

NACHNAME:	SEMESTER: <input type="checkbox"/> M6 <input type="checkbox"/> M7 <input type="checkbox"/> M4 <input type="checkbox"/> M5 <input type="checkbox"/> M8
VORNAME:	VERTIEFUNG: <input type="checkbox"/> FV <input type="checkbox"/> IM

**Aufgabe 1: (11 Punkte)**

Die Klasse `GeometrischeReihe` beschreibt unendliche (konvergente oder divergente) geometrische Zahlen-Reihen:

$$1 + q + q^2 + q^3 + \dots = \sum_{n=0}^{\infty} q^n \quad (q \in \mathbb{R})$$

Die noch ganz rudimentäre Implementierung

```
public final class GeometrischeReihe {  
    double q;  
}
```

soll schrittweise erweitert werden. `javadoc`-Kommentare sind dabei nicht verlangt.

**a)** Schreiben Sie den naheliegendsten und wichtigsten Konstruktor:

 <hr/> <hr/> <hr/> <hr/> <hr/>
---

**b)** Implementieren Sie (im Sinne der Abstraktion) die Methode `istKonvergent`:

 <hr/> <hr/> <hr/> <hr/> <hr/>
---

**c)** Schreiben Sie eine Methode `summe` entsprechend der angegebenen Dokumentation:

```
/**
 * liefert die Summe dieser unendlichen Reihe.
 * Bei einer divergenten Reihe wird eine
 * UnsupportedOperationException geworfen.
 */
```

**Aufgabe 2: (8 Punkte)**

Die Klasse `Reihen` stehe im gleichen package wie die in **Aufgabe 1** entstandene Klasse und enthalte folgende Methode:

```
public static void tuWasMit(GeometrischeReihe g) {
    if (!g.istKonvergent()) g.q = 1. / g.q;
    if (!g.istKonvergent()) g.q = 0.;
}
```

**a)** Aufgrund welches schwerwiegenden Mangels in dem bei **Aufgabe 1** vorgegebenen Code ist diese Implementierung überhaupt erst möglich:

**b)** Wie ändert sich die übergebene Reihe `g` beim Aufruf mit folgenden Werten:

abstrakter Wert vor dem Aufruf	abstrakter Wert nach dem Aufruf
1 + 0 + 0 + 0 + ...	
1 + 3 + 9 + 27 + ...	
1 + 1/2 + 1/4 + 1/8 + ...	
1 - 1 + 1 - 1 +- ...	

**Aufgabe 3: (11 Punkte)**

Die in **Aufgabe 1** entstandene Klasse werde durch folgende Methode erweitert:

```
public Intervall tuWas() {
    if (!istKonvergent()) return new Intervall(1/2, q);
    if (q < 0.) return new Intervall(1 + q, 1);
    return new Intervall(1, summe());
}
```

Dabei ist `Intervall` die aus den Übungen bekannte Klasse.

**a)** Welche Ergebnisse liefert diese Methode bei den nachstehenden Aufrufen? Falls beim Aufruf eine Exception geworfen wird, geben Sie in der Ergebnis-Spalte den Namen der Exception an.

diese GeometrischeReihe	Ergebnis-Intervall
$1 + 1 + 1 + 1 + \dots$	
$1 + 1/2 + 1/4 + 1/8 + \dots$	
$1 - 2 + 4 - 8 + \dots$	
$1 - 1/2 + 1/4 - 1/8 + \dots$	

**b)** Welche drei Methoden von `Object` sollten überschrieben werden, wenn `GeometrischeReihe` letztlich eine gute Klasse werden soll. Geben Sie die vollständige Signatur (inklusive Rückgabebetyp und Modifizierer) dieser Methoden an:
