Hochschule Aalen
Fakultät Elektronik und Informatik
Studiengang Informatik

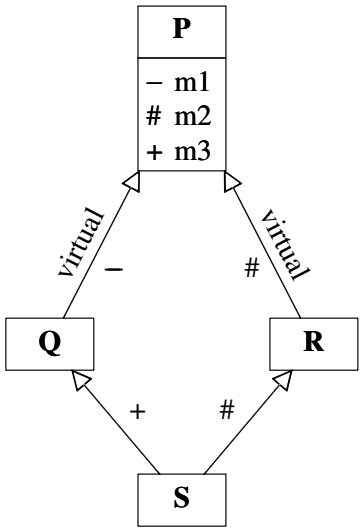
Programmieren in C++

Vorlesung im Wintersemester 2017/2018
Prof. Dr. habil. Christian Heinlein

6. Übungsblatt (7. Dezember 2017)

Aufgabe 10: Zugriffsrechte

Gegeben sei die folgende Klassenhierarchie:



Teilaufgabe 10.a)

Welche Zugriffsrechte besitzen die Elemente `m1`, `m2` und `m3` der Klasse `P` in den abgeleiteten Klassen `Q`, `R` und `S`?

Teilaufgabe 10.b)

Gegeben seien Objekte `p`, `q`, `r` und `s` mit Typ `P`, `Q`, `R` bzw. `S`.

Welche der Elementverwendungen `p.m2`, `q.m2`, `r.m2` und `s.m2` sind innerhalb der Klassen `P`, `Q`, `R` und `S` erlaubt? Begründen Sie Ihre Antwort jeweils anhand der in der Vorlesung besprochenen Regeln!

Welche Änderungen ergeben sich, wenn `R` ein Freund von `Q` und/oder umgekehrt ist?

Aufgabe 11: Schnittstellen- und Implementierungshierarchie

Teilaufgabe 11.a)

Definieren Sie folgende Schnittstellen mit sinnvollen Vererbungsbeziehungen unter Verwendung geeigneter „Schnittstellenbausteine“ (vgl. §4.8.10 der Vorlesung):

- Ein allgemeines geometrisches Objekt `Figure` besitzt Mittelpunktskoordinaten `x` und `y`, eine Breite `width`, eine Höhe `height` sowie eine Fläche `area`.
  `move(dx, dy)` verschiebt den Mittelpunkt in x-Richtung um `dx` und in y-Richtung um `dy`.
  `scale(f)` multipliziert Breite und Höhe jeweils mit dem Faktor `f`.

- Ein regelmäßiges geometrisches Objekt `RegularFigure` besitzt zusätzlich eine Größe `size`, die mit `width` und `height` übereinstimmt.

- Ein Rechteck `Rectangle` ist ein allgemeines geometrisches Objekt, das zusätzlich eine Diagonalenlänge `diag` besitzt.

- Ein Quadrat `Square` ist ein regelmäßiges Rechteck.

- Eine Ellipse `Ellipse` ist ein allgemeines geometrisches Objekt.

- Ein Kreis `Circle` ist eine regelmäßige Ellipse, der zusätzlich einen Durchmesser `diam` und einen Radius `rad` besitzt, wobei `diam` mit `size` übereinstimmt und `rad` die Hälfte davon ist.

Teilaufgabe 11.b)

Erstellen Sie modulare Implementierungskomponenten wie z. B. `CenterComp` zur Implementierung der Funktionen `x`, `y` und `move` für beliebige geometrische Objekte oder `RectangleComp` zur Implementierung der Funktionen `diag` und `area` für Rechtecke!

Teilaufgabe 11.c)

Definieren Sie zu den Schnittstellen `Rectangle`, `Ellipse`, `Square` und `Circle` jeweils eine zugehörige Implementierungsklasse, indem sie die zuvor erstellten Komponenten jeweils geeignet kombinieren!

Die Objekte dieser Klassen sollen nur die wirklich benötigten Datenelemente besitzen, d. h. die Mittelpunktskoordinaten sowie eine bzw. zwei Abmessungen bei regulären bzw. allgemeinen geometrischen Objekten.

Die Konstruktoren der Klassen sollen als Parameter die Abmessung(en) des Objekts erhalten und die Mittelpunktskoordinaten jeweils mit 0 initialisieren.

Teilaufgabe 11.d)

Definieren Sie die Konstruktoren der Implementierungsklassen privat oder halböffentlich, damit sie nicht direkt aufgerufen werden können.

Erstellen Sie stattdessen befreundete globale Erzeugungsfunktionen wie z. B. `make_rectangle`!