



# Bachelorprüfung: OOSE (WI)

HSRM, Wirtschaftsinformatik, Wintersemester 2015/2016

**Erlaubte Hilfsmittel:** Ein Buch sowie ein gedrucktes, fest gebundenes (z.B. Spiralbindung, keine Leitzordner) selbst zusammengestelltes Skript

Die Lösung ist auf den Klausurbögen anzufertigen (eventuell Rückseiten nehmen).

**Bearbeitungszeit:** 90 Minuten

---

Name

---

Matrikelnr.

---

Unterschrift

Benotung

Aufgabe:	1	2	3	4	5		Gesamt	Note
Punkte:	20	25	25	15	15		100	
erreicht:								



---

## Abkürzungsregeln

Um Ihnen unnötige Schreibarbeit zu ersparen, können Sie in der Klausur folgende Abkürzungsregeln für Typen und andere Ausdrücke verwenden.

Folgende Typnamen und Ausdrücke können Sie folgendermaßen abkürzen:

- `System.out.println` → `Sys.out.pln`
- `Sytem.out.printf` → `Sys.out.pf`

Zu den verwendeten Datentypen müssen Sie keine Import-Statements bereitstellen außer es wird in der Aufgabe explizit gefordert.

## Lesehinweise

Damit Sie die die Zusammenhänge in einer Aufgabe besser verstehen können, wurde erklärender Text hinzu-gefügt.

*Diese erklärenden Texte, die nicht explizit etwas fordern, das in der Aufgabe getan werden muss, werden in kursiver Schreibweise dargestellt.*

## Aufgabe 1 (20 Punkte)

### (Programmfluss analysieren und verstehen)

Führen Sie die folgenden Programme von Hand aus.

- a) Was passiert bei Ausführung der folgenden Klasse? Schreiben Sie rechts vom Code die Ausgabe auf der Konsole auf (8 Punkte).

*Hinweis: Den ?:-Operator finden Sie in [Jobst: S. 72] oder auf Folie 7 in 09\_Ergänzungen.*

```
public class Aufg1a{

    public static int calc(int i){
        try{
            return ((i % 2) == 0) ? 0 : i ;
        }
        catch(ArithmeticException ae){
            System.out.println("AEx"); return -1;
        }
        finally{
            System.out.println("F");
        }
    }

    public static void fct(int j){
        if (j >= 6){ return; }

        int z= calc(j);
        if (z == 0){ ++z; }
        else { System.out.println(z);}
        fct(++j);
    }

    public static void main(String [] args){
        fct(1);
    }
}
```

- b) Gib für jede der Programmzeilen in main an (12 Punkte):

- Führt die Programmzeile beim Kompilieren zu einem Fehler (*einfach mit F kennzeichnen*)? *Folgefehler aus einem Fehler können Sie mit FF kennzeichnen.*
- Falls nein: Zu welcher Ausgabe führt die Programmzeile?

```
public interface Interf {
    void falle(); void gehe();
}

public class Ant implements Interf {
    public void falle() { System.out.println("A.f"); }
    public void gehe() { System.out.println("A.g"); }
    public void halte() { System.out.println("A.h"); }
}

public abstract class Xxx extends Ant {
    public void falle() { super.gehe();
                        System.out.println("X.f"); }
    public void halte() { System.out.println("X.h");
                        super.halte(); }
}

public final class Yps extends Ant {
    public void falle() { this.gehe();
                        System.out.println("Y.f"); }
    public void gehe() { super.halte();
                        System.out.println("Y.g"); }
}

public class Main {
    public static void main(String [] args) {
01      Xxx xx    =new Xxx();
02      Yps yps   =new Yps();
03
04      Ant a1 =new Ant();
05      Ant a2 =xx;
06      Interf i1 =new Interf();
07      Interf i2 =yps;
08
09      a1.falle();
10      a1.halte();
11      a2.falle();
12      a2.gehe();
13      a2.halte();
14
15      i1.halte();
16      i1.falle();
17      i2.falle();
18      i2.halte();
    }
}
```

---

## Aufgabe 2 (25 Punkte)

### (Klassen entwerfen)

Sie entwickeln die Basisklassen einer SW zur Produktdatenverwaltung im Onlinehandel.

a) Entwickeln Sie eine Klasse *Prod*. *Ein Objekt dieser Klasse repräsentiert ein Produkt im Sortiment eines Onlineshops.*

Es soll zwei Eigenschaften haben: Einen Namen, der durch eine Zeichenkette dargestellt wird und einen Preis (Gleitkommazahl).

Schreiben Sie einen Konstruktor und überschreiben Sie die Methode `toString` auf sinnvolle Weise (6 Punkte).

---

b) Schreiben Sie eine Unterklasse EProd, die von Prod erbt. *Objekte der Klasse repräsentieren elektrische Produkte.*

EProd soll als zusätzliche Eigenschaft einen boolean haben, der angibt, ob das Produkt über Batterie oder Kabel betrieben wird. Schreiben Sie eine Getter-Methode für diese zusätzliche Eigenschaft.

Des Weiteren überschreiben Sie die Methode toString und den Konstruktor, um die zusätzliche Information zu berücksichtigen (6 Punkte).



---

c) Schreiben Sie eine Klasse Bestand. Diese Klasse soll als Eigenschaft ein Array von 40 Produkten enthalten (*Nur das Array anlegen; Das Hinzufügen müssen Sie nicht machen*). Schreiben Sie innerhalb der Klasse eine Public-Methode void stats(), die das Array analysiert und auf der Konsole eine Produktestatistik in folgender Form ausgibt (13 Punkte):

Gesamt: <Anzahl aller Produkte>

Elek: <Anzahl aller Elektroprodukte>

Batt: <Anzahl der batteriebetriebenen Produkte>



---

### Aufgabe 3 (25 Punkte)

(GUI)

- a) Schreiben Sie ein Panel `HWPan`, das auf Maus-Klick-Ereignisse reagiert und bei einem neuen Klickereignis auf dem Panel den String "Hello World" (*Sie können das mit "H W" abkürzen*) genau an den Koordinaten schreibt, wo vorher geklickt wurde (10 Punkte).  
*Hinweis: Wenn Sie mit `MouseAdapter` arbeiten, sparen Sie sich viel Schreibarbeit.*



- b) Schreiben Sie eine Klasse für ein normales Anwendungsfenster bei dem im Konstruktor dessen Inhalt und Funktionalität erzeugt wird und das in der Größe 400 x 400 Pixel angezeigt wird.

Das Fenster soll folgenden Inhalt und Funktionalität haben:

1. Als Titel soll „H W A“ (Abk. für Hello World App) angezeigt werden. Ein Klick auf den Schließen-Knopf (X-Knopf rechts oben) soll die Applikation beenden.
2. Im oberen Bereich soll ein Label angezeigt werden, das zunächst den Text „Click Button!“ enthalten soll.
3. Im mittleren Bereich soll das Panel aus Aufgabe a eingebunden werden.
4. Im unteren Bereich soll ein Button mit Aufschrift „Btn“ eingefügt werden, der folgendes macht:
  - a. Wenn das Label "Click Button!" anzeigt, soll das Label "Click Panel!" anzeigen und das Panel soll auf Mausklickereignisse reagieren (z.B. indem es auf Mausereignisse registriert wird).
  - b. Wenn das Label "Click Panel!" anzeigt, soll das Label wieder "Click Button" anzeigen und das Panel nicht mehr auf Mausklickereignisse reagieren (z.B. indem es nicht mehr für Mausereignisse registriert wird).

*Hinweis1: Sie können "Click Button! " mit "C B" und "Click Panel! " mit "C P" abkürzen.*

*Hinweis2: Sie können die main-Methode weglassen. Dafür schreiben Sie aber im Konstruktor alles was für die Erzeugung und Anzeige des Fensters notwendig ist, so dass man in der Main-Methode nur den Konstruktor aufrufen müsste, um die Anwendung laufen zu lassen.*

*Hinweis3: Die in 4 gemachte Beschreibung der Funktionalität für den Button soll irgendwie umgesetzt werden. D.h. die dort gemachten Beschreibungen zur Umsetzung sind nur eine Möglichkeit. Sie können es ggf. auch anders umsetzen. Falls es Ihnen irgendwie hilft, können Sie natürlich auch noch in die Panelklasse aus Aufgabe 3a Veränderungen einfügen (15 Punkte).*



## Aufgabe 4 (15 Punkte)

### (Fehlerbehandlung & Rekursion)

a. Gegeben ist folgende Funktion:

```
public static int calc(int i) throws MyExcept1, MyExcept2 {  
    ...  
}
```

Vervollständigen Sie die unten angegebene main-Methode so, dass folgendes auf der Konsole ausgegeben wird (12 Punkte):

- Wenn der Benutzer keine oder mehr als einen Kommandozeilenparameter angegeben hat:  
    "Gz mg" (Abk. für "eine Ganzzahl mitgeben") und es wird das Programm beendet.
- Wenn der Kommandozeilenparameter nicht in einen int konvertiert werden kann:  
    "Gz mg" (Abk. für "eine Ganzzahl mitgeben") und es wird das Programm beendet.
- Ansonsten soll calc mit der aus dem Kommandozeilenparameter gewonnenen Ganzzahl aufgerufen werden und sich daraus folgenden Ausgaben sich ergeben:
  - Wenn der Aufruf von calc keine Exception zur Folge hat: "Erfolg"
  - Wenn der Aufruf von calc eine MyExcept1 zur Folge hat: "MyEx1"
  - Bei allen anderen Ausnahmen: Nachricht der Exception-Klasse + Stacktrace

```
public static void main (String [] args) {  
    ...  
}
```

b. Gegeben ist folgende rekursive Funktion:

```
static int fct(int x){  
    if(x<=0){  
        return 1;  
    }  
  
    return 3*fct(x-1);  
}
```

Schreiben Sie eine neue Funktion fct1, die das Problem iterativ (*mittels Schleife*) löst (3 Pkte).

## Aufgabe 5 (15 Punkte)

### (Allgemeine Fragen)

- a) Erklären Sie was es bedeutet, wenn eine Variable als static final deklariert wurde und ihr ein Wert zugewiesen wurde (1 Punkt).

- b) Analysieren Sie folgende Codezeilen, ob sie kompilieren. Falls nein, geben Sie an was man machen muss, und warum (3 Punkte).

```
01 long lng1= 300;  
02 int  int1= 300;  
03 long lng2= int1;  
04 int  int2= lng1;
```

- c) Angenommen Sie möchten eine Klasse MyClass in einem Unterpaket Utils des Pakets MyFirma definieren. Schreiben Sie was Sie bei der Definition der Klasse MyClass tun müssen und was Sie in anderen Klassen tun müssen, um die Klasse MyClass benutzen zu können (3 Punkte).



- 
- d) Erklären Sie was man unter Garbage Collection versteht, wann sie zum Einsatz kommt und wofür sie wichtig ist (4 Punkte)?
- e) Erklären Sie was im System passiert, wenn wir einen ==-Vergleich bei einem primitiven Datentyp und einem Referenztypen (z.B. Klasse oder Array) anwenden. Was sollte man statt == bei Referenztypen machen und warum (4 Punkte)?

**Bonusaufgabe<sup>1</sup>:**

Was versteht man unter Generischen Datentypen (Generics) und wofür sind diese gut (3 Bonuspunkte)?

---

<sup>1</sup> Nichtbeantwortung geht nicht in die Bewertung ein, jedoch können Sie bei Beantwortung sonst fehlende Punkte gut machen.