**­­­Anforderungsspezifikation:**

Die Klasse Kunde hat das Attribut name (Typ: String, Defaultwert: “Unbekannt“) und das Attribut kundenNr (Typ: int, reguläre Werte zwischen 10000 und 99999, Defaultwert: 0) mit den entsprechenden Setter- und Getter-Verwaltungsmethoden. Irreguläre Wertzuweisungen werden ignoriert. Magic-Values sollen als öffentliche Konstanten deklariert werden.  
Der parameterlose und der vollparametrisierte Konstruktor sind vorzusehen.  
Die überschriebene Methode toString() liefert die Kundennummer und den Namen des Kunden als String zurück (getrennt durch ein Leerzeichen), die überschriebene Methode equals() vergleicht die Kundennummern.

**Vorgegebene Klassen-Fragmente:**

---------------------------------------------------------------

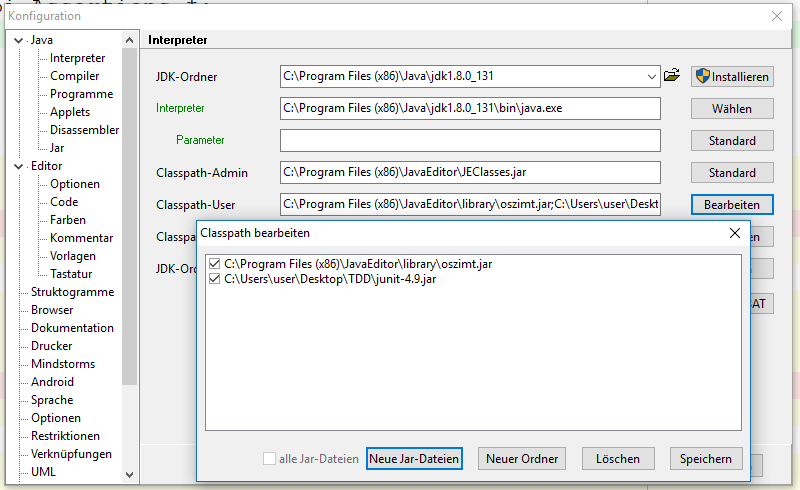
public class Kunde {

}

---------------------------------------------------------------

//import junit.framework.TestCase; // JUNIT 3: KundenTest extends TestCase  
import org.junit.\*; // JUNIT 4: für alle Annotationen  
import static org.junit.Assert.\*; // JUNIT 4: für alle assert-Methoden  
import org.junit.runner.\*; // JUNIT 4: für JUnitCore und Result  
import org.junit.runner.notification.\*; // JUNIT 4: für Failure  
//import org.junit.jupiter.api.\*; // JUNIT 5  
//import static org.junit.jupiter.api.Assertions.\*; // JUNIT 5

public class KundenTest {

 Kunde kunde1;

@Before  
 //@BeforeEach JUNIT 5  
 public void initialize() {  
 kunde1 = new Kunde();  
 }

@Test  
 //@DisplayName("Aaa") JUNIT 5  
 public void testAaa() {  
 assertTrue(true);  
 }

@Test  
 //@DisplayName("Bbb") JUNIT 5  
 public void testBbb() {  
 assertEquals(7,5);  
 }

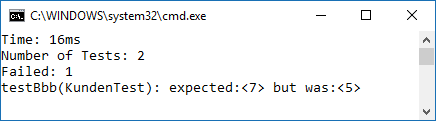
public static void main(String[] args) {  
 //junit.textui.TestRunner.run(KundenTest.class); // JUNIT 3: Eigene TUI  
 Result result = JUnitCore.runClasses(KundenTest.class);  
 printTUI(result);  
 }

public static void printTUI(Result result) {  
 System.out.println("Time: " + result.getRunTime() + "ms");  
 System.out.println("Number of Tests: " + result.getRunCount());  
 System.out.println("Failed: " + result.getFailureCount());  
 for (Failure f: result.getFailures()) {  
 System.out.println(f.toString());  
 }  
 }

}

---------------------------------------------------------------

**Ausgabe der Testklasse:**



**Aufgabe:**

Implementieren Sie die Klasse Kunde mit Hilfe von Test Driven Development!

**Fragen:**

* Wie viele Tests haben Sie geschrieben?
* Wie lang war im Durchschnitt Ihre Zykluszeit (Test schreiben -> Test schlägt fehl -> Produktivcode schreiben -> Test gelingt)?  
  Vereinfacht kann diese ausgerechnet werden mit:   
  ZeitdauerBisDieKlasseKundeFertigWurde / AnzahlDerTests