Exceptions Practice

Step 1: Creating the Student class

- Make name and average data members, they are private!

- Add constructor and getter/setter generation. The name should be read-only, the average should be readable and modifiable.

Step 2: ArrayList

- Let's create some Student instances and an ArrayList, add the students to the list.

- With a for loop: list the students one below the other by overriding toString in the Student class.

Step 3: PriorityQueue

- Create a PriorityQueue, add the students to it. Then print the content of the queue, who is at the lead? How can you reverse the order of the elements in the queue?

Step 4: TreeSet

- Let's create a TreeSet, fill in with the students (you don't need to add them individually, addAll is perfectly fine). Implement Comparable interface: (otherwise you will encounter ClassCastException) we compare based on average. If two students have the same average, add only one of them!

- List those with an average higher than 3.0 in ascending order of average. (The goal is to use either stream().filter or tailSet().)

Step 5: Exception

- Create an IncorrectAverageException class, which should be the subclass of the Exception class.

- Check if the setAverage() method of the Student class receives a negative number, it should throw an IncorrectAverageException exception.