

Leibniz Universität Hannover
Fachgebiet Mensch-Computer-Interaktion

Programmieren 1
Dozent: Prof. Dr. Michael Rohs

Übungstest C-Teil
21./22. Dezember 2015

Dies ist ein **40**-minütiger Übungstest über den C-Teil von Programmieren I. Es sind keine zusätzlichen Hilfen erlaubt. Schalten Sie bitte Ihr Mobiltelefon aus. Für Notizen können Sie diesen Zettel verwenden. Notizen auf diesem Zettel werden nicht mitbewertet.

Ablauf:

1. Bearbeiten Sie die Aufgaben dieses Tests.
2. Melden Sie sich, wenn Sie fertig sind.
3. Falls nach Bearbeitung der Aufgaben noch Zeit ist, nehmen Sie bitte an dieser Umfrage teil:
http://tiny.cc/progl_ctest_umfrage.
4. Verlassen Sie den Raum bitte nicht vor dem Ende der Prüfung.

Hinweise:

- Bitte laden Sie sich die Aufgaben von http://tiny.cc/progl_ctest herunter. Entpacken Sie die zip-Datei auf dem Desktop.
- Sie kommen in das Prüfungsverzeichnis, indem Sie ein Terminal öffnen und `cd Desktop/CTest` eingeben.
- Mit `ls` können Sie die Dateien im Verzeichnis auflisten.
- Die Dokumentation der `progl`lib findet sich unter <http://hci.uni-hannover.de/files/progl1lib/files.html>.
- Als Tools dürfen Sie den Texteditor `KWrite`, `make`, sowie den C-Compiler (`gcc`) verwenden.
- Es empfiehlt sich, Umlaute im Quelltext zu vermeiden.
- Ihre Lösungen müssen mit den jeweiligen Testfällen funktionieren, dürfen aber nicht speziell darauf zugeschnitten sein.

1. bounding_box_shape

Implementieren Sie die Funktion `Rectangle bounding_box_shape(Shape *s)`. Diese bekommt einen Zeiger auf eine Shape übergeben und soll das kleinste achsenparallele Rechteck berechnen, das die Shape enthält.

Verwenden Sie `bounding_box.c`. Anweisungen zum Kompilieren und Ausführen finden Sie dort.

2. scale_shape

Implementieren Sie die Funktion `void scale_shape(Shape *s, double factor)`. Diese bekommt einen Zeiger auf eine Shape übergeben. Die Funktion skaliert die Shape mit `factor`.

Verwenden Sie `bounding_box.c`. Anweisungen zum Kompilieren und Ausführen finden Sie dort.

3. bounding_box_rectangles

Implementieren Sie die Funktion `Rectangle bounding_box_rectangles(Rectangle a, Rectangle b)`. Diese bekommt zwei Rectangles übergeben und soll das kleinste achsenparallele Rechteck berechnen, das beide Rectangles enthält.

Verwenden Sie `bounding_box.c`. Anweisungen zum Kompilieren und Ausführen finden Sie dort.

4. bounding_box_list

Implementieren Sie die Funktion `Rectangle bounding_box_list(List list)`. Diese bekommt eine Liste übergeben, die als Elemente Zeiger auf Shapes enthält. Die Funktion soll das kleinste achsenparallele Rechteck berechnen, das alle Shapes der Liste enthält.

Verwenden Sie `bounding_box.c`. Anweisungen zum Kompilieren und Ausführen finden Sie dort.