

# JAVA 初体验

陈德创 19030500217

2020 年 5 月 29 日

## 1 环境配置

**JDK 安装** JDK 的安装参考 (<https://www.runoob.com/java/java-environment-setup.html>) 由于安装时间较为久远, 当时未考虑版本, 选择了当时的最高版本 (JDK13), 如图:

```
C:\Users\RainCurtain>java -version
java version "13" 2019-09-17
Java(TM) SE Runtime Environment (build 13+33)
Java HotSpot(TM) 64-Bit Server VM (build 13+33, mixed mode, sharing)
```

**JAVA 第一行代码** 在任意目录创建 Hello.java 文件, 用编辑器打开, 输入如下图所示代码:

```
1 public class Hello{
2     public static void main(String [] args) {
3         System.out.println("Hello World");
4     }
5 }
```

即创建了一个 Java 文件, 作用是在控制台输出"Hello World" 字符串, 在 cmd 中进入文件目录, 使用命令 javac Hello.java 编译文件, 得到 Hello.class 文件, 再使用 java Hello 运行预编译好的文件即可, 注意 java 命令跟类名, 不必加".class"。如图:

```
D:\>javac Hello.java
D:\>java Hello
Hello World
```

## 2 IntelliJ IDEA 的使用

**IDEA 的安装** 略..., 不过学生邮箱可以免费申请一年, 不用怪可惜的。

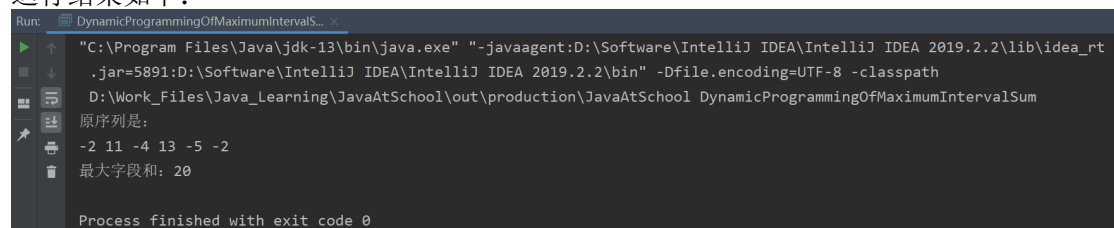
**项目创建** 打开 Idea, 点击”Create New Project”, 点击 JAVA, 在 Project SDK 处可以选择 SDK 文件, 点击 Next, 选择模板 (没有模板), Next, 给项目命名, 选择路径, Finish.

等待 Ide 创建完成新项目之后, 观察目录文件, 我们只需关心在 src 文件夹里面的文件, 里面存放了我们的 Java 源码。

**在 Idea 中编写 JAVA 程序** 在 src 文件夹中创建 JAVA 类, 名为 DynamicProgrammingOfMaximumIntervalSum1。注意, 一个 java 文件中, 只能有一个类被 public 修饰, 这个类名必须和文件名保持一致。复制老师提供代码, 如下:

```
1 public class DynamicProgrammingOfMaximumIntervalSum {
2     public static void main(String[] args) {
3         int[] array = { -2, 11, -4, 13, -5, -2 };
4         System.out.println("原序列是: ");
5         printSegment(0, array.length - 1, array);
6         int maxSum = maxSum(array);
7         System.out.println("最大字段和: " + maxSum);
8     }
9     private static int maxSum(int[] array) {
10         int sum = 0;
11         int b = 0;
12         for (int i = 0; i < array.length; i++) {
13             if (b > 0) {
14                 b += array[i];
15             } else {
16                 b = array[i];
17             }
18             if (b > sum) {
19                 sum = b;
20             }
21         }
22         return sum;
23     }
24     private static void printSegment(int start, int end, int[] array) {
25         for (int i = start; i <= end; i++) {
26             System.out.print(array[i] + " ");
27         }
28         System.out.println();
29     }
30 }
```

运行结果如下:



```
Run: DynamicProgrammingOfMaximumIntervalSum
"C:\Program Files\Java\jdk-13\bin\java.exe" "-javaagent:D:\Software\IntelliJ IDEA\IntelliJ IDEA 2019.2.2\lib\idea_rt
.jar=5891:D:\Software\IntelliJ IDEA\IntelliJ IDEA 2019.2.2\bin" -Dfile.encoding=UTF-8 -classpath
D:\Work_Files\Java_Learning\JavaAtSchool\out\production\JavaAtSchool DynamicProgrammingOfMaximumIntervalSum
原序列是:
-2 11 -4 13 -5 -2
最大字段和: 20
Process finished with exit code 0
```

### 3 对算法的理解

由注释描述中可以看出这是一个算法用来求给定序列的最大连续子序列, 主要实现代码为 maxSum 方法, 如下:

```

private static int maxSum(int[] array) {
2   int sum = 0;
   int b = 0;
4   for (int i = 0; i < array.length; i++) {
       if (b > 0) {
6           b += array[i];
       } else {
8           b = array[i];
       }
10      if (b > sum) {
           sum = b;
12      }
   }
14  return sum;
}

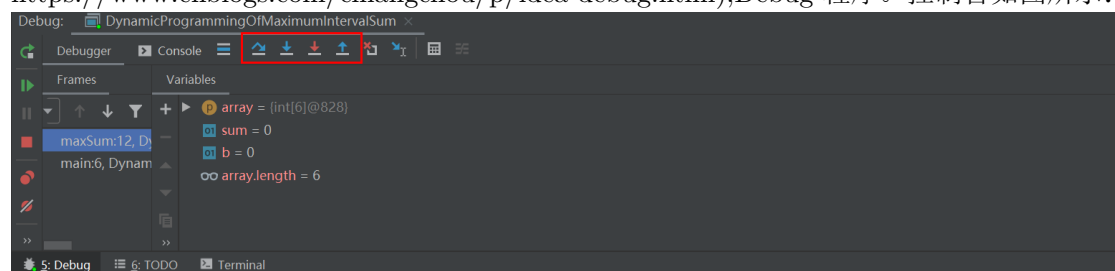
```

#### 3.1 算法猜测

猜测, 当遍历到第  $i$  时,  $sum$  为  $array[0...i]$  的最大子段和,  $b$  为  $array[0...i]$  以  $array[i]$  为结尾的最大子段和。

#### 3.2 通过调试验证算法

为此方法的 for 循环设置断点, 具体方法与在 Dev 中类似 (参考博客: <https://www.cnblogs.com/chiangchou/p/idea-debug.html>), Debug 程序。控制台如图所示:



单步执行, 记录结果如下:

i	0	1	2	3	4	5
b	-2	11	7	20	15	13
array	-2	11	-4	13	-5	-2
sum	-2	11	-4	13	-5	-2

与我们的猜想相符合, 更换 array 的值后, 多次实验均符合猜想, 下面进行粗略的算法的正确性证明。

### 3.3 算法的正确性

**数学归纳法** 对于长度为  $n$  的序列, 我们设其编号为  $0, 1, 2, \dots, n-1$ , 则对于构成最大连续列和的子序列, 其结尾一定是  $0, 1, 2, \dots, n-1$  其中之一。依照代码符号, 约定:

当遍历到  $i$  时,

$b$ : 以  $i$  为结尾的最大子段和

$sum$ :  $0 - i$  的最大字段和

则  $ans = \max\{sum\}, sum = \max\{b\}$

**1) 当  $i$  遍历到 0 时** 显然, 第一次循环体执行过后,  $b = array[0]$ , 而  $sum = \max(0, array[0])$ , 注意到  $sum$  的每次更新只能取大, 所以实际上  $sum$  一定是大于 0 的, 所以对于全部为复数的特殊情况, 程序实际上会给出一个错误答案, 但是为了整个证明的连续性, 我们先假设给定输入不是全为复数的 (即, 至少存在一个  $i$ , 使得  $array[i] > 0$ )。

**2) 假设当  $i$  遍历到  $k$  时, 算法都是正确的, 则当  $i$  遍历到  $k+1$  时** 若当  $i=k$  时,  $b$  大于 0, 那么显然的, 以  $k+1$  为结尾的最大子段和为  $array[k+1]+b$ , (若以  $k+1$  为结尾的最大子段和为  $t > array[k+1] + b$ , 那么显然当  $i=k$  时, 一定存在一个以  $i$  为结尾的最大字段和为  $t - array[k+1] > b$ , 与假设矛盾); 若  $b \leq 0$ , 那么以  $k+1$  为结尾的最大子段和为  $array[k+1]$ , 这个也是显然的, 反证法可证, 否则只会更小。

由此我们证明了  $b$  的正确性, 然后根据  $ans = \max\{sum\}, sum = \max\{b\}$ , 在每次遍历时更新  $sum$  的最大值, 或者说更新  $b$  的最大值 (因为  $sum$  的最值从  $b$  的最值中取出), 即可得到目标答案。

### 3.4 算法的小改进

前面已经说到, 当  $array$  数组全部为负数时, 程序是错误的 (这里我们规定一个序列的子段不能为空), 改进的话只需将  $sum$  的初始值更改一下就可以了, 有两个常用方案, 1 是将  $sum$  定为无穷小 (一个小于数据范围的负数), 2 是将  $sum$  定为  $array[0]$ , 更严谨点可以判断  $array$  的长度后再初始化  $sum$ 。

```

1  private static int maxSum(int[] array) {
2      int sum = 0;
3      if(array.length > 0){
4          sum = array[0];
5      }
6      ...
7      ...
8      return sum;
9  }

```

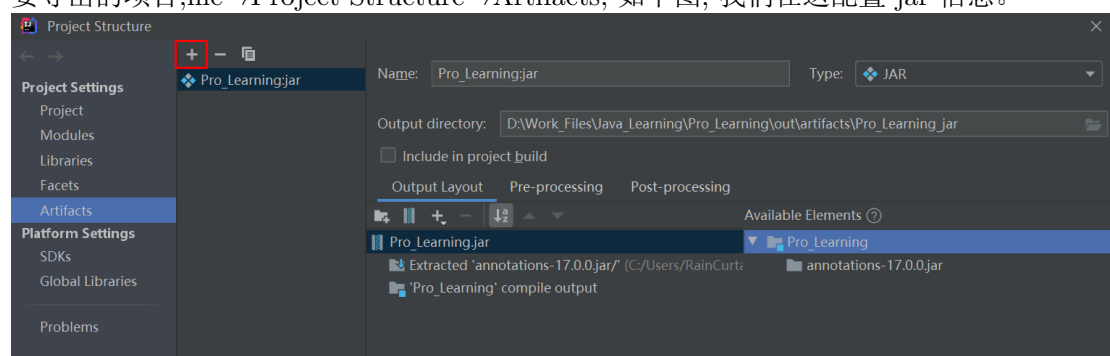
## 4 导入/导出 JAVA

### 4.1 Jar 是什么

这里参考博客 ([https://blog.csdn.net/qq\\_22073849/article/details/77148847](https://blog.csdn.net/qq_22073849/article/details/77148847)), Jar 包是一种特殊的 zip 文件, 并且与 zip 文件兼容, 只是多了清单文件。在我看来就像是一个库文件, 里面包含着别的工程的文件, 通过导入 Jar 包, 我们可以快速完成工程的接洽工作, 有利于分组开发。

### 4.2 导出 Jar 包

参考博客: (<https://blog.csdn.net/gslsxqrj/article/details/82624157>)。在 idea 中打开所需要导出的项目, file→Project Structure→Artifacts, 如下图, 我们在这配置 jar 信息。



点击红框中的“+”→jar→From...。在这里面可以选择要导出的模块, 主类 (运行 jar 时调用的类), 剩下的作用目前不明朗。OK, Apply, OK。然后 Build→Build Artifacts→Build, 就可以了, 然后去先前设置的目录里面找到 jar 文件就可以了。(默认为项目目录/out/artifacts)

### 4.3 运行 Jar 包

在 cmd 中进入 jar 包所在目录, 应用命令“java -jar 包名.jar”即可运行。如下图:

```
D:\Work_Files\Java_Learning\Test\out\artifacts\Test_jar>java -jar Test.jar
Hello World!
```

但是在我第一次进行导出并运行操作时, 控制台报错如下:

```
D:\Work_Files\Java_Learning\JavaAtSchool\out\artifacts\JavaAtSchool_jar>java -jar JavaAtSchool.jar
: Hello ^ LinkageError
    java.lang.UnsupportedClassVersionError: Preview features are not enabled for Hello (class file version 57.65535)
    . Try running with '--enable-preview'
```

上网查询错误原因发现是打包 jdk 与运行的 jre 的版本不一致, 但是在按照提示更改命令为“java -jar --enable-preview 包名.jar”后, 程序运行正常, 如下:

(乱码推测为编码问题, 因为正常输出是中文)

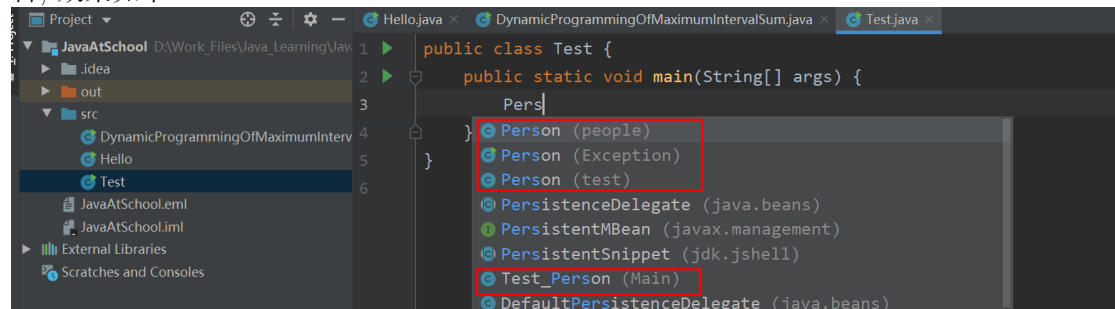
```
D:\Work_Files\Java_Learning\JavaAtSchool\out\artifacts\JavaAtSchool_jar>java -jar JavaAtSchool.jar
: DynamicProgrammingOfMaximumIntervalSum ^ LinkageError
    java.lang.UnsupportedClassVersionError: Preview features are not enabled for DynamicProgrammingOfMaximumIntervalSum (class file version 57.65535). Try running with '--enable-preview'

D:\Work_Files\Java_Learning\JavaAtSchool\out\artifacts\JavaAtSchool_jar>java -jar --enable-preview JavaAtSchool.jar
-2 -11 -4 -13 -5 -2
0
```

#### 4.4 在 idea 中导入 Jar 包

参考博客:(<https://www.jianshu.com/p/82353edd1240>)。仍然是在 Project Structure 中, 选择 Modules, 选择 Dependencies, 在右侧的“+”中选择 jar, 然后找到文件目录添加就可以了, 别忘了 Apply。

添加之后直接在项目中使用就可以了, 十分方便。我导入了一个之前练习用的 Java 项目, 效果如下:



## 5 总结

因为我寒假预习了很多, 环境也早早地配好了, 本想作业应该可以很快完成, 但是很多问题接踵而至, 难以预料。再加上最近在练习 Latex, 更是拖慢了进度, 以至于一篇作业用了两三天时间。在学习了 jar 包有关的知识后, 感觉豁然开朗, “原来小组之间是这样接洽的!”, 我原一直单纯以为是复制粘贴。于是深感自己知识之浅薄, 学习之路任重而道远。希望可以谦虚谨慎, 努力地学好学通 Java 这门课, 使之真正可以切实落实到实践之中。

“路漫漫其修远兮, 吾将上下而求索。”