

Schreibe ein Klasse person mit den Datenfeldern:

```
private String name;  
private String vorname;
```

und mit den Methoden

```
public void setName (String n)  
public String getName()  
public void setVorname (String n)  
public String getVorname()
```

Teste diese Klasse mit der Klasse Testperson:

```
public class TestPerson  
{  
    public static void main (String args[])  
    {  
        String vorname;  
        String name;  
        Person edinson;  
  
        edinson = new Person();  
        edinson.setName ("Edinson");  
        edinson.setVorname ("Thomas");  
  
        name = edinson.getName();  
        vorname = edinson.getVorname();  
        System.out.println ("Vorname : " + vorname);  
        System.out.println ("Name : " + name);  
    }  
}
```

- 1) In einer Klasse **Angestellter** soll das Attribut **gehalt** (dezimale Attributwerte) sowie die Zugriffsmethoden enthalten sein. Schreiben Sie die dafür notwendigen Zeilen in Java!
- 2) In den angegebenen Methodenköpfen sind Fehler. Berichten Sie!
(a) **boolean Public berechneSteuer (double einkommen; double steuersatz)**
(b) **privat void druckeListe (Int anzahl, String überschrift)**
- 3) Entwerfen Sie zu folgenden Problemen einen Methodenkopf und begründen Sie Ihre Entscheidung!
(a) Berechnung des Flächeninhalts eines Rechtecks
(b) Ausgabe aller Schüler, deren Nachnamen innerhalb eines bestimmten Buchstabenbereichs liegen
(c) Berechnung des Terms $\frac{(a + b)^2}{c^3}$
- 4) Für die folgenden Probleme sollen die Methodenköpfe entworfen werden. Dazu müssen Überlegungen zum Ergebnis und zu den Parametern angestellt werden. Der vollständige Methodenkopf soll anschließend notiert werden! Für die Probleme (a) und (d) sollen auch die Methodenrumpfe ausformuliert werden!
(a) Berechnung des Flächeninhalts eines Rechtecks
(b) Ausgabe eines Namens
(c) Berechnung der Einkommenssteuer aus dem Jahresverdienst
(d) Berechnung des Mittelwertes aus drei Noten
- 5) Eine Klasse **Tankstelle** soll erstellt werden. Definieren Sie für die Klasse zwei private Attribute **diesel** und **benzin** und zwei Methoden **kaufe-Diesel** und **kaufeBenzin**!
- 6) Ein Rechteck enthält als Attribute die Länge und die Breite des Rechtecks. Diese Attribute enthalten dezimale Attributwerte. Entwickeln Sie den Kopf der Methoden! Begründen Sie Ihren Entwurf!
(a) Berechnung des Umfangs
(b) Vergleich mit einem anderen Rechteck hinsichtlich der Länge
- 7) Folgende Methoden seien deklariert:
(a) Dient zum Erzeugen einer neuen Person
public Person erzeugen (String name, String vorname)
(b) Berechnet aus dem Einkommen und dem Steuersatz die Einkommenssteuer
public int berechneSteuer (int einkommen, double steuersatz)
Die Anwendung der Methoden ist fehlerhaft. Finden Sie die Fehler und erläutern Sie, warum die Zeilen falsch sind!
 - i) erzeugen ("Meyer")
 - ii) berechneSteuer (2.5, 80000)
 - iii) berechneSteuer (40000, 5)
 - iv) erzeugen ("Meyer", 30)

Nutzen Sie die Dokumentation im Internet.

- a) Nennen Sie mindestens drei Klassen aus dem Paket **java.awt!**
- b) Nennen Sie mindestens zwei weitere Pakete!

Nutzen Sie die Dokumentation im Internet.

- a) Geben Sie eine Kurzbeschreibung für die Funktion der Klasse **java.awt.Button!**
- b) Von welcher Oberklasse erbt diese Klasse?
- c) Wie viele Konstruktoren gibt es? Worin unterscheiden sie sich?

- 1) In einer Klasse **Angestellter** soll das Attribut **gehalt** (dezimale Attributwerte) sowie die Zugriffsmethoden enthalten sein. Schreiben Sie die dafür notwendigen Zeilen in Java!

Lösung

```
public class Angestellter
{
    private double gehalt;
    public double getGehalt()
    {
        return gehalt;
    }
    public void setGehalt(double g)
    {
        gehalt = g;
    }
}
```

- 2) In den angegebenen Methodenköpfen sind Fehler. Berichtigen Sie!

(a) boolean Public berechneSteuer (double einkommen; double steuersatz)

Lösung

```
public boolean berechneSteuer(double einkommen, double steuersatz)
```

(b) privat void druckeListe (Int anzahl, String überschrift)

Lösung

```
private void druckeListe(int anzahl, String ueberschrift)
```

- 3) Entwerfen Sie zu folgenden Problemen einen Methodenkopf und begründen Sie Ihre Entscheidung!

(a) Berechnung des Flächeninhalts eines Rechtecks

Lösung

```
public double flaeche(double laenge, double breite)
```

Zur Berechnung sind dezimale Angaben zugelassen; Das Ergebnis ist der Rückgabewert;

(b) Ausgabe aller Schüler, deren Nachnamen innerhalb eines bestimmten Buchstabenbereichs liegen

Lösung

```
public void ausgeben(String anfang, String ende)
```

Rückgabewert void, da es sich nicht um einen einzelnen Wert handelt; Buchstabenkombinationen für Beginn und Ende des Bereich sind zu übergeben;

(c) Berechnung des Terms $\frac{(a+b)^2}{c^3}$

Lösung

```
public double berechne(double a, double b, double c)
```

berechneter Rückgabewert ist dezimal (double), a, b, c müssen zur Berechnung übergeben werden;

- 4) Für die folgenden Probleme sollen die Methodenköpfe entworfen werden. Dazu müssen Überlegungen zum Ergebnis und zu den Parametern angestellt werden. Der vollständige Methodenkopf soll anschließend notiert werden! Für die Probleme (a) und (d) sollen auch die Methodenrumpfe ausformuliert werden!

(a) Berechnung des Flächeninhalts eines Rechtecks

Lösung

```
public double flaeche(double laenge, double breite)
{
    return breite*laenge
}
```

Zur Berechnung sind dezimale Angaben zugelassen; Das Ergebnis ist der Rückgabewert;

(b) Ausgabe eines Namens

Lösung

```
public void ausgeben(String name)
```

Die Ausgabe berechnet kein Ergebnis, deshalb void. Die auszugebende Information wird als Parameter übergeben;

(c) Berechnung der Einkommenssteuer aus dem Jahresverdienst

Lösung

```
public int berechneSteuer(int verdienst, int steuerklasse, ...)
```

im Allgemeinen wird hier mit ganzzahligen Werten gearbeitet, daher int; vielfältige weitere Angaben sind zur Berechnung erforderlich, die Lösung kann also entsprechend erweitert werden;

(d) Berechnung des Mittelwertes aus drei Noten

Lösung

```
public double mittelw(int not1, int not2, int not3)
{
    return (not1+not2+not3)/3.0; (3.0 damit die Berechnung dezimal erfolgt)
}
```

Der Mittelwert wird im Allgemeinen dezimal sein, daher double, die Noten sind ganzzahlig;

- 5) Eine Klasse **Tankstelle** soll erstellt werden. Definieren Sie für die Klasse zwei private Attribute **diesel** und **benzin** und zwei Methoden **kaufe-Diesel** und **kaufeBenzin**!

Lösung

```
public class Tankstelle
{
    private int diesel;
    private int benzin;
    public void setDiesel(int d)
    {
        diesel=d;
    }
    public int getDiesel()
    {
        return diesel;
    }
    public void setBenzin(int b)
    {
        benzin=b;
    }
    public int getBenzin()

```

```
{
return benzin;
}
public void kaufeDiesel(int liter)
{
    diesel=diesel-liter;
}
public void kaufeBenzin(int liter)
{
    benzin=benzin-liter;
}
}
```

6) Ein Rechteck enthält als Attribute die Länge und die Breite des Rechtecks. Diese Attribute enthalten dezimale Attributwerte. Entwickeln Sie den Kopf der Methoden! Begründen Sie Ihren Entwurf!

(a) Berechnung des Umfangs

Lösung

```
public double berechneU()
```

Der Umfang kann dezimal sein, also ist der Rückgabewert double; Parameter sind nicht erforderlich, weil die Rechtecksdaten innerhalb der Klasse bekannt sind;

(b) Vergleich mit einem anderen Rechteck hinsichtlich der Länge

Lösung

```
public String vergleiche(Rechteck r)
```

Die benötigte Information (länge) gehört zur Klasse des Rechtecks r; der Vergleichsgegenstand wird als Parameter übertragen; Rückgabotyp kann String sein (z.B. „länger“, „kürzer“, „gleich lang“)

7) Folgende Methoden seien deklariert:

(a) Dient zum Erzeugen einer neuen Person

```
public Person erzeugen (String name, String vorname)
```

(b) Berechnet aus dem Einkommen und dem Steuersatz die Einkommenssteuer

```
public int berechneSteuer (int einkommen, double steuersatz)
```

Die Anwendung der Methoden ist fehlerhaft. Finden Sie die Fehler und erläutern Sie, warum die Zeilen falsch sind!

i) *erzeugen ("Meyer")*

Lösung: *zwei Parameter sind erforderlich*

ii) *berechneSteuer (2.5, 80000)*

Lösung: *Erst muß int angegeben werden*

iii) *berechneSteuer (40000, 5)
kein Fehler*

Lösung: *5.0 muß eingegeben werden; allerdings*

iv) *erzeugen ("Meyer", 30)*

Lösung: *zwei Zeichenketten erforderlich*

Leonardo von Pisa, genannt Fibonacci

Leonardo von Pisa wurde zwischen 1170 und 1180 geboren. Bekannt wurde er unter dem Namen Fibonacci, was eine Verkürzung von "Filius Bonacci", also "Sohn des Bonacci" ist. Er lernte auf Handelsreisen nach Algerien, Ägypten, Syrien, Griechenland, Sizilien und in die Provence alle damals bekannten Rechenverfahren kennen. Sein 1202 erschienenes, 459 Seiten starkes Werk *Liber Abaci* machte in Europa die indische Rechenkunst bekannt und führte die heute übliche arabische Schreibweise der Zahlen ein.

In der modernen Mathematik ist sein Name mit der Folge von natürlichen Zahlen verbunden, die sich aus dem berühmten "Kaninchenproblem" aus dem *Liber Abaci* ergibt:

Betrachtet wird die Nachkommenschaft eines (idealisierten) Kaninchenpaares, die bekanntlich sehr groß ist. Für die Simulation werden folgende Annahmen gemacht:

- Jedes Kaninchenpaar wird im Alter von zwei Monaten fortpflanzungsfähig.
- Jedes Kaninchenpaar bringt von da an jeden Monat ein neues Paar zur Welt.
- Alle Kaninchen leben ewig.

1. Wie lauten die ersten 15 Folgenglieder?
2. Wie lautet das rekursive Bildungsgesetz dieser Zahlenfolge?
3. Schreiben sie in Java eine rekursive Methode *long fib(int f)* zur Berechnung der Folgenglieder.
4. Implementieren Sie die Berechnung mit Hilfe einer iterativen Methode *long fib(int f)*. (Sie benötigen drei lokale Hilfsvariable)

Lösungen

Monat	1	2	3	4	5	6	7	8	9
Anz	1	1	2	3	5	8	13	21	34

Bildungsgesetz rekursiv: $a_{n+2} = a_{n+1} + a_n$

Methode in Java rekursiv

```
public long fib(int f){
    if (f==0 ||f==1) return 1;
    else return fib(f-1)+fib(f-2);
}
```

Methode iterativ

```
public long fib(int f)
{
    long fib = 1;
    for (long fib1=1, fib2=1, i=3; i <=f; i++)
    {
        fib=fib1+fib2;
        fib1 = fib2;
        fib2 =fib;
    }
    return fib;
}
```

Bemerkung: im Quelltext einer For-Schleife lassen sich mehrere Variablen deklarieren, sie werden durch Kommata getrennt.

Quelltext für eine einfache Stoppuhr:

```
public class Stoppuhr
{
    private long millis;

    public void starte()
    {
        millis = System.currentTimeMillis();
    }

    public void stoppe()
    {
        millis = System.currentTimeMillis() - millis;
    }

    public long zeit()
    {
        return millis;
    }
} // class Stoppuhr
```

1. Analysieren Sie das folgende Programm! Was wird hier berechnet? Schreiben Sie eine iterative Methode für die gleiche Funktion.

```
public class was {
    public int rekAuf(int n){
        if (n>1) return n+rekAuf(n-1);
        return 1;
    }

    public static void main(String[] args) {
        was rek=new was();
        System.out.println(rek.rekAuf(10));
    }
}
```

Lösung:

*// Berechnung der Summe der ersten 10 natuerlichen Zahlen:
// s = 1+2+...+10 = 55. Alternativer Algorithmus: s = 10*(10+1)/2*

```
public class Was
{
    public int itAuf (int n)
    {
        int wert = 0;
        for(int i=1; i<=n; i++)
        {
            wert = wert+i;
        }
        return wert;
    }

    public static void main (String[] args)
    {
        Was it=new Was();
        System.out.println(it.itAuf(10));
    }
}
```

2. }Welche mathematische Formel berechnet das folgende Programm? Wie lautet das Ergebnis?

```
public class Iteration {
    public int itAuf(int n){
        int wert =1;
        for(int i=2;i<=n;i++){
            wert *=i;
        }
        return wert;
    }

    public static void main(String[] args) {
        Iteration it=new Iteration();
        System.out.println(it.itAuf(4));
    }
}
```

Lösung:

*// Berechnung der Fakultaet des Uebergabeparameters:
// n! = 1*2*...*(n-1)*n. Das Ergebnis fuer die Zahl 4
// lautet: 4! = 1*2*3*4 = 24*

3. Es soll mit Hilfe einer Rekursion die Quersumme einer Zahl berechnet werden. Hierbei soll n von der Tastatur eingelesen werden. Dabei soll sich die Berechnung der Werte auf natürliche Zahlen größer als Null beschränken. Vervollständige hierzu das folgende Programm an den mit gekennzeichneten Stellen.

Quersumme rekursiv

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
public class Quersumme_rek
{
    public static ..... summe_reku(.....)
    {
        .....
    }
    public static void main (String[] args)
    {
        InputStreamReader inp = new InputStreamReader (Sys-
tem.in);
        BufferedReader buffer = new BufferedReader (inp);
        try
        {
            System.out.println ("Gib einen Wert >0 fuer n ein: ");
            Integer n = Integer.valueOf (buffer.readLine());
            int ergebnis = summe_reku (n.intValue());
            System.out.println ("Das Ergebnis ist: " + ergebnis);
        }
        catch (Exception ex)
        {
            System.out.println (ex.toString());}}}
```

Lösung:

```
public class Quersumme_rek
{
    public static int summe_reku(int n)
    {
        if(n<10&&n>=0)return n;
        else return summe_reku(n/10)+n%10;
    }
}
```

Beispiel : $n = 234$

$\text{summe_reku}(234/10 = 23) \quad +234\%10 = 4$

$\text{summe_reku}(23/10 = 2) \quad + 23\%10 = 3$

$= 2 \quad \text{summe_reku}(234) = 2 + 3 + 4 = 9$

$234/10$ Ganzzahlige Division \rightarrow Ergebnis ist 23

$234\%10$ Rest bei der Division durch 10 \rightarrow Ergebnis ist 4

4. Es soll mit Hilfe einer Rekursion die Potenz a^n berechnet werden. Hierbei sollen a und n von der Tastatur eingelesen werden. Dabei soll sich die Berechnung auf Werte größer 0 für a und n beschränken. Vervollständigen Sie hierzu im folgenden Programm die mit `.....` gekennzeichneten Stellen im unten angegebenen Code. Die Potenz berechnet sich beispielsweise für $n = 5$ und $a = 2$ wie folgt: $2*2*2*2*2 = 32$

```
public class PotenzRekursiv
{
    public static . . . . berechnePotenz (. . . .)
    {
        . . . .
    }
    public static void main (String [] args)
    {
        InputStreamReader inp = new InputStreamReader (System.in);
        BufferedReader buffer = new BufferedReader (inp);
        try
        {
            System.out.println ("Gib einen Wert >0 für a
            ein: ");
            Integer a = Integer.valueOf (buffer.readLine());
            System.out.println ("Gib einen Wert >0 für n
            ein: ");
            Integer n = Integer.valueOf (buffer.readLine());
            int ergebnis = berechnePotenz
            (a.intValue(),n.intValue());
            System.out.println ("Das Ergebnis ist: " + ergebnis);
        }
        catch (Exception ex)
        {
            System.out.println (ex.toString());
        }
    }
}
```

Lösung:

```
public class PotenzRekursiv
{
    public static int berechnePotenz (int basis, int exponent)
    {
        if (exponent == 1)
            return basis;
        return basis * berechnePotenz (basis, exponent - 1);
    }
}
```

5. Erweitern Sie die Methode `berechnePotenz()` so, dass sie auch mit 0 Werten für `a` und `n` zurecht kommt. Negative Werte sollen weiterhin nicht betrachtet werden. Hierbei ist zu beachten, dass eine Potenz für die Hochzahl 0 immer 1 liefert, außer für den Fall, dass `a == 0` ist. Bei einem Wert von `a == 0` ist das Ergebnis immer 0.

Lösung:

```
public static int berechnePotenz (int basis, int exponent)
{
    if (basis == 0)
        return 0;
    if (exponent == 0)
        return 1;
    if (exponent == 1)
        return basis;

    return basis * berechnePotenz (basis, exponent - 1);
}
```

6. Schreiben Sie eine neue Klasse `PotenzIterativ`. Diese Klasse soll die Berechnung der Potenz (Lösung von Aufgabe 4.) in einer Iteration berechnen. Verwenden Sie dazu eine `for`-Schleife.

Lösung:

```
public static int berechnePotenz (int basis, int exponent)
{
    int ergebnis;
    if (basis == 0)
    {
        ergebnis = 0;
    }
    else
    {
        ergebnis = 1;
        for (int i = 0; i < exponent; i++)
        {
            ergebnis *= basis;
        }
    }
    return ergebnis;
}
```