**Insertion Sort**

Quellen:

* <http://users.informatik.haw-hamburg.de/~klauck/AlguDat/TIB3-AD-skript.pdf> (S. 57 ff)
* Vorlesung vom 04.11.2015 bei Prof. Dr. Klauck

Ziel: Sortieren einer Zahlenreihe basierend auf dem Sortieralgorithmus: Insertion-Sort

Allgemeiner Ablauf:

Pseudo-Code aus dem Vorlesungsskript von Prof. Dr. Klauck (S.58):

public void insertionSort(){

for ( int i = 2; i <= n ; i++ )

{

int j = i;

Datensatz<T> t = a[i];

int k = t.key;

while(a[j-1].key > k)

{

a[j] = a[j-1];

j = j-1;

}

a[j] = t;

}

}

Anpassung des Pseudo-Codes für das folgende Beispiel:

* Key-Abfrage nicht nötig, da Elemente Zahlen sind
* Erweiterung der while-Bedingung (j>1), um eine sichere Terminierung zu gewährleisten

Erläuterung an einem Beispiel:

1. Gegeben sei eine Zahlenreihe *Z* = [5,3,6,1,2,7,4]. Begonnen wird mit dem Index *i* = 2 (in unserem Beispiel die Zahl 3).
2. Gespeichert wird *i* in *j* und Z[i] in *k*. Dieses ist nötig, da ggf. eine höher-wertige Zahl an diese Position verschoben wird. Im weiteren Verlauf wird nun beginnend bei der Zahl mit dem *Index j-1,* *k* mit dieser verglichen. Wobei j nicht kleiner als 2 sein darf, da j-1 minimal den Index der ersten Zahl der Zahlenreihe darstellen kann.
3. Sollte *j* nicht kleiner als 2 sein, gehe zu (4), sonst zu (6)
4. Sollte Z[j-1] größer als *k* sein, gehe zu (5), sonst zu (6)
5. wird die Zahl mit dem *Index j-1* an die Position *j* kopiert und *j* wird um 1 verkleinert:   
   *Z* = [5,**5**,6,1,2,7,4]

Gehe zu (3)

1. k wird nach Z[j] gespeichert:   
   *Z* = [**3**,5,6,1,2,7,4]
2. Der Index i wird um 1 erhöht und ab (2) wiederholt bis einschließlich i = n

Hinweise zur Implementierung:

Um eine doppelte Abfrage in der while-Schleife zu verhindern, kann an der Position Z[0] ein Dummy (mit kleinstmöglichem Wert) gespeichert werden, welches als Stopper-Element dient und dazu führt, dass die while-Schleife sicher terminiert.