

*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

1) 10P

Mit welchem HTML-Text wird folgende Tabelle realisiert (nur body-Teil angeben)

Die Tabelle muss 2 Zeilen, 3 Spalten und Gitterlinien besitzen.

Deutsch	Mathe	Englisch
1,5	2,4	1,6

2)

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?

5P

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 5P

in eine HTML-Datei eingebunden?

3)

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

a) 10P

Die Rechnung wird von einem Programm auf einem Webserver gemacht.

Nur entsprechendes HTML-Formular schreiben (kein Servlet).

b)

Realisieren Sie das durch ein java-Script in der Datei berechnung.js

20P

Bei der HTML-Datei head- und body-Teil angeben.

Bemerkung:

Mit

```
z=eval(window.document.xxxx.yyyy.value);
```

greift man auf ein Element eines Formulars zu. Dabei ist:

**xxxx** der Name des Formulars und

yyyy der Name des entsprechenden Formularfelds.

Mit

window.document.xxxx.yyyy.value = z

kann man ein Element eines Formulars beschreiben.

Lösungen:

1)

```
<body>
<table border>
  <tr>
    <td>Deutsch</td>
    <td>Mathe</td>
    <td>Englisch</td>
  </tr>

  <tr>
    <td>1,5</td>
    <td>2,4</td>
    <td>1,6</td>
  </tr>
</table>
</body>
```

2)

a) myStyle

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

b) Einbinden in HTML-Datei

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

3) 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>
```

b)

b1) HTML-Datei:

```
<head>
  <script language="JavaScript" src="berechnung.js"
    type="text/javascript">
    </script>
</head>
<body>
<form name="MyTaschenrechner">
1. Zahl: &nbsp; &nbsp; &nbsp;<input name="Zahl1" size=30
maxlength=30><br>
2. Zahl: &nbsp; &nbsp; &nbsp;<input name="Zahl2" size=30
maxlength=30><br>
Ergebnis:&nbsp;<input name="Ergebnis" size=30
maxlength=30><br><br>
<input type=button value="Addition" onClick="addition()">
<input type=button value="Subtraktion"
onClick="subtraktion()"><br>
</form>
</body>
```

b2) Java-Script Datei berechnung.js

```
function addition() {
  var erg=0;
  var z1=0;
  var z2=0;

  z1=eval(window.document.MyTaschenrechner.Zahl1.value);
  z2=eval(window.document.MyTaschenrechner.Zahl2.value);
  erg = z1 + z2;
  window.document.MyTaschenrechner.Ergebnis.value=erg;
}

function subtraktion() {
  var erg=0;
  var z1=0;
  var z2=0;

  z1=eval(window.document.MyTaschenrechner.Zahl1.value);
  z2=eval(window.document.MyTaschenrechner.Zahl2.value);
  erg = z1 - z2;
  window.document.MyTaschenrechner.Ergebnis.value=erg;
}
```

*Name, Vorname:*

Hilfsmittel:

selbstgeschriebenes zweiseitiges DIN A4 Blatt

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 !!!!

## AUFGABE

Erzeugen Sie ein dreispaltiges Framset-Fenster:

1. Spalte bekommt 10% der Gesamtbreite zugewiesen.
2. Spalte bekommt 10% der Gesamtbreite zugewiesen.
3. Spalte bekommt 80% der Gesamtbreite zugewiesen.

Der dazugehörige HTML-Text soll in der Datei main1.html geschrieben werden.

Im 1. Fenster erscheint die Meldung:

"Dieser Link bringt Sie zur E1F1", wobei E1F1 einen Link auf die Datei e1fi.html ist, der den Inhalt der Datei e1fi.html im 3. Unterfenster ausgibt.

Die Meldung: "Dieser Link bringt Sie zur E1F1" muss in der Datei datei1.html gespeichert sein und von dort aus in das 1. Fenster geladen werden.

Der Inhalt der Datei e1fi.html muss die folgende Ausgabe veranlassen:

"Hallo E1FI: Als Einstiegsklasse bekommen Sie die Schulordnung im Sekretariat ausgehändigt."

Im 2. Fenster erscheint die Meldung:

"Dieser Link bringt Sie zur E2F1", wobei E2F1 einen Link auf die Datei e2fi.html ist, der den Inhalt der Datei e2fi.html im 3. Unterfenster ausgibt.

Die Meldung: "Dieser Link bringt Sie zur E2F1" muss in der Datei datei2.html gespeichert sein und von dort aus in das 2. Fenster geladen werden.

Der Inhalt der Datei e2fi.html muss die folgende Ausgabe veranlassen:

"Hallo E2FI: Der Notendurchschnitt der letzten SAE-Klassenarbeit ist 3,4"

Im 3. Fenster erscheint die Meldung:

Dies ist das Ausgabefenster. Klicken Sie den jeweiligen Link an.

Die Meldung: "Dies ist das Ausgabefenster. Klicken Sie den jeweiligen Link an." muss in der Datei datei3.html gespeichert sein und von dort aus in das 3. Fenster geladen werden.

Geben Sie den HTML-Text an, der sich in den folgenden Dateien befindet:

main1.html, datei1.html, datei2.html, datei3.html, e1fi.html, e2fi.html

Lösungen:

**main1.html:**

```
<html>
<head>
<title>Titel des Hauptfensters</title>
</head>

<frameset cols="10%, 10%, 80% ">
  <frame src="datei1.html" name="Unterfenster1">
  <frame src="datei2.html" name="Unterfenster2">
  <frame src="datei3.html" name="Unterfenster3">
</frameset>

<body>
</body>
</html>
```

**datei1.html:**

```
<html>
<head>
  <title>Titel Fenster 1
  </title>
</head>
```

Dieser Link bringt Sie zur E1F1<br>

```
<a href="elfi.html" target="Unterfenster3">elfi.html</a>
```

```
<body>
</body>
```

```
</html>
```

**datei2.html:**

```
<html>
<head>
  <title>Titel Fenster 2
  </title>
</head>
```

```
<body>
```

Dieser Link bringt Sie zur E2F1<br>

```
<a href="e2fi.html" target="Unterfenster3">e2fi.html</a>
```

```
</body>
</html>
```

**datei3.html:**

```
<html>
<head>
  <title>Titel Fenster 3
  </title>
</head>
```

```
<body>
```

Dies ist das Ausgabefenster<br>

Klicken Sie den jeweiligen Link an<br>

```
</body>
</html>
```

**e1fi.html:**

```
<html>
<head>
  <title>Titel Fenster e1fi
</title>
</head>
<body>
Hallo E1FI:<br>
Als Einstiegsklasse bekommen Sie die Schulordnung im
Sekretariat ausgehändigt.<br>
</body>
</html>
```

**e2fi.html:**

```
<html>
<head>
  <title>Titel Fenster e2f1
</title>
</head>
<body>
Hallo E2FI:<br>
Der Notendurchschnitt der letzten SAE-Klassenarbeit ist 3,4<br>
</body>
</html>
```

*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 Punkteabzug !!!!

## AUFGABEN

- 1) 5P  
Welche HTML-Tags sind unbedingt notwendig, um eine Tabelle zu erstellen, die das kleine Einmaleins beinhaltet. (Nur Tags angeben)
- 2) 5P  
Sie wollen mit HTML Daten an einen Server versenden.  
Was brauchen Sie - auf HTML bezogen - dazu unbedingt?
- 3) 10P  
Sie wollen einen Link in einer HTML-Datei erzeugen.  
Was ist der Unterschied zwischen  
a href  
und  
link
- 4) 6P  
Geben Sie ein konkretes Beispiel an, was man mit CSS und HTML im Gegensatz zu reinem HTML machen kann.
- 5) a) 6P  
Geben Sie ein konkretes Beispiel an, was man mit XML im Gegensatz zu HTML machen kann.
- b) 6P  
Mit einem Web-Browser wollen Sie eine XML-Datei auf dem Bildschirm ausgeben.  
Welche zwei prinzipiellen Möglichkeiten gibt es, die Ausgabe zu gestalten?
- c) 6P  
Was ist XSLT, warum wird es benötigt und in welcher Beziehung steht es zu XML?  
Bitte konkrete Erklärung !
- d) 6P  
Was kann man mit XSLT, was im Gegensatz zu CSS machen kann.

Lösungen:

1) <table>, <tr>, <td>

2) Formular

3) 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.

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

5) a)

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

Z.B. den Tag <mytag>

b) Mit CSS oder XSLT

c) Wenn man in einer XML-Datei ein selbstdefiniertes Tag erzeugt hat, muss dieses ja noch irgendwie interpretiert werden. Dies wird in einer XSLT-Datei gemacht.

d) Schleifen, wie beim Programmieren



*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

1) 10P

Erzeugen Sie ein Formular mit den 2 Eingabefeldern "Vorname" und "Nachname", dem Button "Rücksetzen" und "Daten versenden".

2) 8P

Was bedeutet die folgende Formatdefinition ?

```
.meinpink
{
  color: pink
}
```

3)  
Eine mit Hilfe von <h1> erzeugte Überschrift soll komplett mit rot dargestellt werden.  
Dies soll mit Hilfe von CSS realisiert werden.

a) 8P  
Machen Sie dazu ein Beispiel (Format zentral definieren, d.h: Formatdefinition im Header der HTML-Datei).

b) 8P  
Was muss gemacht werden, wenn eine Formatierung (z.B: Überschrift <h1> blau) nur in einem bestimmten Bereich der HTML-Datei gelten soll?

c) 8P  
Was muss gemacht werden, wenn sich die entsprechenden Formatierungen in einer eigenen css-Datei befinden. Machen Sie dazu ein Beispiel.

Lösungen:

1)

```
<html>
<head></head>
<body>
<form action="http://localhost:8080//test1.html" method="get">
Vorname:
<input type="text" name="myvorname" size=60 maxlength=60 value="Erika"><br><br>
Nachname:
<input type="text" name="mynachname" size=60 maxlength=60
value="Mustermann"><br><br>
<input type="submit" value="Daten versenden">
<input type="reset" value="Rücksetzen">
</form>
</body>
</html>
```

2) Es wird eine Format-Unterklasse definiert, die sich auf irgendein HTML-Tag anwenden lassen, wie z.B:

```
<h1 class="meinpink"> Ich bin pink</h1><br>
<p class="meinpink"> so schön ist pink</p>
```

3) a) + b)

```
<html>
<head>
<style type="text/css">
```

```
  h1
  {
    color:red;
  }
</style>
```

```
</head>
<body>
  <div>
    <h1 style="color:blue ">Willkommen</h1>
  </div>
</body>
</html>
```

c)

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

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

Bemerkung: Jede Teilaufgabe entspricht einem Programmteil. Die ganze Aufgabe soll durch **ein** Projekt realisiert werden.

I)

1)

13P

Nichtmathematikinteressierte Fachinformatiker sollen eine Klasse "GeoForm" entwickeln, die eine Methode `getOberfläche()` besitzt, die die Oberfläche (= Fläche der Unterseite + Fläche der Oberseite) eines zweidimensionalen Gebildes (wie z.B. Kreis, Rechteck, Parallelogramm, Dreieck, usw.) berechnet.

Die Methode `getOberfläche()` muß allerdings mit Hilfe der abstrakten Methode `getFlaeche()` berechnet werden, die die Fläche eines zweidimensionalen Gebildes (wie z.B. Kreis, Rechteck, Parallelogramm, Dreieck, usw.) berechnet.

Die abstrakte Methode `getUmfang()` wird auch benötigt.

Erstellen Sie die Klasse "GeoForm" den entsprechenden Methoden und - falls nötig - mit den entsprechenden Attributen.

2)

21P

Mathematikinteressierte Fachinformatiker sollen die Klassen "Rechteck" und "Kreis" entwickeln (unter Zuhilfenahme der Vererbung), mit den entsprechenden Methoden (und Nicht Standard-Konstruktoren) und mit den entsprechenden Attributen.

Bemerkung:

Umfang Kreis =  $2 * \pi * \text{Radius}$

Fläche Kreis =  $\pi * \text{Radius} * \text{Radius}$

3)

16P

Machen Sie im "Hauptprogramm" `main()` folgendes:

- a) Erzeugen Sie ein Rechteck Mit Länge 1 und Breite 2
- b) Erzeugen Sie einen Kreis mit Radius 3
- c) Berechnen Sie die Oberfläche des Rechtecks.
- d) Berechnen Sie den Umfang des Rechtecks.
- e) Berechnen Sie die Oberfläche des Kreises.
- f) Berechnen Sie den Umfang des Kreises.

Name, Vorname:

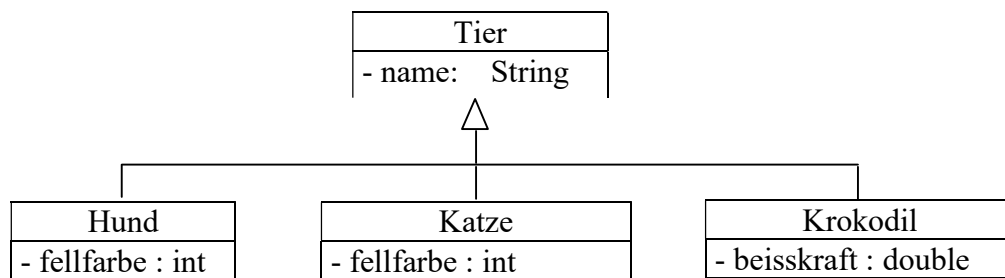
Hilfsmittel:  
keine

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 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

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



a) 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!

39P

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

Hier ist eine Farbe ein Zahlenwert, mehr nicht!

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

11P

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:

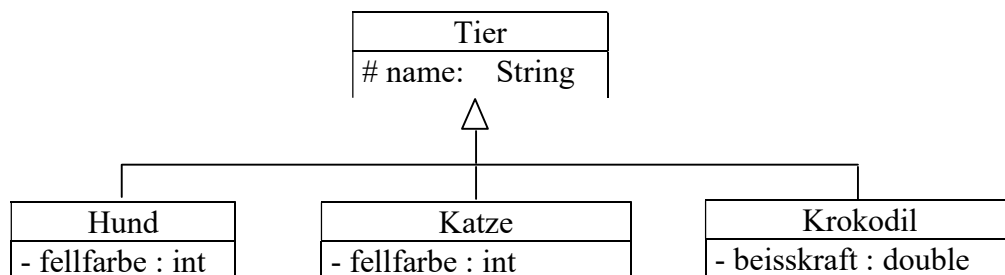
Hilfsmittel:  
keine

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 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

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



a) Erzeugen Sie aus dem folgenden UML-Diagramm die entsprechenden Klassen mit den jeweiligen Konstruktoren, set- und get-Methoden. Keine weiteren Attribute einfügen!  
Wichtig: Jede Farbe wird durch einen Integer-Wert dargestellt, wie z.B:

-100: weiß 1:gelb, usw.

Umgekehrt entspricht jedem Integer-Wert eine Farbe!

39P

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

Hier ist eine Farbe ein Zahlenwert, mehr nicht!

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

11P

void 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.

c)

12,5P

Beachten Sie:

Ein Krokodil hat - nach heutigem biologischen Wissenstand - kein Fell.

Dagegen besitzt eine Katze bzw. ein Hund ein Fell.

In dem Feld "felktiere" sollen die im vorigen Programmteil erstellte Katze und Hund abgespeichert werden und dann mit Hilfe einer Schleife und der Methode getFellfarbe() die Farbe des jeweiligen Tiers auf dem Bildschirm ausgegeben werden.

c1) Was wäre der Nachteil der Lösung, in der man die Methode getFellfarbe() in der Klasse Tier implementiert?

c2) Beurteilen Sie die folgende Lösung:

Die Methode getFellfarbe() wird nur (und sonst nirgends) in die Klasse Hund und Katze implementiert.

c3) Was ist die beste Lösung ? (keine Implementierung, nur verbale Beschreibung und Begründung).

Bewertung:

62,5P	Note 1
50P	Note 2
37,5P	Note 3
25P	Note 4
12,5P	Note 5
0P	Note 6



**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<3;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;
    }
}

```

c1)

4P

Dadurch erbt die Klasse Krokodil u.a. die Methode getFellfarbe().

Dadurch kann man die Fellfarbe eines Krokodils herausfinden, obwohl ein Krokodil keine Fellfarbe besitzt.

c2) Beurteilen Sie die folgende Lösung:

4P

Die Methode getFellfarbe() werden nur (und sonst nirgends) in die Klasse Hund und Katze implementiert.

Da in einer Schleife z.B. mit felltiere. getFellfarbe() die Farbe abgefragt wird, muss die Methode auch in der Klasse Tier existieren. Da dies nicht der Fall ist, meldet der Compiler einen Fehler.

c3)

4,25P

Katze und Hund implementieren das Interface FelltierIF, wo die Methodenköpfe getFellfarbe() und setFellfarbe(...) angegeben werden.

Nur in den Klassen Hund und Katze müssen diese Methoden dann ausprogrammiert werden.