

# Übung 4 – Vorüberlegungen (I)

Copy Konstruktor und Zuweisungsoperator werden während der Laufzeit üblicherweise sehr häufig aufgerufen

- Oft wird das Objekt danach auch nicht mehr verändert (z.B. temporäre Objekte)

Vorgehensweise unserer Klasse ist daher sehr ineffizient, da bei in diesen Fällen

- neuer Speicher für die Zeichenkette allokiert
- und die Zeichenkette kopiert wird



Effizienter wäre es nur bei Veränderungen Speicher zu allokieren bzw. Zeichenketten zu kopieren.

Solange es keine Veränderung gibt, könnten mehrere Objekte (über den Zeiger) die gleiche Zeichenkette referenzieren

Kann in C++ leicht umgesetzt werden:

- Volle Kontrolle über die Operatoren
- Dadurch kann auch ein Referenzzähler eingebaut werden (Anzahl Kopien im Einsatz).

# Übung 4 – Vorüberlegungen (II)

`repStruct` ist die Repräsentation der Daten die in der Klasse verwaltet werden

Als Attribut wird ein Zeiger auf ein Element dieser Struktur verwendet

- D.h. auch hier ist `new/delete` notwendig wenn ein neues Element benötigt wird

Copy Konstruktor/Zuweisungsoperator übernehmen einfach den Zeiger und erhöhen den Referenzzähler (`refCount`)

Bei Veränderungen erzeugen wir ein neues Repräsentationselement und reduzieren den Referenzzähler des bisherigen Elements

- Ist `refCount` 0 gibt es keinen weiteren Nutzer der Repräsentation und wir geben den Speicher für die Repräsentation und die Zeichenkette frei

**BISHER:**

```
class MyString
{
private:
    char *sPtr;
    unsigned len;
...
}
```

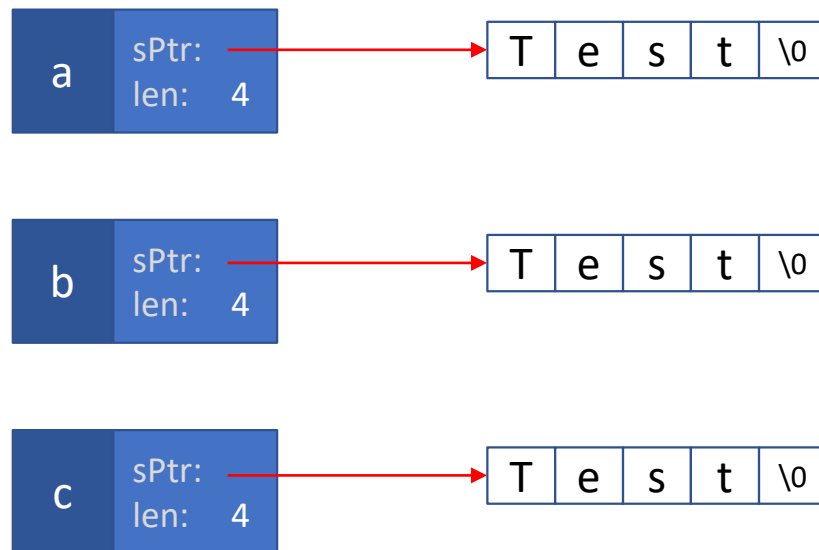
**NEU:**

```
class MyString
{
private:
    struct repStruct
    {
        char* sPtr;
        unsigned len;
        int refCount;
    };
    struct repStruct* repPtr;
...
}
```

# Übung 4 – Vorüberlegungen (III)

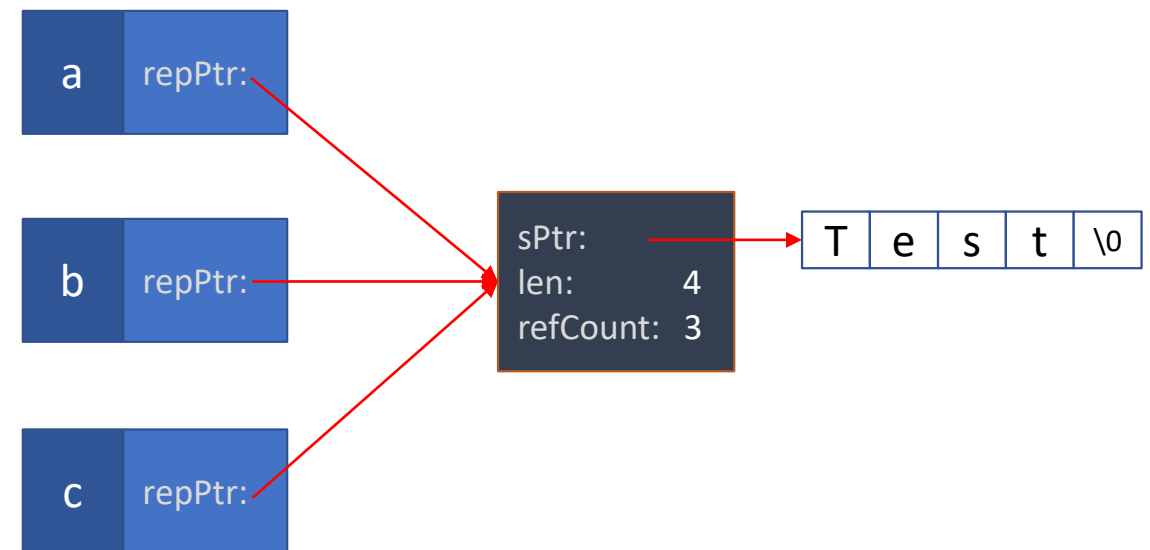
## Basierend Stand MyString Übung 3

```
MyString a = "Test";  
MyString b = a;  
MyString c = b;
```



## Ziel für MyString Übung 4

```
MyString a = "Test";  
MyString b = a;  
MyString c = b;
```



# Übung 4

1. Modifizieren Sie die Klasse mit Hilfe der `repStruct` Konstruktion.  
Hierzu müssen speziell folgende Methoden angepasst werden:
  - Alle Konstruktoren und der Destruktor
  - Zuweisungsoperator
  - Operator `+=` und `+`Zugriff aller Methoden auf `sPtr` muss angepasst werden.
2. Zu Testzwecken modifizieren Sie die `printOn()` Methode um zusätzlich zur Zeichenkette den Referenzzähler auszugeben (z.B. Ausgabe: „Test[2]“ – Repräsentation wird von 2 Objekten verwendet).

Zusammen mit der bereitgestellten „main.cpp“ Datei sollte folgende Ausgabe erfolgen:

```
a:<[1]>, b:<Test[1]>, c:<[1]>  
a:<[2]>, b:<Test[1]>, c:<[2]>  
a:<Test[3]>, b:<Test[3]>, c:<Test[3]>  
a:<Test[2]>, b:<Test[2]>, c:<Testfall[1]>
```

