Welche Datentypen gibt es in Java?

Es gibt folgende Datentypen in **Java**:

- **Primitive Datentypen**
- **Komplexe Datentypen**

Primitive Datentypen

Datentyp	Groeße	Kann darstellen
byte	8 Bit	Ganze Zahlen
short	16 Bit	Ganze Zahlen
int	32 Bit	Ganze Zahlen
long	64 Bit	Ganze Zahlen
float	32 Bit	Kommazahlen
double	64 Bit	Kommazahlen
char	16 Bit	Zeichen
boolean	1 Bit	Wahr & Falsch
void	-	Kein Wert

Komplexe Datentypen

Datentyp	Beschreibung
String	Zeichenkette
Array	Liste von Elementen
Klasse	Eigene Datentypen

Beispiele

```
1        int a;       // Deklaration
2        a = 5;       // Initialisierung
3
4        int b = 10; // Deklaration und Initialisierung
```

Klassen und Methoden

Was ist eine Klasse?

Eine Klasse ist ein Bauplan fuer Objekte. Sie definiert die Eigenschaften und Methoden, die ein Objekt haben soll.

Klassen und Methoden

Eine Methode ist ein Codeblock, der eine bestimmte Aufgabe erfuellt.
Die Methode kann immer wieder neu Aufgerufen werden.

Beispiel:

Hello.java

```java
// Klassenname = Hello
// Dateiname = Hello.java

// Zugriffsklasse = public (Zugriff von ueberall)
public class Hello {

    // Main-Methode
    // Zugriffsklasse   = public (Zugriff von ueberall)
    // Statisch         = Methode gehoert zur Klasse
    // Rueckgabetyp     = void (kein Rueckgabewert)
    // Name             = main
    // Parameter        = String[] args
    public static void main(String[] args) {

        // Ausgabe von "Hello World!"
        System.out.println("Hello World!");
    }
}
```

1 Kontrollstrukturen

Was sind Kontrollstrukturen?

Kontrollstrukturen sind Anweisungen, die den Programmfluss steuern.
Sie entscheiden, welche Anweisungen ausgefuehrt werden und welche nicht.

If/Else-Statement

Das If-Else-Statement ist eine Kontrollstruktur, die eine Bedingung ueberprueft.

Sie benutzen Boolsche Ausdruecke, um zu entscheiden, ob ein Codeblock ausgefuehrt wird oder nicht.

Zur besseren Lesbarkeit kann man auch eine Variable des Datentyps **boolean** benutzen und in If/Else benutzen.

Beispiel:If-/Else-Statement

```
1 // If-Else-Statement
2 if(12 > 10) {
3     System.out.println("Wahr");
4 } else {
5     System.out.println("Falsch");
6 }
```

Beispiel:If-/Else-Boolean

```
1 // If-Else-Statement mit boolean
2 boolean istWahr = 12 > 10;
3 if(istWahr) {
4     System.out.println("Wahr");
5 } else {
6     System.out.println("Falsch");
7 }
```

Ternary Operator

Der Ternary Operator ist eine Kurzschreibweise fuer If-Else-Statements.
Er besteht aus einer Bedingung, einem Fragezeichen und zwei Moeglichkeiten.

Syntax: `Bedingung ? Ausdruck1 : Ausdruck2`
Beispiel:

Beispiel:Ternary-Operator

```
int a = 10;
int b = 12;

// Groessere Zahl wird in groesser gespeichert
int groesser = a > b ? a : b;

system.out.println(groesser); // Ausgabe: 12
```

Verglichen mit dem If Else saehe das so aus:

Beispiel:Ternary-Operator

```
int a = 10;
int b = 12;
int groesser;

if(a > b) {
    groesser = a;
} else {
    groesser = b;
}

system.out.println(groesser); // Ausgabe: 12
```

Switch-Case

Der Switch-Case ist eine Kontrollstruktur, die eine Variable auf verschiedene Werte ueberprueft. Es wird ein Ausdruck ausgewertet und mit den verschiedenen Cases verglichen.

Syntax:

Syntax:Switch-Case

```
switch(Ausdruck) {
    case Wert1:
        // Code
        break;
    case Wert2:
        // Code
        break;
    default:
        // Code
}
```

Außerdem gibt es noch die LLambda Likeßchreibweise, die ab Java 12 verfuegbar ist.

Syntax:

Syntax:Switch-Case

```
switch(Ausdruck) {
    case Wert1 -> // Code;
    case Wert2 -> // Code;
    default ->    // Code;
}
```

Schleifen

Schleifen sind Kontrollstrukturen, die eine Anweisung wiederholt ausfuehren.
Es gibt drei Arten von Schleifen in Java:

1. **For-Schleife**

   & **'for-feach' Schleife**

2. **While-Schleife**

3. **Do-While-Schleife**

Beispiele für Schleifen

Beispiel:For-Schleife

```
// For-Schleife
for(int i = 0; i < 5; i++) {
    System.out.println(i);
}
```

Beispiel: für Datenmengen (For-Each-Schleife)

Beispiel:For-Each-Schleife

```
// For-Each-Schleife
int[] zahlen = {1, 2, 3, 4, 5};
for(int zahl : zahlen) {
    System.out.println(zahl);
}
```

While-Schleife

Beispiel:While-Schleife

```
1 // While-Schleife
2 int i = 0;
3 while(i < 5) {
4     System.out.println(i);
5     i++;
6 }
```

Do-While-Schleife

Beispiel:Do-While-Schleife

```
1 // Do-While-Schleife
2 int i = 0;
3 do {
4     System.out.println(i);
5     i++;
6 } while(i < 5);
```

2 Zurueck zu Methoden

Methoden werden genutzt, um Code zu strukturieren und wiederzuverwenden.
Sie koennen Parameter entgegennehmen und einen Rueckgabewert haben.

Beispiele für Methoden, anhand von einem Additionsverfahren

Beispiel: Addition

```
// Methode add
public static int add(int a, int b) {
    i = a;
    s = b;
    while(i > 0) {
        s++; // s = s + 1
        i--; // i = i - 1
    }
    return s;
}
```

Diese Methode, könnte nun innerhalb der Main-Methode oder anderer Methoden aufgerufen werden. und Wir muessen nicht jedes Mal den Code neu schreiben.

Beispiel: Aufruf der Methode

```
public class Main {

    public static void main(String[] args) {

        int summe = add(5, 3);
        System.out.println(summe); // Ausgabe: 8
    }
}
```

Aufgabenstellungen

Aufgabe 1

Ihre Aufgabe ist es, eine Klasse zu erstellen die verschiedene Methoden beinhaltet. folgende Vorgaben müssen Sie erfüllen:

a) Klassenname: **Kalkulator**

b) Methoden:

(a) Addition

Addiert Zwei Zahlen und gibt das Ergebnis zurueck.

(b) Subtraktion

Subtrahiert Zwei Zahlen und gibt das Ergebnis zurueck.

(c) Multiplikation

Multipliziert Zwei Zahlen und gibt das Ergebnis zurueck.

(d) Division

Dividiert Zwei Zahlen und gibt das Ergebnis zurueck.

(e) getHappyBirthdayString (param: String name)

Gibt einen Glueckwunsch für ¡name¿ zurueck.

(f) printStuff (param: ¡beliebiger Datentyp¿ stuff)

Gibt verschiedene Werte aus.

c) Testen Sie, jede Methode mit folgenden Werten

(a) Addition(5, 3)

(b) Subtraktion(5, 3)

(c) Multiplikation(5, 3)

(d) Division(5, 3)

(e) getHappyBirthdayString("Max")

und geben Sie die Ergebnisse aus.

d) Testen Sie die Methode printStuff mit verschiedenen Werten, indem Sie, die Ergebnisse der Zahlenrückgaben als String übergeben. Benutzen Sie hierfür Google und die Java-Dokumentation.

Erweiterung der Aufgabe 1

Erweitern Sie die Klasse kalkulator folgendermaßen:
Importieren Sie die Klasse Scanner und lesen Sie zwei Zahlen von der Konsole ein. Testen Sie ihre Methoden und die Ausgabe mit diesen Werten.

Import einer Klasse

```
// Import
import java.util.Scanner;

// Scanner-Objekt erstellen
Scanner <Variablenname> = new Scanner(System.in);

// Einlesen von Werten
int a = <Variablenname>.nextInt();
String b = <Variablenname>.nextLine();
```

Noch eine Erweiterung

Erweitern Sie die Klasse Kalkulator um eine Methode, die den Benutzer nach einer Zahl fragt und diese zurückgibt sollte Sie, größer als eine 10 sein, und keine ',' oder Buchstaben enthalten.
Benutzen zur Recherche google, und schauen Sie nach Begriffen wie:

1. `Java Scanner`
2. `Java String Methoden (Matches)`
3. `Cast Operatoren`
4. `Regular Expression`