Chapter 11

Exercise 1

In this exercise, you will practice some basics of using GUIs.

1. Write an application that displays a frame with a single button.
2. Make sure that the application closes when  is clicked.
3. Reset the size of the window in your program. Keep in mind how the setSize, setPreferredSize and pack functions work (together).
4. Add labels to the frame.
5. Try out different layouts for the components (i.e., your labels and buttons), such as no layout, a FlowLayout, a BorderLayout and a GridLayout.
6. Implement an ActionListener for the button, that makes the application close when it is clicked.

Exercise 2

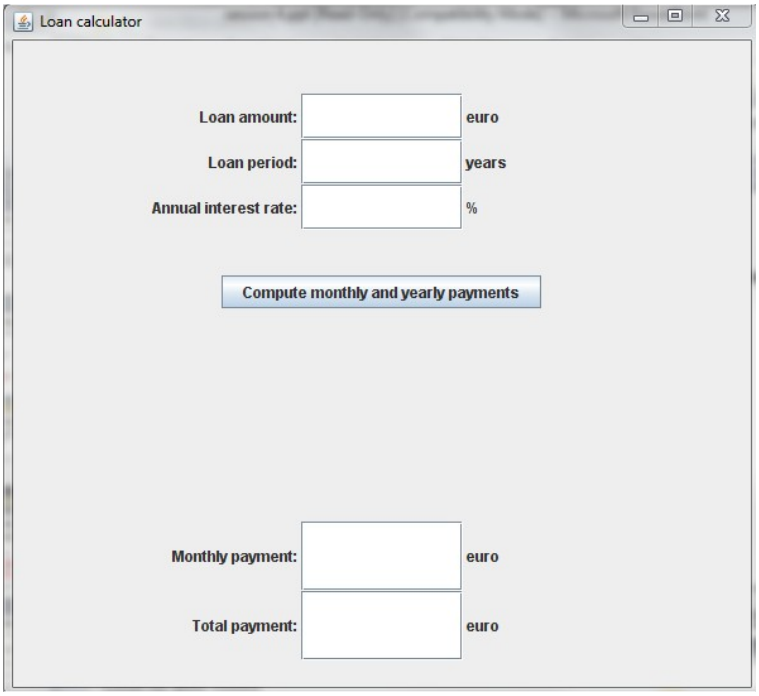
Create a graphical user interface for the BMI Calculator. The interface should contain two fields where the user can enter his height and weight, and a button, which when pressed, displays the users BMI. Make sure to handle unexpected (or empty) input.

Exercise 3

Develop a Java GUI application that computes both monthly and total payments for a given loan amount, annual interest rate, and loan period. The application should be event driven, which means that the user can click, fill out a number, resize the window, etc.

Implement a “Reset” button, which clears all fields when clicked.

You can use the figure as inspiration for how the calculator can look like.



Note: Formulas for calculating monthly payments are included below.

$$Monthly\ Interest\ Rate = \frac{Annual\ Interest\ Rate}{100 \cdot Months \in Year}$$

$$Number\ of\ Payments = Loan\ Period \cdot Months \in Year$$

$$Montly\ Payment = \frac{Loan\ Amount \cdot Monthly\ Interest\ Rate}{1 - (1 + Montly\ Interest\ Rate)^{-Number\ of\ Payments}}$$

$$Total\ Payment = Monthly\ Payment \cdot Number\ of\ Payments$$

Exercise 4

Design and implement a GUI for a quiz with at least 6 questions.

Each question of the quiz should appear in a separate frame, with a button to go to the next question. The questions can be multiple choice (in which case there's only one correct answer) or open (in which case a text field is provided to write the answer (and exception handling should be implemented to take care of unwanted input)). Make sure you have at least one question of each type. If you can, try to incorporate different GUI components for answering questions, such as Combo Boxes and Spinners.

At the end of the quiz, the user should be presented with his final score.

Example of questions and answers

1. What color is the sun?
    a. Blue
    b. Red
    c. Yellow
2. What year is it?
    a. 2016
    b. 2017
    c. 2018
3. How high is Mount Everest in meters?
    - Spinner with values from 0-10000?
4. In Friends, what is the name of Monica's brother?
    - Text Field