

Java 语言程序设计 作业 2-Java 基础

封钰震 (1951362)

2021 年 10 月 24 日

1. 编程环境

硬件环境

1. 型号名称: MacBook Pro
2. 处理器名称: Dual-Core Intel Core i5
3. 内存: 8 GB

软件环境

系统版本: macOS 10.15.7

运行环境

1. JDK 14.0.2

2. 设计思想

走台阶——动态规划

采用动态规划算法。对于第 n 个台阶而言, 可以从第 $n-1$ 个台阶处跨一个台阶到达, 也可以从第 $n-2$ 个台阶处跨两个台阶得到。因此, 有:

$$f(n) = \begin{cases} 1, n = 1 \\ 2, n = 2 \\ f(n-1) + f(n-2) \end{cases}$$

其余问题不需要使用特别的算法，详见“执行过程”一节。

3. 执行过程

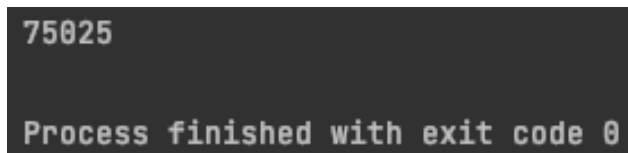
第一问

首先，编写计算走台阶方法数的walkOnStep函数。

```
public class Main {  
    private static int walkOnStep(int numOfStep) {  
        int n1 = 1, n2 = 2, temp = n2;  
        //  $f(n) = f(n - 1) + f(n - 2)$   
        if (numOfStep == 1) {  
            return 1;  
        }  
        else if (numOfStep == 2) {  
            return 2;  
        }  
        else {  
            for (int i = 3; i <= numOfStep; i = i + 1) {  
                temp = n2;  
                n2 = n2 + n1;  
                n1 = temp;  
                // System.out.println(i + ": " + n2);  
            }  
            return n2;  
        }  
    }  
}
```

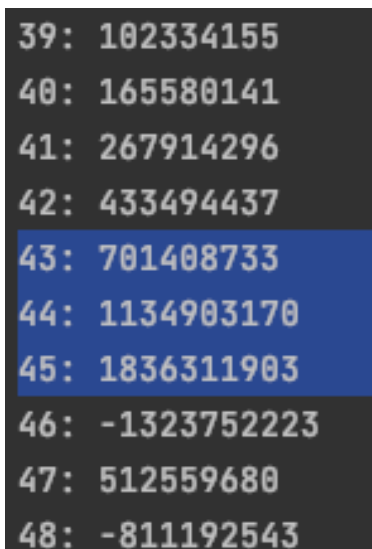
再在main函数中调用walkOnStep函数并输出。

```
public static void main(String[] args) {  
    int num = Main.walkOnStep(24);  
    System.out.println(num);  
}
```



```
75025
Process finished with exit code 0
```

图 1: 输出结果



```
39: 102334155
40: 165580141
41: 267914296
42: 433494437
43: 701408733
44: 1134903170
45: 1836311903
46: -1323752223
47: 512559680
48: -811192543
```

图 2: 输出结果

得到结果 $f(24) = 75025$ ，程序输出如图1所示。

第二问

首先，在walkOnStep函数添加输出：

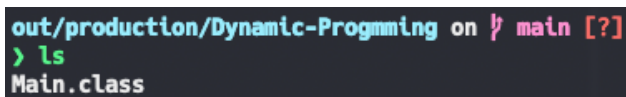
```
System.out.println(i + ": " + n2);
```

(即前一问所示代码中被注释掉的一行)。

再在main函数中调用walkOnStep函数，给以较大的参数，如：

```
int num = Main.walkOnStep(80);
```

最后，得到输出结果，如图2所示。因此，int数据类型可以处理的台阶数的最大值为 45。



```
out/production/Dynamic-Programming on % main [?]  
> ls  
Main.class
```

图 3: 输出结果

第三问

针对该问，在前两问程序的基础上增加了输入。用户可以选择运行该程序后输入台阶数（对应代码中if分支），也可以选择在使用命令行运行该程序时输入台阶数（对应代码中else分支）。修改后的main函数如下：

```
public static void main(String [] args) {  
    int n = 1;  
    try {  
        if (args.length == 0) {  
            System.out.print("Please input the number of steps: ");  
            Scanner in = new Scanner(System.in);  
            n = in.nextInt();  
        }  
        else {  
            n = Integer.parseInt(args[0]);  
        }  
    } catch (Exception e) {  
        e.printStackTrace();  
        return;  
    }  
  
    int num = Main.walkOnStep(n);  
    System.out.println(num);  
}
```

下面使用命令行运行编译完成的.class文件。

首先，找到Main.class文件所在目录，如图3所示。

使用java程序启动 Java 虚拟机，虚拟机执行Main.class文件中的字节码，未输入和输入参数的情形分别如图4和图5所示。

```
out/production/Dynamic-Programming on 7 main [?]  
> java Main  
Please input the number of steps: 24  
75025
```

图 4: 输出结果

```
out/production/Dynamic-Programming on 7 main [?]  
> java Main 24  
75025
```

图 5: 输出结果

第四问

首先通过ssh登录远程服务器，再通过scp将本地的Main.java文件上传至服务器，查看服务器上的该文件，如图6所示。

再通过javac程序将Main.java文件编译成Main.class文件，如图7所示。

最后，使用java程序启动 Java 虚拟机，虚拟机执行Main.class文件中的字节码，输入参数24，结果如图8所示。

```
[root@openeuler Java]# cd ~/Java/Homework-3/  
[root@openeuler Homework-3]# ls  
Main.java
```

图 6: 输出结果

```
[root@openeuler Homework-3]# javac Main.java  
[root@openeuler Homework-3]# ls  
Main.class Main.java
```

图 7: 输出结果

```
[root@openeuler Homework-3]# java Main 24  
75025
```

图 8: 输出结果