NOVEMBER 2018

French-Azerbaijani University

SWING MINI-PROJECT REPORT

Emin Sultanov
Kamal Aghayev

MAIN IDEA

The main idea of this project is a clear understanding of how to work with GUI in Java and displaying the work of the threads

Problems to solve

- With the usage of graphics draw the "steps" of threads' working.
- Make the animation of threads' working with changing of colors.

Steps of solving:

The implementation of this project is made by 4 classes:

- DataSender, which is an abstract class
- Thread
- Buffer
- Main

We started by creating our abstract class (**DataSender**). There we defined the color variable that is by default green, which is the color that means the thread or buffer is empty. We also defined the variable toSend to know how many messages a thread or a buffer has. To create the variable of DataSender class we made constructors to define the position and diameter of our figures. Afterward, there had to be a receiver to make information "move". The last method of this class is the draw method that draws a thread or a buffer.

The second step was the creation of the **Thread** class that extends DataSender class.

The animation is implemented by three methods: update, save and receive. The first, update, method updates the state of a thread. The second, send, method checks if there is any information to send and sends, updates receiver, make receiver receive the information. and make the current thread empty if there is. The last, receive, method changes the color of the thread to the red which means that it has the information to send (toSend=1).

The next step was the creation of the Buffer class that extends DataServer class. The implementation of the meanings of the methods is the same but the implementations a little differ. We added in each method if clause that compares nextReceive seconds with current time seconds. It needs in the update method to check if the last message has processed the buffer is ready for a new one, in the send it checks if it is free and then send a message and the same for receive method.

Main implementation

The last step was the creation of Main class where our main method and all the process are. The Main class implements ActionListener interface for repainting animation and extends the JPanel class to to be able to draw. Afterward, we defined the threads array of the Thread class and the buffers array of the Buffer class. Also, we set two constant variables of the width and height variable for our window. For full creation the window and all figures we needed to create a method to paint on a canvas and there we used a timer for making animation. To show the direction of sending the information we used arrows and created them with the arrow method, where we defined the position the beginning and the end of it. In the main method, we have set the position of each thread and buffer while defining them and set the order of receiving. Also, we made window "visible" for users. At the end, we defined the action that will be performed at each frame which is a repaint.