

Nachname	Vorname	Matrikelnummer	Studienkennzahl



Schriftliche Prüfung



- LV-Titel: Grundzüge der Programmierung
- LV-Nr.: 76
- LV-Typ(en): LVP
- LV-Leiter: Gmainer, Johannes

- LV-Titel: Grundzüge der Programmierung für SBWL Informationswirtschaft
- LV-Nr.: 1180
- LV-Typ(en): LVP PI AG PS
- LV-Leiter: Holmes, Ta'id

Studienassistent: Loibl, Wilhelm
 Institut: Department für Informationsverarbeitung und Prozessmanagement,
Institut für Informationswirtschaft,
Wirtschaftsuniversität Wien
 Augasse 2-6, A-1090 Wien

Datum: 28. Juni 2008
 Zeit: 09:30
 Dauer: 60 min (3.5 ECTS) / 75 min (5.0 ECTS)

Inhaltsverzeichnis

Software-Entwicklung (10).....	2
Programmiersprachen (10).....	2
Variablen, Datentypen und Operationen (10).....	2
Kontrollstrukturen (10).....	3
Methoden (10).....	4
Klassen (45).....	4
Pakete (15).....	7
Zugriffsbeschränkungen (8).....	7

Software-Entwicklung

(10)

1. Beschreiben Sie die Phase der Implementierung eines Softwareprojekts. Welche Phase geht ihr voraus? ×10

Programmiersprachen

(10)

2. Java ist aus der Programmiersprache C/C++ entstanden. Welche Innovationen hat Java mit sich gebracht? Erläutern Sie z.B. den Slogan „Write once, run everywhere!“ ×10

Variablen, Datentypen und Operationen

(10)

3. Korrigieren Sie den Syntax des folgenden Programmfragments, ohne Zeichen zu entfernen. Schreiben Sie korrigierte Anweisungen neben die ursprünglichen. ×10

4. Beschreiben Sie, wie Operationen in Java ausgewertet werden. Nennen Sie einen Operator, der – angewandt auf verschiedene Datentypen – eine jeweils unterschiedliche Semantik hat und geben Sie dafür je ein Beispiel. ×10

Kontrollstrukturen (10)

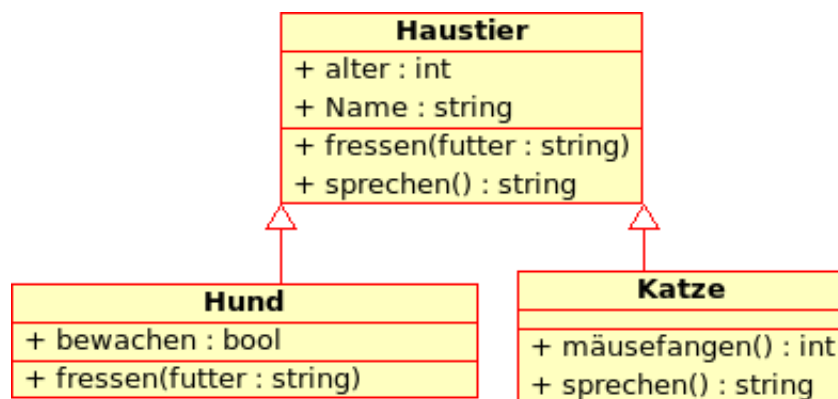
5. Wie lautet die Ausgabe von folgendem Algorithmus? ×10

```
class Test {
    public static void main(String[] args) {
        String[] ICE_STOPS = { "Linz", "Regensburg",
                               "Nürnberg", "Würzburg",
                               "Hanau" };
        boolean gerade = true;
        while (true) {
            for (int i=0; i<ICE_STOPS.length; i++) {
                if (gerade && (i%2 == 0)) {
                    System.out.print(ICE_STOPS[i]);
                } else if (!(gerade) && (i%2 != 0)) {
                    System.out.print(ICE_STOPS[i]);
                }
                System.out.print(" ");
            }
            System.out.println();
            gerade = ! (gerade);
        }
    }
}
```

6. Schreiben Sie ein Java-Programm, das alle übergebenen Programmargumente (= Parameter der main Methode) in umgekehrter Reihenfolge ausgibt.

×10

Sie haben unten stehendes UML-Klassendiagramm erhalten. Die Klasse Haustier wurde bereits erstellt.



7. Erstellen Sie die Klasse Hund und schreiben Sie einen entsprechenden Konstruktor. Überschreiben Sie weiters die Methode fressen(String futter). Hier soll sichergestellt werden, dass ein Hund auch nur „Hundefutter“ bekommt. Etwas anderes frisst er nicht. Geben Sie entsprechende Kommentare des Hundes aus! Der Aufruf schaut folgendermaßen aus: ×30

```
public class TestHund {  
    public static void main(String[] args) {  
        Hund fester = new Hund(10, "Fester", true);  
        Hund phil = new Hund(3, "Phil", false);  
        fester.fressen("Hundefutter");  
        fester.fressen("aus der Mülltonne");  
        fester.fressen("Gras");  
        System.out.println(fester.sprechen());  
    }  
}
```

und erzeugt folgende Ausgabe:

```
Hhmmm das ist gut!  
Nein, aus der Mülltonne mag ich nicht!  
Nein, Gras mag ich nicht!  
Mein Name ist: Fester
```

8. Welche der folgenden Aussagen sind richtig?

×15

- In Java kann eine Klasse genau ein Interface implementieren.
- Bei Initialisierung eines Objekts muss zuerst der Konstruktor der Superklasse aufgerufen werden.
- Beim Überschreiben von Methoden muss die Funktionalität der entsprechenden Methode in der Superklasse innerhalb der überschriebenen Methode aufgerufen werden.
- Im Gegensatz zum Konstruktor kann der Aufruf einer Supermethode zu beliebigem Zeitpunkt erfolgen.
- Eine abstrakte Methode muss immer von einer abgeleiteten Klasse implementiert werden.

Die restlichen Fragen sind ausschließlich für ECTS 5.0 vorgesehen!

Packete

(15)

9. Wie setzt sich ein voll qualifizierter Klassenname zusammen? Geben Sie ein Beispiel. ×5

10. Wann müssen Sie Klassennamen nicht voll qualifizieren? Nennen Sie verschiedene Möglichkeiten? ×10

Zugriffsbeschränkungen

(8)

11. Welche der folgenden Aussagen sind richtig? ×10

- Um auf ein Attribut ausschließlich innerhalb der Klasse zugreifen zu können, kann dieses als `private` deklariert werden.
- Standardmässig können Methoden innerhalb von Packeten und Unterpaketten aufgerufen werden.
- Get/Set Methoden werden implizit in Java als `public` deklariert.
- Zugriffskontrollen können nach Vererbung in Java nicht weiter eingeschränkt werden.