Lab 02 Assignment

班级：202111
学号：19241027
姓名：胡峻诚

Question1: 简答题

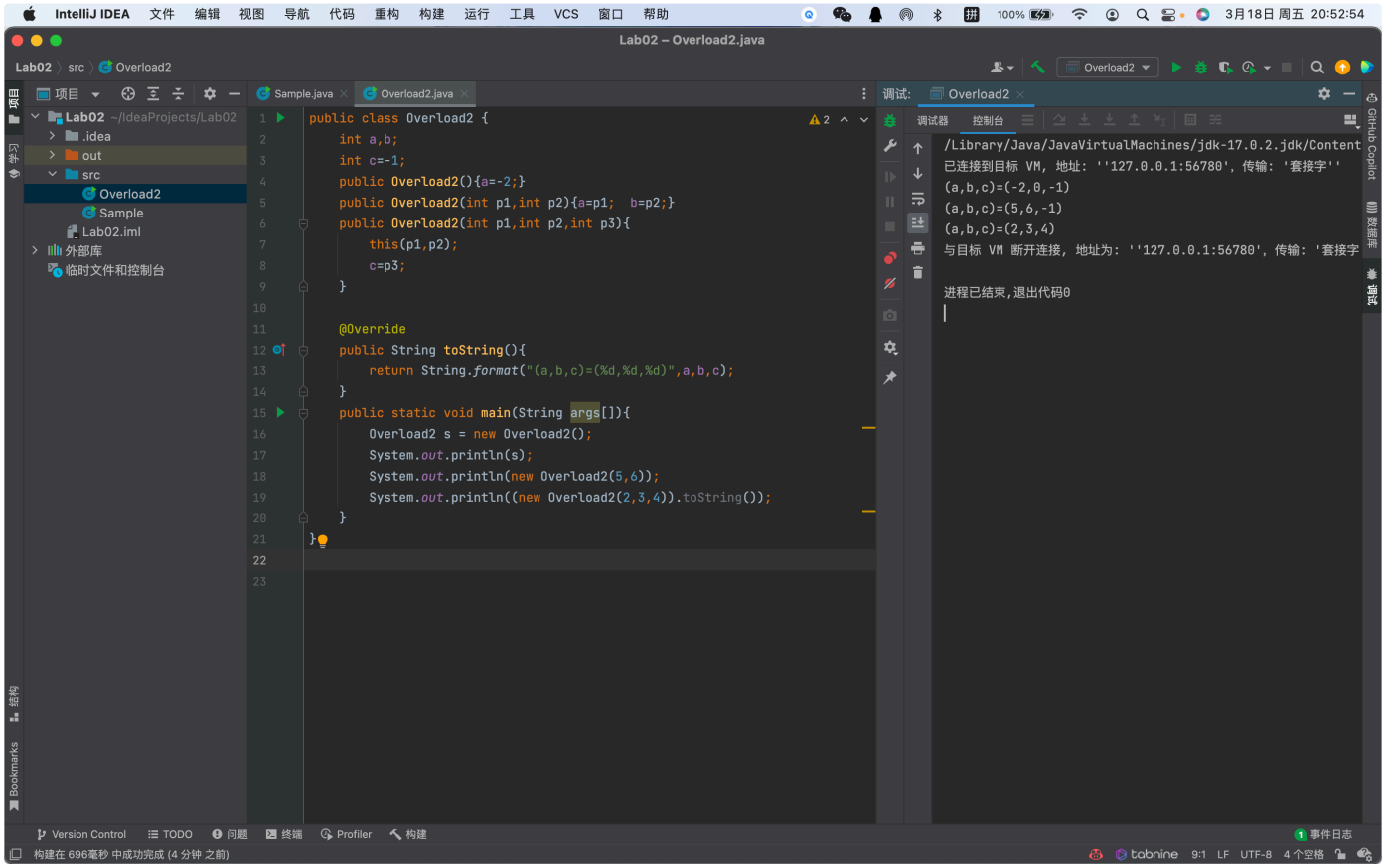
标记的第4行编译会导致编译错误：原因 `java: 可能尚未初始化变量m`

Question2: 简答题 overload

- a. 对
- b. 对
- c. 对
- d. 错。只有 `Overload(int m)` 与 `Overload(double m)` 才是构造方法。

Question3: 简答题 overload2

```
(a,b,c)=(-2,0,-1)
(a,b,c)=(5,6,-1)
(a,b,c)=(2,3,4)
```



Question4: 简答题 finalize

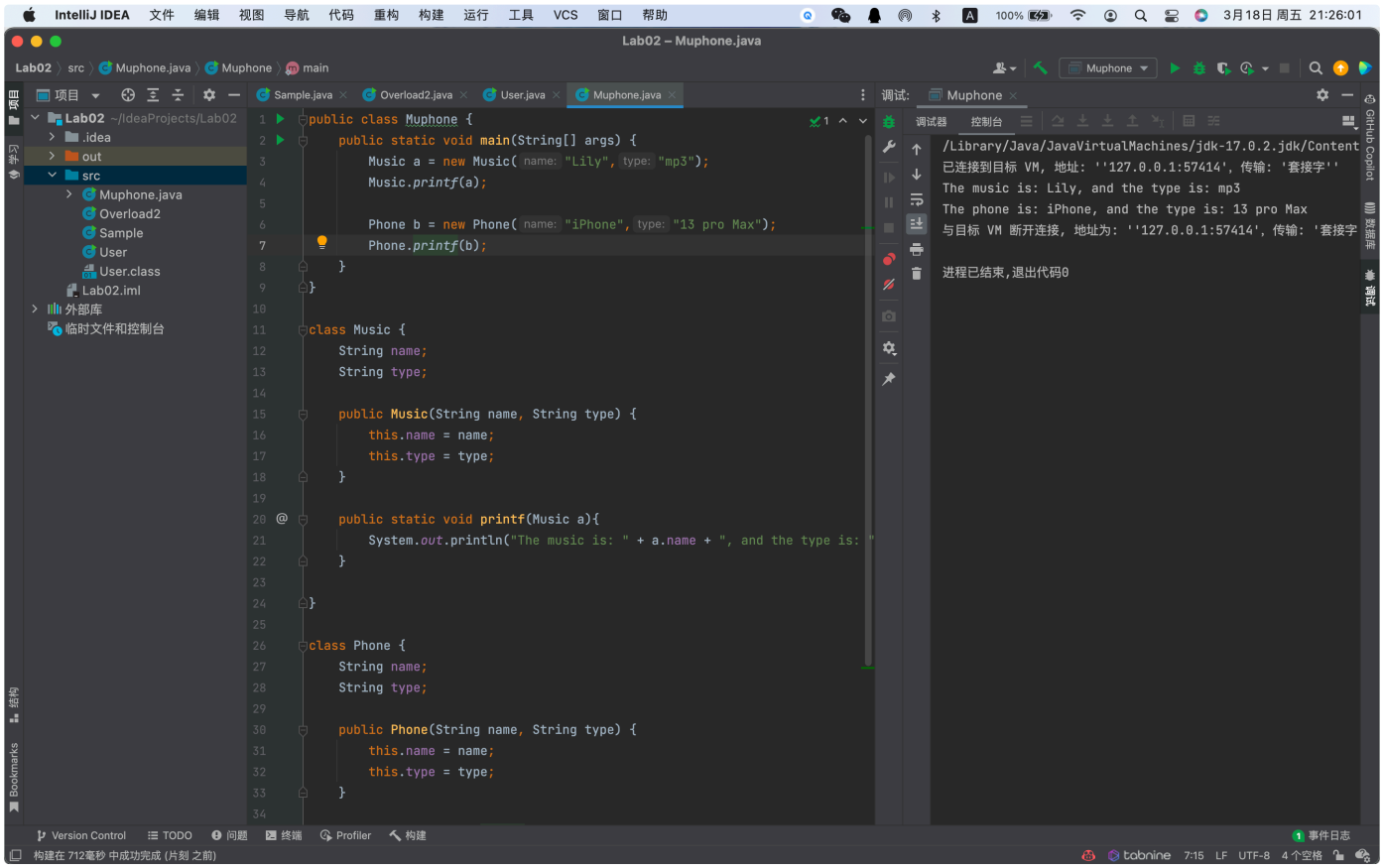
注释标记行	可能的输出内容
1	0
2	1
3	0/1
4	0/1

有什么办法可以让 `finalize()` 在 `main()` 结束前一定被调用吗?

不能保证一定被调用。
因为Java无法保证`finalize()`会被及时执行并且根本无法保证它会被执行 （因为程序中其他线程的优先级远远高于`finalize()`函数线程的优先级)。

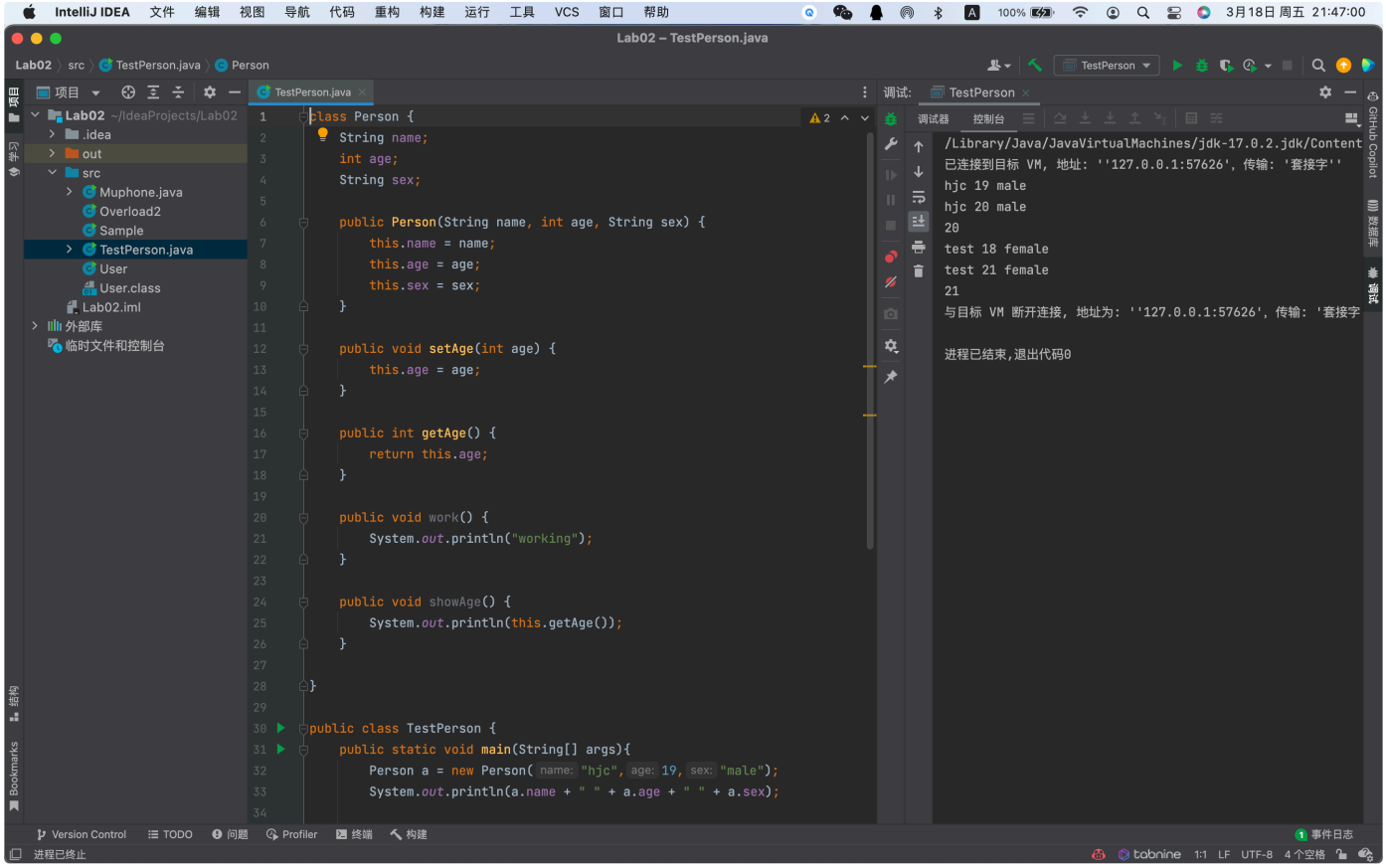
Question5: 编程题

```
目录名为：Question5
文件名有：Muphone.java
```



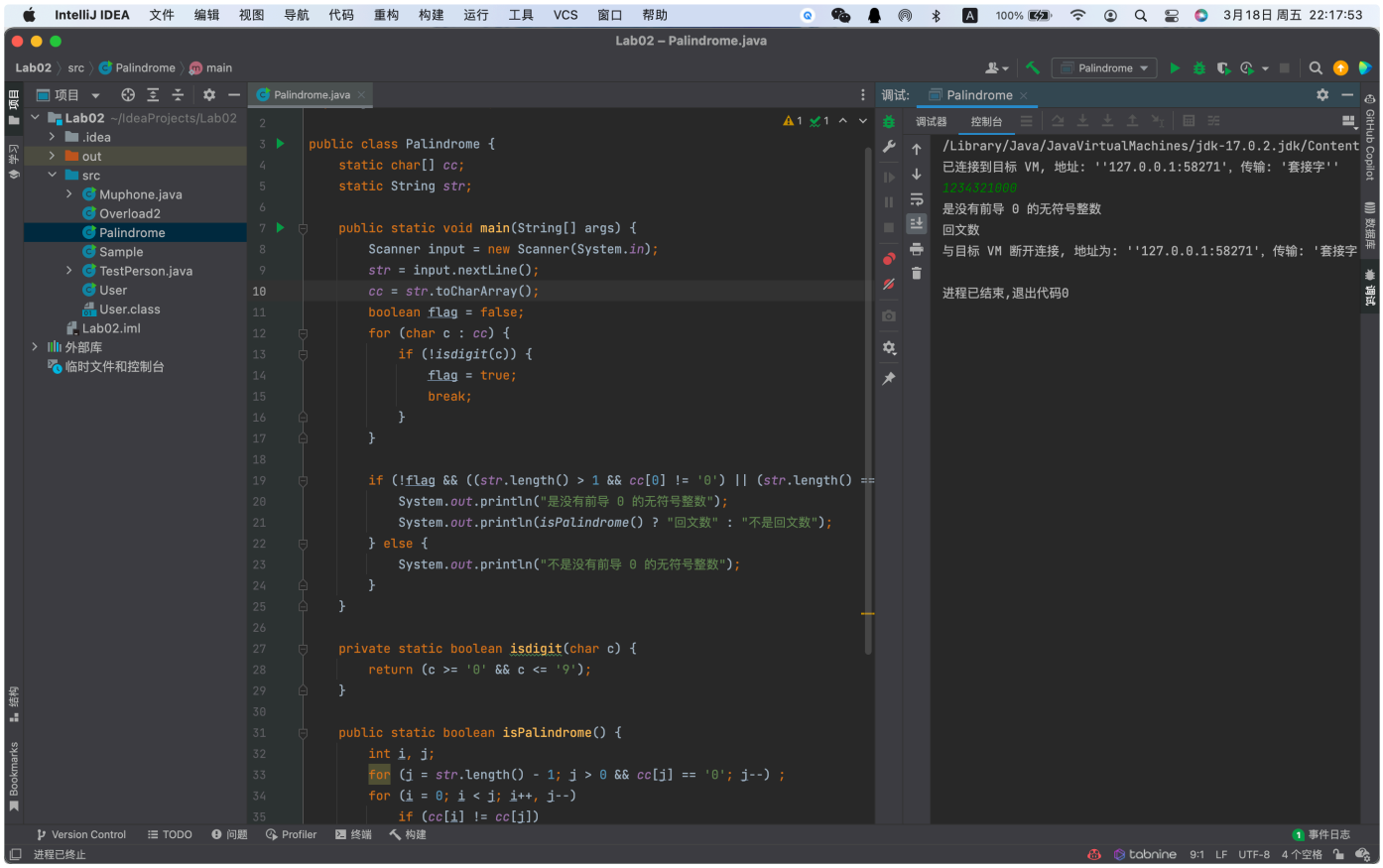
Question6: 编程题

```
目录名为：Question6
文件名有：TestPerson.java
```



Question7: 编程题

```
目录名为：Question7
文件名有：Palindrome.java
```


IntelliJ IDEA 文件 编辑 视图 导航 代码 重构 构建 运行 工具 VCS 窗口 帮助
3月18日 周五 22:17:53
Lab02 – Palindrome.java
Lab02 〉 src 〉 Palindrome 〉 main
Palindrome
项目
Lab02 ~/IdeaProjects/Lab02
.idea
out
src
Muphone.java
Overload2
Palindrome
Sample
TestPerson.java
User
User.class
Lab02.iml
外部库
临时文件和控制台
Palindrome.java
public class Palindrome {
static char[] cc;
static String str;
public static void main(String[] args) {
Scanner input = new Scanner(System.in);
str = input.nextLine();
cc = str.toCharArray();
boolean flag = false;
for (char c : cc) {
if (!isdigit(c)) {
flag = true;
break;
}
}
if (!flag && ((str.length() > 1 && cc[0] != '0') || (str.length() ==
System.out.println("是没有前导 0 的无符号整数");
System.out.println(isPalindrome() ? "回文数" : "不是回文数");
} else {
System.out.println("不是没有前导 0 的无符号整数");
}
}
private static boolean isdigit(char c) {
return (c >= '0' && c <= '9');
}
public static boolean isPalindrome() {
int i, j;
for (j = str.length() - 1; j > 0 && cc[j] == '0'; j--) ;
for (i = 0; i < j; i++, j--)
if (cc[i] != cc[j])
调试: Palindrome
调试器 控制台
/Library/Java/JavaVirtualMachines/jdk-17.0.2.jdk/Content
已连接到目标 VM, 地址: ''127.0.0.1:58271', 传输: '套接字''
1234321000
是没有前导 0 的无符号整数
回文数
与目标 VM 断开连接, 地址为: ''127.0.0.1:58271', 传输: '套接字
进程已结束,退出代码0
Version Control TODO 问题 Profiler 终端 构建
事件日志
进程已终止
tabnine 9:1 LF UTF-8 4 个空格