

Universität Potsdam
Institut für Informatik

Sommersemester 2014

Praxis der Programmierung

Aufgabenblatt Woche 10

1. Schreiben Sie ein **C++**-Programm, das eine Stringvariable definiert und mit einer E-Mail-Adresse an der Uni Potsdam initialisiert. Anschließend wird durch Aufruf von Methoden der Klasse `string` folgende Ausgabe erreicht:

```
Adresse: mklose@uni-potsdam.de
Benutzer ist mklose
Domaene ist uni-potsdam.de
```

2. Schreiben Sie Klassen `Date` und `HighScore`, jeweils in separaten Dateien.
 - `Date` repräsentiert ein Datum mit Tag, Monat und Jahr. Kapseln Sie die Datenelemente und stellen Sie geeignete Methoden und Konstruktoren bereit.
 - `HighScore` hat ein Datenelement `score` vom Typ `int` und ein Datenelement `date` vom Typ Pointer auf `Date`, das auf dem Heap angelegt wird. Ein `HighScore`-Objekt repräsentiert einen an `date` erreichten höchsten Punktestand `score` in einem Spiel. Auch hier sind die Datenelemente zu kapseln. Stellen Sie geeignete Konstruktoren, einen Destruktor und geeignete Methoden zur Verfügung.
Warum ist ein Destruktor sinnvoll?

 - Testen Sie Ihre Klassen in einer Anwendung, die wiederum in einer separaten Datei gespeichert ist.

Hinweise:

- (a) Denken Sie daran, was Sie über das Verteilen von Quellcode auf mehrere Dateien in **C** gelernt haben. Splitten Sie die Klassendefinition so, dass Sie die Schnittstellen-Beschreibung (welche Datenelemente, welche Methodensignaturen) in Header-Dateien vereinbaren und die Implementierungen in `.cpp`-Dateien halten.
- (b) Importieren Sie Dateien (`#include`) dort, wo sie gebraucht werden. Das gilt für Ihre eigenen (Header-)Dateien und für Bibliotheksdateien.
- (c) Achten Sie darauf, dass nichts doppelt definiert wird (durch ungünstige Importe).

3. Schreiben Sie eine Klasse `Rational` rationaler Zahlen, die durch gemeine Brüche repräsentiert werden. Sie hat zwei gekapselte Datenelemente `num` (*numerator*, *Zähler*) und `denom` (*denominator*, *Nenner*) vom Typ `int`. Ferner soll die Klasse über folgende öffentliche Konstruktoren verfügen:

- `Rational()` zum Erzeugen der 0,
- `Rational(int)` zum Erzeugen der als Parameter angegebenen ganzen Zahl,
- `Rational(int, int)` zum Initialisieren beider Datenelemente.

Die Methoden der Klasse, alle öffentlich, sind:

- `int gcd()` zum Ermitteln des g.g.T. von Zähler und Nenner,
- `void reduce()` zum Kürzen des Bruches, so dass Zähler und Nenner von nun an teilerfremd sind (Überschreiben der Werte der Datenelemente!),
- `void extend(int)` zum Erweitern des Bruches mit dem als Parameter angegebenen Faktor (Überschreiben der Werte der Datenelemente!),
- `string toString()`, die den Bruch in der Form a/b zurückgibt,
- `string toStringReduced()`, die den gekürzten Bruch zurückgibt.

Rufen Sie, wenn immer es möglich ist, bereits definierte Methoden auf.

Testen Sie Ihre Klasse mit Hilfe von `useRational.cpp` (Kopie aus `/home/rlehre/Woche_10`)!