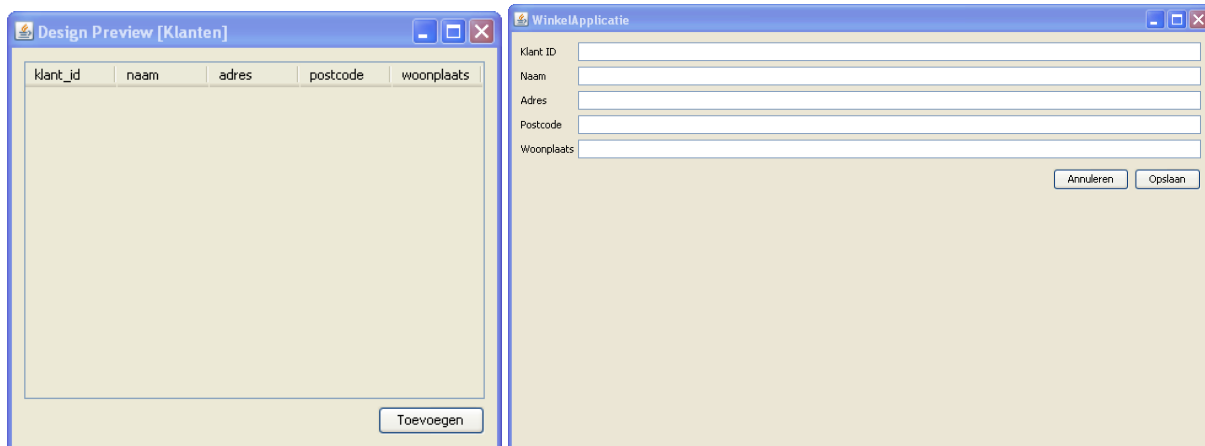


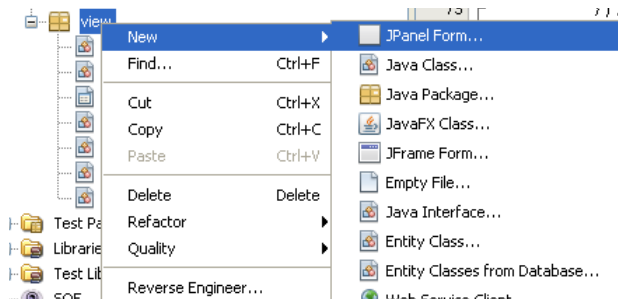
WinkelApplicatie KlantenBeheer

Na het maken van een menu, kunnen we beginnen met het maken van nieuwe functionaliteiten. Een van de onderdelen die nog gemaakt moet worden is het beheer van klanten. We gaan een scherm maken met een overzicht van alle klanten en maken een formulier om nieuwe klanten toe te voegen.

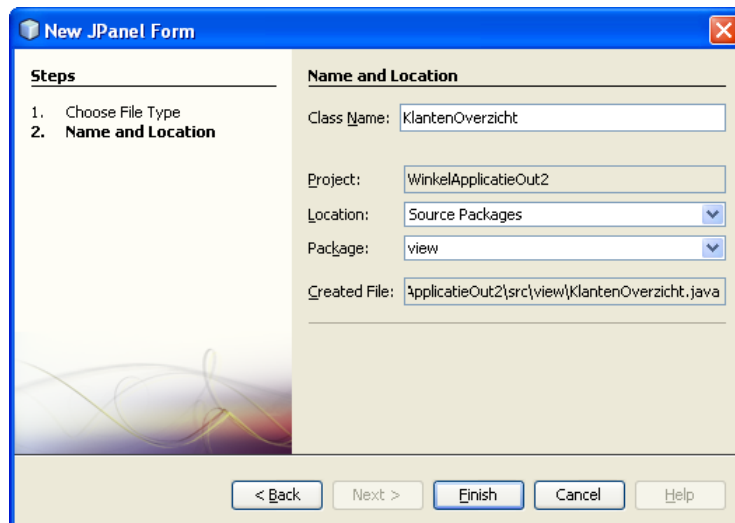


Klanten overzicht

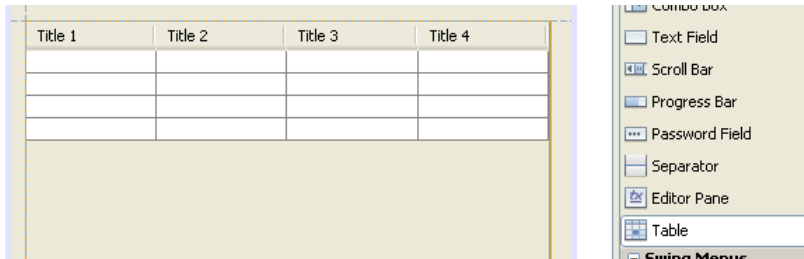
Het scherm KlantenOverzicht zal alle klanten weergeven in een tabel.



1. Klik met de rechtermuisknop op de package view en kies voor New -> JPanel Form...



2. Geef de klasse de naam KlantenOverzicht en klik op Finish



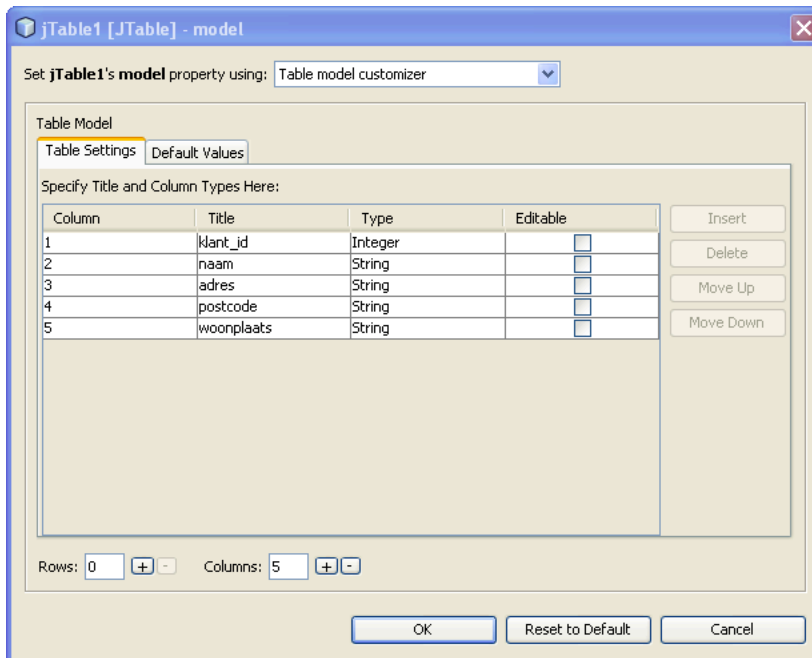
- Plaats een Table.



- Plaats onder in het scherm een button.



- Klik met de rechtermuisknop op de table en open de properties. Klik op ... om het model aan te passen.



- Pas het model aan zoals hierboven weergegeven.



- Verander de tekst van de button door hem te selecteren en op F2 te drukken.



- Klik op de knop Source, om de code van het scherm te openen.

```

26  /** Creates new form KlantenOverzicht */
27  public KlantenOverzicht () {
28      initComponents();
29      List<Klant> klanten = WinkelApplication.getQueryManager().getKlantList();
30      DefaultTableModel model = (DefaultTableModel) this.jTable1.getModel();
31      for(Klant klant : klanten) {
32          model.addRow(new Object[] {new Integer(klant.getKlantId()),
33              klant.getNaam(),
34              klant.getAdres(),
35              klant.getPostcode(),
36              klant.getWoonplaats()});
37      }
38  }

```

9. Pas de constructor aan, zodat de klanten worden toegevoegd aan de tabel.

```

127 public List<Klant> getKlantList () {
128     List<Klant> klanten = new ArrayList<Klant>();
129     try {
130         String sql = "SELECT * FROM klant";
131         ResultSet result = dbmanager.doQuery(sql);
132         while (result.next()) {
133             klanten.add(new Klant(result.getInt("klant_id"),
134                 result.getString("naam"),
135                 result.getString("adres"),
136                 result.getString("postcode"),
137                 result.getString("woonplaats")));
138         }
139     } catch (SQLException e) {
140         System.out.println(Dbmanager.SQL_EXCEPTION + e.getMessage());
141     }
142     return klanten;
143 }
144

```

10. Open de klasse QueryManager en voeg de methode getKlantList toe.

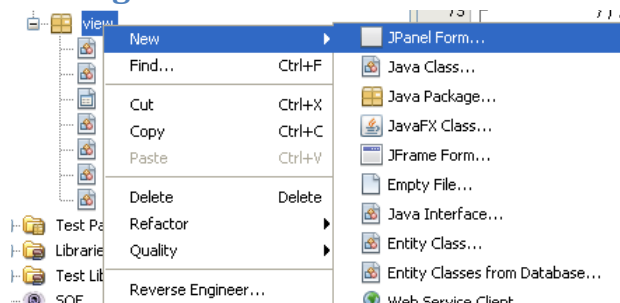
```

82 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
83     WinkelApplication.getInstance().showPanel(new KlantenOverzicht());
84 }

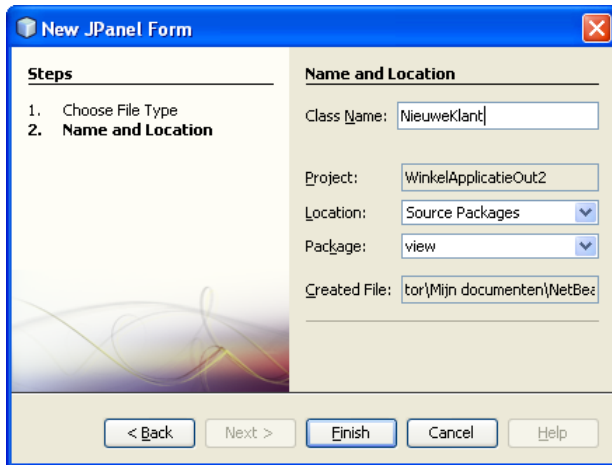
```

11. Open de klasse MainMenu en dubbelklik op de knop Klanten. Pas de methode aan, zodat na het klikken op de knop Klanten het klantenoverzicht wordt getoond.

Klant toevoegen

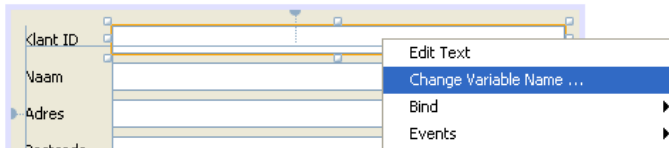


1. Klik met de rechtermuisknop op de package view en kies voor New -> JPanel Form...



2. Geef de klasse de naam NieuweKlant en klik op Finish.

3. Plaats labels, textfields en buttons om een invoer formulier van een klant te maken.



4. Klik met de rechtermuisknop op een textfield en kies voor Change Variable Name... Verander de namen naar tfKlantId, tfNaam, tfAdres, tfPostcode en tfWoonplaats.

```

126 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
127     WinkelApplication.getInstance().showPanel(new MainMenu());
128 }

```

5. Dubbelklik op de knop Annuleren en pas de code aan.

```

136 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
137     Klant klant = new Klant();
138     klant.setKlantId(Integer.parseInt(tfKlantId.getText()));
139     klant.setNaam(tfNaam.getText());
140     klant.setAdres(tfAdres.getText());
141     klant.setPostcode(tfPostcode.getText());
142     klant.setWoonplaats(tfWoonplaats.getText());
143     WinkelApplication.getQueryManager().addKlant(klant);
144     WinkelApplication.getInstance().showPanel(new MainMenu());
145 }

```

6. Open de design view door op de knop Design te klikken. Dubbelklik op de knop opslaan en pas de methode aan.

```

143 public void addKlant(Klant klant) {
144     String SQL_klant = "INSERT INTO `klant` (klant_id, naam, adres, postcode, woonplaats)"
145         + " VALUES(' " + klant.getKlantId() + ", ' " + klant.getNaam() + ", ' "
146         + klant.getAdres() + ", ' " + klant.getPostcode() + ", ' "
147         + klant.getWoonplaats() + ")";
148     dbmanager.insertQuery(SQL_klant);
149 }

```

7. Open de klasse QueryManager en voeg de methode addKlant toe.

```

107 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
108     WinkelApplication.getInstance().showPanel(new NieuweKlant());
109 }

```

8. Open het scherm KlantOverzicht en dubbelklik op de knop Toevoegen. Pas de methode aan, zodat het scherm NieuweKlant wordt weergegeven.