### 1. Summe der Preise

## Umgangsprache

Setze eine Variable "s" (für Summe) auf 0.

Wiederhole für jedes Element des Arrays:

 Addiere den Wert des aktuellen Elements zu s

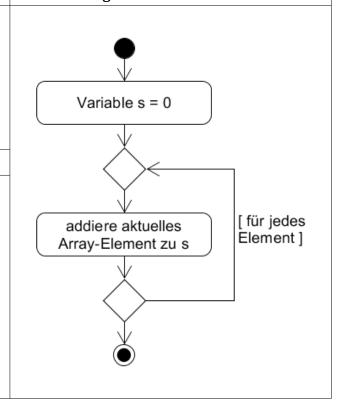
### Programmcode

```
public double summe()
{
    int i;
    double s;

    s = 0.0;
    for (i=0; i<liste.length; i++)
    {
        s = s + liste[i];
    }

    return s;
}</pre>
```

### Aktivitätsdiagramm



### 2. Mittelwerte der Preise

### Umgangsprache

Setze eine Variable "s" (für Summe) auf 0.

Wiederhole für jedes Element des Arrays:

 Addiere den Wert des aktuellen Elements zu s

Teile s durch die Anzahl der Elemente

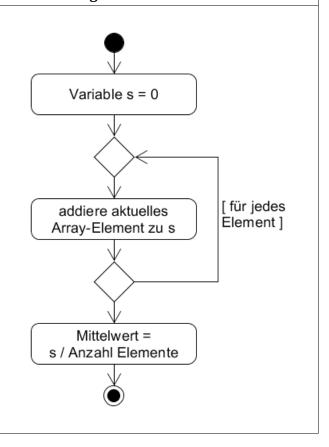
#### Programmcode

```
public double mittelwert()
{
   int i;
   double s, m;

   s = 0.0;
   for (i=0; i<liste.length; i++)
   {
      s = s + liste[i];
   }

   m = s / liste.length;
   return m;
}</pre>
```

### Aktivitätsdiagramm



return min;

# 3. Niedrigster Preis ("Minimum")

Der Algorithmus für das Maximum funktioniert analog.

## Umgangsprache Aktivitätsdiagramm Setze eine Variable "min" auf das erste Element Wiederhole für alle Elemente des Arrays ab dem zweiten: Variable min = erstes Array-Element Falls das aktuelle Element kleiner ist als min, setze min auf das aktuelle Element **Programmcode** public double minimum() [ akt. Elem. < min ] [sonst] int i; [ für jedes double min; Element setze min auf das ab dem 2.] aktuelle Element min = liste[0];for (i=1; i<liste.length; i++)</pre> if (liste[i] < min)</pre> min = liste[i];

#### 4. Anzahl Nullen

#### Umgangsprache

Setze eine Variable "anzahl" auf 0.

Wiederhole für jedes Element des Arrays:

 Falls das aktuelle Element gleich 0 ist, erhöhe anzahl um 1.

#### Programmcode

```
public int anzahlNullen()
{
   int i, anzahl;

   anzahl = 0;
   for (i=0; i<liste.length; i++)
   {
      if (liste[i] == 0)
      {
        anzahl = anzahl + 1;
      }
   }
   return anzahl;
}</pre>
```

### Aktivitätsdiagramm

