

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

Aufgabe 1: (15 Punkte)

Im gleichen package wie die aus den Übungen bekannte Klasse `Intervall` stehe die folgende, noch ganz rudimentäre Klasse `IntervallBisUnendlich`. Sie soll abgeschlossene, unendliche Intervalle der Form $[a, \infty)$ beschreiben:

```
public final class IntervallBisUnendlich {  
    private double mVon;  
}
```

Die Klasse soll nun schrittweise erweitert werden. javadoc-Kommentare sind dabei nicht verlangt.

a) Schreiben Sie den naheliegendsten und wichtigsten Konstruktor:

--

b) Schreiben Sie eine Methode `enthaelt` entsprechend der angegebenen Dokumentation:

```
/**  
 * prueft, ob das angegebene (endliche) Intervall ganz  
 * in diesem IntervallBisUnendlich enthalten ist.  
 */
```

--

c) Durch den Aufruf

```
System.out.println(a);
```

soll der abstrakte Wert des Objekts `a` vom Typ `IntervallBisUnendlich` ausgegeben werden. Nehmen Sie eine nötige Erweiterung der Klasse vor:

d) Um dem allgemeinen Vertrag von `Object` zu genügen, müssen weitere Methoden überschrieben werden. Geben Sie die vollständige Signatur (inklusive Rückgabotyp und Modifizierer) dieser Methoden an:

Aufgabe 2: (7 Punkte)

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

```
public Intervall tuWasMit(Intervall i) {
    if (enthaelt(i)) return i;
    if (!i.enthaelt(mVon)) return null;
    return new Intervall(mVon, i.obereGrenze());
}
```

a) Welche Ergebnisse liefert diese Methode bei den nachstehenden Aufrufen:

dieses <code>IntervallBisUnendlich</code>	angegebenes <code>Intervall i</code>	Ergebnis
$[0, \infty)$	$[2, 5]$	
$[2, \infty)$	$[0, 3]$	
$[4, \infty)$	$[-2, 1]$	
$[6, \infty)$	$[-4, 6]$	

b) Um das Verhalten im Fehlerfalle zu verbessern, braucht nur eine (einzige) Zeile geändert zu werden. Geben Sie die neue Zeile an:

Aufgabe 3: (8 Punkte)

Die in **Aufgabe 2** entstandene Klasse werde durch folgende Methode erweitert:

```
public void tuWasMit(IntervallBisUnendlich i) {
    if (i.mVon > mVon) mVon = i.mVon;
    else i.mVon = mVon;
}
```

a) Geben Sie an, wie die Variablen **a** und **b** beim folgenden Aufruf verändert werden:

```
a.tuWasMit(b);
```

	a	b
vorher	$[3, \infty)$	$[1, \infty)$
nachher		
vorher	$[2, \infty)$	$[4, \infty)$
nachher		

b) Welche zuvor vorhandene gute Eigenschaft der Klasse würde durch diese Methode zerstört?

--

Durch welche Änderung in einer einzigen Zeile der Klasse könnte das versehentliche Zerstören dieser Eigenschaft verhindert werden? Geben Sie die neue Zeile an:

--