



Babeș-Bolyai Universität  
Cluj Napoca  
Fakultät für Mathematik und Informatik  
*Dr. Cătălin Rusu*

## Klausur *Fortgeschrittene Programmierungsmethoden*

20 Februar 2016

**Kandidat/in:**

Name: \_\_\_\_\_

Email: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_

### Allgemeine Hinweise:

- Neben Papier und Schreibutensilien sind keine weiteren Hilfsmittel erlaubt
- **Verwenden Sie keine roten Stifte und keine Bleistifte.**
- Sie dürfen alle Aufgaben in beliebiger Reihenfolge lösen. Konzentrieren Sie sich jeweils auf eine Aufgabe, aber teilen Sie sich Ihre Zeit ein.
- Alle Mobiltelefone müssen vollständig ausgeschaltet sein.
- **Vergessen Sie nicht, Ihren Namen und die Matrikelnummer auf jedes Blatt zu schreiben.**
- Blätter ohne diese Angaben werden nicht gewertet.
- Bitte schreiben Sie in Ihrem eigenen Interesse deutlich. Unverständliche oder nicht begründete Antworten werden nicht bewertet.
- In den letzten 20 Minuten der Prüfung kann der Raum nicht mehr verlassen werden.
- **Im Fall von Täuschungsversuchen wird die Klausur sofort mit 0 Punkten bewertet. Eine Vorwarnung erfolgt nicht.**

Aufgabe	max. Punkte	erreicht
1	7.5	
2	7.5	
3	25	
4	15	
5	25	
6	10	
Summe	90	

**GL&HF!**

## Aufgabe 1

Der folgende Java-Quellcode enthält verschiedene Fehler, die entweder von einem Java-Compiler erkannt werden oder zur Laufzeit in Form einer Exception auftreten. Tragen Sie in der unten stehenden Tabelle die fehlerhaften Zeilen ein (5) und geben Sie dazu jeweils eine kurze Begründung an. (7.5 Punkte)

### Code-Fragment

```
1. public abstract class C {
2.     int i;
3.         private static int j;
4.
5.         public static void main(String[] args) {
6.             C c = new C();
7.             C c = new C();
8.             c = d;
9.             i = 3;
10.            j = 3;
11.        }
12.    abstract void m();
13.}
14.
15. class D extends C {
16.     int k = j;
17.}
18.
```

Zeile	Begründung

## Aufgabe 2

Gegeben sei der folgende Java-Quellcode. Geben Sie zu den in der unten stehenden Tabelle angegebenen Zeilen jeweils an, ob und wenn welche Bildschirmausgabe erfolgt. (7.5 Punkte)

Zeile	Bildschirmausgabe
41	
42	
44	
45	
46	
47	

---

## Code-Fragment

---

```
1. interface A {
2.     void v();
3. }
4. interface B extends A {
5.     void x(B b);
6. }
7. class E implements A {
8.     public void v() {
9.         System.out.println("v in E");
10.    }
11.    public void x(G g) {
12.        System.out.println("x in E");
13.    }
14. }
15. class F extends E implements B {
16.    public void x(A a) {
17.        System.out.println("xA in F");
18.    }
19.    public void x(B b) {
20.        System.out.println("xB in F");
21.    }
22. }
23. class G extends F {
24.    public void v() {
25.        System.out.println("v in G");
26.    }
27.    public void x(B b) {
28.        System.out.println("x in G");
29.    }
30. }
31. public class Test {
32.     public static void main (String args[]) {
33.         A a = new F();
34.         B b = new G();
35.
36.         E e = new E();
37.         F f = new F();
38.         G g = new G();
39.
40.         a.v();
41.         b.v();
42.
43.         g.x(e);
44.         g.x(f);
45.         g.x(g);
46.         g.x(b);
47.     }
48. }
49. }
50. }
```

## Aufgabe 3

(a) Aus welchen Elementen kann eine Interface-Deklaration bestehen? (3 Punkte)

(b) Einer Ihrer Kollegen sagt Ihnen: „Interfaces sind überflüssig. Statt ihnen könnte man genauso abstrakte Klassen verwenden.“ Was sagen Sie dazu? Begründen Sie! (7 Punkte)

(c) Gegeben sind eine Klasse Super und ihre Subklasse Sub (7.5 Punkte)

---

## Code-Fragment

---

```
1. class Super {}
2. class Sub extends Super {}
```

Geben Sie für folgende Zuweisungen an, ob diese zu einem Compilierfehler, Laufzeitfehler (in Form einer Exception) oder zu keinem Fehler führen.

- Super test1 = (Super) new Sub(); \_\_\_\_\_
- Super test2 = (Sub) new Super(); \_\_\_\_\_
- Super test3 = (Sub) new Sub(); \_\_\_\_\_
- Sub test4 = (Sub) new Super(); \_\_\_\_\_
- Sub test5 = (Super) new Sub(); \_\_\_\_\_
- Sub test6 = (Super) new Super(); \_\_\_\_\_

Gegeben sei folgendes Programmstück. Begründen Sie, warum der Compiler dieses zu Recht wegen fehlender Typsicherheit mit einem Compilerfehler in der zweiten Zeile versieht.

---

### Code-Fragment

---

```
1. java.util.Stack
2. subStack = new java.util.Stack();
3. java.util.Stack superStack = subStack;
4. superStack.push(new Super());
5. Sub sub = subStack.pop();
```

(d) Für das Schlüsselwort **this** gibt es in **Java/C#** mehrere Verwendungen. **(7.5 Punkte)**

Was bezeichnet this im Rumpf einer Methode?

---

---

---

An welchen Stellen im Programm ist ein Aufruf this() möglich und was bedeutet er?

---

---

---

Was bezeichnet this innerhalb eines Konstruktors?

---

---

---

Warum führt die Verwendung von this innerhalb des Rumpfs einer statischen Methode (mit Ausnahme von Konstruktoren) zu einem Compilerfehler?

---

---

---

### Aufgabe 4

GRASP (Beschreibung, anwendung, beispiele). **(15 Punkte)**

### Aufgabe 5

(a) **(15 Punkte)**

1. Schreibe drei Klassen zu folgendem Problem: Es gibt Bücher, zu denen der Autorenname, der Titel und die Sprache festgehalten werden soll. Gedruckte Bücher und Elektronische Bücher sind Spezialisierungen von Büchern. Gedruckte Bücher haben zusätzlich eine Seitenzahl und sind entweder ein Taschenbuch oder nicht. Elektronische Bücher haben eine Dateigröße und ein Format.

Die Klassen sollen folgende abstrakte Methoden haben:

- Buch: drucken()
- ElektronischesBuch: speichern()
- GedrucktesBuch: kopieren()

2. Es sei eine Klasse nicht-abstrakte WebArtikel als Erweiterung von ElektronischesBuch gegeben, sowie eine nicht-abstrakte Klasse Lehrbuch, die GedrucktesBuch erweitert.

Erstelle nun ein Buch[]-Array und fülle es mit vier Instanzen von Klassen, die Du in diesem Array speichern kannst.

3. Welche Methode besitzen alle in (2) erzeugten Instanzen? Schreibe eine Schleife, die für alle Elemente des Arrays diese Methode ausführt.
4. Wann wird festgestellt, aus welcher Klassendefinition die auszuführende Methode herangezogen wird? Wie nennt man das dahinterstehende Konzept?

(b) Implementieren Sie fünf Typen Spinne, Fisch, Tisch, MitBeinen und Tier, so dass die folgenden Bedingungen erfüllt sind:

- Fisch ist Subtyp von Tier
- Spinne ist Subtyp von Tier
- Spinne ist Subtyp von MitBeinen
- Tisch ist Subtyp von MitBeinen
- Fisch ist kein Subtyp von MitBeinen
- Tisch ist kein Subtyp von Tier
- Es kann keine Instanzen von Tier und von MitBeinen geben

Weiterhin soll der Typ MitBeinen eine Methode public int anzahlBeine() und der Typ Tier eine Methode public String nameDerTierart() bekommen, die geeignet zu implementieren sind. **(10 Punkte)**

## Aufgabe 6

Welche der folgenden Aussagen trifft zu? Begründen Sie Ihre Antwort. **(10 Punkte)**

(a) Abstrakte Klassen müssen mindestens eine abstrakte Methode besitzen.

☐

---

---

---

---

---

(b) Eine Klasse kann von beliebig vielen Klassen erben.

☐

(c) Ein Konstruktor kann alle nichtprivaten Methoden der aktuellen Klasse und ihrer Oberklasse aufrufen.

☐

---

---

---

(d) Unter einer „überladenen“ Methode versteht man zwei Methoden gleichen Namens in derselben Klasse, mit gleichen Rückgabetypen und unterschiedlichen Parametern

☐

---

---

---

---

---

---