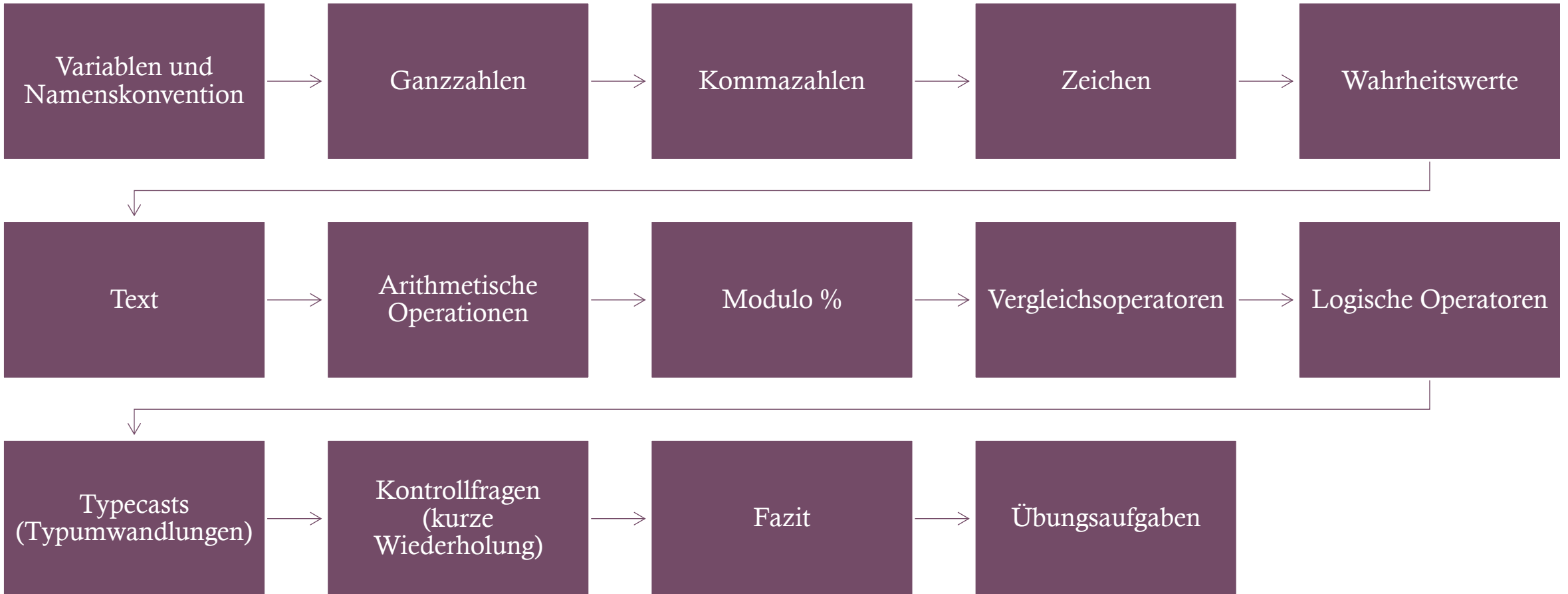

PROGRAMMIEREN LERNEN MIT JAVA

Said Cetin

KAPITEL 1 AGENDA



VARIABLEN & NAMENSKONVENTION

Was ist eine Variable?

- Container für einen Wert
- Hat immer:
 - **Datentyp** (z. B. int, String)
 - **Name** (z. B. seitenAnzahl)

```
int seitenAnzahl = 200;
```

Namenskonventionen in Java

- **Sprechende Namen:** seitenAnzahl statt x
- **Kleinbuchstabe am Anfang:** alter, nicht Alter
- **camelCase für zusammengesetzte Namen:**
 - meinVorname
 - maximaleGeschwindigkeit

```
int seitenAnzah;  
seitenAnzahl = 200;
```

GANZZAHLEN

Typ	Größe	Wertebereich
byte	8 Bit	-128 bis 127
short	16 Bit	-32.768 bis 32.767
int	32 Bit	-2.147.483.648 bis 2.147.483.647
long	64 Bit	sehr große Ganzzahlen

- Speichert ganze Zahlen ohne Nachkommastellen
- Wertebereich hängt vom Typ ab
- Deklaration und Initialisierung:
 - `Int zahl;`
 - `Zahl = 10;`

KOMMAZAHLEN

Double & Float

- Speichern Zahlen mit Nachkommastellen
- Unterschied in Genauigkeit und Speicherbedarf

TYP	Genauigkeit	Speicher	Beispiel
double	hoch	64 Bit	double pi = 3.14;
float	geringer	32 Bit	float temp = 23.5f;

ZEICHEN & ZEICHENKETTE

CHAR & STRING

Char:

- Speichert einzelne Zeichen (Unicode ASCII- Tabelle)
- Werden in einfachen Anführungszeichen geschrieben

```
char buchstabe = 'A';  
char sonderzeichen = '§';  
char umlaut = 'ü';
```

String:

- Ist keine primitive Datenart sondern eine Klasse
- Ist eine Zeichenkette
- Strings können miteinander verbunden werden
- Strings vergleicht man nicht mit == sondern mit .equals()

```
String gruss = "Hallo Welt!";  
String vorname = "Anna";  
String nachname = "Müller";  
String vollerName = vorname + " " + nachname; // "Anna Müller,"  
String a = "Hallo";  
String b = "Hallo";  
boolean gleich = a.equals(b);    // true
```

WAHRHEITSWERTE

BOOLEAN

- Kann nur die Werte `true` oder `false` annehmen
- Wird häufig in bedingungen oder Vergleichen gesetzt

```
boolean istVolljaehrig = true;
```

```
int alter = 20;
```

```
boolean istErwachsen = alter >= 18; // true
```

ARITHMETISCHE OPERATIONEN

Java Unterstützt:

Addition /
Subtraktion

Multiplikation

Division

Modulo (%)

Achtung bei Ganzzahldivision:

```
int ergebnis = 5 / 2; // ergibt 2, nicht 2.5
```

Wenn einer der Operanden eine Kommazahl ist, erfolgt eine Fließkommadivision:

```
double ergebnis = 5.0 / 2; // ergibt 2.5
```

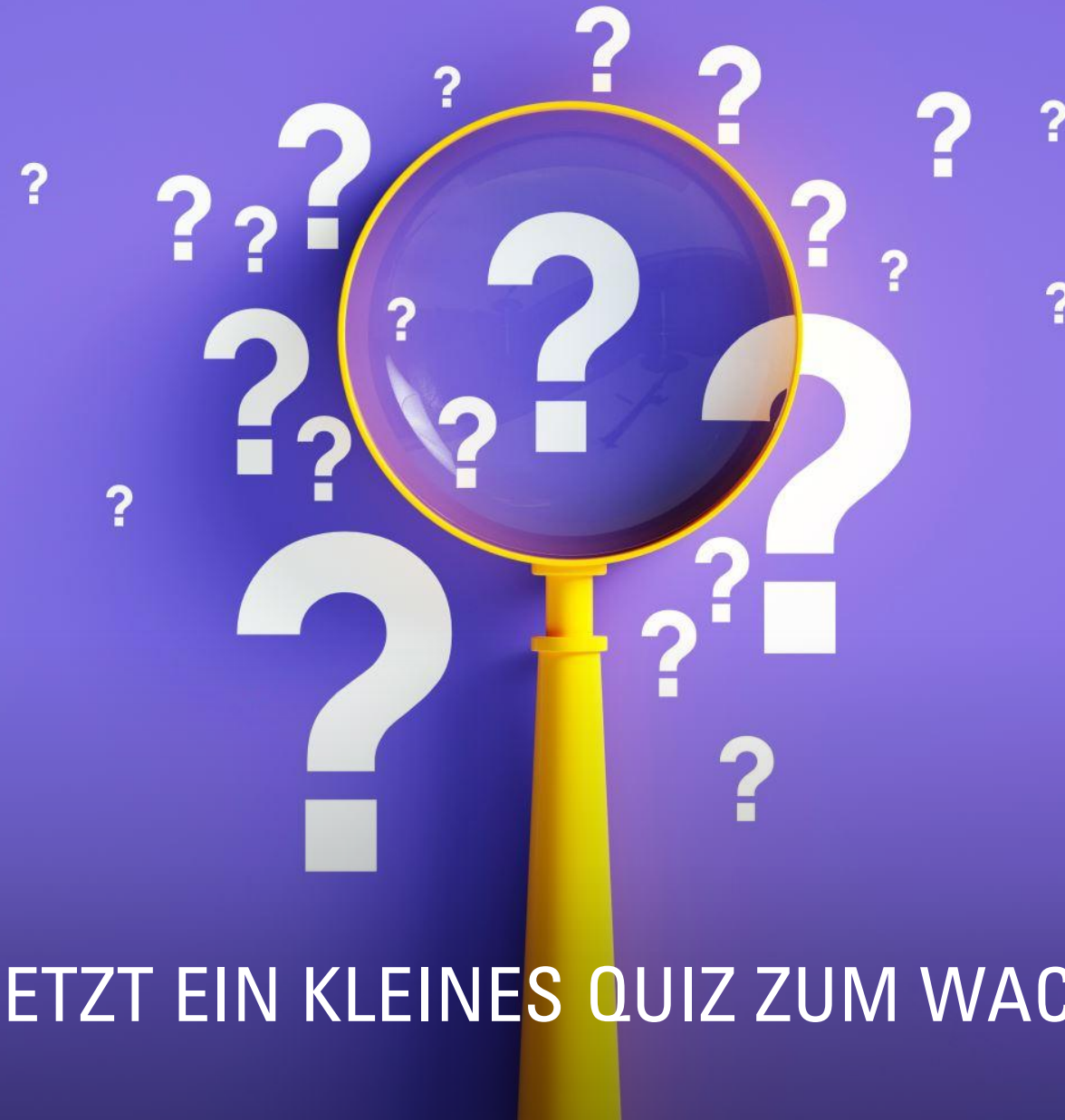
MODULO (%)

Der **Modulo-Operator** `` liefert den **Rest** einer Division:

```
int rest = 5 % 2; // ergibt 1
```

Anwendungsfall: Überprüfen, ob eine Zahl gerade ist:

```
if (zahl % 2 == 0) {  
    // gerade Zahl  
}
```



JETZT EIN KLEINES QUIZ ZUM WACH WERDEN

```
1 ► public class test {
2 ►     public static void main(String[] args) {
3         int ergebnis = 1 + 1;
4         System.out.println("ergebnis1: " + ergebnis);
5         double ergebnis2 = 2.0 - 1;
6         System.out.println("ergebnis2: "+ergebnis2);
7         int ergebnis3 = 2 * 2;
8         System.out.println("ergebnis3: "+ergebnis3);
9         int ergebnis4 = 5 / 2;
10        System.out.println("ergebnis4: "+ergebnis4);
11        double ergebnis5 = 5 / 2;
12        System.out.println("ergebnis5: "+ergebnis5);
13        double ergebnis6 = 5.0 / 2.0;
14        System.out.println("ergebnis6: "+ergebnis6);
15        int ergebnis7 = 5 % 2;
16        System.out.println("ergebnis7: "+ergebnis7);
17        double ergebnis8 = 5.0 % 2.0;
18        System.out.println("ergebnis8: "+ergebnis8);
19    }
20 }
```

VERGLEICHSOPERATOREN

Operator	Bedeutung
<code>==</code>	gleich
<code>!=</code>	ungleich
<code><</code>	kleiner als
<code><=</code>	kleiner oder gleich
<code>></code>	größer als
<code>>=</code>	größer oder gleich

```
int alter = 20;  
boolean istErwachsen = alter >= 18; // true
```

LOGISCHE OPERATOREN

Diese kombinieren mehrere `boolean`-Ausdrücke:

- `&&` (UND)
- `||` (ODER)
- `!` (NICHT)

Beispiele:

```
true && true    // true
true && false   // false
true || false   // true
!true           // false
```

TYPECASTS (TYPUMWANDLUNGEN)

Implizite Umwandlung (automatisch):

```
int i = 5;  
double d = i; // d ist 5.0
```

Explizite Umwandlung (manuell):

```
double d = 3.9;  
int i = (int) d; // i ist 3 (Nachkommastelle abgeschnitten)
```

KONTROLLFRAGEN (WIEDERHOLUNG)

1

Was passiert bei
5/2 und warum ?

2

Wie unterscheidet
sich float von
double?

3

Wozu dient der
Modulo-Operator
% ?

4

Warum vergleicht
man Strings mit
.equals() und
nicht mit == ?

5

Was ergibt
(int) 4.9 ?

LERNZIELE

Was eine Variable ist

- du weißt kennst die Namenkonvention

Primitive Datentypen in Java

- du kennst jetzt die häufigsten primitiven Datentypen, die benutzt werden

Nicht-primitiver Datentyp: String

- du weißt das String eine Klasse ist und kein primitiver Datentyp
- du kannst Texte verketteten und speichern
- du kennst den unterschied zwischen == und .equals()

Operatoren anwenden

Typumwandlung

- implizit (automatisch)
- explizit (manuell)