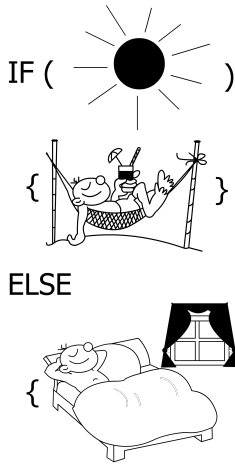




Thema: Informationsblatt Kontrollstrukturen



```
public String aufladen(double pBetrag){
    String mMeldung="";
    if (pBetrag + this.getGuthaben() <= 200)
    {
        this.setGuthaben(this.guthaben + pBetrag);
        mMeldung = "Ihr Guthaben beträgt " + String.valueOf(this.getGuthaben());
    }
    else
    {
        mMeldung = "Ladebetrag zu groß, Vorgang abgebrochen!";
    }
    return mMeldung;
}
```

Fall1: richtige Eingabe

Fall2: falsche Eingabe

Für den Test die Main-Methode der Test-Klasse erweitern:

```
System.out.println("#####");
System.out.println("#####");
System.out.println("#####Test der Methode aufladen() -- IF-Zweig###");
System.out.println("Aufladen 1.Fall: " +geldkartel.aufladen(20.00));

System.out.println("#####");
System.out.println("#####");
System.out.println("#####Test der Methode aufladen() -- ELSE-Zweig###");
System.out.println("Aufladen 2.Fall: " +geldkartel.aufladen(250.00));
```

Ergebnis:

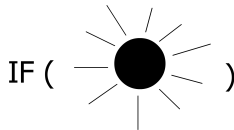


```
C:\E:\WINDOWS\system32\cmd.exe
#####
##### Objekt: geldkarte2#####
Inhaber: Christian Wulff
Guthaben: 75.0
#####
#####Test der Methode aufladen() -- IF-Zweig###
Aufladen 1.Fall: Ihr Guthaben beträgt 95.0
#####
#####Test der Methode aufladen() -- ELSE-Zweig###
Aufladen 2.Fall: Ladebetrag zu groß, Vorgang abgebrochen!

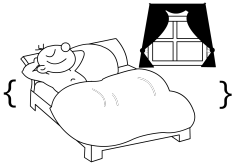
G:\Referendariat\ZweiteExamensarbeit\E-Learning\SRC\Geldkarte
Drücken Sie eine beliebige Taste . . .
```



Thema: Informationsblatt Kontrollstrukturen



ELSE



```
public String beZahlen (double pBetrag){
    String mMeldung="";
    if (pBetrag <= this.getGuthaben())
    {
        this.setGuthaben(this.getGuthaben() - pBetrag);
        mMeldung = "Ihr Guthaben beträgt " + String.valueOf(this.getGuthaben());
    }
    else
    {
        mMeldung ="Guthaben reicht nicht aus, Vorgang abgebrochen!";
    }
    return mMeldung;
}
```

Für den Test die Main-Methode der Test-Klasse erweitern:

```
System.out.println("#####");
System.out.println("#####");
System.out.println("#####Test der Methode beZahlen() -- IF-Zweig###");
System.out.println("Bezahlen 1.Fall: " +geldkartel.beZahlen(15.00));

System.out.println("#####");
System.out.println("#####");
System.out.println("#####Test der Methode beZahlen() -- ELSE-Zweig#");
System.out.println("Bezahlen 2.Fall: " +geldkartel.beZahlen(75.00));
```

Ergebnis:



```
C:\E:\WINDOWS\system32\cmd.exe
#####
#####
##### Objekt: geldkarte2#####
Inhaber: Christian Wulff
Guthaben: 75.0
#####
#####Test der Methode aufladen() -- IF-Zweig###
Aufladen 1.Fall: Ihr Guthaben beträgt 95.0
#####
#####Test der Methode aufladen() -- ELSE-Zweig###
Aufladen 2.Fall: Ladebetrag zu groß, Vorgang abgebrochen!
#####
#####Test der Methode beZahlen() -- IF-Zweig###
Bezahlen 1.Fall: Ihr Guthaben beträgt 80.0
#####
#####Test der Methode beZahlen() -- ELSE-Zweig#
Bezahlen 2.Fall: Ihr Guthaben beträgt 5.0

G:\Referendariat\ZweiteExamensarbeit\E-Learning\SRC\Geldkarte\C
Drücken Sie eine beliebige Taste . . .
```

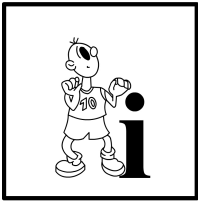


ÜBERBLICK KONTROLLSTRUKTUREN



Anweisungen

Name	Java Quellcode	Struktogramm																					
if ... else	<pre>if (Bedingung) { Anweisungsblock 1; } else { Anweisungsblock 2; }</pre>																						
else if	<pre>if (Bedingung 1) { Anweisungsblock 1; } else if (Bedingung 2) { Anweisungsblock 2; } else if (Bedingung 3) { Anweisungsblock 3; } else if (Bedingung n) { Anweisungsblock n; } else { Anweisungsblock; }</pre>																						
switch ... case	<pre>int note = 6; String pNote; switch (note) { case 1: pNote = "sehr gut"; break; case 2: pNote = "gut"; break; case 3: pNote = "befriedigend"; break; ... default: pNote = "keine gültigen Note"; break; }</pre>	<p>Programmname: NotenAusgabeSwitchCase.java Beschreibung: Das Programm prüft die eingegebene Note mit der Switch-Case-Anweisung und gibt sie in Worten (verbal) aus. Ungültige Eingaben erzeugen die Ausgabe: "Keine gültige Note"</p> <p>Deklaration: Integer: note</p> <p>Ausgabe: Bitte Note eingeben:</p> <p>Eingabe: note</p> <table border="1"> <thead> <tr> <th colspan="6">prüfe note</th> <th>sonst</th> </tr> <tr> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th>5</th> <th>6</th> <th></th> </tr> </thead> <tbody> <tr> <td>Ausgabe: "sehr gut"</td> <td>Ausgabe: "gut"</td> <td>Ausgabe: "befriedigend"</td> <td>Ausgabe: "ausreichend"</td> <td>Ausgabe: "mangelhaft"</td> <td>Ausgabe: "ungenügend"</td> <td>Ausgabe: "Keine gültige Note"</td> </tr> </tbody> </table>	prüfe note						sonst	1	2	3	4	5	6		Ausgabe: "sehr gut"	Ausgabe: "gut"	Ausgabe: "befriedigend"	Ausgabe: "ausreichend"	Ausgabe: "mangelhaft"	Ausgabe: "ungenügend"	Ausgabe: "Keine gültige Note"
prüfe note						sonst																	
1	2	3	4	5	6																		
Ausgabe: "sehr gut"	Ausgabe: "gut"	Ausgabe: "befriedigend"	Ausgabe: "ausreichend"	Ausgabe: "mangelhaft"	Ausgabe: "ungenügend"	Ausgabe: "Keine gültige Note"																	



Schleifen (Wiederholstruktur)

Name	Java Quellcode	Struktogramm
for	<pre>for(int i=0; i<10; i++){ System.out.print(i); }</pre>	
while	<pre>while(int x <= 10) { System.out.println(x); x++; }</pre>	

BEDINGUNGEN UND OPERATOREN

Rechenoperator, um Werte zu berechnen. In der Regel rechnet man mit Werten vom Typ integer, long, double oder float.

A1 + A2	Addition zweier Werte: Wert in <i>Attribut1</i> ... plus... Wert in <i>Attribut2</i> ?
A1 - A2	Subtraktion zweier Werte: Wert in <i>Attribut1</i> ... minus... Wert in <i>Attribut2</i> ?
A1 * A2	Multiplikation zweier Werte: Wert in <i>Attribut1</i> ... mal... Wert in <i>Attribut2</i> ?
A1 / A2	Division zweier Werte: Wert in <i>Attribut1</i> ... durch... Wert in <i>Attribut2</i> ?



Der String-Vergleich mit der Methode „equals ()“

A1.equals (A2)	Vergleich zweier Werte mit komplexem Datentyp (u.a. String): z.B. A1 = „grün“, A2 = „rot“ Wert in <i>Attribut1</i> ... ist gleich... Wert in <i>Attribut2</i> ? Rückgabewert ist „false“
-----------------------	---

Operatoren im Fall eines numerischen Wertevergleichs

Bedingungen enthalten in der Regel Vergleichsoperatoren. Bedingungen liefern den Rückgabewert „true“ (für wahr) oder „false“ (für falsch).

A1 == A2	Vergleich zweier numerischer Werte: z.B. A1 = 5, A2 = 10 Wert in <i>Attribut1</i> ... ist gleich... Wert in <i>Attribut2</i> ? Rückgabewert ist „false“
B1 && B2	UND-Verkettung mehrerer Bedingungen: <i>Bedingung1</i> ... und ... <i>Bedingung2</i> ... muss korrekt sein. Nur dann ist der Rückgabewert „true“
A1 < A2	Vergleich zweier Werte: z.B. A1 = 5, A2 = 10 Wert in <i>Attribut1</i> ... ist größer als... Wert in <i>Attribut2</i> ? Rückgabewert ist „true“
A1 > A2	Vergleich zweier Werte: z.B. A1 = 3.50, A2 = 5.20 Wert in <i>Attribut1</i> ... ist kleiner als... Wert in <i>Attribut2</i> ? Rückgabewert ist „false“
A1 >= A2	Vergleich zweier Werte: z.B. A1 = 10, A2 = 7 Wert in <i>Attribut1</i> ... ist größer oder gleich... Wert in <i>Attribut2</i> ? Rückgabewert ist „true“
A1 <= A2	Vergleich zweier Werte: z.B. A1 = 20, A2 = 25 Wert in <i>Attribut1</i> ... ist kleiner oder gleich... Wert in <i>Attribut2</i> ? Rückgabewert ist „false“