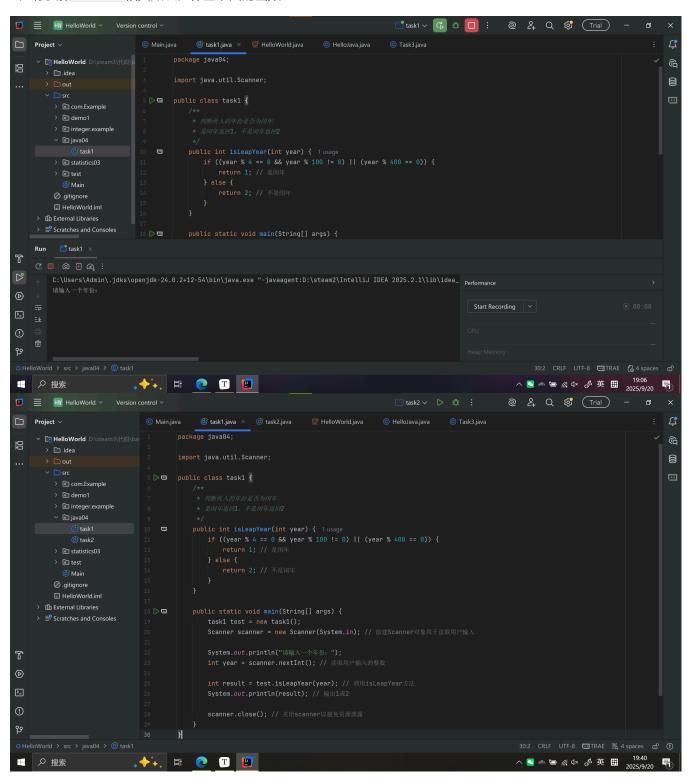
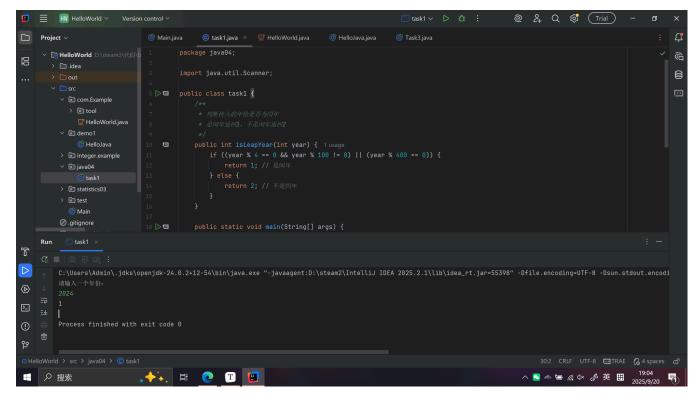
### Task1.if-else

1、请了解 if-else 相关知识。补全下面的函数:





2、你知道 switch-case 吗,有人说 switch-case 是 if-selse 的语法糖( syntax suger )这是正确的吗?你知道他们的底层实现原理吗?

switch-case 可以被视为 if-else 的语法糖。它简化了多分支条件判断的代码,使其更易读。

底层实现上,编译器会根据情况将 switch-case 转换为跳转表 (jump table) 或一系列比较和跳转指令,类似于 if-else 语句。对于连续且数量较少的 case , 通常使用跳转表提高效率 ; 否则可能生成类似 if-else 的结构。

3、请了解 for-while 相关知识。补全下面函数:

```
@ 24 Q & Trial
 C Task3.java
       package java04;
铝
                                                                               •
              System.out.print(" ");
  Run
T
Ð
လူ
                                                            へ № ← 恒 🦟 Ф× 🔗 英 胼 19:37
              ,++. 🛱 😢 🗉 📵
  ♪ 搜索
@ 24 Q & (Trial)
  ක
                                                                               Ø
လှ
                                                            へ № № ㎞ 🦨 Ф 競 競 🖽 19:40
             ,++, 🛱 😢 T 📳
₩ 夕 搜索
```

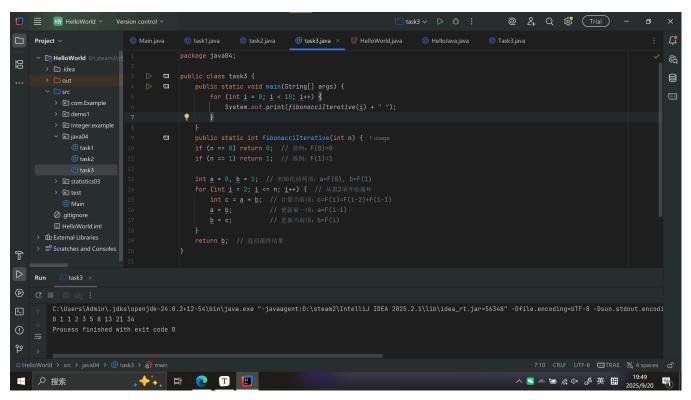
# Task3.递归和迭代

4、请了解递归和迭代相关知识,为斐波那契数列实现两个版本

### 1. 递归版本

```
HW HelloWorld V Version control V
                                                                                     @ 24 Q 8 Trial
☐ Project ∨
                                      C Task3.java
                         package java04;
    ∨ 🛅 HelloWorld 🛭
                                                                                                                 @
品
                                                                                                                 public static void main(String[] args) {
   > Scratches and Console
   Run
Ð
      0 1 1 2 3 5 8 13 21 34
                                                                                       へ 🔊 📤 🖅 🦧 d× 🔗 中 🖽 19:44 🍕
                   .↓. ♯ @ T 🖺
  ₽ 搜索
```

## 2. 迭代版本



5、迭代和递归不同处在于哪里,为什么一般偏好迭代?循环是否能完全用递归来取代?

### 递归和迭代的区别

#### 不同点:

- 执行方式: 递归通过函数调用自身解决问题, 迭代使用循环结构。
- 内存使用: 递归需要额外栈空间, 可能导致栈溢出; 迭代不需要额外栈空间。
- 效率: 迭代通常比递归更高效, 避免重复计算。

#### 偏好迭代原因:

- 更高效,避免函数调用开销和重复计算。
- 内存使用少,更安全。

#### 循环与递归关系:

- 理论上,任何循环都能用递归实现,反之亦然。
- 实际中,某些问题更适合用迭代(如线性问题),某些更适合递归(如树形结构)。

# Task4.汉诺塔

6、通过 if-else, for-while 等等能够对程序进行简单的流程控制,下面的问题是一个名为汉诺塔的游戏。现在有 A,B,C三个柱子,在A柱上有n块铁饼,现在要把A柱上的n块铁饼都移动到C柱上,**请补全函数**,使得函数打印出移动的过程。要求给出你代码的具体思路,可以写在文档或者代码注释中

