

Nachname	Vorname	Matrikelnummer	Studienkennzahl



## Schriftliche Prüfung



- LV-Titel: Grundzüge der Programmierung  
 LV-Nr.: 59  
 LV-Typ(en): LVP

- LV-Titel: Grundzüge der Programmierung für SBWL Informationswirtschaft  
 LV-Nr.: 1699  
 LV-Typ(en): PI AG PS

LV-Leiter: [Holmes, Ta'id](#)  
 Institut: Department für Informationsverarbeitung und Prozessmanagement,  
[Institut für Informationswirtschaft,](#)  
[Wirtschaftsuniversität Wien](#)  
 Augasse 2-6, A-1090 Wien

Datum: 24. November 2007  
 Zeit: 09:30  
 Dauer: 60 min

### Inhaltsverzeichnis

Software - Entwicklung (10).....	2
Programmiersprachen (8).....	2
Variablen und Datentypen (10).....	2
Operationen (10).....	3
Kontrollstrukturen (10).....	3
Methoden (12).....	4
Klassen (30).....	4
JavaDoc - Anhang.....	6

## **Software-Entwicklung**

---

**(10)**

- 1. Definieren Sie den Begriff Algorithmus .**

**×10**

---

## **Programmiersprachen**

---

**(8)**

- 2. Worauf bezieht sich eine Programmierkonvention und wofür dient eine solche?**

**×10**

---

## **Variablen und Datentypen**

---

**(10)**

- 3. Deklarieren Sie eine Variable mit primitivem Datentyp und weisen Sie ihr einen Wert zu. Weisen Sie anschließend diesen Wert unter Benutzung der Variablen einer zusätzlichen Variablen mit unterschiedlichem Datentyp zu. Worauf müssen Sie achten?**

**×10**

---

## Operationen

(10)

4. Prüfen Sie die Variablen **eee** und **bbb** auf Ungleichheit und speichern Sie das mit logischem ODER und dem Operand **ppp** verknüpfte Ergebnis in einer von Ihnen zu definierenden Variablen mit Bezeichner **fff** in Java.

×10

## Kontrollstrukturen

(10)

5. Zeichnen Sie ein Kontrollflussdiagramm zur graphischen Visualisierung von folgendem Algorithmus.

×10

```
if (this.artikel.contains(artikel)) {  
    if (wert >= artikel.wert) {  
        this.artikel.remove(artikel);  
        this.wert += wert;  
    } else {  
        System.out.println(„Zuwenig bezahlt!“);  
    }  
} else {  
    System.out.println(„Kein Artikel!“);  
}
```

## Methoden

(12)

6. Die Klasse `java.lang.Integer` (siehe JavaDoc im Anhang) hat eine parameterlose Methode `hashCode`. Schreiben Sie ein beliebig korrektes Programm, das diese Methode aufruft und den Rückgabewert ausgibt. ×10

## Klassen

(30)


7. Nennen und beschreiben Sie stichwortartig verschiedene Arten von Beziehungen zwischen Klassen. ×10

8. Zeichnen Sie ein UML Klassendiagramm für die Klasse **Panda**. Was wurde in der Implementierung wie realisiert? Beschreiben Sie die Klasse. ×15

```
public class Panda {  
    private Panda mutter, vater;  
    public void blinzeln() { System.out.println("*"); }  
    public void setMutter(Panda mutter) { this.mutter = mutter; }  
    public void setVater(Panda vater) { this.vater = vater; }  
}
```

9. Erzeugen Sie eine **Panda** Instanz mit Bezeichner **baby**. Setzen Sie für dieses Objekt die Eltern durch die bereits definierten und sichtbaren **Panda** Objekte **yangYang** (Mutter) und **longHui** (Vater). ×5

## java.lang Class Integer

Constructor Summary	
<a href="#">Integer</a> (int value)	Constructs a newly allocated Integer object that represents the specified int value.
<a href="#">Integer</a> (String s)	Constructs a newly allocated Integer object that represents the int value indicated by the String meter.
	
Method Summary	
static int	<a href="#">bitCount</a> (int i) Returns the number of one-bits in the two's complement binary representation of the specified int value.
byte	<a href="#">byteValue</a> () Returns the value of this Integer as a byte.
int	<a href="#">compareTo</a> ( <a href="#">Integer</a> anotherInteger) Compares two Integer objects numerically.
static <a href="#">Integer</a>	<a href="#">decode</a> (String nm) Decodes a String into an Integer.
double	<a href="#">doubleValue</a> () Returns the value of this Integer as a double.
boolean	<a href="#">equals</a> ( <a href="#">Object</a> obj) Compares this object to the specified object.
float	<a href="#">floatValue</a> () Returns the value of this Integer as a float.
static <a href="#">Integer</a>	<a href="#">getInteger</a> (String nm) Determines the integer value of the system property with the specified name.
static <a href="#">Integer</a>	<a href="#">getInteger</a> (String nm, int val) Determines the integer value of the system property with the specified name.
static <a href="#">Integer</a>	<a href="#">getInteger</a> (String nm, <a href="#">Integer</a> val) Returns the integer value of the system property with the specified name.
int	<a href="#">hashCode</a> () Returns a hash code for this Integer.
static int	<a href="#">highestOneBit</a> (int i) Returns an int value with at most a single one-bit, in the position of the highest-order ("leftmost") one-bit in the specified int value.
int	<a href="#">intValue</a> () Returns the value of this Integer as an int.
long	<a href="#">longValue</a> () Returns the value of this Integer as a long.
static int	<a href="#">lowestOneBit</a> (int i) Returns an int value with at most a single one-bit, in the position of the lowest-order ("rightmost") one-bit in the specified int value.
static int	<a href="#">numberOfLeadingZeros</a> (int i) Returns the number of zero bits preceding the highest-order ("leftmost") one-bit in the two's complement binary representation of the specified int value.
static int	<a href="#">numberOfTrailingZeros</a> (int i) Returns the number of zero bits following the lowest-order ("rightmost") one-bit in the two's complement binary representation of the specified int value.
static int	<a href="#">parseInt</a> (String s) Parses the string argument as a signed decimal integer.