

## Selection Sort

### Algorithmus in Umgangssprache

#### Äußere Schleife:

Wiederholt die Suche des Minimums mehrmals.

*Wo fängt die Schleife an, wo hört sie auf?*

#### Innere Schleife:

Suche das Minimum im verbleibenden Array.

*Wo fängt das verbleibende Array an, wo hört es auf?*

*Was braucht es für die Suche nach dem Minimum? (Variablen etc.)*

*Achtung: es muss nicht nur das Minimum selbst, sondern auch sein Index gespeichert werden, damit man es anschließend mit der Zahl am Anfang des restl. Arrays tauschen kann*

#### Nach der inneren Schleife:

Tausche Zahl am Anfang des restl. Arrays mit dem Minimum.

*Wie tauscht man zwei Werte in einem Array?*

### Gerüst für die Java-Methode

```
int i, j;
int min, minIndex;
for (i = ???; i < ???; i++)
{
    // Suche Minimum und dessen Position (vom aktuellen Index an)
    min = ???
    minIndex = ???;
    for (j = ???; j < ???; j++)
    {
        if (???)
        {
            ???
        }
    }
    // Vertausche die kleinste Zahl mit der aktuellen
    ???
}
```

## Bubble Sort

### Algorithmus in Umgangssprache

#### Äußere Schleife:

Wiederholt das Vertauschen der Nachbarn mehrmals.

*Wo beginnt die Schleife, wo hört sie auf?*

*Nach jeder Wiederholung der äußeren Schleife sind einige Elemente des Arrays schon an der richtigen Stelle. An welchen Indizes stehen sie?*

#### Innere Schleife: Vertauschen der Nachbarn

*Wo beginnt die Schleife, wo hört sie auf?*

Vergleiche zwei Elemente und tausche sie gegebenenfalls.

*Welche Elemente? Wann werden sie vertauscht?*

### Gerüst für die Java-Methode

```
for (???; ???; i++)
{
    // Die letzten i Elemente sind schon richtig sortiert.
    // Gehe also vom Anfang bis Ende - i.
    for (???; ???; j++)
    {
        // Vergleiche jedes Element mit seinem nächsten Nachbarn
        // Falls es kleiner als der Nachbar ist, tausche
        if (???)
        {
            ???
        }
    }
}
```

## Insertion Sort

### Algorithmus in Umgangssprache

#### Äußere Schleife:

Wiederhole das Einfügen der aktuellen Zahl nach links.

*Wo beginnt die Schleife, wo hört sie auf?*

#### Innere Schleife:

Fügt die aktuelle Zahl nach links an der richtigen Stelle ein

*Wo beginnt die Schleife?*

*In welche Richtung läuft ihr „Index“?*

*Welche Zahlen werden jeweils verglichen?*

*Welche Zahl wird wohin verschoben?*

*Wann bricht die innere Schleife ab?*

#### Nach der inneren Schleife

*Was passiert, wenn die innere Schleife endet?*

### Gerüst für die Java-Methode

```
for (i = ???; i < ???; i++)  
{  
    // Merke das aktuelle Element  
    ???  
  
    // Gehe so lange nach links bis ein Element gefunden ist,  
    // das kleiner oder gleich dem aktuellen Element ist oder  
    // ODER bis das Ende der Liste erreicht ist.  
    while (???)  
    {  
        // Schiebe jedes Element, das größer als das aktuelle ist,  
        // eins nach rechts.  
        ???  
    }  
  
    // An dieser Stelle wird das aktuelle Element eingefügt.  
    ???  
}
```