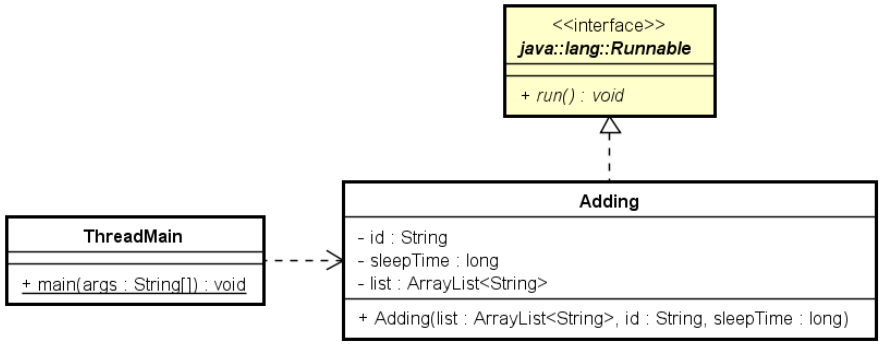
(Exercise 01.01)

Implement the following system (see below)



A class `Adding`:

- Three instance variables, an ArrayList, an id, and a sleep time in milliseconds.
- A constructor setting all instance variables
- A method `run()` adding 5 String elements. In a loop, add a string marked with id and number, print out the id and the entire list, sleep for `sleepTime` milliseconds and go to the next loop cycle.

A main class `ThreadMain`:

- Create an ArrayList for string elements
- Create one thread with `Adding` as the Runnable
- Start up the thread and wait for the thread to finish before you print out the entire list and the number of elements (the list size)

Run the program a few times and inspect the output.

Exercise 01.02 A

Change the previous exercise such that you start up 3 threads in `main` (give different id's and different sleep times) and then run the program a few times to inspect the output (add print statements in the `run` method to see which thread is adding an item and print the full queue also)

Can you anticipate the order? Is the list size always 15?

Example run:

```
id=A, list=[A#1]
id=B, list=[A#1, B#1]
id=A, list=[A#1, B#1, A#2]
id=C, list=[A#1, B#1, A#2, C#1]
...
[A#1, B#1, A#2, C#1, A#3, B#2, A#4, A#5, B#3, C#2, B#4, C#3, B#5, C#4, C#5]
count=15
```

Exercise 01.02 B

Change the previous exercise such that you start up 3 threads all with the same sleep time. Run the program a couple of times and explain why the result is the way it is. (Are any of the updates lost?)