Informatik I WS 2005/2006

Übungsblatt 8

Abgabe: 9.1.06 - 11.1.06

Aufgabe 1 (Typsysteme und Ausdruck-Bäume) Punkte: 1+2

Zeichnen Sie einen Ausdruck-Baum (Folie 24 im Kapitel Typkonvertierung) inklusive der notwendigen Typ-Konvertierungen:

a) für den Ausdruck `('c'+1)==('f'+1)` (`'c'` und `'f'` sind vom Typ `"char"`) gemäss der Promotion-Hierarchie

b) und für den Ausdruck `strcmp(imag(1+f), ('a'+1.0))`. Nehmen Sie dabei an, daß das Typsystem zusätzlich zu den üblichen eingebauten Typen noch die Typen `Complex` und `String` enthält. Die Promotion-Hierarchie sei um die Konvertierungen `float->Complex->string` erweitert. Die Funktion `strcmp` vergleicht zwei Strings und gibt einen boolschen Wert zurück. Die Funktion `imag` liefert den imaginären Anteil einer Komplexen Zahl vom Typ `Complex` als `float` zurück.

Aufgabe 2 (Typsysteme) Punkte: 2+2

Vergleichen Sie die Typisierung in

a) Python und

b) C/C++:

Seien x und y vom selben Typ. Welcher Typ muss das sein, damit die folgenden Ausdrücke korrekt sind? Welchen Wert haben die Ausdrücke? Welche Ausdrücke sind auf jeden Fall korrekt? Welchen Typ haben x und y nach Auswertung der Ausdrücke? Welche der Ausdrücke sind in der jeweiligen Sprache gar nicht zulässig?

- x==y=x
- (x==y)=x
- x==(y=x)
- x=y==x
- (x=y)==x
- x=(y==x)

Aufgabe 3 (Klassen, Dokumentation) Punkte: 2+5+2+4

In dieser Aufgabe soll eine Klasse `GeoInfoSystem` zur Verwaltung von Geo-Informationen entworfen werden. Das können Städte sein, oder Namen von Bergen oder Tiefseegräben oder auch bestimmte Sehenswürdigkeiten.

a) Zuerst soll eine Klasse `GeoLocation` zur Repräsentation eines einzelnen Ortes implementiert werden.

Jeder dieser Orte wird durch folgende Attribute charakterisiert:

- Regionalcode = Eine Zahl zwischen 1 und 6 mit 1 (Westeuropa und Amerika), 2 (Osteuropa), 3 (Afrika und Mittlerer Osten), 4 (Zentralasien), 5 (Asien und Pazifik), 6 (Vietnam)
- Breitengrad = Gradzahl als Floatingpoint-Zahl:
  - Vorzeichen + bedeutet: nördliche Breite
  - Vorzeichen - bedeutet: südliche Breite
- Längengrad = Gradzahl als Floatingpoint-Zahl:
  - Vorzeichen + bedeutet: östliche Länge
  - Vorzeichen - bedeutet: westliche Länge
- Objektart = Ein Buchstabe aus der Menge {A, P, V, L, U, R, T, H, S} mit
  - A (Administrative Region) = Administrative Gruppenbezeichnung (Kreisname)
  - P (Populated Place) = Siedlungsname (Stadt, Dorf)
  - V (Vegetation) = Flurname (Wald, Feld)
  - L (Locality or Area) = Auffallende Gebäude und Plätze
  - U (Undersea) = Von Wasser bedeckt
  - R (Roads) = Strassenwege, Schienenwege
  - T (Hypsographic) = hochgelegene Objekte (Luft)
  - H (Hydrographic) = tiefgelegene Objekte (Wasser)
  - S (Spot feature) = Sehenswürdigkeit
- Ländercode = String mit Abkürzung des Landes: z.B.: GB (Great Britain), SP (Spain), GM (Germany)
- Name = String mit dem Namen des Ortes

Die Attribute sollen per Konstruktor übergeben werden.
Beispiel: `clz = GeoLocation(1, 51.18, 10.34, 'P', "GM", "Clausthal")`.
Für jedes der Attribute soll es eine `get`-Methode zur Abfrage des Attributs und eine `set`-Methode zum Ändern eines Attributs geben (Also z.B.: `getCountryCode()`, `setCountryCode("US")`).

b) Als nächstes soll die Klasse `GeoInfoSystem` zur Verwaltung einer Menge von `GeoLocation`-Objekten implementiert werden.
Ein `GeoInfoSystem`-Objekt besteht aus einer Liste von `GeoLocation`-Objekten sowie einer Karte (als Bilddatei).
Im Konstruktor wird einem `GeoInfoSystem`-Objekt diese Bilddatei, sowie deren Koordinaten in Gradzahlen übergeben:
Für eine Weltkarte müßte der Konstruktor dementsprechend mit den Werten
`GeoInfoSystem( "welt.jpg", (-180, 180), (-90, 90) )` aufgerufen werden.
Außerdem besitzt ein `GeoInfoSystem`-Objekt eine Funktion `readGeoLocations("Dateiname")` mit der eine Datei mit Geo-Daten eingelesen werden kann.
Das genaue Format der Datei entnehmen Sie bitte dem Link auf der Vorlesungs-Homepage. Zum Einlesen werden lediglich die Spalten 1, 4, 5, 10, 13 und 24 benötigt. Achten Sie beim Einlesen darauf, dass die Daten durch ein Tab (entpricht dem character `'\t'`) getrennt sind und nicht durch ein Leerzeichen, wie es auf dem vorletzten Übungszettel bei der Datei kant.txt der Fall war. Übergeben Sie beim Aufteilen einer Zeile in einzelne Strings einfach der `split`-Methode das `'\t'` als Trennzeichen.

Außerdem besitzt die Klasse eine Funktion
`getLocations((from_breite, to_breite), (from_laenge, to_laenge))`, die eine Liste mit allen `GeoLocation`-Objekten zurückliefert, die in dem entsprechenden Landstrich liegen. Zum Anzeigen die Orte besitzt die Klasse eine Funktion
`showLocations((from_breite, to_breite), (from_laenge, to_laenge))`. Diese Funktion liefert eine Kopie der Karte (als Image-Objekt) zurück, in der alle Orte im entsprechenden Landstrich markiert sind.

Schreiben Sie zusätzlich noch eine für die `showLocations`-Funktion nützliche Funktion
`getPixelLocation( geoLocationObjekt )`, die für ein gegebenes `geoLocation-Objekt` dessen Koordinaten (gegeben als Längen- und Breitengrad) in eine Pixelkoordinate auf der Karte umrechnet.

c) Testen Sie die Funktion, indem Sie auf einer Weltkarte (zu finden auf der Vorlesungs-Homepage) alle Orte aus der Datei Russia.txt einzeichnen, die sich nördlich des (nördlichen) Polarkreises befinden.

d) Erstellen Sie mit Doxygen eine Dokumentation der Klassen. Ein entsprechendes Config-File steh auf der Homepage zum Download bereit.

Hinweis: Das Geoinformationssystem wird später im Kapitel „Suchen und Sortieren“ noch gebraucht werden.