FACH:	**CAD-Projekt**	**NAME:**	
DATUM:	Donnerstag, 18.12.2008		
ZEIT:	15:45 – 17:15	**SEMESTER:**	
PRÜFER:	Prof. Dr. Wolfgang Erben		
		STUDIENGANG:	☐ **D**iplom Mathematik ☐ **B**achelor Mathematik

ANLAGEN: **HILFSMITTEL:**	• Quelltext einiger Klassen (**bereits vorab ausgegeben**). • Handschriftliche Ergänzungen auf der vorab ausgegebenen Anlage
UNBEDINGT BEACHTEN:	• Tragen Sie **jetzt gleich** auf dieser Seite Ihren **Name** und ihr **Semester** ein und kreuzen Sie Ihren **Studiengang** an. Bearbeitungen ohne diese vorherige Kennzeichnung sind ungültig. • Führen Sie die Bearbeitung direkt auf diesen Aufgabenblättern in den dafür vorgesehenen Rahmen aus. Der dort vorhandene Platz sollte ausreichen.
HINWEISE ZUM QUELLTEXT:	• Bei dem angegebenen Quellcode handelt es sich um Auszüge, welche sinnvoll ergänzt zu denken sind. Insbesondere trifft dies auf das Importieren benötigter Klassen in anderen Packages zu. • `Intervall` und `VereinigungVonIntervallen` sind zwei Klassen, welche den Anforderungen der (zweiten) Studienarbeit genügen. Insbesondere sind beide Klassen kanonisch. • An vielen Stellen ist der Quelltext absichtlich schlecht. Insbesondere werden häufig nichtssagende Namen verwendet.
EMPFEHLUNGEN ZUR BEARBEITUNG:	• Lesen Sie die Aufgaben und Aufgabenteile bitte in der vorgegebenen Reihenfolge durch. An manchen Stellen ist dies für das Verständnis der Aufgabenstellung wichtig. • Einzelne Bearbeitungen können aber zurückgestellt werden. Die vollständige Lösung einer Aufgabe ist für die folgenden Aufgaben nicht erforderlich.

Aufgabe 1: (12 Punkte)

In dieser Aufgabe geht es um allgemeine Zusammenhänge.

a) Welche der folgenden Aussagen sind richtig?

	ja	nein
Jede eigene Klasse besitzt eine Basisklasse.		
Im Konstruktor einer Klasse wird immer (explizit oder implizit) ein Konstruktor der Basisklasse gerufen.		
Jede Klasse besitzt eine öffentliche `equals` Methode.		
Konstruktoren können überladen werden.		
Konstruktoren können überschrieben werden.		

b) Welche der folgenden Aussagen sind richtig?

	ja	nein
Eine Klasse kann höchsten ein `interface` implementieren.		
Ein `interface` kann nicht als Rückgabetyp einer Methode verwendet werden.		
Ein `interface` muss mindestens eine Methode enthalten.		
Ein `interface` kann nur öffentliche Methoden haben.		
Die Methoden eines `interface` müssen explizit als `public` gekennzeichnet werden.		

c) Welche der folgenden Aussagen sind richtig?

	ja	nein
Eine abstrakte Klasse kann keinen Konstruktor besitzen.		
Eine abstrakte Klasse besitzt mindestens eine abstrakte Methode.		
Eine abstrakte Klasse darf nur abstrakte Methoden besitzen.		
Eine Klasse mit einer abstrakten Methode muss explizit als abstrakt deklariert werden.		
Von einer abstrakten Klasse kann nicht abgeleitet werden.		

d) Welche der folgenden Empfehlungen sollten möglichst befolgt werden?

	ja	nein
In einem Konstruktor sollten keine virtuellen Methoden gerufen werden.		
`hashCode` sollte für gleiche Objekte den gleichen Wert liefern.		
`hashCode` muss für verschiedene Objekte verschiedene Werte liefern.		
`equals` muss eine Äquivalenzrelation sein.		
Bei Klassen mit `double`-Werten sollte `equals` die Gleichheit mit Toleranz überprüfen.		

Aufgabe 2: (12 Punkte)

In dieser Aufgabe geht es um Design-Entscheidungen bei den Klassen `Intervall` und `VereinigungVonIntervallen` aus den Studienarbeiten.

a) In beiden Klassen wird die Methode `equals` überschrieben. Warum?

Mit `equals` wird auch die Methode _____ überschrieben.

b) Die Methode `clone` wird nur in `VereinigungVonIntervallen` überschrieben. Warum nicht in der Klasse `Intervall`?

c) Die beiden Klassen `Intervall` und `VereinigungVonIntervallen` könnten ein gemeinsames `interface`, etwa `Menge` besitzen. Was spricht dagegen?

Aufgabe 3: (27 Punkte)

In dieser Aufgabe geht es um die Parameter-Übergabe. Aus den Studienarbeiten wird die Klasse `Intervall` und aus dem **Anhang** die Klasse `Tool` verwendet.

```
public class Murks {
    private Intervall i = Tool.gibMir();
    public final Murks tuWasMit(Murks a) {
        if (i.istDisjunktZu(a.i)) a = this;
        i = i.untereHaelfte();
        a.i = a.i.obereHaelfte();
        if (i.equals(a.i)) return this;
        return new Murks();
    }
}
```

Der abstrakte Wert eines Objektes dieser Klasse ist der des enthaltenen Intervalls. Nach den drei Initialisierungen

```
Murks a = new Murks();
Murks b = new Murks();
Murks c = a;
```

haben die Objekte `a`, `b` und `c` die (abstrakten) Werte

a	[−1,15]	b	[14,30]	c	[−1,15]

a) Wie verändern sich diese Werte durch folgende Anweisung?

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

a		b		c	

b) Wie verändern sich diese neuen Werte durch folgende Anweisung?

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

a		b		c	

c) Damit erfolge nachstehender Aufruf. Welchen Wert haben `a`, `b` und `c` dann?

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

a		b		c	

Aufgabe 4: (21 Punkte)

In dieser Aufgabe geht es um das Kopieren von Objekten der in der **Anlage** definierten Typen. Eine `JUnit`-Testklasse enthält die Hilfs-Methode

```
private void kopiere(Pool p) {
        Pool q = p.clone();
        assertTrue(p.equals(q));  // A
        assertTrue(q.equals(p));  // B
        assertFalse(p == q);      // C
        q.gibMir();
        assertFalse(p.equals(q)); // D
        assertFalse(q.equals(p)); // E
        assertTrue(p == q);       // F
}
```

Verschiedene – in der ersten Spalte nachfolgender Tabelle angegebene – Testmethoden bestehen jeweils aus einem einzigen Aufruf dieser Hilfs-Methode. Keine dieser Testmethoden läuft fehlerfrei ab. Geben Sie die Stelle (A, B, …, F) an, bei welcher der Fehler auftritt:

`public void testeKopierenToolPool() { kopiere(new ToolPool()); }`	
`public void testeKopierenCoolPool() { kopiere(new CoolPool()); }`	
`public void testeKopierenBoolPool() { kopiere(new BoolPool()); }`	
`public void testeKopierenFoolPool() { kopiere(new FoolPool()); }`	
`public void testeKopierenToolPoolPool() { kopiere(new PoolPool(new ToolPool())); }`	
`public void testeKopierenFoolPoolPool() { kopiere(new PoolPool(new FoolPool())); }`	
`public void testeKopierenFoolPoolPoolPool() { kopiere(new PoolPool( new PoolPool(new FoolPool()))); }`	

Aufgabe 5: (18 Punkte)

In dieser Aufgabe geht es um die Verwendung der Klassen aus der **Anlage**.

a) Kreuzen Sie jeweils an, ob die angegebene Verwendung korrekt ist oder aber zu einem Compile-Fehler, einer Warnung (inkl. Checkstyle) oder zu einer Exception führt:

Verwendung	C.-F.	War.	Exc.	OK
`Pool p = new ToolPool();` `Intervall s = (Intervall) p.gibMir();`				
`Pool p = new FoolPool();` `Intervall s = (Intervall) p.gibMir();`				
`FoolPool p = new FoolPool();` `Intervall s = (Intervall) p.gibMir();`				
`Pool p = new CoolPool();` `Intervall s = p.gibMir();`				
`CoolPool p = new BoolPool();` `Intervall s = p.gibMir();`				
`Tool p = new Tool();` `Intervall s = p.gibMir();`				

b) Die Anweisungen in der linken Tabellenspalte werden hintereinander ausgeführt. Geben Sie in der rechten Spalte den Hash-Code von `p` nach der Ausführung des jeweiligen Statements an:

	`p.hashCode()`
`Pool p = new BoolPool();`	
`p.gibMir();`	
`p.nimmDir(2);`	