

Universität Potsdam
Institut für Informatik

Sommersemester 2015

Praxis der Programmierung

Aufgabenblatt Woche 12

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 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** (Kopieren Sie diese Datei)!

3. Definieren Sie eine Klasse **Gcd**, die eine statische Methode **gcd(int, int)** bereitstellt, mit der der größte gemeinsame Teiler der beiden **int**-Parameter ermittelt und zurückgegeben wird. Es soll der Euklid'sche Algorithmus verwendet werden.

4. Modifizieren Sie die Klasse `Rational` und die Testanwendung `useRational.cpp` so, dass die Klasse den größten gemeinsamen Teiler nirgends mehr selbst berechnet, sondern statt dessen die Klasse `Gcd` benutzt.
5. Implementieren Sie zwei Unterklassen der Klasse `Point` wie folgt:
 - (a) `Square` hat ein zusätzliches gekapseltes Datenelement `a` vom Typ `Integer`, das die Kantenlänge des Quadrates bezeichnet. Die Datenelemente von `Point` bezeichnen den linken oberen Eckpunkt des Quadrates. Es gibt einen Konstruktor zum Initialisieren aller Datenelemente und drei Methoden `getSize()` zum Abfragen und `resize(int)` zum Ändern der Kantenlänge sowie `area()`, die den Flächeninhalt von `Square`-Instanzen zurückgibt.
 - (b) `Circle` als Klasse von Kreisen (mit Mittelpunkt und Radius) in Analogie zu `Square` (also auch mit nur einem zusätzlichen Datenelement, einem Konstruktor und den Methoden `getSize()`, `resize(int)` und `area()`).
 - (c) Benutzen Sie Header-Dateien für die Klassendefinitionen!
6. Entwickeln Sie eine alternative Klassendefinition von `Circle`, wobei `Circle` jetzt von `Square` ableitet (Unterklasse von `Square` ist). Testen Sie!