Image Analysis Lab
Jun.-Prof. Dr.
Olaf Ronneberger
Claudius Korzen

Algorithmen und Datenstrukturen (ESE + IEMS) WS 2014 / 2015

http://lmb.informatik.uni-freiburg.de/lectures/

UNI FREIBURG

Übungsblatt 6

Abgabe für ESE: bis Donnerstag, den 4. Dezember um 10:00 Uhr

Abgabe für IEMS: bis Donnerstag, den 18. Dezember um 10:00 Uhr

Aufgabe 1 (20 Punkte)

Implementieren Sie eine Klasse *PriorityQueue* mit den folgenden in der Vorlesung erklärten Methoden: *insert* (6 Punkte), *getMin* (1 Punkt), *deleteMin* (6 Punkte), *changeKey* (6 Punkte), *size* (1 Punkt).

Für die genaue Spezifikation dieser Methoden (insbesondere der Grenzfälle) sowie nützlicher Hilfsmethoden, siehe den Code Design Vorschlag auf der Webseite. Wenn Sie noch keine Erfahrung mit C++ template oder Java Generics Programmierung haben (und die Gelegenheit, das zu Lernen ungenutzt lassen wollen), dürfen Sie statt der templates/Generics auch einen string als Datentyp im *PriorityQueueItem* benutzen.

Schreiben Sie wie gehabt für jede dieser Methoden einen Unit Test. Wie immer gilt: Immer erst den Test schreiben, dann die Methode implementieren. Es ist ok, wenn Sie alle Methoden gemeinsam in einer Testfunktion testen.

Beachten Sie, dass Sie hier Teile von Ihrem Code oder der Musterlösung vom 2. Übungsblatt (HeapSort) wiederverwenden können. Wenn Sie das tun, kopieren Sie den Code bitte, und versuchen Sie nicht, irgendwie bestehenden Code aus *uebungsblatt_02* einzubinden.

Committen Sie, wie gehabt, Ihren Code in das SVN, in einen neuen Unterordner *uebungsblatt_06*, sowie, ebendort, Ihr Feedback in einer Textdatei *erfahrungen.txt*. Insbesondere: Wie lange haben Sie ungefähr gebraucht? An welchen Stellen gab es Probleme und wieviel Zeit hat Sie das gekostet?

Tipps C++ templates:

- Template-Funktionen gehören in **.h* oder **–inl.h* Dateien (siehe Coding-styles), sonst kann der Compiler keinen Code für ein Objekt mit einem neuen Datentyp erzeugen.
- Ein *template< typename T>* vor der Klassen-Deklaration reicht aus, danach kann T innerhalb der Klasse so benutzt werden, als sei es z.B. mit einem *#define T float* definiert worden.
- Wenn Sie die Implementation der Methoden außerhalb der Klassendeklaration (also z.B. in der **–inl.h* Datei) vornehmen, muss dort das *template< typename T>* vor jeder Funktion wiederholt werden, also z.B.

```
template <typename T>
size_t PriorityQueue<T>::size() {
  return heap_.size();
}
```

- Von außerhalb benutzen Sie die template-Klasse dann wie die STL Container, z.B. *PriorityQueueItem<float>*

Tipps Java Generics:

- Ihre Klassenname (in PriorityQueueItem.java) muss nur das generics Argument in spitzen Klammern enthalten, also *public class PriorityQueueItem<T> { ...*
- Den Platzhalter *T* können Sie dann in der Klasse wie einen normalen Datentyp benutzen
- Von außerhalb benutzen Sie die Klasse, wie sie es von den Standard-Containern her kennen, also z.B. mit *PriorityQueueItem<String>*