ADSOF ASSIGNMENT 1

**Part 3:**

3.1 ) Execute the program with different parameters e.g. numerical or not, negative, etc., without parameters or several ones. What happens? Do you see any problem or improvement to implement in this source code?

Tests:

No parameter:

Whenever the program is executed without parameters, an error saying “At least one number is expected” shows, this is because of the error control line written just after the main function.

Numerical:

The program prints two lines, one that says the number written as input and the amount of divisors it has, and the other one where it shows the treeset containing the divisors of such number

Decimal:

The program throws an exception java.lang.NumberFormatException because it considers the parameter as something which is not a numeric number.

Non numerical:

If a non numeric character is written as input, the program throws an exception java.lang.NumberFormatException that indicates that the program was expecting a number but got a non numeric character instead.

Negative:

The negative numbers are not valid since the program only computes the divisors from positive numbers, so it’s the same as putting a 0 as a parameter (which has 0 divisors and it is the unique number without divisors if we consider the negative numbers as correct parameters).

Operations:

The program does not accept operations as input. This is because it processes them as a string so it doesn’t execute the operation but tries to read the whole operation as a number instead resulting in an exception as the one it is given when a non numeric character is written as input.

More than one input:

The program supports more than one parameter since it calculates the divisors of each parameter one by one and then prints them.

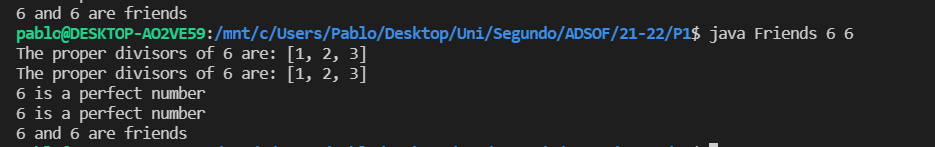
Problem and improvements:

* When generating the javadoc it generates a warning because in the constructor there is no parameter named number (as indicated in the comment) but it is named n instead.
* When negative numbers are written as input, the program does not return their divisors but returns an empty set instead. We could make it so the program also detects and processes the negative numbers.
* We should make catch the exception thrown by the invalid inputs and identify between the different types of errors shown (non numeric character, operation, decimal number…) showing a more comprehensible message of why the input written is not correct for the program.

**Part 4:**

4.1) Are perfect numbers friends of themselves? (provide argumentation using execution examples)

Yes, perfect numbers are always friends of themselves because the result of adding the divisor is the number itself so they both are the same number. Here we can see an example with the perfect number 6:



4.2) Are isPerfect and isFriend methods class or object methods? Justify your answer

They are object methods since the object methods are those that depend individually on each object while the methods class are the ones that depend on all the objects that the class has. An example of method class would be to get all the common divisors between all the Divisors elements created.