

## Beispiel 1: Strings (24 Punkte)

Entwerfen Sie eine Klasse namens `String`, die einen privaten Zeiger auf einen dynamischen Speicherbereich besitzt. Als weitere Attribute sollen `len` für die Stringlänge und `bufsize` für die Größe des Stringbuffers angelegt werden. Schreiben Sie drei Konstruktoren `String(void)`, `String(const char*)` und `String(const char)`, die den nötigen Speicherbereich reservieren, und zwar soll der für den String verwendete Speicherbereich immer um 15 Zeichen größer sein als benötigt. Schreiben Sie drei Zuweisungsoperatoren `operator=(String)`, `operator=(const char*)` und `operator=(const char)` sowie einen copy-Konstruktor `String(const String&)`. Vergessen Sie den Destruktor nicht.

Um die vom Benutzer ansprechbaren Funktionen von der tatsächlichen Verwaltung des Speichers zu kapseln, entwerfen Sie eine private Methode `replace`, die einen neuen String übernimmt und alle dazu nötigen Maßnahmen ergreift.

Des Weiteren soll die Klasse mit überladenen `<<`- und `>>`-Operatoren ausgestattet werden, wobei die Eingabe vorerst mit einer maximalen Größe arbeiten darf.

Implementieren Sie für `operator+` und `operator +=` alle nötigen Funktionen, um die Datentypen `const String&`, `const char*` und `const char` verarbeiten zu können.

Überlegen Sie, welche Funktionen als Elementfunktionen deklariert werden können und welche als Nicht-Elementfunktionen deklariert werden müssen.

Benutzen Sie zur Implementierung der `operator`-Funktionen eine Funktion `insert(unsigned long pos, unsigned long len, const char*)`, die an der Stelle `pos` im Stringspeicher die ersten `len` Zeichen des Strings `s` einfügt. `insert` soll dabei Gebrauch vom zusätzlichen Speicher des Stringpuffers machen.

Überladen Sie den Operator `[]`, um auf die einzelnen Zeichen des `String`-Objekts genau so zugreifen zu können wie auf ein `char`-Feld und überladen Sie für die Klasse `String` die Vergleichsoperatoren (`<`, `<=`, `==`, `!=`, `>=`, `>`).

Trennen Sie die Deklarationen und Definitionen voneinander, und speichern Sie diese in zwei Dateien namens `String.h` und `String.cpp`.

Achten Sie bitte darauf, bei Ihrer schriftlichen Abgabe im Header Ihre Account-Nummer anzugeben, damit wir Ihre elektronische Abgabe finden können.

**Abgabe:** bis 14.6. 23:59 Uhr elektronisch (Verzeichnis `uebung9` einrichten!) und am 15.6. bzw. 16.6. jeweils zu Beginn des Praktikums auf Papier (Listing der Quelldateien).