

# Homework Assignment

## 6 Reaction Experiment

You are going to build a program to experiment with reaction times.

The user is presented with a small red circle. After some time (the length varies), the circle turns yellow. The user has to press a mouse button as quickly as possible after he sees the yellow circle.

The program runs five of these tests. At the fifth test, the program shows a really big yellow circle instead of a small one. This will startle the user and has in many cases the effect that the reaction time is a little bit shorter than the previous ones. An explanation is that in the first reaction you make a conscious reaction, whereas in the startled reaction you react directly with the brain stem. You can try to train yourself to use this type of reaction and be faster in sports and games. Even after a few experiments with this program, you may observe this effect.

Make an application of one window (a `JFrame`) with a button that starts or resets the experiment, a large panel that shows the circles (the so-called reaction panel), and an areas (on six labels, a `JTextArea`, etc.) that shows the reaction times and the average reaction time. The reaction time is the time in milliseconds between the moment the yellow circle is shown and the moment the user clicks in the large panel. One experiment consists of five tests. Each test consists of:

1. showing a small red circle in the reaction panel;
2. replacing the red circle with a yellow circle after a random period of so-called delay time;
3. showing the reaction time and starting a new test, except after the fifth test;
4. after the fifth test, the program is inactive and just showing the reaction times and their average, waiting for the start button to be pressed.

Experiment with the delay time. When testing your program, it may be practical to use a short period as a delay time. In the final version, the delay should be longer and have a relatively large variation, to make the reaction test harder. Think of a minimum of five seconds.

### 6.1 Hints about class design and other things

Use a central class (e.g., `ReactionExperiment`) for the application where most of the work is done. Use a separate class for the reaction panel, where `paintComponent` is overridden.

In `ReactionExperiment`, make separate methods for the phases as described above. Have these call methods in `ReactionPanel` to change the picture (the circle). Note that this change is always indirect: instance variables of `ReactionPanel` are changed and then `repaint` is called, which will have the effect that the Swing system calls `paintComponent`.

A time stamp of the current time can be obtained by `System.currentTimeMillis()`. This method returns a `long` (a variant of `int` that can store larger values).

The timer is started when the red circle appears and at the first firing of the timer the yellow circle should appear. The time of the first firing is determined by `setInitialDelay`. Use this method, not `setDelay`. Use a different (random) delay at each test.

The method call `Math.random()` gives a random value between 0 (inclusive) and 1 (exclusive). By adding and multiplying, a random value between any two numbers can be obtained. It returns a `double` value. When an `int` value is needed, use a so-called cast (`int`) that rounds a `double` down to an `int`.

For instance, `(int)(1 + Math.random() * 6)` gives a random integer number between 1 and 6 (like a roll with a dice).

## **6.2 Approach**

Start with a version that does not show the big yellow circle and simply prints the reaction times to the console.

When that works, add the big yellow circle and later add the labels with the reaction times and the average reaction time.

## **6.3 Grading**

A good class design with short, coherent methods will be rewarded. Up to 1.5 points will be awarded for extras in design or functionality.