Lehrstuhl für Algorith.
und Datenstrukturen
Prof. Dr. Hannah Bast
Robin Schirrmeister

Programmieren in C++ (ESE) SS 2013

http://ad-wiki.informatik.uni-freiburg.de/teaching

UNI FREIBURG

Übungsblatt 11

Abgabe bis Montag, den 15. Juli um 12:00 Uhr

Aufgabe 1 (20 Punkte)

Füllen Sie den Evaluationsbogen zur Veranstaltung aus unter dem auf dem Wiki angegebenen Link.

Nehmen Sie sich bitte Zeit für das Ausfüllen, vor allem für die Freitextfelder. Schreiben Sie zur Bestätigung, dass Sie den Bogen abgegeben haben, einen Satz in Ihre *erfahrungen.txt* für dieses Übungsblatt.

Aufgabe 2 (20 Punkte)

Entscheiden Sie sich für eines der in der Vorlesung vorgestellten und auf dem Wiki näher spezifizierten Projekte. Sie müssen ihr Projekt **alleine**, ohne Pair Programming, programmieren. Führen Sie für ihr Projekt für dieses Blatt **eine** der unten angegebenen Möglichkeiten durch.

Wenn Sie Ihr Projekt zu einem selbstgewählten Thema machen möchten, wenden Sie sich bitte so früh wie möglich an Ihren Tutor, allerspätestens aber bis Donnerstag, den 11. Juli um 14 Uhr. Erläutern Sie dabei Ihr Projekt, kurz und prägnant, aber trotzdem so, dass Art und Umfang ersichtlich werden. Warten Sie dann die Rückmeldung von Ihrem Tutor ab, bevor Sie mit dem Schreiben der Header-Dateien beginnen.

Möglichkeit 1 - Header-Files

Schreiben Sie die möglichst vollständigen *.h* Dateien von allen Klassen, die Sie am Ende zur Realisierung des Projektes benötigen werden, samt möglichst allen Methoden und Membervariablen. Gehen Sie dabei davon aus, dass in der *main* Funktion am Ende nur sehr wenig Code steht, die gesamte Funktionalität soll durch die Klassen realisiert werden.

Ihre *.h* Dateien sollten bereits dokumentiert sein (so weit wie in diesem Stadium möglich), kompilieren und *checkstyle* ohne Fehler passieren. Schreiben Sie dazu eine *...Main.cpp*, die alle Ihre *.h* Dateien inkludiert und sonst nichts macht.

Tests brauchen Sie in dieser Phase noch keine zu schreiben. Können Sie aber, wenn Ihnen das hilft (indem die Tests beispielhafte Deklarationen Ihrer Objekte und Aufrufe Ihrer Methoden enthalten, das kann manchmal ganz erhellend sein).

[bitte umblättern]

Möglichkeit 2 - Teil-Funktionalität per TDD

Alternativ können Sie auch einen sinnvoll testbaren Teil des Programms mit vollständigen Tests per Test-Driven-Development implementieren.

So sollten Sie für das Snake-Projekt bereits die Bewegung der Schlagen innerhalb des Spielfeldes mit vollständigen Tests implementieren. Inklusive des richtigen Behandelns einer Kollision zwischen den Schlagen. Sie müssen noch keine zufälligen, essbaren, Symbole einbauen, nichts auf der Konsole zeichnen und keine Benutzereingaben behandeln.

Bei dem Virensacanner sollten Sie schonmal reine *Hex-Patterns* sowie Patterns mit *?* erkennen und dies mit entsprechenden Tests sicherstellen.

Abgabe

Laden Sie wie gehabt alle Code-Dateien, das Makefile und Ihre *erfahrungen.txt* in unser SVN hoch, in einem neuen Unterverzeichnis *uebungsblatt-11*, und stellen Sie sicher, dass nach Ihrem letzten commit auf Jenkins alles durchläuft.