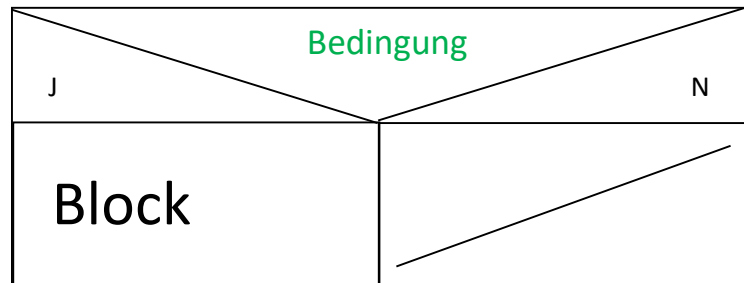


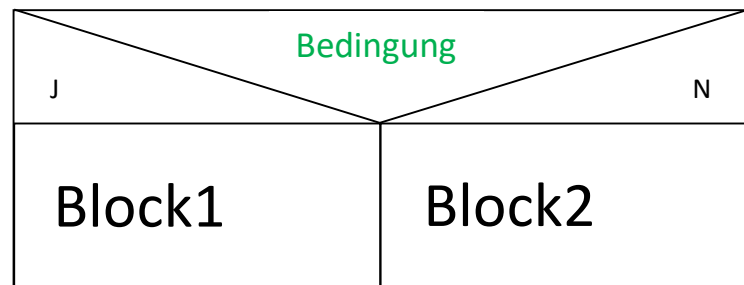
7. Verzweigung / Auswahl / Selection statements

7.1. Einfache Verzweigung (if)

```
if (Bedingung) {  
    Block  
}
```



```
if (Bedingung) {  
    Block1  
} else {  
    Block2  
}
```



7.1.1. Vergleichs Operatoren (Bedingungen)

==	gleich	=	Float.compare(f1,f2)	Double.compare(d1,d2)
!=	ungleich	≠		
<	kleiner	<		
<=	kleiner gleich	≤		
>	größer	>		
>=	größer gleich	≥		

7.1.2. Logische Operatoren (Verknüpfungen von Bedingungen)

Bedingung 1	Bedingung 2	oder V 	und ∧ &&
falsch	falsch	falsch	falsch
falsch	wahr	wahr	falsch
wahr	falsch	wahr	falsch
wahr	wahr	wahr	wahr

Beispiel: am Beispiel Rechner

```
public static void main(String[] args) {
    var scan = new java.util.Scanner(System.in);
    int zahl1, zahl2, erg=0;
    char rz;

    System.out.print("Bitte 1.Zahl eingeben ");
    zahl1 = scan.nextInt();
    System.out.print("Bitte 2.Zahl eingeben ");
    zahl2 = scan.nextInt();
    System.out.print("Bitte Rechenzeichen eingeben ");
    rz = scan.next().charAt(0);

    if((rz == '+') || (rz == 'a')) {
        erg = zahl1 + zahl2;
    } else if ((rz == '-') || (rz == 's')) {
        erg = zahl1 - zahl2;
    } else if ... {
        ...
    } else {
        System.out.println("Falsches Rechenzeichen!!!");
    }

    System.out.println("Ergebnis: "+zahl1+rz+zahl2+" = "+erg);
}
```

Struktogramm zeichnen

7.2. Mehrfach Verzweigung (switch)

Einschränkung gegenüber der „einfachen Verzweigung (if)“:

- **NUR** mit Variablen vom Datentyp `byte`, `short`, `int`, `char` oder `Strings` (**NICHT** `long`)
- **IMMER** nur dieselbe **VARIABLE** auf **GLEICH**
- **NUR „ODER“** als logische Operation

Beispiel: am Beispiel Rechner

```
switch(rz) {
    case '+':
    case 'a':
        erg = zahl1 + zahl2;
        break;
    case '-':
    case 's':
        erg = zahl1 - zahl2;
        break;
    ...
        break;
    default:
        System.out.println("Falsches Rechenzeichen!!!");
}
```

Beispiel: am Beispiel Rechner (ab JAVA 14)

```
switch(rz) {
    case '+', 'a' -> {
        erg = zahl1 + zahl2;
    }
    case '-', 's' -> {
        erg = zahl1 - zahl2;
    }
    ...
    default ->
        System.out.println("Falsches Rechenzeichen!!!");
}
```

rz				
'+', 'a'	'-', 's'	'*', 'm'	'/', 'd'	
erg ← zahl1 + zahl2	erg ← zahl1 + zahl2	erg ← zahl1 + zahl2	erg ← zahl1 + zahl2	O: „Falsches Rechenzeichen“

Ganzes Struktogramm zeichnen

8. Schleifen / Iteration statements (loops)

8.1. Kopfgesteuerte Schleife (while)

```
while (Bedingung) {
```

Block

```
}
```

solange **Bedingung**

Block

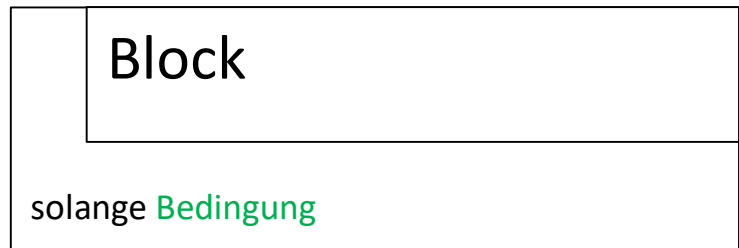
min 0x max ∞

8.2. Fußgesteuerte Schleife (do- while)

```
do {
```

Block

```
} while (Bedingung);
```



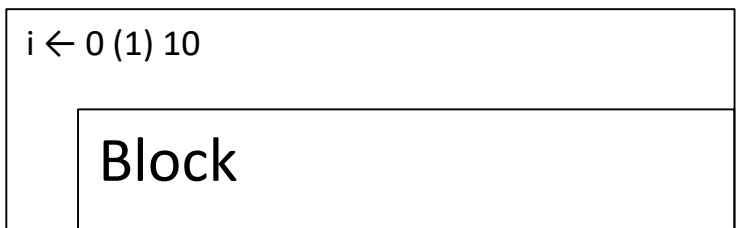
min 1x max ∞

8.3. Zählschleife (for)

```
for(i=0; i<=10; i++) {
```

Block

```
}
```



min 0x max ∞

8.3.1. Java Kurzschreibweisen

```
i = i + 1;  
i = i + 3;  
i = i - 1;  
i = i - 4;  
i = i * 5;
```

```
i += 1;  
i += 3;  
i -= 1;  
i -= 4;  
i *= 5;
```

```
i++;
```

```
i--;
```

mit allen Grundrechnungsarten