

## Übungsblatt 5

Revision: 1.0

---

### Strukturtests zur Anweisungsüberdeckung

In diesem Übungsblatt stehen Strukturtests zur Codeüberdeckung im Vordergrund. Für einen gegebenen Testling sollen unter Verwendung des RT-Testers Softwareintegrations- und Modultests entwickelt werden, mit dem Ziel, eine vollständige Anweisungsüberdeckung zu erreichen.

Als Testling dient die Datei `replace.cpp`<sup>1</sup>, welche sich im online zur Verfügung gestellten RT-Tester-Projekt *TA1-Uebung-05* im Verzeichnis `./software/replace/` befindet. Sie enthält - neben anderen - 3 Hauptfunktionen, mit deren Hilfe ein Textpattern gegen ein Muster ausgetauscht werden kann:

- `int getpat( char* arg, char* pat )` erhält das zu suchende Textpattern in `arg` und schreibt es in aufbereiteter Form nach `pat`.
- `bool getsub( char* arg, char* sub )` erhält den Substitutionsstring in `arg`, kappt ungültige führende und endende Zeichen und schreibt den bereinigten String nach `sub`.
- `void subline( char* lin, char* pat, char* sub, char* resStr )` erhält eine Textzeile `lin`, das Pattern `pat` und den Substitutionsstring `sub` als Eingabe und schreibt den modifizierten String, in welchem alle Vorkommen von `pat` in `lin` durch `sub` ersetzt wurden, in den Ergebnisstring `resStr`.

Der Speicher für die Ausgabestrings `pat`, `sub` und `resStr` muss vom Aufrufer in der erforderlichen Größe bereitgestellt werden.

### Aufgabe 1: Softwareintegrationstests

Entwickeln Sie Softwareintegrationstests, mit denen die maximal mögliche Anweisungsüberdeckung erreicht wird.

- Für jeden Testfall soll die Aufruffolge

```
getpat( arg1, pat );  
getsub( arg2, sub );  
subline( lin, pat, sub, resStr );
```

mit geeignet gewählten Eingabedaten ausgeführt werden. In bestimmten Fällen kann es allerdings sinnvoll sein auf Aufrufe nachfolgender Funktionen zu verzichten, wenn der Aufruf einer vorausgehenden Funktion erwartungsgemäß fehlschlägt.

---

<sup>1</sup>Die Quellen stammen in leicht modifizierter Form aus einer von Siemens Corporate Research publizierten Bibliothek von Beispielprogrammen, die als "Benchmarks" zur Prüfung der Wirksamkeit von Testverfahren dienen. Die vollständige Bibliothek ist unter <http://www.cc.gatech.edu/aristotle/Tools/subjects> verfügbar.

- Definieren Sie für jeden Testfall geeignete erwartete Resultate und prüfen Sie das Testlingverhalten gegen diese mittels `@rttAssert`.
- Dokumentieren Sie alle Testfälle im RT-Tester-Format (`@tag`, `@condition`, `@event`, `@expected` und ggf. `@note`).

## Aufgabe 2: Modultests

Durch Softwareintegrationstests, wie sie in Aufgabe 1 beschrieben sind, wird keine 100%ige Anweisungsüberdeckung erreicht werden können. Führen Sie deshalb ergänzende Modultests durch. Analog zur Aufgabe 1 müssen auch für die Modultests geeignete erwartete Resultate definiert und das Testlingverhalten gegen diese geprüft werden. Jeder Testfall ist wieder im RT-Tester-Format zu dokumentieren.

## Hinweise zum Testprojekt TA1-Uebung-05

- Mit `TA1-Uebung-05/replace/test1` existiert ein Beispiel für einen Softwareintegrationstest gemäß Aufgabe 1. Die Testprozedur enthält einen Testfall, welcher als Ausgangspunkt für weitere Testfalldefinitionen verwenden werden kann.
- Um den aktuellen Stand der Anweisungsüberdeckung zu erhalten, kann nach einer Testausführung im RTTUI der Button *“Prepare the code coverage”* verwendet werden. Hierdurch wird (im der Testprozedur übergeordneten Verzeichnis) ein Verzeichnis namens `code-coverage` angelegt, das die entsprechenden Informationen enthält.

**Abgabe: 02.02.2010 bis 16:00 Uhr**

Die Abgabe erfolgt per E-Mail an die Tutorin.