

Dokumentation der Klasse LectureWithSortAlgorithms

Zugriffsrecht	Kategorie ¹ → Rückgabetypp	Signatur	Beschreibung
(+) → public	Konstruktor → LectureWithSortAlgorithms	<code>public LectureWithSortAlgorithms (String t, int cap)</code>	Erzeugt ein neues <i>LectureWithSortAlgorithms</i> -Objekt, dass den Veranstaltungsnamen, der als <i>String t</i> übergeben wurde hat und eine maximale Anzahl an <i>int cap</i> Studenten in dem Feld <i>students</i> aufnehmen kann.
(+) → public	Auftrag → void	<code>public LectureWithSortAlgorithms (String t, int cap)</code>	Falls noch Platz in dem Feld <i>students</i> ist, wird das als Argument übergebene <i>Student</i> -Objekt an den nächsten freien Index im Feld <i>students</i> eingefügt. Falls nicht genug Platz ist, passiert nichts.
(+) → public	Auftrag → void	<code>public void show()</code>	Erzeugt eine angemessene Ausgabe aller <i>Student</i> -Objekte im Feld <i>students</i>
(+) → public	Auftrag → void	<code>public void isSorted()</code>	Die Methode wirft eine <i>Exception</i> , falls das Feld <i>students</i> nicht <i>Aufsteigend sortiert</i> ist. Wenn das Feld <i>aufsteigend sortiert</i> ist, passiert nichts.
(-) → private ²	Auftrag → void	<code>private void showStudents()</code>	Ruft für alle <i>Student</i> -Objekte im Feld <i>students</i> die <code>public String toString()</code> -Methode der Klasse <i>Student</i> auf.
(-) → private	Auftrag → void	<code>private void show (String explain, int low, int high)</code>	Gibt die <i>Student</i> -Objekte im Bereich von <i>low</i> bis <i>high</i> mithilfe der <code>public String toString()</code> -Methode der Klasse <i>Student</i> aus.

SelectionSort

Zugriffsrecht	Kategorie → Rückgabetypp	Signatur	Beschreibung
(+) → public	Auftrag → void	<code>public void selectionSortByName() public void selectionSortByNumber()</code>	Durchläuft das gesamte Feld und prüft für jedes Element, anhand der <code>searchForMinimalName(int start)</code> oder <code>searchForMinimalNumber(int start)</code> Methode, an welchem Index das Minimumobjekt, bezüglich des sortierkriteriums steht und <i>vertauscht</i> das Objekte am <i>aktuellen Index</i> mit dem Objekt am Index, wo das <i>Minimumobjekt</i> steht mithilfe der <code>void swapStudents(int i, int j)</code> Methode. Das Minimumobjekt steht nun an der richtigen Position und ist einsortiert.
(-) → private	Anfrage → int	<code>private int searchForMinimalName (int start) int searchForMinimalNumber (int start)</code>	Durchläuft das gesamte Feld und liefert den <i>Index zurück</i> , an dem sich das Objekt befindet, dass bezüglich des Sortierkriteriums das kleinste ist.
(-) → private	Auftrag → void	<code>private void swapStudents (int i, int j)</code>	Vertauscht das <i>Students</i> -Objekt am <i>Index i</i> mit dem <i>Students</i> -Objekt am <i>Index j</i>

InsertionSort

Zugriffsrecht	Kategorie → Rückgabetypp	Signatur	Beschreibung
(+) → public	Auftrag → void	<code>public void insertionSortByNumber() public void insertionSortByName()</code>	Bestimmt den allgemeinen Zwischenzustand und ruft für den aktuellen Index, der noch nicht sortiert ist, die <code>shiftStudentsByNumber(int start)</code> oder <code>shiftStudentsByName(int start)</code> Methode auf, die das Objekt am <i>aktuellen Index</i> an seiner <i>richtigen Position</i> einfügt und alle Objekte nach rechts schiebt, die bezüglich des Sortierkriteriums größer als das einzufügende Objekt sind. Danach ist das Objekt einsortiert.
(-) → private	Auftrag → void	<code>private void shiftStudentsByNumber (int start) private void shiftStudentsByName (int start)</code>	Verschiebt alle Objekte, die bezüglich des Sortierkriteriums größer sind als das Objekt am Index start jeweils ein Index nach rechts und fügt anschließend das Objekt, dass sich am Index start befand an der <i>richtigen Position</i> des Feldes ein.

¹ Die Gängigsten Kategorien sind 'Konstruktor', 'Anfrage' und 'Auftrag' → Eine 'Anfrage' gibt etwas zurück, während ein 'Auftrag' den Rückgabetypp void hat. Eine Anfrage könnte zum Beispiel den Rückgabetypp boolean erwarten.

² Beachte, dass man auf private Methoden nur innerhalb der Klasse zugriff hat.

Quicksort

Zugriffsrecht	Kategorie → Rückgabetyt	Signatur	Beschreibung
(+) → public	Auftrag → void	<pre>public void quickSortByNumber () public void quickSortByName ()</pre>	<p>Diese Methode ist notwendig, da sie den <i>ersten Aufruf</i> von <code>private void groupByNumber (int leftBound, int rightBound)</code> oder <code>private void groupByName (int leftBound, int rightBound)</code> übernimmt.</p> <p>Dieser Umweg ist erforderlich, da Parameterwerte für die Größe und die Belegung des Feldes (firstUnused) ebenfalls privat deklariert sind.</p> <p>Der erste Aufruf findet mit den Werten (0, firstUnused-1) statt</p>
(-) → private	Auftrag → void	<pre>private void groupByNumber (int leftBound, int rightBound) private void groupByName (int leftBound, int rightBound)</pre>	<p>Rekursive Methode</p> <p>Stellt eine Teilsortierung bezüglich des Pivot-Elements (Das Pivot-Element ist das Element, dass am Index rightBound steht) her:</p> <p>Alle Elemente, die kleiner als das Pivot sind stehen links vom Pivot. Alle Elemente die größer als das Pivot sind, stehen rechts vom Pivot. Das Pivot wird mit dem Element am ersten Index der größeren Elemente getauscht (da dieses offensichtlich größer sein muss, als das Pivot).</p> <p>Nach dem Tausch (Mithilfe der <code>private void swapStudents (int i, int j)</code> Methode) ist <i>das Pivot einsortiert</i> und steht an der richtigen Stelle (Die Teilfolgen sind in sich nicht sortiert! Sie sind nur kleiner/größer als das Pivot. Beispiel: (Wenn das Pivot 5 gewesen ist)</p> <p> 4 1 3 2 4 1 2 5 8 9 7; kleiner als Pivot, Pivot, Größer als Pivot)</p>
(-) → private	Auftrag → void	<pre>private void swapStudents (int i, int j)</pre>	<p>Vertauscht das <i>Students</i>-Objekt am Index <i>i</i> mit dem <i>Students</i>-Objekt am Index <i>j</i></p>