

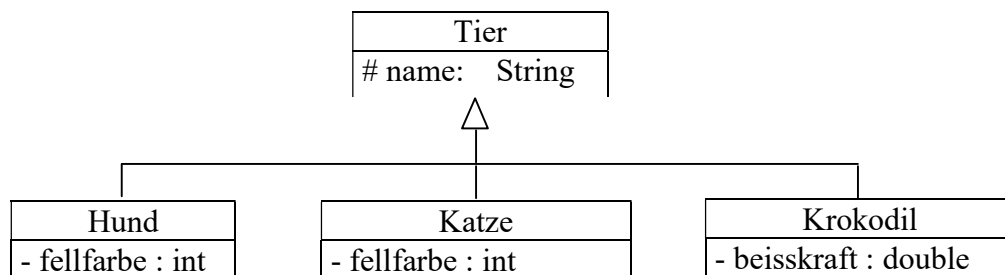
*Name, Vorname:*Hilfsmittel:
keine

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß in DRUCKSCHRIFT auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Sichtbarkeits- und Zugriffsmodifizierer wie public, usw. dürfen nicht weggelassen werden.
- In allen Klassen dürfen in keiner Methode Ein- oder Ausgaben (auf dem Bildschirm) vorkommen, außer es handelt sich um reine Ausgabe- oder Eingabe Methoden.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punktabzug !!!!

AUFGABEN

1) Gegeben ist das folgende abgespeckte, **fest vorgegebene** UML-Diagramm.
Die Klassenhierarchie darf also **nicht** verändert werden.



a) 39P

Erzeugen Sie aus dem folgenden UML-Diagramm die entsprechenden Klassen mit den jeweiligen Konstruktoren, set- und get-Methoden. Keine weiteren Attribute einfügen!
Die Konstruktoren müssen jeweils Parameter enthalten.

Wichtig: Jede Farbe wird durch einen Integer-Wert dargestellt, wie z.B:

... -100: weiß -99: gelb,

Umgekehrt entspricht jedem Integer-Wert eine Farbe!

Bitte keine "Umrechnungstabelle" der Zahlenwerte in tatsächliche Farben erstellen!

Hier ist eine Farbe ein Zahlenwert, mehr nicht!

b) 11P

Das Programm soll multipersonal entwickelt werden: Zu Testzwecken sollen die Entwickler der einzelnen Klassen gezwungen werden, die Methode
String getBeschreibung()

zu entwickeln, die die Namen der Attribute mit den zugehörigen Werten auf dem Bildschirm ausgibt.

In einem Feld sollen verschiedene Tiere (Hunde, Katzen, Krokodile) abgespeichert werden.

Danach sollen die Werte der Attribute mit der Methode getBeschreibung() ausgegeben werden. Machen Sie folgendes in der Methode main()

b1) Erzeugen Sie einen Hund, eine Katze und ein Krokodil.

b2) Speichern Sie diese 3 Tiere in dem Feld.

b3) Geben Sie mit Hilfe einer Schleife und der Methode getBeschreibung() die Eigenschaften der jeweiligen Tiere auf dem Bildschirm aus.

Lösung:

public class MainKlassenarbeit {		11P
public static void main(String[] args) {		
int i;		
Katze katze;		
Hund hund;		
Krokodil krokodil;		
Tier tiere[];	1P	
tiere = new Tier[3];	1P	
katze = new Katze("Ute", 2);	1P	
tiere[0] = katze;	1P	
hund = new Hund("Rex", 3);	1P	
tiere[1] = hund;	1P	
krokodil = new Krokodil("Krok", 30);	1P	
tiere[2] = krokodil;	1P	
System.out.println("Beschreibung der Tiere:");		
for(i=0; i<tiere.length; i++){	3P	
System.out.println(tiere[i].getBeschreibung());		
}		
}		
}		
 abstract class Tier{	1P	11P
protected String name;	1P	
 public Tier(String pName){	2P	
name = pName;		
}		
 public void setName(String pName){	2P	
name=pName;		
}		
 public String getName(){	2P	
return(name);		
}		
 abstract public String getBeschreibung();	3P	
}		
 class Katze extends Tier {		10P
private int fellfarbe;		
 public Katze(String pName, int pFellfarbe){		
super(pName);		
fellfarbe = pFellfarbe;		
}		
 public String getBeschreibung(){		
String s;		
s="Name="+name+" Fellfarbe="+fellfarbe;		
return s;		
}		
 public void setFellfarbe(int pFellfarbe){		
fellfarbe = pFellfarbe;		
}		

```

    public int getFellfarbe() {
        return fellfarbe;
    }
}

```

```

class Hund extends Tier {
    private int fellfarbe;

```

10P

```

    public Hund(String pName, int pFellfarbe) {
        super(pName);
        fellfarbe = pFellfarbe;
    }

```

```

    public String getBeschreibung() {
        String s;
        s="Name="+name+" Fellfarbe="+fellfarbe;
        return s;
    }

```

```

    public void setFellfarbe(int pFellfarbe) {
        fellfarbe = pFellfarbe;
    }

```

```

    public int getFellfarbe() {
        return fellfarbe;
    }
}

```

```

class Krokodil extends Tier{
    private double beisskraft;

```

10P

```

    Krokodil(String pName, double pBeisskreaft){
        super(pName);
        beisskraft = pBeisskreaft;
    }

```

```

    public String getBeschreibung() {
        String s;
        s="Name="+name+" Beisskraft="+beisskraft;
        return s;
    }

```

```

    public void setBeisskraft(double pBeisskraft) {
        beisskraft = pBeisskraft;
    }

```

```

    public double getBeisskraft() {
        return beisskraft;
    }
}

```

Name, Vorname:

Hilfsmittel:

zwei DIN A4-Seiten handschriftliche Aufschriebe (keine Kopien).

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß in DRUCKSCHRIFT auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Sichtbarkeits- und Zugriffsmodifizierer wie public, usw. dürfen nicht weggelassen werden.
- In allen Klassen dürfen in keiner Methode Ein- oder Ausgaben (auf dem Bildschirm) vorkommen, außer es handelt sich um reine Ausgabe- oder Eingabe Methoden.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punktabzug !!!!

AUFGABEN

I)

1) 22P

Es soll ein Taschenrechner (Addition und Subtraktion) mit Hilfe eines HTML-Formulars erzeugt werden.

Die Rechnung wird von einem Programm auf einem Webserver gemacht.

Nur entsprechendes HTML-Formular schreiben (kein Servlet).

2) Sie wollen einen Link in einer HTML-Datei erzeugen. 4P

Was ist der Unterschied zwischen

a href

und

link

3) Geben Sie ein konkretes Beispiel an, was man mit CSS und HTML im Gegensatz zu reinem HTML machen kann. 4P

4)

Sie wollen alle ihre HTML-Dateien mit der Hintergrundfarbe rot versehen, indem in alle HTML-Dateien die Datei mycssstyle.css eingebunden wird.

a) Welchen Text muss die Datei myCssStyle.css enthalten? 4P

Benutzen Sie dazu die Formateigenschaft background und die Farbe red.

In reinem HTML (ohne CSS) würde man es so machen:

`<body bgcolor=red>`

b) Wie (mit welchem Text) wird die Datei myCssStyle.css in eine HTML-Datei eingebunden? 4P

5) a) Geben Sie ein konkretes Beispiel an, was man mit XML im Gegensatz zu HTML machen kann. 4P

Gegeben ist die folgenden Datei demo3.xml:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="c.xsl" ?>
<EUROPA>
  <LAND>
    <NAME>Deutschland</NAME>
    <EINWOHNERZAHL EINHEIT="Millionen">82.4</EINWOHNERZAHL>
    <HAUPTSTADT>Berlin</HAUPTSTADT>
    <KFZ-KENNZEICHEN>D</KFZ-KENNZEICHEN>
    <TEL-VORWAHL>0049</TEL-VORWAHL>
  </LAND>
  <LAND>
    <NAME>Frankreich</NAME>
    <EINWOHNERZAHL EINHEIT="Millionen">58.5</EINWOHNERZAHL>
    <HAUPTSTADT>Paris</HAUPTSTADT>
    <KFZ-KENNZEICHEN>F</KFZ-KENNZEICHEN><TEL-VORWAHL>0033</TEL-VORWAHL>
  </LAND>
</EUROPA>
```

1) Aus der Datei demo3.xml wird mit der folgenden Datei demo3.xsl durch XSLT die Datei demo3.html erzeugt:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns="http://www.w3.org/TR/REC-html40">
  <xsl:output method="html" version="4.0" indent="yes"
    encoding="ISO-8859-1"/>
</xsl:stylesheet>
```

Welche Ausgabe erscheint auf dem Bildschirm?

2) Aus der Datei demo3.xml wird mit der folgenden Datei demo3.xsl durch XSLT die Datei demo3.html erzeugt:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns="http://www.w3.org/TR/REC-html40">
  <xsl:output method="html" version="4.0" indent="yes"
    encoding="ISO-8859-1"/>
  <xsl:template match="/">
    <h1>DEUTSCHLAND</h1>
  </xsl:template>
</xsl:stylesheet>
```

Welche Ausgabe erscheint auf dem Bildschirm?

Lösungen:

I)

1) a)

<body>

Taschenrechner

```
<form action="http://localhost:8080//test1.html" method="get">
1. Zahl:
<input type="text" name="Zahl1" size=60 maxlength=60 value=""><br><br>
2. Zahl:
<input type="text" name="Zahl2" size=60 maxlength=60 value=""><br><br>
<input type="submit" value="Addition">
<input type="submit" value="Subtraktion">
<input type="reset" value="Rücksetzen">
</form>
</body>
```

2) Mit a href kann man einen Link erzeugen

Mit link erzeugt man einen Verweis auf eine Ressource (z.B. eine css-Datei), die sich außerhalb der HTML-Datei befindet.

3) Der Überschrift <h1> ein anderes Erscheinungsbild geben, als dies standardmäßig vorgesehen ist.

4)

a) Inhalt der Datei mycssstyle.css:

```
body {background:red;}
```

b) Einbinden in HTML-Datei

```
<link rel=stylesheet type="text/css" href="myCssStyle.css">
```

5) a)

Man kann selbst ein Tag erstellen, das es in HTML nicht gibt.

Z.B. den Tag <mytag>

II)

1) Deutschland 82.4 Berlin D 0049 Frankreich 58.5 Paris F0033

2) DEUTSCHLAND

Name, Vorname:

Hilfsmittel:

zwei DIN A4-Seiten handschriftliche Aufschriebe (keine Kopien).

Hinweise (unbedingt beachten):

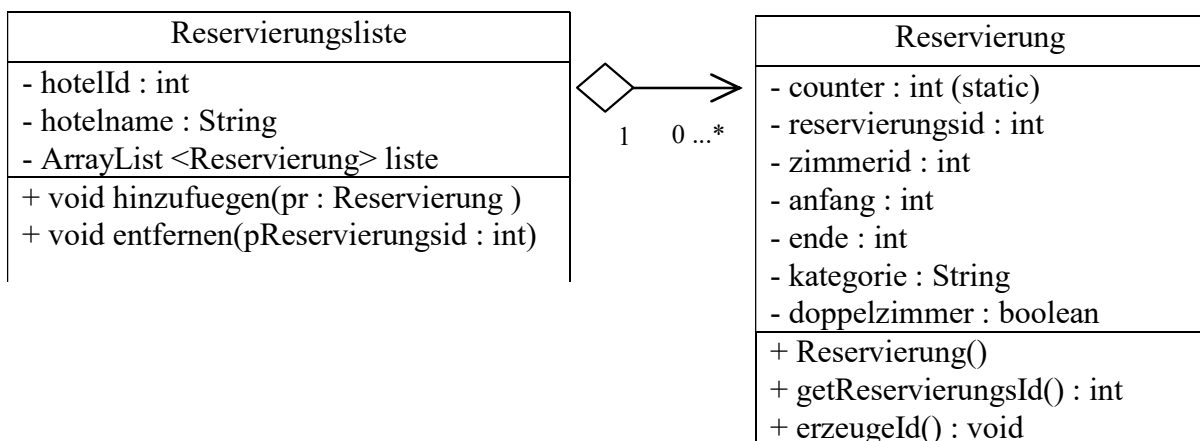
- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß in DRUCKSCHRIFT auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programnteile.
- Sichtbarkeits- und Zugriffsmodifizierer wie public, usw. dürfen nicht weggelassen werden.
- In allen Klassen dürfen in keiner Methode Ein- oder Ausgaben (auf dem Bildschirm) vorkommen, außer es handelt sich um reine Ausgabe- oder Eingabe Methoden.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punktabzug !!!!

AUFGABEN

I)

Für ein Hotel soll die Reservierung der Zimmer durch ein Programm realisiert werden.

Dazu wurde schon das folgende UML-Klassendiagramm entworfen:



1) Beschreiben Sie in kurzen Worten die im UML-Diagramm dargestellte Beziehungsform

2) Implementieren Sie die Klassen **Reservierungsliste** und **Reservierung** in der Programmiersprache Java.

3) Erstellen Sie in `main()` eine **Reservierungsliste**. Fügen Sie dieser Liste zwei **Reservierungen** hinzu. Löschen Sie dann die **Reservierung** mit der `reservierungsid 1`

Bitte Rückseite beachten

Hinweise:

1) Die Methoden haben die folgenden Aufgaben:

hinzufuegen(...) : speichert die übergebene Reservierung in der Reservierungsliste.

entfernen(...) : entfernt die Reservierung mit der angegebenen reservierungsid aus der Reservierungsliste.

Der Konstruktor der Klasse Reservierung setzt die eindeutige reservierungsid mit Hilfe der Methode erzeugeId().

2) Es genügt die im UML-Diagramm dargestellten Elemente zu implementieren.

Zusätzliche Zugriffsmethoden (get/set-Methoden) für die Attribute sind nicht notwendig.

3) Mit Hilfe der sich in der Entwicklungsumgebung von Java befindlichen Klasse ArrayList können dynamische Listen erzeugt werden (siehe auch die entsprechenden Methoden in der Klasse ArrayList).

Lösung:

Siehe Abschlussprüfung Anwendungsentwickler 2010

Erstellen Sie ein Struktogramm dieser Methode.

Lösung:

Siehe Abschlussprüfung Systemintegratoren 2010

Name, Vorname:

Hilfsmittel:

keine

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß in DRUCKSCHRIFT auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Sichtbarkeits- und Zugriffsmodifizierer wie public, usw. dürfen nicht weggelassen werden.
- In allen Klassen dürfen in keiner Methode Ein- oder Ausgaben (auf dem Bildschirm) vorkommen, außer es handelt sich um reine Ausgabe- oder Eingabe Methoden.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

AUFGABEN

1) 25P

Es soll die Summe der Zahlen $1 + 2 + \dots + 100$ berechnet werden (mit Hilfe einer Schleife).
Erstellen Sie dazu das entsprechende Struktogramm.

2) 25P

Es soll das Maximum (die größte Zahl) eines Integer-Feldes der Länge LEN berechnet werden. Erstellen Sie dazu das entsprechende Struktogramm.