Inhalt

[Aufgabe: Bankkunden (sortieren 2.Semester) 2](#_Toc96928283)

[Formeln in Java 4](#_Toc96928284)

[IF 4](#_Toc96928285)

[Debugger 8](#_Toc96928286)

[Primitive Datentypen 14](#_Toc96928287)

[Fibonacci Zahlen 14](#_Toc96928288)

# Aufgabe: Bankkunden (sortieren 2.Semester)

Classe: Bankkunde eine Klasse ist eine Vorlage ein Bauplan und besteht aus:

Attributen

|  |  |  |
| --- | --- | --- |
| Classe | Bankkunde |  |
| Name: String |  |  |
| id: int |  |  |
| alter: int |  |  |
| kontostand: int |  |  |
| Usw. |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Methoden (auch Konstruktoren)  
Name Aendern (nameNeu) String  
Einzahlen (betrag int  
Abheben (betrag int  
getName () String

Code Stil : keine Umlaute

Datentypen:   
Zahlen: long (mehr zahlen) int c.a. +/- 2\*109  
Zeichenketten: String “Max“  
Boolsche Werte: boolean: true false

Public\_ class\_ Bankkunde

{

Private String name =

Private int id =

Private int alter=

Private int kontostand =

Public void nameAendern ( String nameNeu)

{

Name = nameNeu;

{

}

public class Bankkunde

{

// Attribute

private String name;  
private int alter;  
private long kontostand;

// Konstruktor wenn das was nach public kommt gleich mit dem Klassen namen ist

public Bankkunde ()

{

//3 Befehle, Zuweisungen: Variable = Wert, Variable1 = Variable2

name=“Max Muster“;

alter=31;

kontoStand=2000;

}

Public Bankkunde(String name, int alter, int kontoStand)

{

// wir brauchen this. da die Parameternamen gleich dem eigenschaftsnamen

this.name = name;

this.alter = alter;

this.kontoStand = kontoStand;

}

// get methoden

public String getName ()

{

return name;

}

//set Methoden

public void setName (String name)

{

This.name= name;

}

Public void einzahlen(int betrag)

{

kontoStand = kontoStand + betrag;

}

}

# Formeln in Java

Mathe in Java:

Mathe Java Hinweis

X=3

Int x = 3 Variable x vom Typ int definiert und den Wert 3 zugefügt

Int a,b = 2; a=2\*b Variablen a,b als int definiert und Wert 2 zugefügt. A = 2\*b gerechnet.

Double m=1,5; Später Math.pow(..)  
Double c = 300 000;  
double e = m\*c\*c;

Int a= 5; int b=3;

Int flaeche = (a\*b)/2; flaeche ist 7 da int und nicht double oder float

A=2b

E=m\*c2

A=a\*b/2

Bsp Parameter + lokal Variablen

Public void Rechnen (, double u, double x)

{

Double a = 2\*u+3\*x;

Double y = a\*x\*3-2\*u\*a;

System.out.println (“Y ist „+y);

}

# IF

If([Bedingung])

{

Teil A

}

Else

{

Teil B

}

Bsp

Int a=5;

Boolean b = a>6;

If ( a < 3)

{

// ….

}

Int c = 8;

b=c>3 && a<2;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a | b | A&&B | aIIb | !a |
| T | T | t | t | F |
| T | F | f | T | F |
| F | T | f | t | f |
| f | f | f | f | t |

^

Bsp

Int a= 3, b=5 , c=6;

If (! (a>1 && b<6 II c<=5)) // = false

Public Boolean FuehrerscheinErlaubt() Frauen = 14

{ Männer = 20

If (weiblich == true)

{

Return alter >= 14;

}

Else

{

Return alter >= 20;

}

}

Public class Mathematik

{

Int ziffer=5;

Public Mathematik (int ziffer)

{

setZiffer(ziffer);

}

Public void setZiffer(int ziffer)

{

If(ziffer >= 0 && ziffer < 10)

{

This.Ziffer = ziffer;

}

Else

{

System.out.println(„Ziffer darf nur von 0-9 gehen“);

}

}

}

Public class Eintrittskarte

{

Int preisKinder = 5;

Int preisErwachsene = 10;

Public void Zahlen (boolean kind, int geld)

{

If (kind)

{

Int rest = geld-preisKind;

}

Else

{

Int rest = geld-preisErwachsener;

}

If (rest <0)

{

System.out.println(“Es fehlen „+ (-rest) + “ €“);

}

If (rest == 0)

{

System.out.println(„Danke“);

}

Else

{

System.out.println(„Returgeld: „+rest);

}

}

}

Public void setSemester(int semester)

{

If (semester >=1 && semester <=10)

{

This.semester=semester;

}

Else

{

This.semester=1;

System.out.println(“NÖ”);

}

}

Public void setAlter (Boolean abendSchueler, int alter, int semester)

{

If(abendSchueler && semester )

{

If(semester >=1 && <= 4)

{

}

}

}

# Debugger

Der Debugger dient zur Fehler suche. Das Programm kann im einzelschritt durchlaufen werden. Start des Debuggers erfolgt durch das setzen eines Breakpoints (Haltepunkt) neben einem Befehl. Dann die Methode ausführen.

|  |  |
| --- | --- |
| setName(String name)  {  This.name=name;  } |  |
| // | Statische variable |
| This.name | Instanz Variable |
| name | Lokale Variable |

Schritt über: Methoden Aufruf wird ausgeführt

Schritt hinein: Methode wird geöffnet.

Public class Beispiel

{

Public int max2(int a, int b)

{

If(a<b)

{

Return b;

}

Else

{

Return a;

}

}

}

Public int max3(int a, int b, int c)

{

If(a>b)

{

If(a>c)

{

Return a;

}

Else

{

Return c;

}

}

Else

{

If(b>c)

{

Return b;

}

Else

{

Return c;

}

}

}

Public class LKW

{

}

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

# Primitive Datentypen

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Speicher (Byte) | Standadwert | Bereich |
| boolean | 1 (8Bit) | False |  |
| byte | 1 | 0 | -27 – 27-1  -128 - 127 |
| short | 2 | 0 | -215 – 215-1 |
| char | 2 | u /0000 | Alle Unicode Zeichen |
| int | 4 | 0 | -231 – 231-1 |
| float | 4 | 0f | +/- 3,48 \* 1038 |
| long | 8 | 0 | -263 – 263-1 |
| double | 8 | 0.0 | +/- 1,79\*10308 |
|  |  |  |  |

Achtung: Die Anzahl der Nachkommastellen ist natürlich begrenzt!  
 D.h. man kennt nicht jede long Zahl verlustfrei in einen double umwandeln.

# Fibonacci Zahlen

Folgendes Problem (Kaninchen Vermehrung)

* Kein Tier stirbt
* Jedes Monat Nachkommen
* Nach 2 Monaten bekommt der Nachwuchs Nachkommen

|  |  |  |  |
| --- | --- | --- | --- |
| Monat | Kaninchen | k | K |
| 1 | 2 | 1 | 1 |
| 2 | 3 | 1 | 2 |
| 3 | 5 | 2 | 3 |
| 4 | 8 | 3 | 5 |
| 5 | 13 | 5 | 8 |
| 6 | 21 | 8 | 13 |
| 7 | 34 | 13 | 21 |

public void Fibonacci (int n){

//x=große Kaninchen, y=kleine Kaninchen

int x=1,y=0;

System.out.println("Start bei Monat 0 ende bei Monat "+n);

for(int i=0; i<=n; i++){

x=x+y;

System.out.println("Monat "+i);

System.out.println("k: "+y);

System.out.println("K: "+x);

y=x;

}

}

## Primzahl

Boolean istPrim( int x){