Tutorial problems – Wed 13 Nov and Fri 15 Nov 2019

10min Ask whether there are any questions. Collect these by writing them on the board and decide with the whole group which of these to address in the 10 minutes available for question answering. The answering of the question may also be put to the end of the tutorial.

10min Students to work in small groups (of 2 or 3). The method `public static double min(ArrayList<Measurable> a)` is given as

```
public static double min(ArrayList<Measurable> a) {
    double min = a.get(0).getMeasure();
    /* The loop invariant is that the variable min always contains
     * the minimal value seen so far.
     */
    for (int i = 1; i < a.size(); i++){
        if (a.get(i).getMeasure() < min) {
            min = a.get(i).getMeasure();
        }
    }
    return min;
}
```

They should compare the following two methods to compute the values above a minimum (e.g., in order to display the values in some graphics).

```
public static ArrayList<Double> aboveMin1(ArrayList<Measurable> a) {
    double min = min(a);
    ArrayList<Double> result = new ArrayList<Double>();
    for (Measurable el : a) {
        result.add(el.getMeasure() - min);
    }
    return result;
}
public static ArrayList<Double> aboveMin2(ArrayList<Measurable> a) {
    ArrayList<Double> result = new ArrayList<Double>();
    for (Measurable el : a) {
        result.add(el.getMeasure() - min(a));
    }
    return result;
}
```

(It took the code for a randomly generated list of 100000 elements 26ms for the first method and 19090ms for the second, for instance. Explain.)

[There is a second example in the code in which the mean and the standard deviation of an `ArrayList` of type `Measurable` is determined. This example should be presented only to advanced tutorial groups, e.g., to a group with mathematics background.]

p.t.o.

30min The students should again work in small group of 2 or 3 to address the following problem:

Assume the `CreditCard` class from the tutorial of week 6 with fields `private String name`, `private String accountNumber`, and `private int amount` and the `GoldCard` subclass with the additional field `private int fee`.

- Write a method

  ```
  public static double average(ArrayList<CreditCard> cList)
  ```

  which computes the average debt on the list of credit cards. Furthermore write a method

  ```
  public static ArrayList<CreditCard>
        filter(ArrayList<CreditCard> cList,
               Function<CreditCard, Boolean> predicate)
  ```

  which filters a list of credit cards with the given predicate.

  Furthermore write two predicates to determine sublists of all those credit cards which are gold cards and all those which are not.

  Discuss with the students the difference of modifying the existing `ArrayList` vs creating a new `ArrayList`. In this case a new `ArrayList` should be created since we do not want to lose the original list.