Informatik

Algorithmische Strukturen

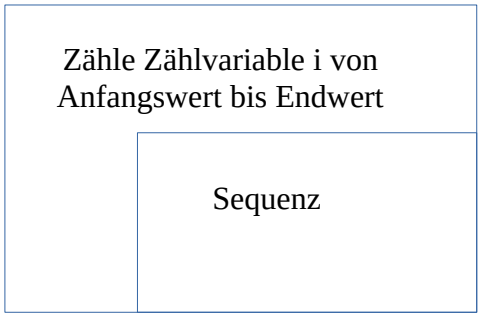
Bedingte Wiederholung mit Endbedingung

Bedingte Wiederholung mit Anfangsbedingung

21.01.2016

Bedingte Wiederholung mit Zähler

Struktogramm:



Zu Beginn hat die Zählvariable einen bestimmten Anfangswert. Nach jedem Durchlauf durch die Sequenz wird der Wert der Variablen um eins erhöht bzw. verringert. Beim letzten Durchlauf hat die Zählvariable den Endwert.

Java-Syntax:

```
for (int i = Anfangswert; i <= Endwert; i++) // i++ - Abkürzung für i = i + 1
        {Sequenz;}
for (int i = Anfangswert; i >= Endwert; i--) // i-- - Abkürzung für i = i + 1
        {Sequenz;}
```

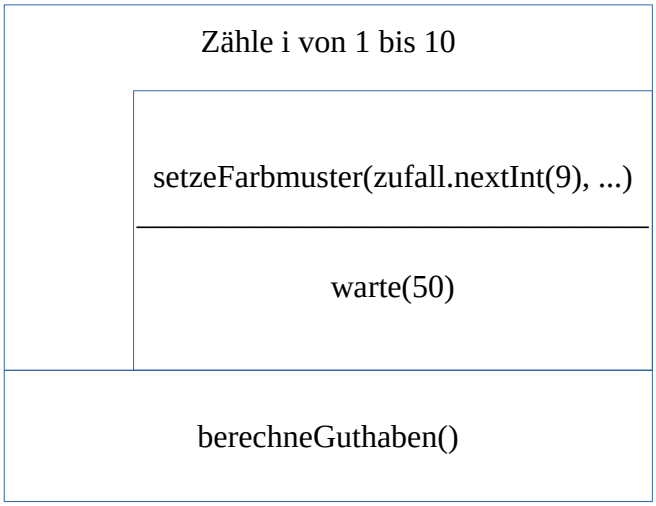
i++ i--

Der Ausdruck „i++“ sorgt dafür, dass, nach dem Durchlaufen der Sequenz, der Wert der Zählvariablen um eins erhöht wird. Bei „i--“ wird er um eins verringert.

Motivation:

Der Spielautomat soll zunächst mehrmals die Farben der Walzen wechseln und erst dann seinen Endzustand erreichen.

Konkretes Struktogramm:



Java-Syntax:

```
public void laufe()
{       for(int i = 1; i <= 10; i++)
            {       setzeFarbmuster(zufall.nextInt(9), …);
                    ZEICHENFENSTER.gibFenster().warte(50);
            }
        berechneGuthaben();
}
```

18.02.2016

Aufgaben

- Schreibe eine Methode, die die Summe aller natürlichen Zahlen bis n angibt.

```
Public int berechneSumme(int n)
{       int summe=0;
        for(int i=1; i<=n; i++)
        {       summe=summe+i;
        }
        return summe;
}
```

oder

```
public int berechneSumme(int n)
{       int a=1;
```

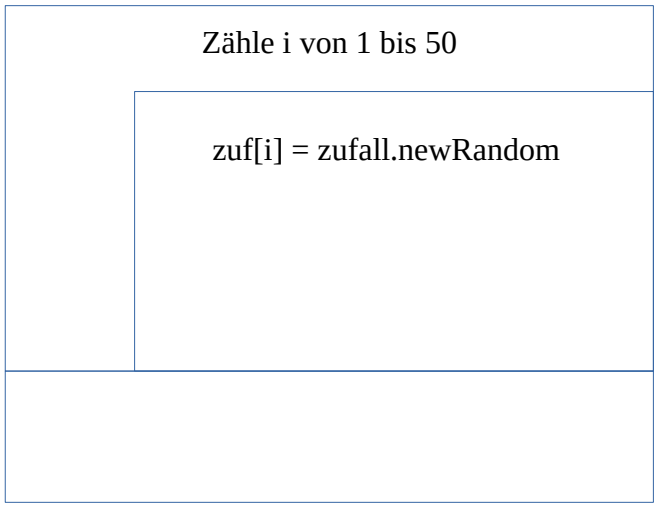
Oder:

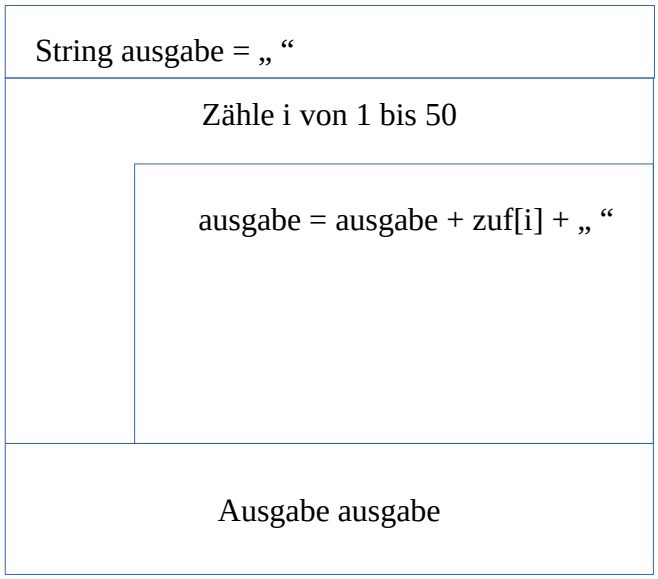
```
Public int berechneSumme(int n)
{       int summe = (n*n+1)/2;
        return summe;
}
```

```
        int summe=0;
        do
        {       summe=summe+a;
                a++;
        }
        while(a<n)
        return summe;
}
```

Aufgaben – AB 18.02.2016-01

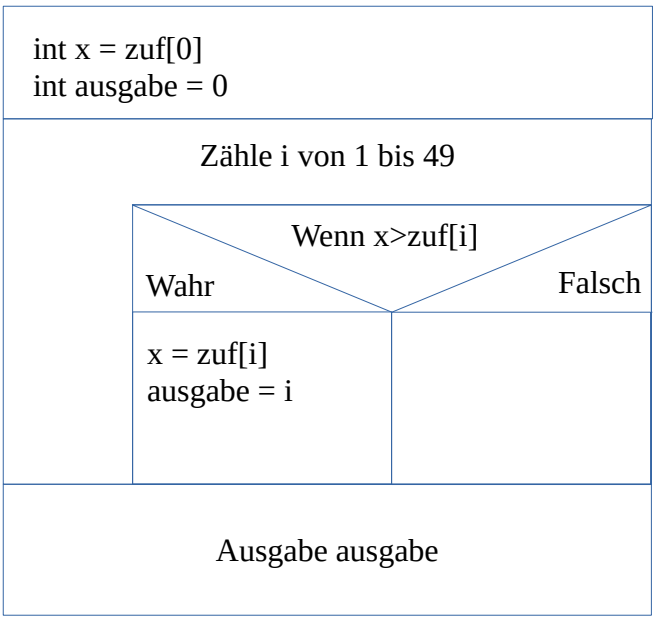
```
public class SORTIEREN
{private int[] zuf;
```

1. 

2. 

3. …
4. …

5.



gibMinStelle()

public int gibMinStelle()

{if …

Elemente Graphischer Oberflächen

Graphische Oberflächen sind aus Komponenten wie Schaltknopf, Textanzeige, Texteingabe aufgebaut. Alle diese Komponenten werden in das Zeichenfenster eingefügt. Zu jeder Komponente stellt das Paket javax.swing eine geeignete Klasse zur Verfügung.

Bsp.: JButton (Schaltknopf)

JLabel (Textanzeige)

Die Klasse Zeichenfenster hat eine Methode zum einfügen graphischer Komponenten:

komponenteHinzufuegen(JKomponent element, string Position)

… fügt die im Parameter übergebene graphische Komponente in einem Bereich rechts oder unter der eigentlichen Zeichenfläche ein. (Position = rechts bzw. unten)

Java-Syntax:

```
Import javax.swing.*; //“*“ importiert alle Klassen aus dem Paket
public class SPIELAUTOMAT
{…
    private JButton knopf;
    private JLabel textanzeige;
    public SPIELAUTOMAT
    {       …
```

```
                knopf = new JButton(„Spielen“);
                textanzeige = new JLabel(„ “);
                ZEICHENFENSTER.gibFenster().komponenteHinzufuegen(knopf, „unten“)
                ZEICHENFENSTER.gibFenster().komponenteHinzufuegen(textanzeige,
„unten“;
        }
    }
```

Projekt Petrus

Im neuen Projekt sollen unterschiedliche Niederschlagsarten am Bildschirm simuliert werden. Zunächst wird nur eine Niederschlagsart durch fallende Tropfen dargestellt.

PETRUS ^{1}– $^{\text{steuert}>}$ → ^{1} WOLKE ^{1}– $^{\text{verwaltet}>}$ → ^{n} NIEDERSCHLAG

Die Klasse Niederschlag

Die Regentropfen werden als Kreise dargestellt, die sich mit konstanter Geschwindigkeit nach unten bewegen.

NIEDERSCHLAG
private double x private double y private double vx private double vy
public NIEDERSCHLAG(double xStart, double yStart) public void bewege(double zeit) public void zeichne()

Java-Syntax:

```
Public class NIEDERSCHLAG
{       …
    public NIEDERSCHLAG(double xStart, double yStart)
    {       x = xStart;
            y = yStart;
            vx = 1;
            vy = 10;
    }
    public void zeichne()
    {       ZEICHENFENSTER.gibFenster().zeichneKreis((int)x, (int)y, 10)
```

```
            // „(int)“ = Typecast(Typumwandlung)
        }
}
```