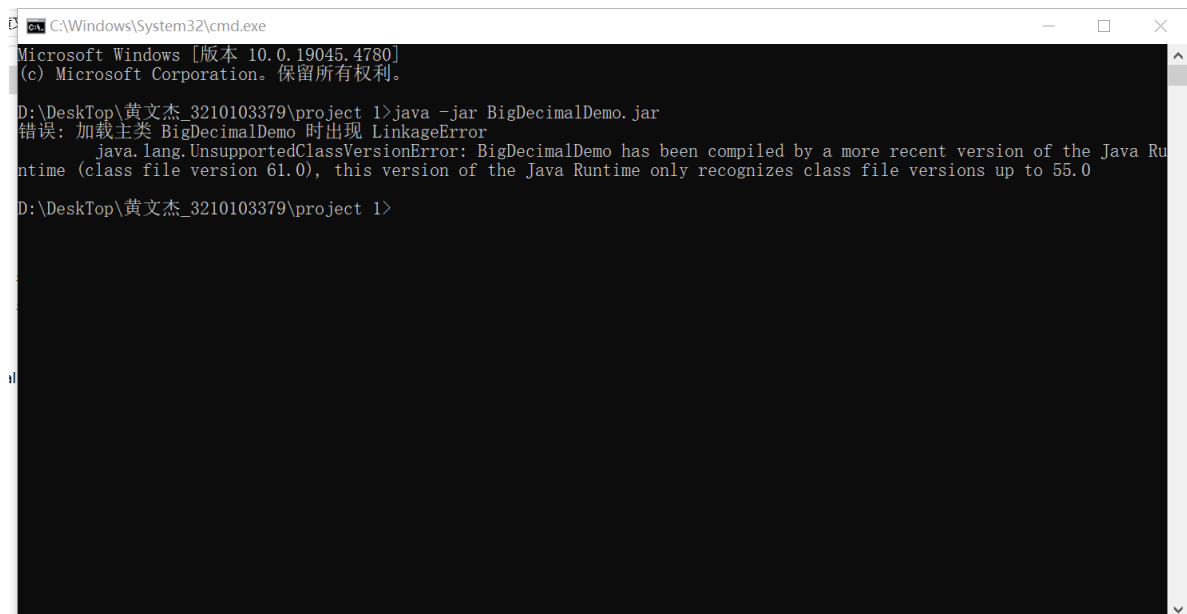


JDK 环境

本实验所用的JDK环境是 **JDK 17**，请确保本机已经安装 **JDK17** 以上的环境，并做了相关配置

如果本机的JDK版本不到17可能会运行不了 `jar` 文件，具体错误可能如下图所示：



运行步骤

在文件夹下已经包含了 `BigDecimalDemo.java`, `BigDecimalDemo.jar`, `BigDecimalDemo.class` 三个核心文件（本实验相关的代码文件），如果本机已经安装过JDK环境并且做了相关配置，只需要文件目录下打开cmd窗口，并运行如下命令即可执行jar文件：

```
1 | java -jar BigDecimalDemo.jar
```

如果运行不了，可以尝试检查是否安装和配置了JDK环境，然后尝试重新生成jar文件，步骤如下：

1. 编译 Java 文件

首先，需要将 `.java` 源文件编译成 `.class` 字节码文件。

```
1 | javac -encoding UTF-8 BigDecimalDemo.java
```

这会在当前目录下生成对应的 `.class` 文件。

2. 创建 JAR 文件

使用 `jar` 命令将编译好的 `.class` 文件打包成JAR文件。

```
1 | jar cvf BigDecimalDemo.jar BigDecimalDemo.class
```

其中：

- `c` 表示创建 JAR 文件
- `v` 表示详细输出过程
- `f` 指定输出文件的文件名

这个命令会把当前目录下的 `.class` 文件全部打包到 `BigDecimalDemo.jar` 文件中。

3. 指定入口点（创建可执行 JAR）

如果需要生成一个可执行 JAR 文件（包含 `main` 方法的入口类），可以指定一个 `Manifest` 文件或直接通过命令行指定入口类。

通过命令行指定主类：

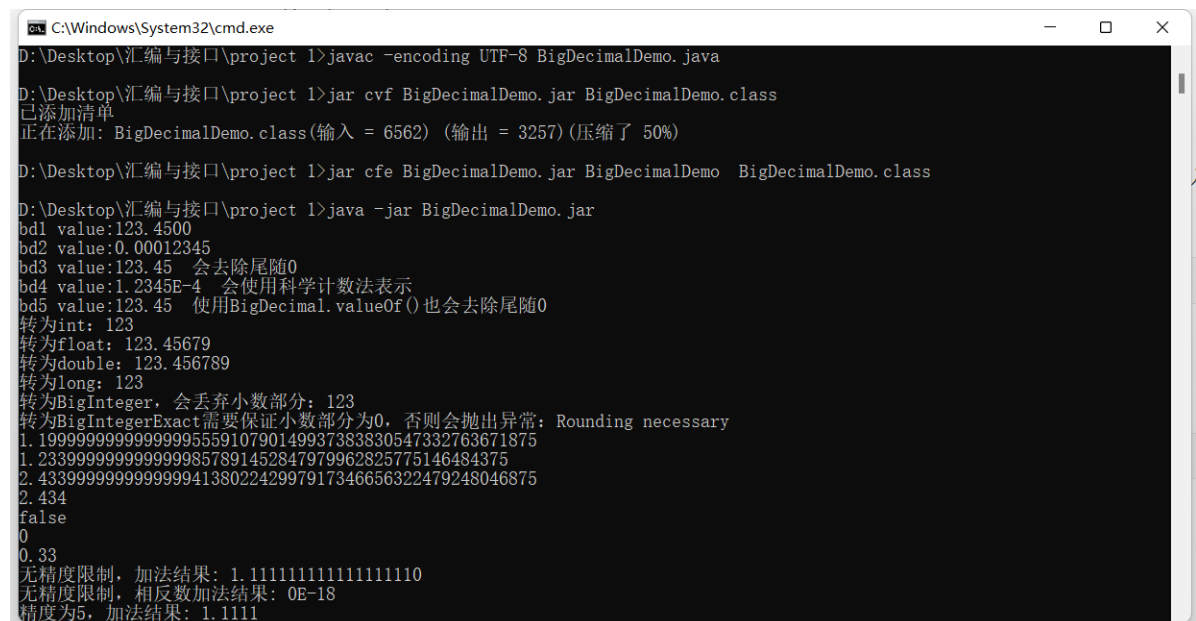
```
1 | jar cfe BigDecimalDemo.jar BigDecimalDemo BigDecimalDemo.class
```

4. 运行 JAR 文件

在当前目录下的 `cmd` 窗口中执行如下命令

```
1 | java -jar BigDecimalDemo.jar
```

提供一张我打包运行 `jar` 文件的 `cmd` 窗口命令截图：



```
C:\Windows\System32\cmd.exe
D:\Desktop\汇编与接口\project 1>javac -encoding UTF-8 BigDecimalDemo.java
D:\Desktop\汇编与接口\project 1>jar cvf BigDecimalDemo.jar BigDecimalDemo.class
已添加清单
正在添加: BigDecimalDemo.class(输入 = 6562) (输出 = 3257) (压缩了 50%)
D:\Desktop\汇编与接口\project 1>jar cfe BigDecimalDemo.jar BigDecimalDemo BigDecimalDemo.class
D:\Desktop\汇编与接口\project 1>java -jar BigDecimalDemo.jar
bd1 value:123.4500
bd2 value:0.00012345
bd3 value:123.45 会去除尾随0
bd4 value:1.2345E-4 会使用科学计数法表示
bd5 value:123.45 使用BigDecimal.valueOf()也会去除尾随0
转为int: 123
转为float: 123.45679
转为double: 123.456789
转为long: 123
转为BigInteger, 会丢弃小数部分: 123
转为BigIntegerExact需要保证小数部分为0, 否则会抛出异常: Rounding necessary
1.1999999999999999555910790149937383830547332763671875
1.2339999999999999857891452847979962825775146484375
2.433999999999999413802242997917346656322479248046875
2.434
false
0
0.33
无精度限制, 加法结果: 1.1111111111111110
无精度限制, 相反数加法结果: 0E-18
精度为5, 加法结果: 1.1111
```