

Java 语言程序设计 作业 1-Java 基础

封钰震 (1951362)

2021 年 10 月 3 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 (19H1030)

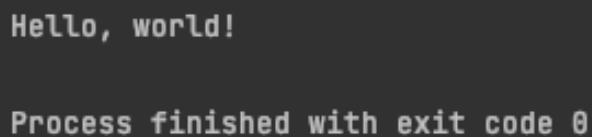
运行环境

1. C++ Language Dialect: GNU++14[-std=gnu++14]
2. C++ Standard Library: libc++
3. JDK 14.0.2

2. 设计思想

JDK 与 IDE

1. 选择最简单的Hello, world!程序, 在 IDE 中编译并运行;
2. 找到安装的 JDK, 重命名其中的 Unix 可执行文件, 再次在 IDE 中编译、运行Hello, world!程序;
3. 将 Unix 可执行文件改回原来的名字, 再次在 IDE 中编译、运行Hello, world!程序。



```
Hello, world!  
  
Process finished with exit code 0
```

图 1: 输出结果

数据类型实验

对于不同数据类型的变量, 依次计算 $2! = 2 \times 1, 3! = 3 \times 2!, \dots, k! = k \times (k - 1)!$ 并输出, 判断是否越界。

3. 执行过程

JDK 与 IDE

实验代码

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Hello , world!");  
    }  
}
```

输出结果

首先, 未对 JDK 中的程序进行修改, 在 IntelliJ IDEA 中编译并运行, 得到输出结果如图1。

接着, 找到安装的 JDK, 如图2所示, 将其中的 Unix 可执行文件 `java` 重命名为 `java1`。

然后, 再次在 IntelliJ IDEA 中编译并运行, 得到如图3所示的报错, 显示 `No such file or dictionary`, 找不到 JDK 中的 `java` 文件。

最后, 将文件 `java1` 重新命名为 `java`, 再次编译并运行, 运行成功并输出。

实验过程中事件日志的截图如图4所示。

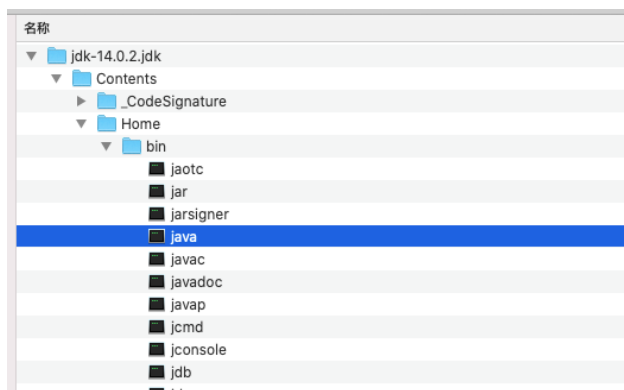


图 2: 安装的 JDK

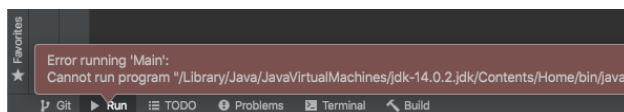


图 3: 报错

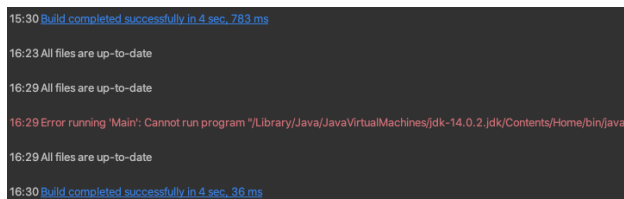


图 4: 实验事件日志

实验结论

1. IDE 是方便开发人员开发的集成开发工具，JDK 是 Java 开发工具包。
2. IDE 能够帮助我们调用已安装的 JDK 提供的工具。若没有 JDK 或 JDK 损坏，则 IDE 不能顺利编译、解释。

数据类型实验

C++ 部分

实验代码

```
#include <iostream>
const int MAX = 20;
int main(int argc, const char * argv[]) {
    short factorial = 1;
    //    unsigned short factorial = 1;
    //    int factorial = 1;
    //    unsigned int factorial = 1;
    //    long factorial = 1;
    //    unsigned long factorial = 1;
    //    long long factorial = 1;
    //    unsigned long long factorial = 1;
    //    float factorial = 1;
    //    double factorial = 1;
    //    long double factorial = 1;
    for (int n = 2; n < MAX; ++n)
    {
        factorial = factorial * n;
        std::cout << n << "!=" << factorial << std::endl;
    }
}
```

输出结果 改变变量factorial的数据类型，观察输出。例如，对于short类型的变量，得到如图5所示的输出。因此，要使 C++ 的short类型变量不溢出，n的最大值为 7。

经过实验，得到结果如表1所示。

2! = 2
3! = 6
4! = 24
5! = 120
6! = 720
7! = 5040
8! = -25216

图 5: C++/short 类型

	n_{\max}
short	7
unsigned short	8
int	12
unsigned int	12
long	20
unsigned long	20
long long	20
unsigned long long	20
float	34
double	170
long double	1754

表 1: C++ 数据类型实验结果

	n_{\max}
byte	5
short	7
int	12
long	20
float	34
double	170

表 2: Java 数据类型实验结果

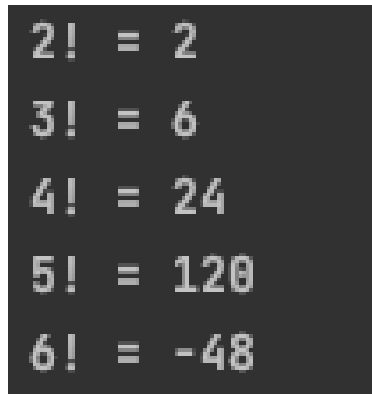
Java 部分

实验代码

```
public class Main {  
  
    public static void main(String[] args) {  
        final int MAX = 20;  
        byte factorial = 1;  
        //    short factorial = 1;  
        //    int factorial = 1;  
        //    long factorial = 1L;  
        //    float factorial = 1f;  
        //    double factorial = 1d;  
        for (int n = 2; n < MAX; ++n)  
        {  
            factorial = (byte) (factorial * n);  
            System.out.print(n);  
            System.out.print("!=");  
            System.out.println(factorial);  
        }  
    }  
}
```

改变变量 `factorial` 的数据类型，观察输出。例如，对于 `byte` 类型的变量，得到如图 6 所示的输出。因此，要使 Java 的 `byte` 类型变量不溢出，`n` 的最大值为 5。

经过实验，得到结果如表 2 所示。



2! = 2
3! = 6
4! = 24
5! = 120
6! = -48

图 6: Java/byte 类型

文件输出

在上述代码后添加文件输出即可。

实验代码

```
package com.company;

import java.io.*;

public class Main {
    public static void main(String[] args) {
        final int MAX = 100;
        double factorial = 1d;
        try {
            BufferedWriter out = new BufferedWriter(new FileWriter("
                factorial.csv"));
            for (int n = 1; n < MAX; ++n)
            {
                factorial = factorial * n;
                out.write(String.valueOf(n) + ',' + String.valueOf(
                    factorial) + '\n');
            }
            out.close();
            System.out.println("Succeed to create the file.");
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

	A	B	C
1	1	1	
2	2	2	
3	3	6	
4	4	24	
5	5	120	
6	6	720	
7	7	5040	
8	8	40320	
9	9	362880	
10	10	3628800	
11	11	3.99E+07	
12	12	4.79E+08	
13	13	6.23E+09	
14	14	8.72E+10	
15	15	1.31E+12	
16	16	2.00E+13	

图 7: csv 文件

```
    } catch (IOException e)
    {
        System.out.println("Fail to create the file.");
    }
}
```

输出结果

程序新建的 csv 文件可用 Excel 打开，如图7所示。