

## Übungsblatt 6

Abgabe bis Dienstag, den **12. Juni 2018** um **12:00 Uhr**

Bei diesem Übungsblatt geht es darum, eine einfache *String* Klasse zu schreiben. Die Tests sind für diese Aufgabe vollständig vorgegeben, in einer Datei *StringTest.cpp* auf dem Wiki. Sie müssen also nur noch die Dateien *String.h* (mit der Klassendeklaration) und *String.cpp* (mit den Implementierungen der Methoden) erstellen. Sie sollten dabei die Dateien aus der Vorlesung als Ausgangspunkt nehmen. Die *StringMain.cpp* brauchen Sie für dieses Übungsblatt nicht.

Die Tests sollten am Ende zusammen mit Ihrem Code fehlerfrei kompilieren und durchlaufen. Außerdem sollte bei der Benutzung der Klasse unter keinen Umständen ein *memory leak* entstehen können. Insbesondere sollte *valgrind -leak-check=full* auf dem Testprogramm ohne Fehler durchlaufen.

Ihr Code sollte außerdem „const-correct“ sein. Das heißt, Sie sollten überall da *const* verwenden, wo es sinnvoll und möglich ist. Sie sollten sich nicht darauf verlassen, dass der vorgegebene Code diese Eigenschaft schon erfüllt.

Falls Sie sich über die Auswahl der Methoden wundern: die Funktionalität wird auf dem nächsten Übungsblatt benötigt (mehr wird noch nicht verraten).

### Aufgabe 1 (4 Punkte)

Vervollständigen Sie die Deklaration der Klasse *String*, so dass Sie zu den vorgegebenen Tests passt. Die zusätzlichen Methoden sind auch noch einmal kurz in den nächsten Aufgaben beschrieben. Die Membervariablen sollten *private* sein. Damit die Tests trotzdem Zugriff darauf haben, müssen Sie auf geeignete Weise *FRIEND\_TEST* verwenden.

### Aufgabe 2 (3 Punkte)

Implementieren Sie eine Methode *int count(char)*, so dass der entsprechende Test durchläuft. Die Methode zählt, wie oft das gegebene Zeichen in der von dem Objekt repräsentierten Zeichenkette vorkommt.

### Aufgabe 3 (3 Punkte)

Implementieren Sie eine Methode *float toFloat()*, so dass der entsprechende Test durchläuft. Die Methode versucht, die von dem Objekt repräsentierte Zeichenkette in eine Fließkommazahl umzuwandeln und gibt diese zurück. Sie dürfen für diese Aufgabe die eingebaute Funktion *atof* benutzen (siehe man page).

### Aufgabe 4 (10 Punkte)

Implementieren Sie eine Methode *void split(char, int, String[])*, so dass die entsprechenden Tests durchlaufen. Die Methode zerlegt die von dem Objekt repräsentierte Zeichenkette in Stücke. Das erste Argument ist das Zeichen, an dem zerlegt werden soll. Das zweite Argument ist die maximale Anzahl Stücke (an den Tests sieht man, wie es gemeint ist). Das dritte Argument ist ein Feld mit den Stücken (verstehen Sie, dass hier ein Zeiger übergeben wird).

Laden Sie wie gehabt alle Code-Dateien und das Makefile in unser SVN hoch, in einem neuen Unterverzeichnis *blatt-06*. Es gelten weiterhin die 10 Gebote von der letzten Seite des Ü1 und nehmen Sie die Ratschläge Ihres Tutors / Ihrer Tutorin ernst.

Laden Sie wie gehabt auch eine Datei *erfahrungen.txt* hoch (im Unterordner *blatt-06*), in der sie kurz Ihre Erfahrungen mit dem Ü6 und der Vorlesung dazu beschreiben.

Wenn es keinen freien Willen gibt, inwiefern ist man dann noch verantwortlich für seine Handlungen?