

NACHNAME:	SEMESTER: <input type="checkbox"/> M5 <input type="checkbox"/> M6 <input type="checkbox"/> M3 <input type="checkbox"/> M4 <input type="checkbox"/> M7
VORNAME:	VERTIEFUNG: <input type="checkbox"/> FV <input type="checkbox"/> IM

**VERWENDETE
KLASSEN:**

- Als **Anlage** erhalten Sie den Ausdruck des vorab bekannt gemachten Quelltextes von vier Versionen einer Klasse `Polygon`. Ein `Polygon` soll (in allen vier Versionen) ein **geschlossenes**, ebenes Polygon aus zwei oder mehr Punkten beschreiben.
 - Darüber hinaus wird die ebenfalls vorab bekannte Klasse `Punkt` der letzten Prüfung (Sommer-Semester 2012) verwendet. Die Klasse wird (wie damals) als bekannt vorausgesetzt.
-

**UNBEDINGT
BEACHTEN:**

- **Bevor** Sie mit der Bearbeitung beginnen, müssen die Angaben zur Person auf dieser Seite **vollständig** ausgefüllt sein.
- Es sind **keinerlei Hilfsmittel** zugelassen.

**GENERELLE
VORGABEN:**

- Es sind keinerlei Kommentare verlangt, weder `javadoc`-Kommentare noch andere.
 - Programmier-Richtlinien (insbesondere Checkstyle) sind zu beachten. Bei Testklassen dürfen aber *magic numbers* verwendet werden.
-

Alle Aufgaben beziehen sich auf die Konstanten des folgenden Interfaces:

```
package w12pruefung;
import w12pruefung.v2.Polygon;
public interface TestKonstanten {
    Punkt A = new Punkt(4, 3);
    Punkt B = new Punkt(-4, 3);
    Punkt C = new Punkt(-4, -3);
    Polygon P = new Polygon(A, B, C);
    Object Q = new Polygon(A, B, C, A);
    Polygon R = new Polygon(A, C, B);
    Polygon S = new Polygon(A, C, A, B, A);
    Polygon T = new Polygon(B, C, A);
    Object U = new Polygon(C, A, B);
}
```

Aufgabe 1: (30 Punkte)

a) Geben Sie an, welchen Wert dann die in der linken Spalte angegebenen Ausdrücke jeweils haben. Abkürzungen (etwa T und F) sind nicht erlaubt. (Der benötigte statische Import soll als vorhanden vorausgesetzt werden.)

	v1	v2	v3	v4
P.equals(Q)				
P.hashCode() == Q.hashCode()				
R.equals(S)				
P.equals(T)				
T.equals(R)				
U.equals(T)				

b) In einer Methode der Klasse wird ein Container für Polygone angelegt:

```
Set<Polygon> p = new HashSet<Polygon>();
```

Dann werden **nacheinander** einige Polygone in diesen Container aufgenommen. Geben Sie nach jeder Operation die Größe p.size() des Containers bei Verwendung der angegebenen Version von Polygon an.

	v1	v2	v3	v4
p.add(P);				
p.add((Polygon) Q);				
p.add(R);				
p.add((Polygon) S);				
p.add(T);				
p.add((Polygon) U);				

