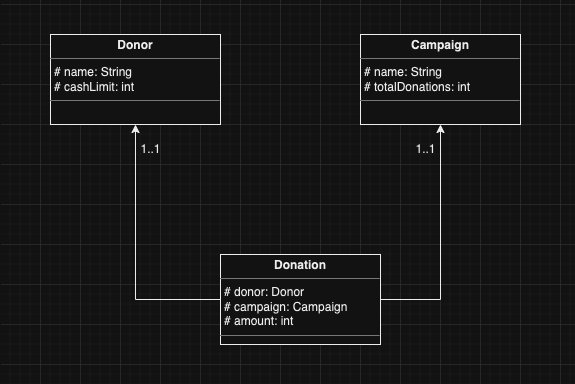
**GFM Code Submission**

**Candidate: Angel Matias Bouin**

***Objects relationship design or class diagram***



The program is designed with OOP and uses Java as a programming language.

We have 3 main classes: Donor, Campaign and Donation. A Donor will relate with a Campaign through a Donation. Each Donation will have one Donor and one Campaign.

For the rest of the classes we will have:

> Reader which will take care of reading the commands

> Processor which will take care of handling the data and processing the commands

> Logger which will log the final outcome of the run

> Utils with helper functions

First three arrays are initialized, Donors, Campaigns and Donations. The Reader class will intake the commands either from STDIN or file path through args[0]. Then it will extract the commands building an array and will pass that array to the Processor class, which will read the commands and populate the Donor, Campaign and Donation arrays. Once processing is finished these arrays will be passed to the Logger for it to build up the final output and print it to the console.

Included Unit Testing under UtilsTest covering two simple functions getDonor() and getCampaign() to make sure they always retrieve the right value.

***To run the program***

You will need Java to run this program. Open your Terminal and check your Java version with command:

java -version

It should return something like this:

java version "20.0.1" 2023-04-18

Java(TM) SE Runtime Environment (build 20.0.1+9-29)

Java HotSpot(TM) 64-Bit Server VM (build 20.0.1+9-29, mixed mode, sharing)

If you don’t have Java installed, follow [this documentation](https://www.java.com/en/download/help/download_options.html) to install it and continue with the following steps.

From your Terminal, navigate to the gfm-recurring folder I sent, then navigate to src/

Here you will see the source code for the implementation, including a couple UnitTests made with JUnit.

I included a sample test input.txt file with several commands for the program to run.

To run the test as it is, compile the program with command:

javac Main.java

Then you can run the program with STDIN with the following command:

cat input.txt | java Main

Or run it with an argument being the file path to the input.txt:

java Main input.txt

You can alter the given input.txt to your desired testing needs.

Adding a screenshot of how the output should display after compiling and running both STDIN mode and Argument mode.

