COA2021-programming03

1 实验要求

在ALU类中实现4个方法，具体如下

1.计算两个32位二进制整数补码真值的和

```
public DataType add(DataType src, DataType dest)
```

2.计算两个32位二进制整数补码真值的差，dest表示被减数，src表示减数(即计算dest - src)

```
public DataType sub(DataType src, DataType dest)
```

3.实现整数的二进制乘法(要求使用布斯乘法实现)。 输入和输出均为32位二进制补码，计算结果直接截取低32位作为最终输出

```
public DataType mul(DataType src, DataType dest)
```

4.实现整数的二进制除法 (dest ÷ src)。 输入为32位二进制补码，输出为32位商，并且将32位余数正确储存在余数寄存器remainderReg中。

注意：**请使用不恢复余数除法实现**；除数为0，且被除数不为0时要求能够正确抛出ArithmeticException异常

```
public DataType div(DataType src, DataType dest)
```

2 实验指导

2.1 代码实现要求

有些同学可能注意到，将传入的参数通过transformer转化为int，再通过整数的加减乘除运算后，将结果重新转化为DataType即可轻松完成实验。在此，我们**明确禁止**各位采用这种方法来完成本次实验。

2.2 数据封装

从本次实验开始，我们采用统一的类DataType来封装32位的二进制数，包括二进制补码整数、NBCD码与IEEE754浮点数。核心数据结构如下

```
    private final char[] data = new char[32];
```

采用这样的数据封装将保证DataType类中存放的一定是32位二进制数。并且我们提供了构造函数与toString函数，使得DataType对象与String对象之间的转化变得更加方便，具体可阅读DataType类源码。 需要注意的是，使用构造函数实现String对象向DataType对象转化时，如果你传入的参数不是32位的01串，构造函数会抛出NumberFormatException异常。