1. Programmieren Sie eine Klasse EinfachesAuto.java mit folgenden Eigenschaften und Methoden.

Teil 1:

```
001 /** EinfachesAuto.java
002 *   Klasse zur Beschreibung eines Autos
003 *   @ author Ihr Name, e-Mail-Adresse
004 *   Datum
005 **/
006 public class EinfachesAuto{
007 // Attribute
008   public String besitzer;
009   public String autotyp;
010   public String farbe;
011   public int erstzulassung;
012   public int leistung;
013   public int kmStand;
014
015 // Konstruktor
016   public EinfachesAuto(String besitzer, String autotyp, String farbe,
017                        int erstzulassung, int leistung, int kmStand){
018
019     this.besitzer = besitzer;
020     this.autotyp = autotyp;
021     this.farbe = farbe;
022     this.erstzulassung = erstzulassung;
023     this.leistung = leistung;
024     this.kmStand = kmStand;
025   }
026
027 // Methoden
028 /** Berechnung des Alters des Autos
029 *  @param ohne
030 *  @return int alter
031 */
032 public int alter(){
033     return 2014 - erstzulassung;
034 }
035
036 /** Einfache Ausgabe auf den Bildschirm
037 *  @param ohne
038 *  @return void
039 */
040 public void meldung(){
041   System.out.print ("Hier gruesst das "+ farbe );
042   System.out.print ("Auto von" + besitzer);
043 }
044 }
```

Schreiben Sie eine **Testklasse AutoTest.java**, mit der Sie **EinfachesAuto.class** testen, in dem Sie drei Objekte erzeugen und deren Methoden aktivieren.

Teil 2:
Erweitern Sie den Quellcode Ihrer Klasse **EinfachesAuto.java** um folgende Attribute:

String standort;
long fahrgestellnummer;

und um folgende Methoden:

- weiteren **Konstruktor**, der die neuen Attribute ebenfalls initialisiert.
- **void alleDatenAusgeben()**
  Schreibt die Werte der Attribute in übersichtlicher Form auf den Bildschirm
- **int faehrtNach(String Ziel, int Entfernung)**
  Methode schreibt eine Meldung auf den Bildschirm:
  Auto fährt von Berlin nach Hamburg 250 km
  Speichert den neuen Standort und den neuen Kilometerstand
  Rückgabewert: kmStand

Speichern Sie die Datei unter **ErweitertesAuto.java** ab. Lösen Sie die Aufgabe über die Vererbung, indem Sie die Klasse **ErweitertesAuto** von der Klasse **EinfachesAuto** ableiten. Beachten Sie, dass in der abgeleiteten Klasse zwei Konstruktoren zur Verfügung gestellt werden müssen. Einer muss die gleichen Parameter enthalten wie der der Klasse **EinfachesAuto.**

2. Gegeben ist die Klasse **Punkt.java**:

```
public class Punkt {
007   protected double x,y;
008
009   public Punkt(double x, double y){
010     this.x = x;
011     this.y = y;
012   }
013
014 /**
015   * @return x,y Koordinaten eines Punktes werden zurueckgegeben.
016   */
017    public String text(){
018      return new String("("+x+", "+y+")");
019    }
020
021 /** Gibt die Koordinaten des Punktes (x,y) aus.
022   * Ruft dazu die Methode text().
023   * @return Formatierte Ausgabe der Punktkoordinaten.
024   * @see Punkt#text()
025   */
026   public String toString(){
027     return new String("Punkt: "+text());
028   }
029
030 }
```

Teil 1:
Programmieren Sie 3 weitere Klassen Quadrat.java, Rechteck.java und Kreis.java, die von der Klasse Punkt direkt oder indirekt abgeleitet werden. Folgende Aufgaben sollen die Klassen erfüllen:

Quadrat.java:

- Subklasse von Punkt.
- verwendet double-Werte x und y aus der Klasse Punkt und speichert zusätzlich einen double-Wert deltaX, der die Hälfte einer Seitenlänge des Quadrates darstellt·
- Initialisierung erfolgt im Konstruktor der Klasse Quadrat und im Konstruktor der Superklasse·
- die Klasse enthält eine getUmfang-Methode, die den Umfang des Quadrates bestimmen soll

Rechteck.java:

- Subklasse von Quadrat.·
- verwendet double-Werte x und y aus der Klasse Punkt, deltaX aus der Klasse Quadrat und speichert zusätzlich einen double-Wert deltaY, der die Hälfte der anderen Seitenlänge darstellt.
- Initialisierung erfolgt im Konstruktor der Klasse Rechteck und im Konstruktor der Superklassen
- die Klasse enthält eine getFlaeche-Methode, welche die Fläche des Rechteckes berechnen soll

Kreis.java:

- Unterklasse von Punkt
- verwendet double-Werte x und y aus der Klasse Punkt und speichert zusätzlich einen double-Wert radius

  radius - Radius des Kreises
- Initialisierung erfolgt im Konstruktor der Klasse Kreis und im Konstruktor der Superklasse
- die Klasse enthält außerdem noch die beiden Methoden: getFlaeche und getUmfang, die Fläche und Umfang des Kreises berechnen sollen

Teil 2
Programmieren Sie eine Testklasse, in der jeweils ein Objekt der oben beschriebenen Klassen Punkt, Rechteck, Kreis und Quadrat erzeugt wird.

In den einzelnen Klassen werden folgende Attribute mit folgenden Werten initialisiert:

Punkt.java	x=20	y=20		
Rechteck.java	x=20	y=20	deltaX=30	deltaY=5
Quadrat.java	x=20	y=20	deltaX=20	
Kreis.java	x=20	y=20	Radius =12;	

Der Umfang des Quadrates, die Fläche des Rechtecks sowie Umfang und Fläche des Kreises sollen berechnet und in der Konsole ausgegeben werden.