|  |
| --- |
| **Lab09: GUI with Netbeans** |

To gain experience with:

1. **Graphical User Interface in Netbeans**
2. **Ohm's Law Formula**
3. **Multiplication Table**
4. **Food Menu**
5. **Exercise for practice**

1. Graphical User Interface in Netbeans

**Properties:**

Properties of an object are used to specify its appearance on the form. For example to set the background colour of a textfield you change its background property; to set its font you change its font property; and so on.

**Methods:**

Methods are used to perform some action on the object. For example to display something in a textfield you can use its setText() method, to extract the contents of a textfield you can use its getText() method. Methods can be divided into two categories getters and setters.

**Events:**

Events are the actions which are performed on controls. Examples of events are: mouseClick, mouseMoved, keyPressed etc. When the user performs any action on a control, an event happens and that event invokes (sends a call to) the corresponding part of the code and the application behaves accordingly.

2. Ohm’s Law Formula Calculation

|  |  |
| --- | --- |
| Create a Java GUI application that provides the user usage of Ohm’s Law formula. In electrical circuits, Ohm's law states that the current through a conductor between two points is directly proportional to the potential difference or voltage across the two points, and inversely proportional to the resistance between them, provided that the temperature remains constant.  The mathematical equation that describes this relationship is:    where V is the potential difference measured across the resistance in units of volts; I is the current through the resistance in units of amperes and R is the resistance of the conductor in units of ohms. More specifically, Ohm's law states that the R in this relation is constant, independent of the current.  The law was named after the German physicist Georg Ohm, who, in a treatise published in 1827, described measurements of applied voltage and current through simple electrical circuits containing various lengths of wire. He presented a slightly more complex equation than the one above to explain his experimental results. The above equation is the modern  form of Ohm's law. |  |
|  |

Double-click on the Calculate JButton—this should take you in the Source code calculateJButtonActionPerformed method. Delete the comment line and type the following code:

|  |
| --- |
| // Calculate the current given voltage and resistance  DecimalFormat numberFormatter = new DecimalFormat("#,##0.0##");  float volts, resistance, current;  volts = Float.parseFloat(voltsJTextField.getText());  resistance = Float.parseFloat(resistanceJTextField.getText());  current = volts/resistance;  currentJTextField.setText(numberFormatter.format(current)); |

Double-click on the Clear button and add the following code:

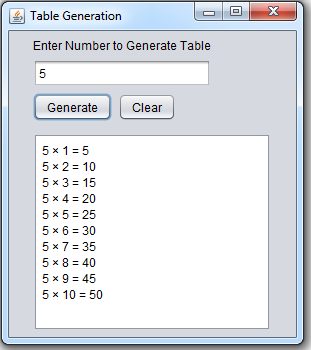
|  |
| --- |
| // Clear all text fields to reset the form  voltsJTextField.setText("");  resistanceJTextField.setText("");  currentJTextField.setText("");  voltsJTextField.requestFocus(); |

Double-click on the Quit button and add the following code:

|  |
| --- |
| // Terminate the application  System.exit(0); |

3. Multiplication Table

Create a GUI application that prints the multiplication table of any given number.



Double-click on the Generate button and add the following code:

|  |
| --- |
| private void btnGenerateActionPerformed (java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  int base, i, n;  base = Integer.parseInt(txtTableValue.getText());  txtAreaTable.setText("");  for (i=1; i<=10; i++) {  n = base\*i;  txtAreaTable.append(base + " × " + i + " = " + n + "\n"); }  } |

Double-click on the Clear button and add the following code:

|  |
| --- |
| private void btnClearActionPerformed(java.awt.event.ActionEvent evt){  // TODO add your handling code here:  txtTableValue.setText("");  txtAreaTable.setText("");  } |

4. Food Menu

Create a GUI application that calculate the total price of items with respect to quantity selected.



Double-click on the Burger checkbox and add the following code:

|  |
| --- |
| private void chkBurgerActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  if(chkBurger.isSelected()){  cmbBurgerQuantity.enable(true);  int a = Integer.parseInt(lblBurgerPrice.getText());  int b = cmbBurgerQuantity.getSelectedIndex()+1;  txtBurgerTotalPrice.setText(""+(a\*b));  }  else {  txtBurgerTotalPrice.setText("0");  cmbBurgerQuantity.setSelectedIndex(0);  cmbBurgerQuantity.enable(false);  }  } |

Double-click on the Burger combobox and add the following code:

|  |
| --- |
| private void cmbBurgerQuantityActionPerformed (java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  if(chkBurger.isSelected()){  int a = Integer.parseInt(lblBurgerPrice.getText());  int b = cmbBurgerQuantity.getSelectedIndex()+1;  txtBurgerTotalPrice.setText(""+(a\*b));  }  } |

Double-click on the Clear button and add the following code:

|  |
| --- |
| private void btnClearActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  chkBurger.setSelected(false);  txtBurgerTotalPrice.setText("0");  cmbBurgerQuantity.setSelectedIndex(0);  cmbBurgerQuantity.enable(false);  } |

Note: Repeat the above code for the other items of menu like chips, rice and coke.

5. Exercise: Netbeans Editor Design

Design an exactly the same GUI of Netbeans editor as shown in the figure below:

