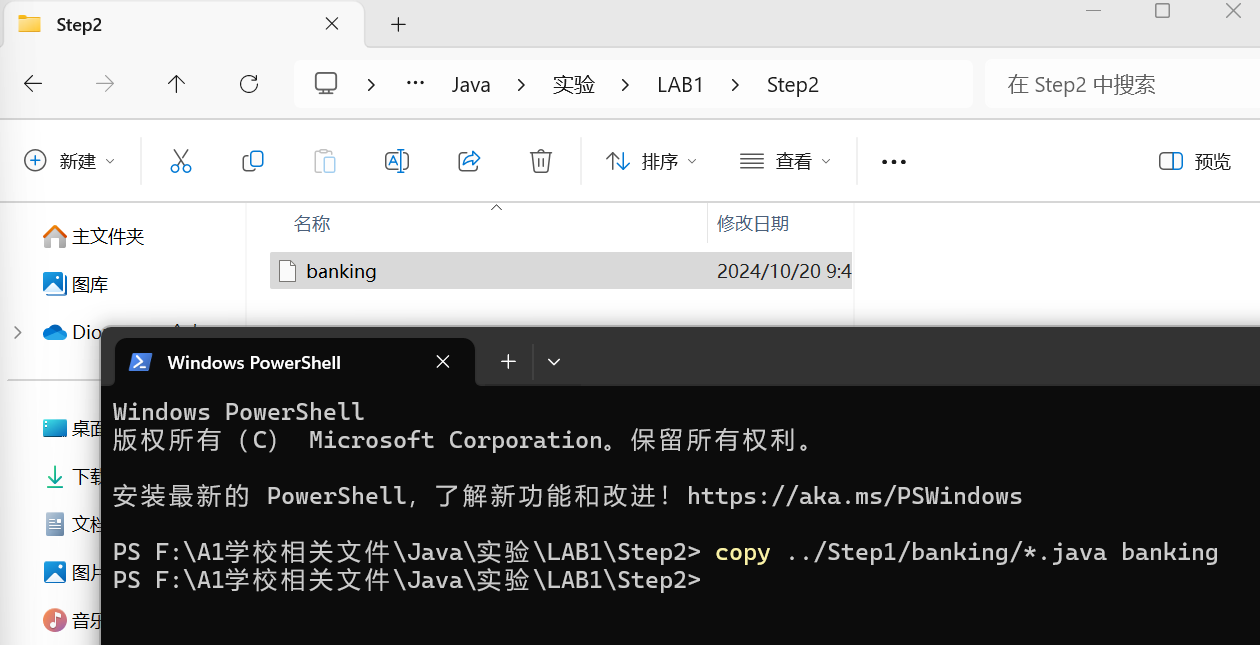
运行结果：



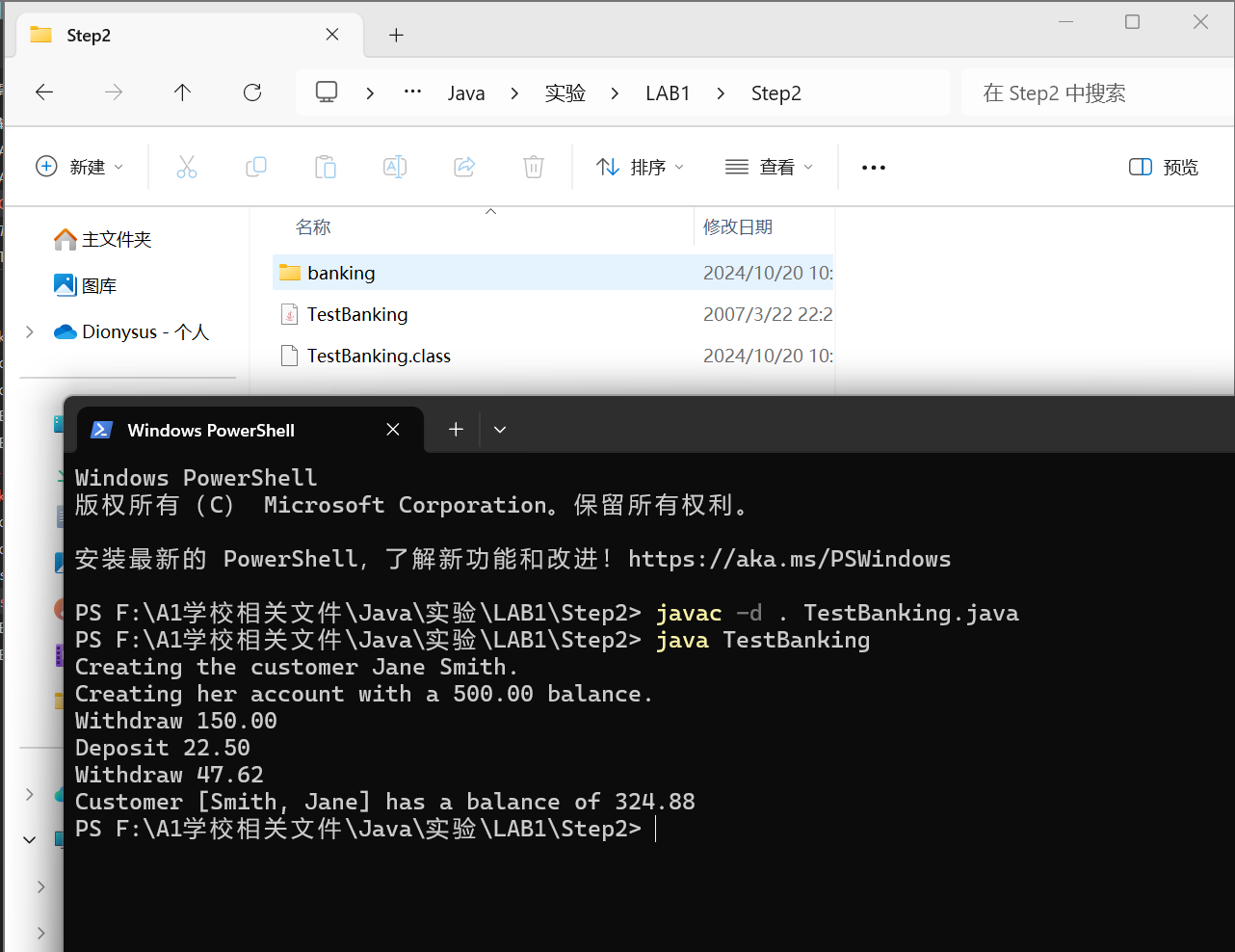
代码：



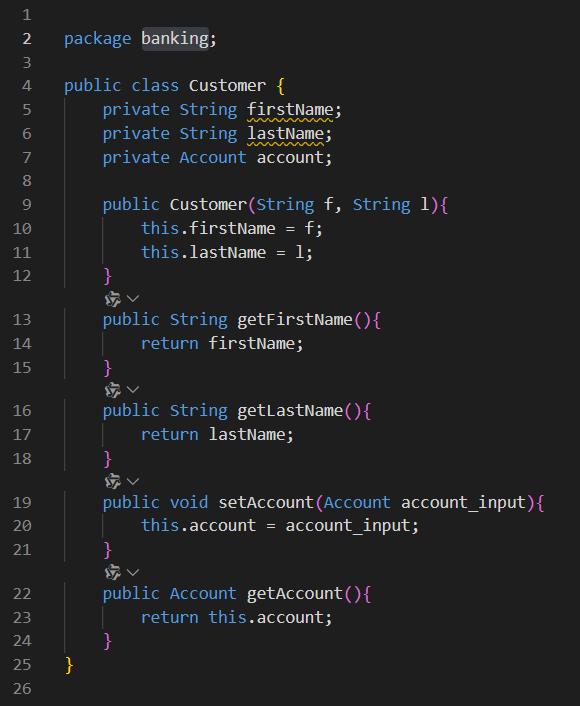
**STEP2**



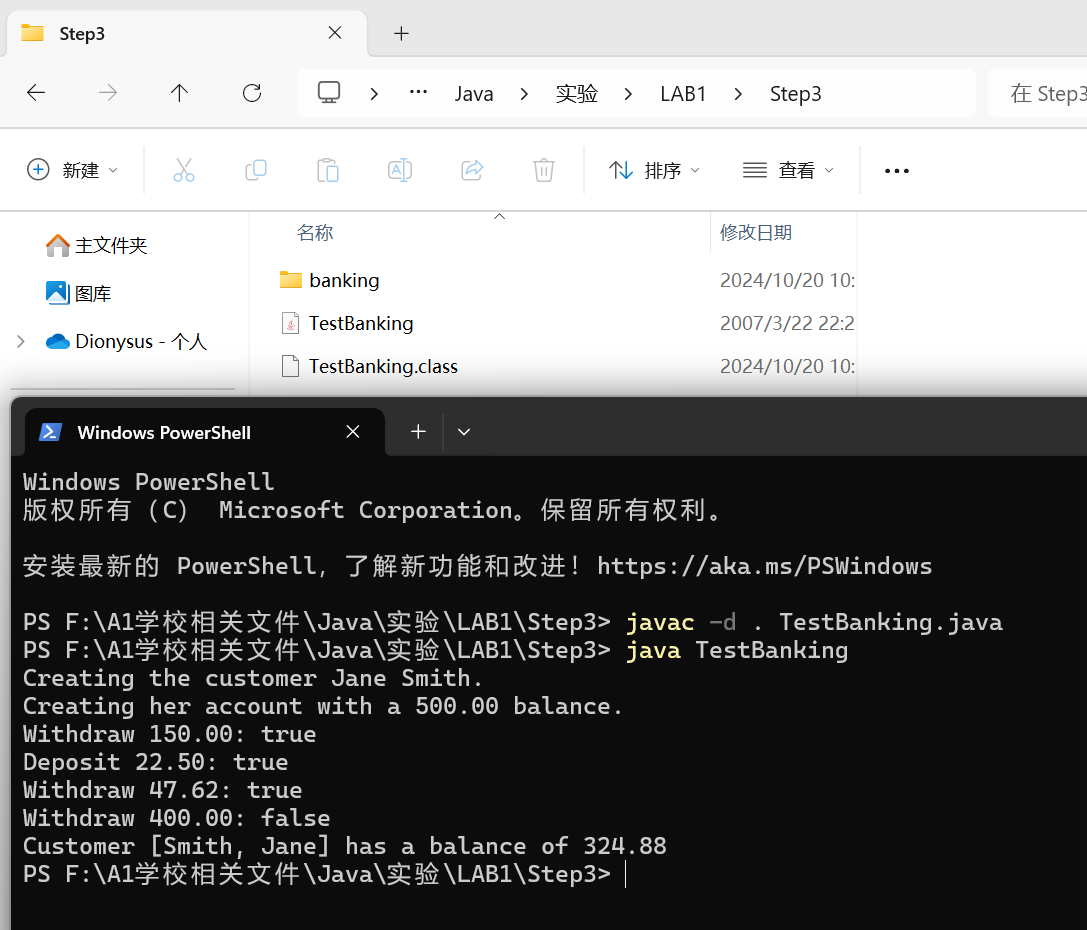
运行结果



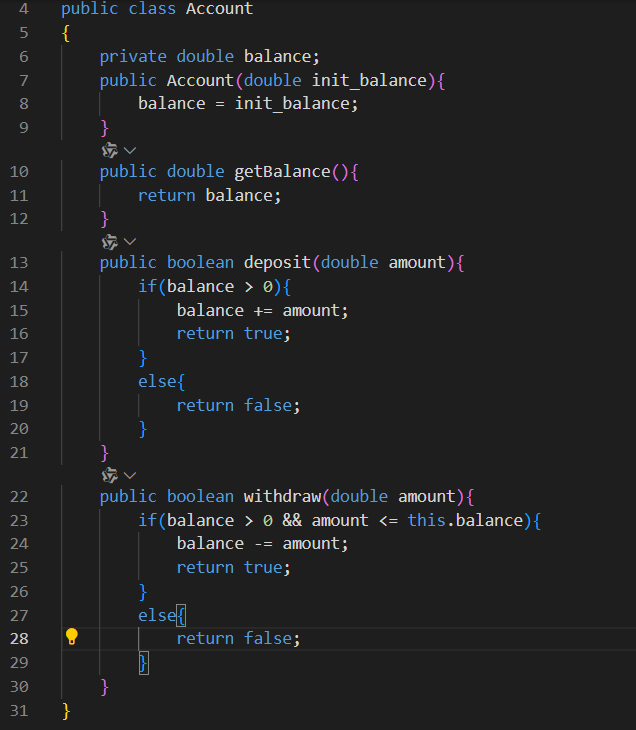
代码



**STEP3**

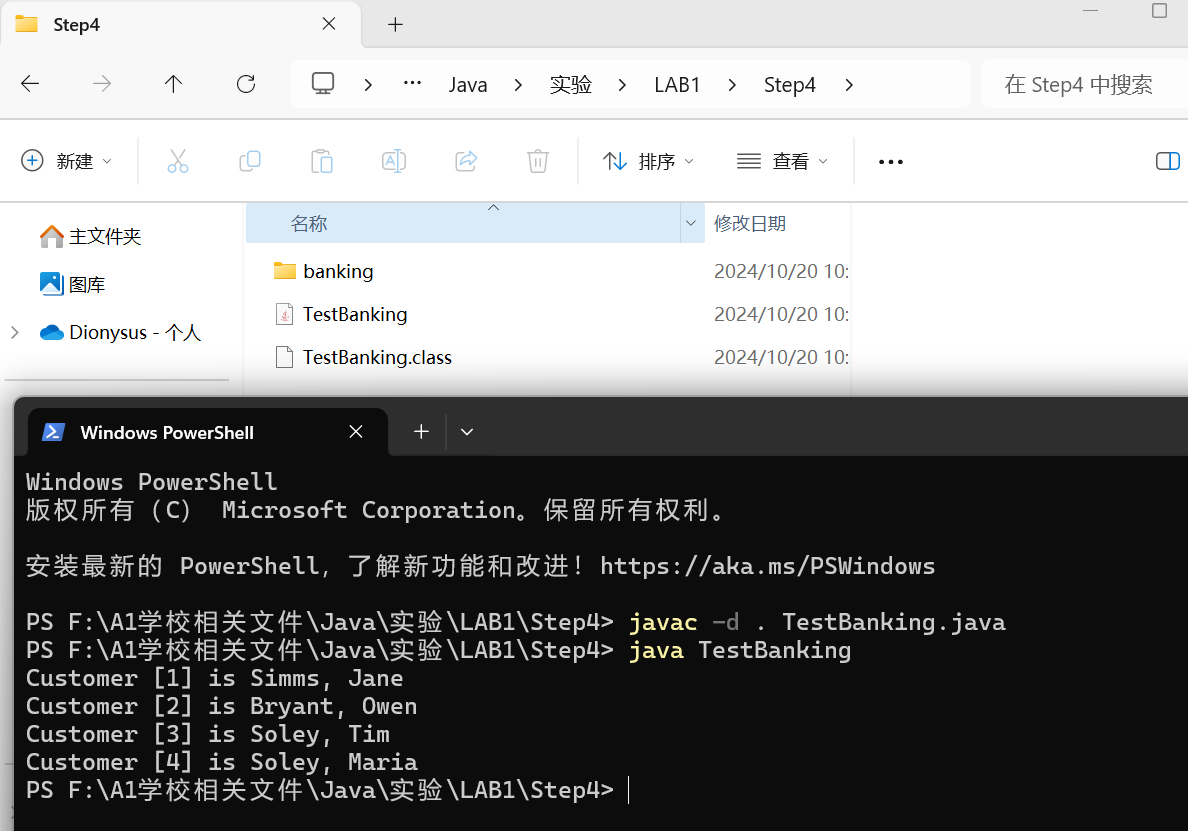


代码

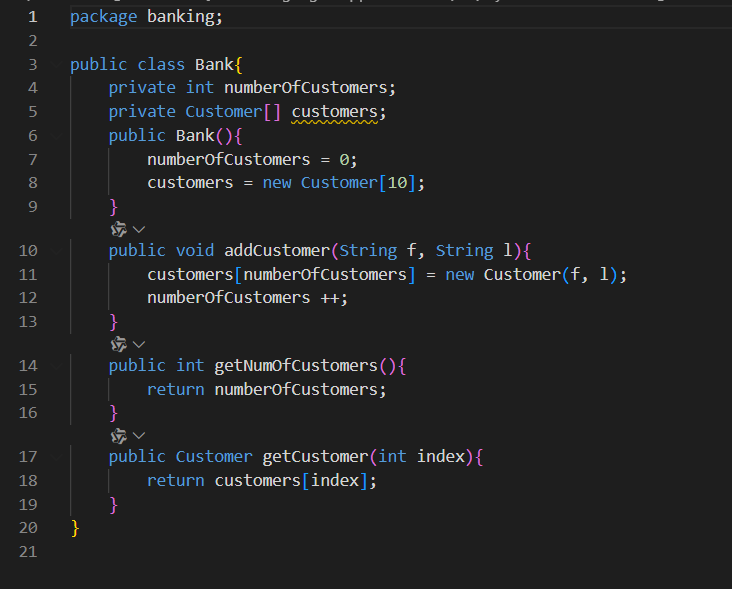


**STEP4**

运行结果：



代码：



1. **问题思考**

1、假设在TestBanking类中引用了Account类和Customer类，在编辑了三个类的源文件后，仅编译TestBanking.java 源文件同时也生成了其他类的类文件，为什么？进一步归纳，当编译器需要某个类的信息时，如何搜索定义该类的源文件或类文件？

答：

（1）为什么

Java 编译器 (javac) 在编译某个源文件时，会检查该文件中引用的类。如果这些引用的类还没有被编译成 .class 文件（或者 .class 文件不存在或过期），编译器会自动去查找并编译这些依赖类。这是因为 TestBanking.java 依赖 Account.java 和 Customer.java，所以编译器需要确保所有引用的类都能正确编译并生成对应的 .class 文件。

因此，当编译 TestBanking.java 时，编译器发现它依赖 Account 和 Customer，于是编译器会尝试找到这两个类的源文件（Account.java 和 Customer.java），并生成相应的 .class 文件。

（2）归纳

当 Java 编译器需要某个类的信息时，搜索顺序如下：

1、当前目录和包路径下的 .class 文件：首先查找同级目录中是否已经有最新的 .class 文件。

2、CLASSPATH 中的 .class 文件：如果当前目录没有找到，编译器会按照 CLASSPATH 变量中的路径去搜索其他目录中的 .class 文件。

3、源文件 .java 文件：如果没有找到合适的 .class 文件，编译器会尝试在源代码目录中查找同名的.java文件并自动编译它，生成.class文件

4、报错：如果没有找到合适的 .class 文件或 .java 文件，编译器会抛出错误。

2、假设已定义了Banking.Customer类和Banking.Account类，并且Customer类中引用了Account类，若直接在该类所在目录（/banking子目录下）运行 Javac customer.java 会出错，请分析出错原因。

答：出错原因如下

1. 包路径不匹配

在 Java 中，包路径和文件路径必须严格匹配。如果 Customer.java 和 Account.java 声明了 package banking;，那么编译器期望它们位于 banking 目录下，但应该从项目的根目录（而不是 banking 目录）运行编译命令。

如果直接在 banking 目录下运行 javac Customer.java，编译器会认为你当前目录就是 banking，而不会识别 package banking 这个声明。编译器期望在项目的根目录下运行，并通过指定完整的路径来找到源文件。

2. 引用类的包未找到

在 Customer.java 中，可能会使用 import banking.Account; 来引用 Account 类，或直接使用它。但如果在 banking 目录下运行 javac Customer.java，编译器会找不到 banking.Account，因为它假定当前目录是 banking，而不是从根目录查找包路径。

为了解决如上问题，应该在根目录运行，确保编译器能够正确处理包声明和类引用，且每个step应该要有独立的工作区

3、描述 -classpath 类路径的作用，思考JDK的编译器和虚拟机是如何定位到所需的类文件的？归纳编译器搜索类的顺序以及JVM搜索类的顺序。

答：

（1）如何定位

编译器：javac 编译器在编译时会检查源代码中引用的所有类，并在类路径中搜索这些类的 .class 文件或 .java 源文件。如果找到了 .class 文件，它会直接使用这些文件；如果找到了 .java 文件，它会先编译这些文件生成 .class 文件。

虚拟机：JVM 在加载类时会按照类路径中的顺序搜索 .class 文件。如果一个类在多个位置被找到，JVM 会按照类路径的顺序选择第一个找到的类。

（2）归纳搜索类顺序

编译器搜索类的顺序：

①当前目录：如果类路径中包含.（当前目录），编译器会首先在当前目录中搜索类文件。

②指定的类路径：编译器会按照类路径中指定的顺序搜索每个目录和JAR文件。

③系统类路径：如果系统类路径（由 JAVA\_HOME 环境变量中的 jre/lib 目录定义）中的类被引用，编译器也会使用这些类。

JVM搜索类的顺序：

①应用程序类路径：JVM 首先在应用程序的类路径中搜索类文件，这包括通过 -classpath 或 -cp 指定的路径，以及通过系统属性 java.class.path 设置的路径。

②扩展目录：如果应用程序类路径中没有找到类，JVM 会在JRE的扩展目录（通常是 $JAVA\_HOME/jre/lib/ext）中搜索。

③系统目录：最后，JVM 会在JRE的系统目录（通常是 $JAVA\_HOME/jre/lib）中搜索类文件。