

6.4 Wiederholung

In Java haben wir die Möglichkeit Sequenzen wiederholen zu lassen, ohne diese durch mehrfache Nennung im Programm-Code zu erzwingen. Dabei gibt es verschiedene Weisen, wie man dies tun kann.

6.4.1 Mit Anfangsbedingung

Möchten wir erreichen, dass eine Anweisung oder eine Sequenz solange wiederholt, bis eine bestimmte Bedingung erfüllt wird, nutzt man die *kopfgesteuerte* Wiederholungsanweisung.

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

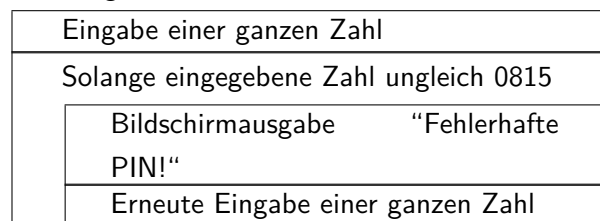
Wir überprüfen also **bevor** die Sequenz ausgeführt wird, ob die Abbruchbedingung erfüllt ist. Dabei kann es dann passieren, dass die Sequenz garnicht ausgeführt wird.

Beispiel kopfgesteuerte Wiederholung:

```
int pin = input();

while(pin != 0815){
    System.out.println("Fehlerhafte PIN!");
    pin = input();
}
```

Struktogramm



6.4.2 Mit Endbedingung

Die in 6.4.1 erwähnte Wiederholungsanweisung prüft die Abbruchbedingung, wie es der Name sagt, **bevor** die Sequenz ausgeführt wird.

Es kann aber natürlich auch mal passieren, dass eine solche Sequenz mindestens einmal ausgeführt werden soll, bevor die Abbruchbedingung überprüft wird. Um diese Anforderung zu erfüllen nutzt man die entsprechende *fußgesteuerte* Wiederholungsanweisung.

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

Bei dieser wird die Sequenz einmal ausgeführt und die Bedingung wird **nach** jeder Wiederholung überprüft.

Beispiel fußgesteuerte Wiederholung:

```
do{  
    System.out.println(number);  
    number--;  
} while(number > 0);
```

Struktogramm

