

**Universität Potsdam
Institut für Informatik**

Sommersemester 2014

Praxis der Programmierung

Aufgabenblatt Woche 11

1. Kopieren Sie die Datei `/home/rlehre/Woche_11/static.cpp`. Analysieren Sie dieses Programm zur Demonstration von Effekten mit Klassenvariablen und beantworten Sie folgende Fragen.
 - (a) Die Klasse `Static` besitzt ein statisches Datenelement `exCounter`. Welche Aussage können Sie über den Wert von `exCounter` treffen, wenn ein Exemplar von `Static` mit dem Konstruktor der Klasse erzeugt wird?

 - (b) Weshalb ist es sinnvoll, dass `increase()` und `output()` nicht statisch definiert sind?

 - (c) Weshalb kann der Aufruf von `setCounter(10)` in der `main`-Methode ohne Qualifikation mit dem Klassennamen erfolgen?

 - (d) Kann der Aufruf von `setCounter(100)` in der `main`-Methode auch mit einer Qualifikation mit einem Objekt erfolgen?

2. 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.
3. 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.
4. Kopieren Sie aus `/home/rlehre/Woche_11/` die Dateien `point.h`, `point.cpp` sowie `square_0.cpp` und `circle_0.cpp`. Analysieren Sie diese Klassendefinitionen. Was stellen Sie fest? Ist das Vorgehen effizient?
Kopieren Sie auch `/home/rlehre/Woche_11/geometry_0.cpp`, um die Klassen zu testen.
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. Kopieren Sie aus `/home/rlehre/Woche_11/` die Dateien `autos.cpp` und `cars.cpp`. Übersetzen Sie die Programme **nicht** und führen Sie sie **nicht** aus!

- Welche Ausgabe erwarten Sie bei `autos`?

- Welche Ausgabe erwarten Sie bei `cars`?

- Übersetzen Sie jetzt, führen Sie aus und vergleichen Sie die Ausgaben!

7. Entwickeln Sie eine alternative Klassendefinition von `Circle`, wobei `Circle` jetzt von `Square` ableitet (Unterklasse von `Square` ist). Testen Sie!