

Sommersemester 2010

## Übungen zu Informatik II - Blatt 2

Abgabe in der Übung am 20. 04 / 21. 04. 2010

Bitte beachten Sie, dass die Programmieraufgaben von ihnen in der Übung vorgeführt und erklärt werden müssen. Zusätzlich senden Sie die Lösung, unter Angabe ihres Namens, an **dm@tu-clausthal.de**.

### Aufgabe 1 (Typsysteme und Ausdruck-Bäume, 2 Punkte)

Zeichnen Sie einen Ausdruck-Baum (Folie 27 im Kapitel Typsysteme) inklusive der notwendigen Typ-Konvertierungen für den Ausdruck  $( 'c'+1 ) == ( 'f'+1.0 )$  ( $'c'$  und  $'f'$  sind vom Typ `"char"`) gemäß der Promotion-Hierarchie (Folie 25 im Kapitel Typsysteme).

#### Hinweise:

Die Knoten bei einem Ausdruck-Baum können Operanden wie 1 und 2 oder Operatoren wie + und \* sein. Operanden sind Blattknoten. Operatoren enthalten Verweise zu ihren Operanden. (Alle diese Operatoren sind binär, haben also genau zwei Operanden.) Es ist darauf zu achten, dass die Operatoren nur Operanden gleichen Types bearbeiten können.

Die Promotion-Hierarchie ist nach dem C++-Standard wie folgt definiert:

`bool → char → int → unsigned int → long int → float → double → long double`

In einem Ausdruck werden die Blätter mit den Typen der Operanden annotiert. Danach werden die Typen im Baum nach oben propagiert, wobei eine geeignete Promotion eingefügt werden muß, falls die beiden Operanden eines Operators nicht vom selben Typ sind.

### Aufgabe 2 (Klassen, 5 Punkte)

Ihre Aufgabe besteht darin, eine Klasse `Rectangle` zu erstellen.

- Erstellen Sie einen Konstruktor, welcher ein Rechteck aus den übergebenen Werten `x_min`, `y_min`, `x_max` und `y_max` erzeugt (siehe Abbildung1).
- Des Weiteren sollen für Rechtecke die Operatoren `*` und `+` definiert werden. Hierfür müssen Sie in ihrer Klassen die Funktionen `__mul__` für den Operator `*` und die Funktion `__add__` für den Operator `+` definieren. Der "neue" Operator `*` soll das Rechteck um den übergebenen Wert skalieren. Hier ist eine einfache Multiplikation der `x_min`, `y_min`, `x_max` und `y_max`-Werte des Rechteckes mit dem übergebenen Wert ausreichend. Der Operator `+` soll die Vereinigung aus zwei Rechtecken berechnen (siehe Abbildung2).
- Um die aktuellen Koordinaten des Rechteckes ausgeben zu können, muss die Funktion `__repr__` definiert werden. Der Rückgabewert muss der String sein, welcher ausgeben werden soll, falls `print RECHTECK` aufgerufen wird.

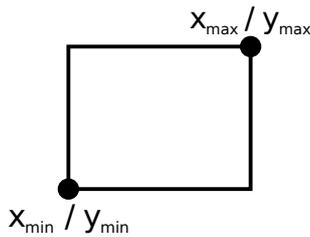


Abbildung 1: Begrenzung eines Rechteckes

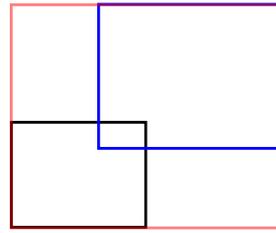


Abbildung 2: Vereinigung zweier Rechtecke

### Aufgabe 3 (Dictionaries, Strings, Listen, Umgang mit Dateien, 6 Punkte)

Die Osterzeit ist die Zeit der Geschenke. Nun gibt es gute Geschenke, schlechte Geschenke und Bücher. Von der letzten Sorte bekommt man meist mehr, als man lesen kann oder will. Peinlich wird es aber, wenn der Schenkende dann am Telefon nachfragt: „Du hast doch sicher schon das Buch „Kritik der reinen Vernunft“ gelesen, das ich dir letzte Weihnachten geschenkt habe und über das du dich so sehr gefreut hast. Was hältst du denn eigentlich von Kants Meinung zu Leibnitz?“.

Der Nichtinformatiker müsste an dieser Stelle verschämt zugeben, das Buch nie gelesen zu haben, während der Informatiker lässig seinen Python-Interpreter anwirft und sich einen Index des Buches erstellen lässt. Darin kann er dann blitzschnell nachschlagen, an welchen Stellen des Buches sich Kant über Leibnitz äussert und somit weiterhin als belesen gelten.

In dieser Aufgabe soll ein Programm zum Indizieren eines Textes erstellt werden. Das Programm erhält eine Textdatei per Kommandozeile. Dann soll für jedes Wort des Textes gespeichert werden, in welcher Zeile es vorkommt. Beispiel:

```
(‘ab’, [2101, 4059, 5933, 6451, 10647, 10835, 12415, 15038, 16016, 17878, 19758])
(‘abaenderung’, [6454])
(‘abaenderungen’, [1022, 19833])
(‘abbruch’, [5149, 7715, 14154, 14254, 14455, 15045, 17938, 18978, 19866, 19879, 21562, 21653])
...
(‘zwischenzustandes’, [7037])
(‘zwist’, [12564, 13431])
(‘zwoelf’, [1509, 1513])
```

Verwenden Sie dazu ein Dictionary.

Das zeilenweise Einlesen einer Datei in einen String wird bereits dem Framework zur Verfügung gestellt (den Code hierfür finden Sie auf der Vorlesungs-Homepage).

Jede eingelesene Zeile muss in Einzelwörter unterteilt werden. Python bietet hierfür die `split()`-Methode. Leider bleiben nach dem Aufspalten mit der `split()`-Methode die Satzzeichen erhalten. So wird die Zeile *„machen soll, einhellig zu machen: so!“* in die Teilstrings *„machen“*, *„soll,“*, *„einhellig“*, *„zu“*, *„machen:“* und *„so!“* aufgespalten. Beim einfachen Einfügen in den Index führt das dazu, dass *„machen“* und *„machen:“* als zwei unterschiedliche Schlüssel gespeichert würden. Dies wird mit Hilfe der Funktion `rem_sc(String)` schon für Sie im Framework korrigiert.

Die Ausgabe des Index soll, wie im Beispiel oben, in alphabetischer Reihenfolge erfolgen. Um die Sortierung zu vereinfachen, wurden in der Funktion `rem_sc(String)` alle Buchstaben in Kleinbuchsta-

ben umgewandelt. Zum Testen des Programms steht die Datei `kant.txt` auf der Vorlesungs-Homepage bereit.

Tips: Sie benötigen die Listen-Methode `sort()` und die Methode `items()` für Dictionaries. Die genauen Funktionsweisen dieser Methoden werden in der *Python Library Reference* auf [www.python.org](http://www.python.org) erklärt.