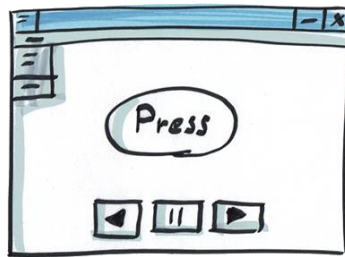


## Software Construction 2 (IE-B2-SO2)

### Lab Exercise 3: Geographic Routes (GUI)



## 1 General Notes

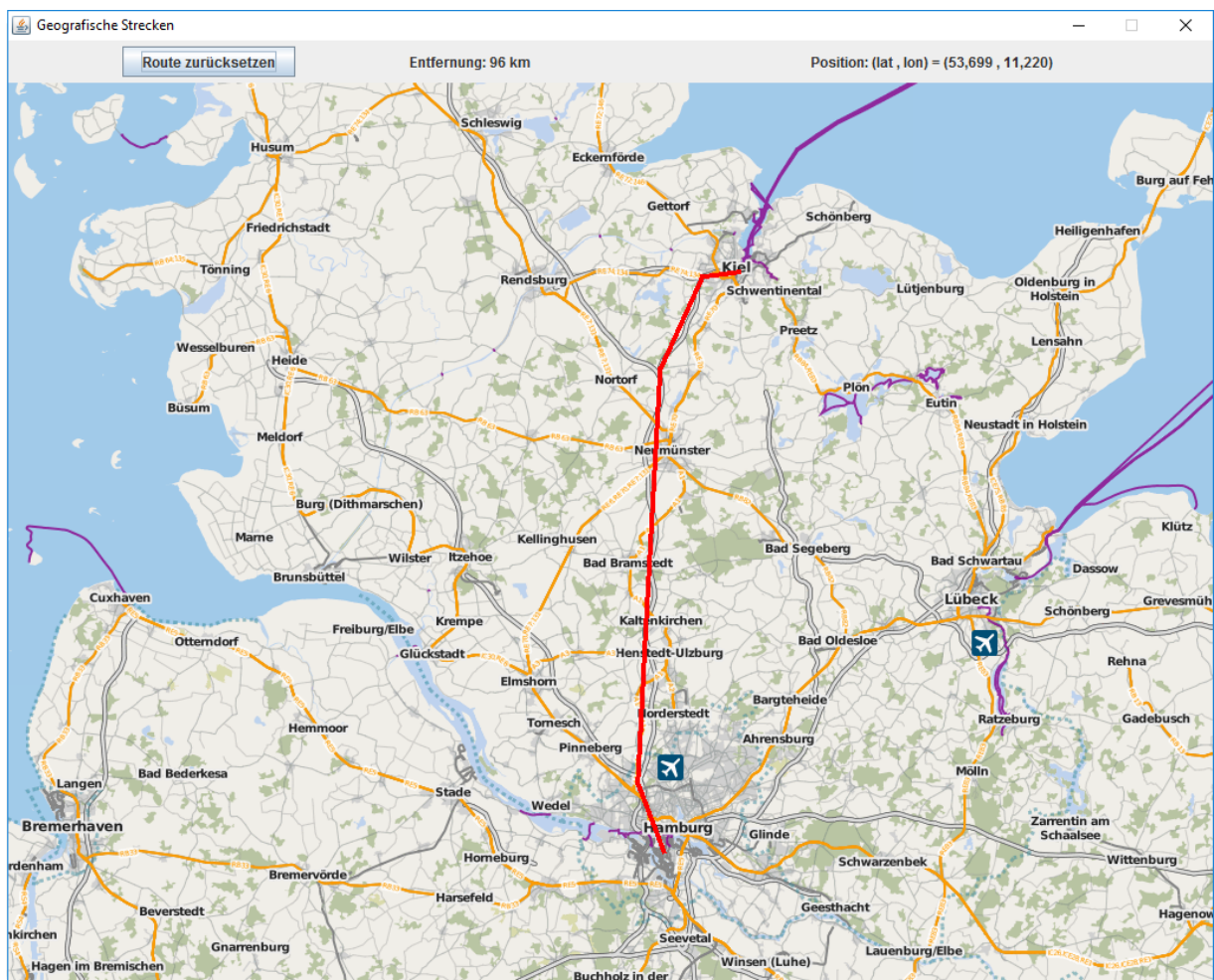
The instructions given in lab exercise 1 apply. Please note also:

- Primary Learning Objectives: Graphical User Interfaces (GUI).
- For the source code, please create a package called *lab3.geoPosition*.

## 2 Overview

In this lab exercise you will develop a *Graphical User Interface (GUI)*. There are no requirements regarding both the classes to be implemented and the exact design of the GUI. Instead, it is only important that your program fulfills the functionality described in Section 3.

The following figure shows one possible solution for the application to be developed. The GUI includes a map in which the user can define a route consisting of one or several parts (here a route from Hamburg to Kiel is shown in red color). A defined route may be deleted. It is your decision whether you implement this functionality, e.g., using a button and/or a menu item. The example solution additionally displays the total length of the route defined as well as the mouse position in geographical coordinates.



### 3 Exercise

Create a GUI with the following functionalities. Please note that only the requirements defined in Section 3.1 are mandatory. Please also consider the hints given in Section 3.3 as well as the solution strategy described in Section 4.

#### 3.1 Mandatory

- The application uses the map provided in file *OSM\_Map.png*<sup>1</sup>.
- Users can select waypoints by left clicking the mouse on a position on the map.
- Waypoints create a route whereas new waypoints are added at the end of the route.
- The route will be displayed correctly in the map.
- Users can delete the defined route.

#### 3.2 Optional

The following requirements are not needed to be fulfilled to pass the lab. However, they are a good way to practice if you have some time left and are looking for a challenge.

- The application displays the current position ( $x, y$ ) of the mouse on the map.
- The application displays the current mouse position in geographical coordinates ( $lat, lon$ ).
- The application displays the total length of the route in kilometers.

#### 3.3 Hints

- Map:
  - Read the file with *ImageIO.read()* creating an object of type *BufferedImage* and use *drawImage()* for drawing.
  - Eclipse uses the directory of the Java project for reading and writing. Thus, please add the image file (*OSM\_Map.png*) using for example drag&drop (from explorer to package *lab3.geoPosition*) and use the relative file path (*lab3/geoPosition/OSM\_Map.png*) for reading.
- Mouse events: Implement a *MouseListener* and add it to your panel with *addMouseListener()*. To respond to mouse movements, use *MouseMotionListener* and *addMouseMotionListener()*.
- Conversion to geographic coordinates: The file *OSM\_Map.txt* includes the longitude and latitude of both the upper left and lower right corner of the map.
- Modify the line width: Cast the *Graphics*-reference to a reference of type *Graphics2D*. Modify the line width with *setStroke(new BasicStroke(x))* whereas  $x$  is the new width.

---

<sup>1</sup> Free OSM-map (<http://www.openstreetmap.de/>), created with MOBAC (<http://mobac.sourceforge.net/>)

## 4 Solution Strategy

- First, design and implement the appearance of the graphical interface (i.e. the type and layout of the elements).
- Implement the required functionality in the second step.
- Do not get into details about the appearance of the graphical interface. (For particularly creative and aesthetic solutions, there is admiration and enthusiasm, but no beauty award.)

## 5 Compliance with Coding Guidelines (Quality)

You already know it: Ensure that all quality criteria given by the software quality checklist (provided in EMIL) are met.

In this experiment, it is also advisable to have your source code reviewed by another group and in turn read their source code. Discuss your impressions with the other group in particular:

- Was it easy to understand the source code of the other group?
- What was the key to making the source code understandable or not easy to understand?
- How could the understandability be increased?