

COA2023-programming01

初窥门径

0.前言

1. 获取作业的方法

编程作业使用**seecoder**平台（和软工一使用相同平台），选课码**2023COA**，加入课程后以相同的操作clone代码到本地，写完后push到作业仓库。

请注意：本次编程作业截止时间为2023-10-20 20:00

编程作业严禁抄袭！

附：seecoder平台网址：<http://114.212.203.205/exam>

有任何平台相关问题请联系编程助教解决。

1. 实验要求

本次作业在Transformer类中实现以下6个方法，实现数据表示的转换。所有数据均以String类型存储表示。

1. 将整数真值（十进制表示）转化成补码表示的二进制，默认长度32位

```
public String intToBinary(String numStr)
```

2. 将补码表示的二进制转化成整数真值（十进制表示）

```
public String binaryToInt(String binStr)
```

3. 将十进制整数的真值转化成NBCD表示（符号位用4位表示）

```
public String decimalToNBCD(String decimal)
```

4. 将NBCD表示（符号位用4位表示）转化成十进制整数的真值

```
public String NBCDToDecimal(String NBCDStr)
```

5. 将浮点数真值转化成32位单精度浮点数表示

- 负数以"-"开头，正数不需要正号
- 考虑正负无穷的溢出（"+Inf", "-Inf", 见测试用例格式）

```
public String floatToBinary(String floatStr)
```

6. 将32位单精度浮点数表示转化成浮点数真值

- 特殊情况同上

```
public String binaryToFloat(String binStr)
```

2. 实验攻略

本次实验推荐使用的库函数有

```
Integer.parseInt(String s)

Float.parseFloat(String s)

String.valueOf(int i)

String.valueOf(float f)
```

本次实验不允许使用的库函数有

```
Integer.toBinaryString(int i)

Float.floatToIntBits(float value)

Float.intBitsToFloat(int bits)
```

3. 实验指导

如果你对于实验要做的内容一头雾水，可以来参考实验指导的注意事项，但请务必独立完成！

下面的序号与实验要求中的题目一一对应

1. 可以考虑将某些操作过程封装，以便在不同方法间重复利用，减少代码冗余。
请注意int表示的范围和32位长度的范围。
2. 注意符号的正负表示，同时和1中一样，考虑int表示是否会溢出
3. NBCD（又叫8421码），保证提供数据不会溢出，请确保NBCD表示一定是32位。
4. 请注意32位NBCD码的格式。
5. 请参考IEEE 754标准下的32位浮点数格式：
维基百科提供的[IEEE 754](#)标准参考
请注意各种32位浮点数中可能存在的特殊情况。
6. 请同样注意各种32位浮点数中可能存在的特殊情况。

4. 调试指南

这将是你在日后coding中几乎最常使用的功能，甚至往往debug都会消耗比coding本身更长的时间，请你善用这一工具。

以下内容是如何使用IDEA debugger的简易说明，如果使用其他ide，请自行搜索如何启动调试功能。其他ide的调试类型应该与IDEA大致相同，以下内容也可以作为参考。

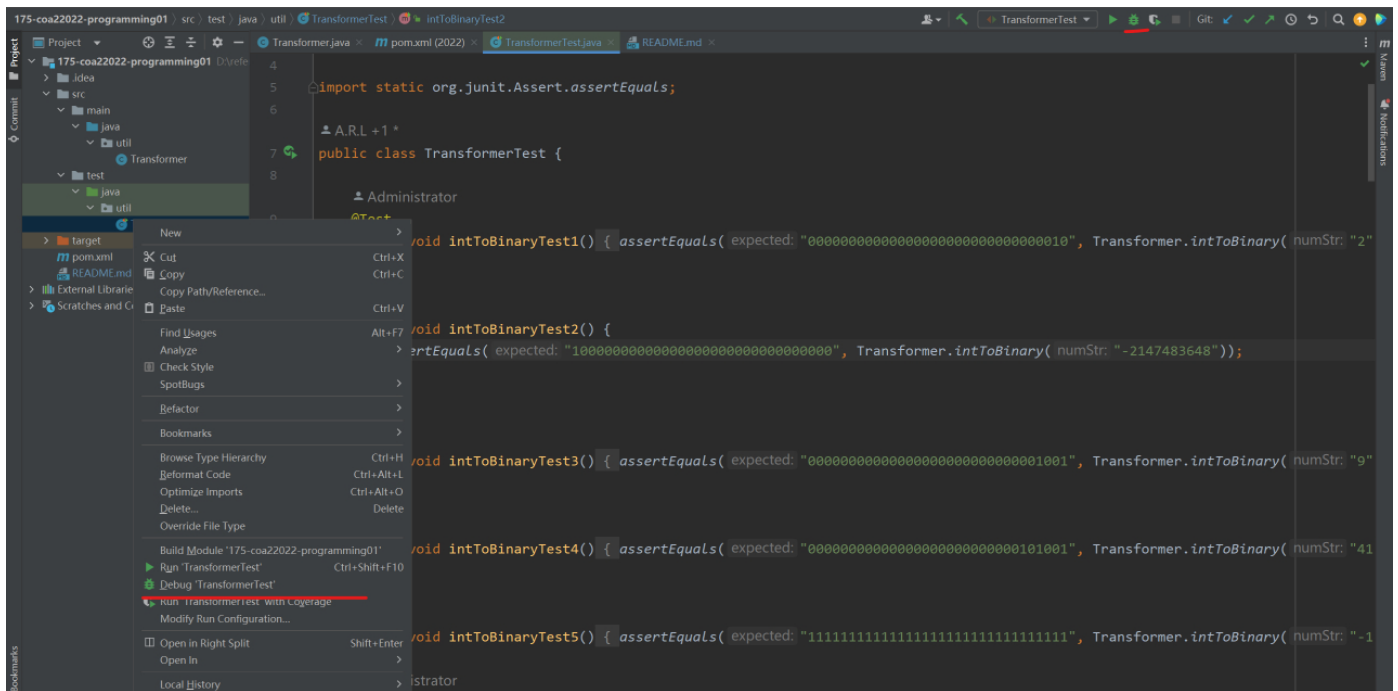
鼓励自己在测试文件中撰写用例对自己的代码进行更全面的测试！

1) 启动debugger

1. 在需要调试的地方点击行号（或行号右侧）打上红色断点。

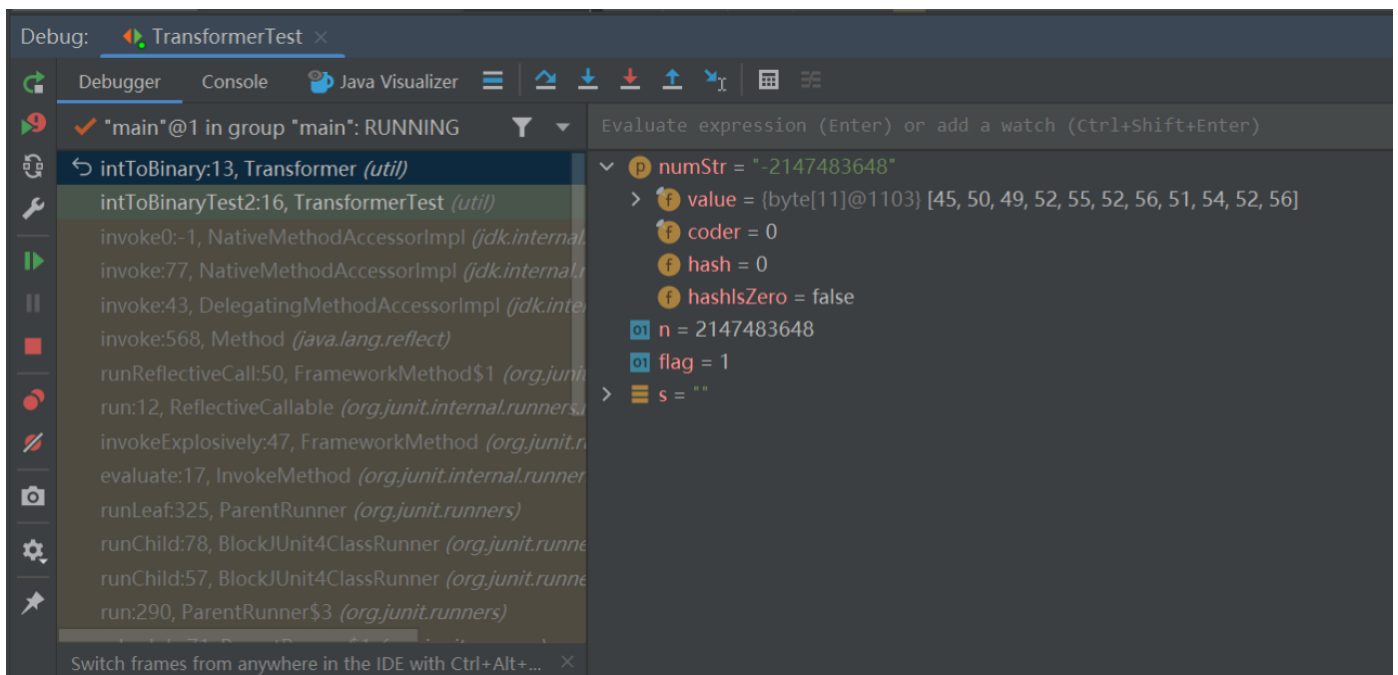
断点解释：启动debug程序后，程序会在设置断点的位置停止，即不执行当前行代码，详细内容：<https://www.jetbrains.com/help/idea/using-breakpoints.html>

2. 点击右上角三角（run）旁边的小虫子（debug）按钮，或者在展示文件的左边栏中，右键点击打断点的测试文件选择debug选项。

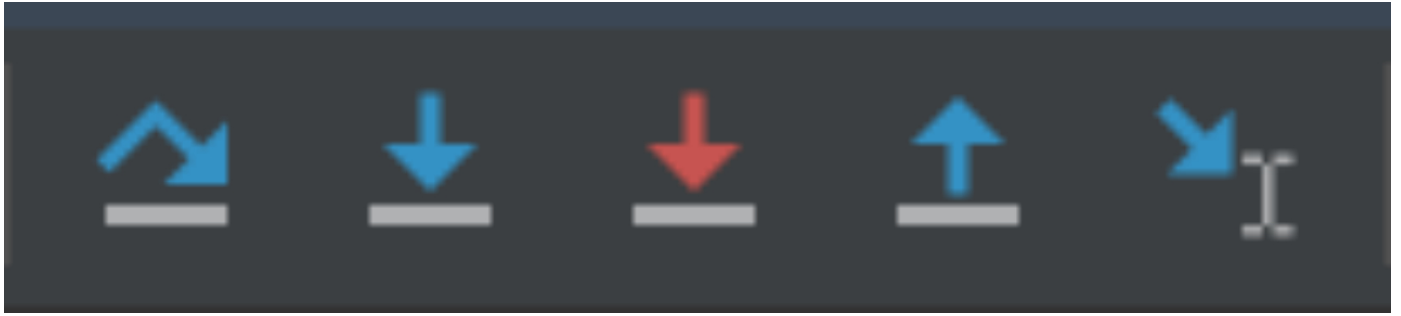


2) 调试过程

启动debug功能后界面如下图



Idea会检测当前变量值，通过点击下方这些按钮进行调试



左一：step over，执行当前这一步，如果存在函数不会跳入函数执行

左二：step into，跳入当前行包括的函数进行一步一步的调试

左三：force step into，强制跳入执行

左四：step out，用于跳出当前正在执行的函数

右一：run to cursor，跳到光标定位所在行

在实际debug操作中探索这些功能！ Good luck and have fun！