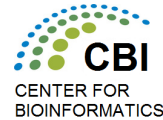




Chair for Clinical Bioinformatics
Professor Dr. Andreas Keller
Dr. Christina Backes
Dipl.-Ing. Thomas Großmann
B. Sc. Yvonne Saara Gladbach



Bioinformatik 1 - Übungsblatt 1

Abgabe bis Dienstag 04.11.2014 um 11:59

Vorname, Nachname:

Matrikelnummer:

Anmerkungen :

Die Übungen dürfen in Gruppen bestehend aus max. 2 Personen abgegeben werden. Der Quellcode ist als Quellcodepaket abzugeben, muss auf den Cip-Pool Rechnern kompilieren, gut dokumentiert und getestet sein (Testfälle sind mit abzugeben!). Schicken Sie Ihre Abgabe als tar.gz Paket per Mail an Ihre Tutorin Yvonne Gladbach (yvonne.gladbach@ccb.uni-saarland.de) mit Betreff: "Übung X, Name Y, Name Z". Das Paket darf keine temporären oder binären Dateien enthalten. Beim Entpacken des Paketes soll ein übergeordnetes Verzeichnis erstellt werden, welches den Quellcode enthält. Der Quellcode muss durch ein mitgeliefertes Makefile kompilierbar sein.

Aufgabe 1 (4P. + 6 P.):

- (a) Informieren Sie sich über die Speicherverwaltung in C++, insbesondere über den Unterschied zu Java und beantworten Sie folgende Fragen: Wie unterscheidet sich die Allokation von Speicher auf dem Stack von der auf dem Heap? Was ist bei der jeweiligen Technik zu beachten? Gehen Sie insbesondere auf Sichtbarkeit, Lebensdauer und Freigabe ein.
- (b) Informieren Sie sich über die Bestandteile einer Klasse in C++ und geben Sie zu den einzelnen Bestandteilen eine Beschreibung ihrer Funktion an:
- Konstruktor
 - Copy-Konstruktor
 - Destruktor
 - Zuweisungsoperatoren
 - Methoden
 - virtuelle Methoden
 - Membervariablen
 - Zugriffsebenen

Aufgabe 2 (40 P.):

Implementieren Sie eine C++-Klasse `Sequence`. Erstellen Sie ein vollständiges Klasseninterface mit Konstruktoren (Default-Konstruktor, Copy-Konstruktor, detaillierter Konstruktor, dem die spätere Sequenz als `std::string seq` übergeben wird), Destruktor und Standardoperatoren (Zuweisungsoperator, Gleichheitsoperator, Ungleichheitsoperator und Stream-Input-Output-Operatoren). Leiten Sie davon eine Klasse `NucleicAcid` und eine Klasse `Protein` ab. Für diese Klassen sollen folgende Funktionen implementiert werden:

- `bool compare(const Sequence& seq, unsigned int i) const` testet, ob die Sequenz ab der Stelle `i` gleich der übergebenen Sequenz `seq` ist. Beginnen Sie beim Zählen der Positionen in der Sequenz bei 0.
- `unsigned int overlap(const Sequence& seq) const` berechnet die maximale Überlappung des Endes der Sequenz mit dem Anfang der übergebenen Sequenz `seq`.
- `bool isValid() const` testet, ob es sich um eine korrekte Protein- oder Nukleinsäuresequenz handelt (*Hinweis: Verwenden Sie das Konzept einer abstrakten, virtuellen Methode*).

Verwenden Sie `std::string sequence_` zur internen Speicherung der Sequenz. Schreiben Sie für jeden Konstruktor, jeden Operator und jede Methode einen Testfall (z.B. in der Methode `main`).