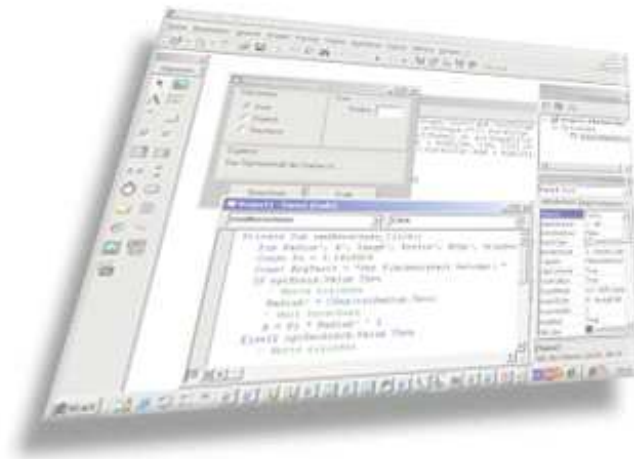


Wiederholungen ohne feste Anzahl





while-Beispiel I

1.

Schleifendurchlauf

//Beispiel: Ausgabe der Zahlen 2 bis 20:

```
int num=1;
while (num <= 10) {
    System.out.println(2*num);
    num++;
}
```

Num ist 1; daher $\text{Num} \leq 10$ ist true, daher Ausführung des while-Konstruktes

Ausgabe von $2=1*2$; da $\text{num}=1$

Der Inhalt von num wird um eins erhöht, also auf 2 gesetzt!

Springe zur Zeile „while num<=10“ und überprüfe die Bedingung.

Syntax:

```
while (<Bedingung>){
    <Anweisungen>
}
```



while-Beispiel II

2.

Schleifendurchlauf

//Beispiel: Ausgabe der Zahlen 2 bis 20:

```
int num=1;
while (num <= 10) {
    System.out.println(2*num);
    num++;
}
```

Num ist 2; daher $\text{Num} \leq 10$ ist true, daher wird in die Schleife „reingesprungen“

Ausgabe von $4=2*2$; da $\text{num}=2$

Der Inhalt von num wird um eins erhöht, also auf 3 gesetzt!

Springe zur Zeile „while $\text{num} \leq 10$ “ und überprüfe die Bedingung.

Syntax:

```
while (<Bedingung>){
    <Anweisungen>
}
```



while-Beispiel III

10.

Schleifendurchlauf

//Beispiel: Ausgabe der Zahlen 2 bis 20:

```
int num=1;
while (num <= 10) {
    System.out.println(2*num);
    num++;
}
```

Num ist 10; daher $\text{Num} \leq 10$ ist true, daher wird in die Schleife „reingesprungen“

Ausgabe von $20 = 2 * 10$; da $\text{num} = 10$

Der Inhalt von num wird um eins erhöht, also auf 11 gesetzt!

Springe zur Zeile „while $\text{num} \leq 10$ “ und überprüfe die Bedingung.
 $(\text{num} \leq 10) = \text{false}$, führe die nächste Anweisung hinter der Schleife aus

Syntax:

```
while (<Bedingung>){
    <Anweisungen>
}
```



while-Beispiel IV

1. Programm eingeben

2. Programm mit F7 schrittweise ausführen

```
1 public class Aufgabe {  
2  
3     public static void main(String[] args)  
4         int num=1;  
5         while (num <= 10) {  
6             System.out.println(2*num);  
7             num++;  
8  
9         }  
10    }  
11 }
```



while-Beispiel V

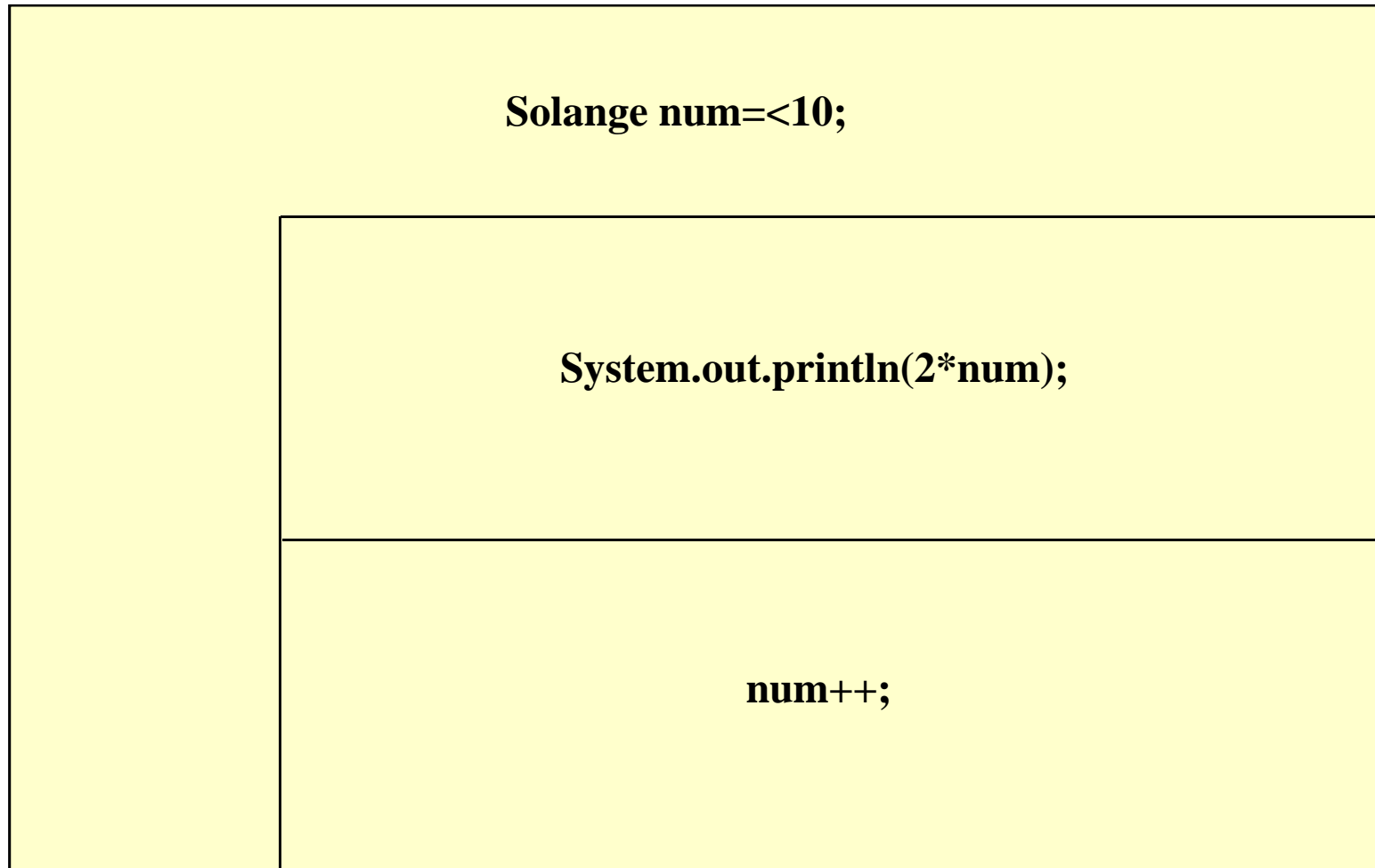
//Beispiel: Ausgabe der Zahlen 2 bis 20:

```
int num=1;
while (num <= 10) {
    System.out.println(2*num);
    num++;
}
```

Schleifen- durchlauf	Num am Anfang der Schleife	num2 <=10	Ausgabe	Num am Ende der Schleife
1	1	true	2	2
2	2	true	4	3
3	3	true	6	4
4	4	true	8	5
5	5	true	10	6
6	6	true	12	7
7	7	true	14	8
8	8	true	16	9
9	9	true	18	10
10	10	true	20	11
11	11	false	-	-



while: Struktogramm





while: Unendliche Schleifen

//Vorsicht vor unendlichen Schleifen!

```
int num=7;
while (num != 0) {
    System.out.println(num);
    num=num-2;
}
```

Die Schleife wird niemals verlassen, da num! wie folgt heruntergezählt wird:
7, 5, 3, 1, -1, -3, -5 usw.

Besser: num>0 statt num=0

Schleifen- durchlauf	Num am Anfang der Schleife	Ausgabe	Num am Ende der Schleife
1	7	7	5
2	5	5	3
3	3	3	1
4	1	1	-1
5	-1	-1	-3
6	-3	-3	-5
7	-5	-5	-7
8	-7	-7	-9
9	-9	-9	-11
10	-11	-11	-13



while: Zahlenfolgen summieren I

Summiere die Zahlen von 1 bis 10!

Benötigt werden:

num: Variable zum Abspeichern des Zählers

sum: Variable zum Abspeichern der Summe

```
int num,sum;
```

```
num=1;
```

```
sum=0;
```

```
while (num <= 10) {
```

```
sum=sum+num;
```

```
num=num+1;
```

```
}
```

```
System.out.println("Summe:"+sum);
```

Initialisierung der Variablen

Addiere zu der Variable sum den Inhalt der Variablen num

Addiere zu der Variable num die Zahl 1



while: Zahlenfolgen summieren II

Werte der Schleifendurchläufe:

```

int num,sum;
num=1;
sum=0;
while (num <= 10) {
    sum=sum+num;
    num=num+1;
}
System.out.println("Summe:"+sum);

```

Schleifen- durchlauf	Num am Ende der Schleife	Neuer Wert von sum	Num am Anfang der Schleife
1	2	1	1
2	3	3	2
3	4	6	3
4	5	10	4
5	6	15	5
6	7	21	6
7	8	28	7
8	9	36	8
9	10	45	9
10	11	55	10



while: Zahlenfolgen summieren III

„Optimierte Version“:

```
int num=1,sum=0;
while (num <= 10) {
    sum+=num;
    num++;
}
System.out.println("Summe:"+sum);
```



do-while- Beispiel I

//Beispiel: Ausgabe der Zahlen 2,4,6,...,18, 20:

```
int num;
num=1;
do {
    System.out.println("Summe:"+2*num);
    num++;
} while (num <=10);
```

1.

Schleifendurchlauf

Ausgabe von $2=2*1$; da $num=1$

Der Inhalt von num wird um eins erhöht, also auf 2 gesetzt!

Überprüfe die Bedingung; da $num=2 > 10$ und damit falsch(false) ist, gehe zur Zeile „Do“

Syntax:

do

<Anweisungen>

while (<Bedingung>)



do-while - Beispiel II

//Beispiel: Ausgabe der Zahlen 2,4,6,...,18,20:

```
int num;
num=1;
do {
    System.out.println("Summe:"+2*num);
    num++;
} while (num <=10);
```

2.

Schleifendurchlauf

Ausgabe von $4=2*2$; da $num=2$

Der Inhalt von num wird um eins erhöht, also auf 3 gesetzt!

Überprüfe die Bedingung; da $num=3 > 10$ und damit falsch ist, gehe zur Zeile „Do“

Syntax:

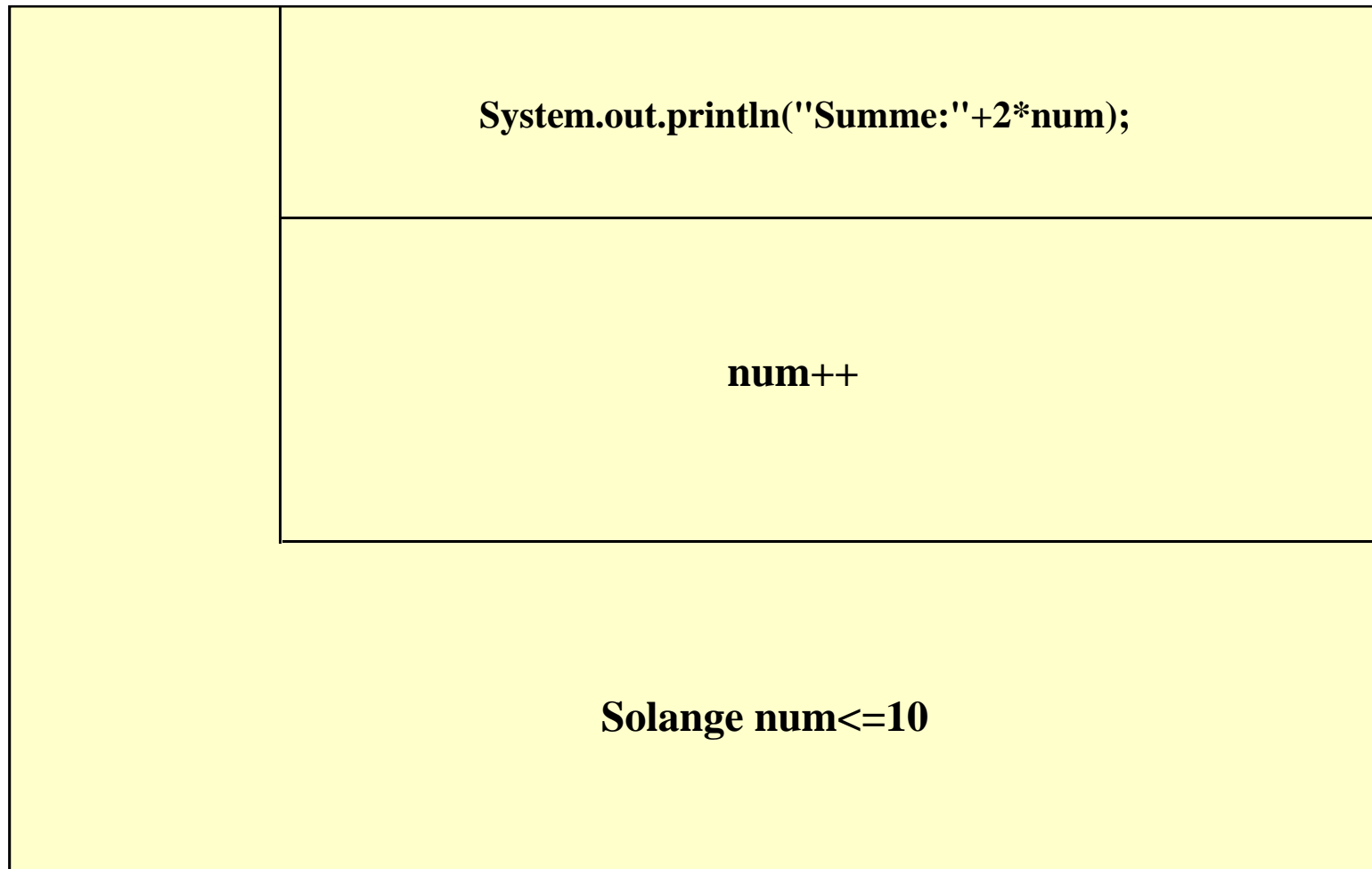
Do

<Anweisungen>

while (<Bedingung>)



do-while: Flußdiagramm





Vergleich der beiden Schleifen

```
while (<Bedingung>) {  
    i++  
    .....  
}
```

„**Kopfgesteuert**“: Die Schleifen-Bedingung steht am Anfang der Schleife

```
do {  
    .....  
    i++  
} while (<Bedingung>)
```

„**Fußgesteuert**“: Die Schleifen-Bedingung steht am Ende der Schleife



Wiederholung - Vergleich

//Beispiel: Ausgabe der Zahlen 2 bis 20:

```
int i;  
for (i=1; i<=10; i++) {  
    System.out.println("Summe:"+2*i);  
}
```

```
int i;  
i=1;  
do {  
    System.out.println("Summe:"+2*i);  
    i++;  
} while (i<=10);
```

```
int i;  
i=1;  
while (i<=10) {  
    System.out.println("Summe:"+2*i);  
    i++;  
}
```