

3. Praktikum "Algorithmen und Programmierung II"

SS 2014

Vorbemerkungen:

Die vorgefertigten Klassen zu den Praktikumsaufgaben finden Sie wieder auf der AP2-Seite unter www.gm.fh-koeln.de/ehses/ap_in in der Datei *vorlage3.zip*. Bei den Aufgaben brauchen Sie nur die jeweils angegebenen Dateien zu verändern.

Grundsätzlich: Wenn in einer Klasse (noch) nicht benutzte private Methoden stehen, so hat das seinen Sinn: es ist hilfreich sie zu verwenden!

Die Aufteilung der Aufgaben gibt eine sinnvolle Vorgehensweise an. Am Ende haben Sie ein Eclipse-Projekt, dass Sie bei der Abgabe vorzeigen!

Aufgabe 1.

Die Klasse `FinancialHistory` ist erste Beispielklasse in dem legendären „Smalltalk-80 The Language and its Implementation“ (blue book) aus dem Jahre

Betrachten Sie zunächst nur die Klassen `FinancialHistory` und `FinancialHistoryTest`. An einigen Stelle ist in `FinancialHistory` per `TODO` angegeben, was zu tun ist. An anderen Stellen geht es darum, die durch den Test gefundenen Fehler sinnvoll zu korrigieren. Wenn man auch die Meldungen von `JUnit` liest, sollte es einfach sein, die Fehlerursachen zu erkennen.

Wenn der Test erfolgreich läuft, können Sie die Anwendung mit der Klasse `Application` starten. In *Anleitung.pdf* finden Sie dazu ein paar Hinweise. Zugegeben die Oberfläche ist kein Beispiel für ein gelungenes Design. Dafür ist sie halt sehr einfach. Im Programm ist eingestellt, dass die deutsche Regel für Dezimalzahlen verwendet wird.

Aufgabe 2.

Zu ändernde Klasse: `LinearMap`. Zu beachten sind das Interface: `IMap` und die Klassen `BasicLinearMapTest`, `MapRemoveTest`, `MapExceptionTest`.

Hinweis: Ein Java-Interface deklariert abstrakte Methoden, die von der implementierenden Klasse definiert werden müssen! Die Kommentaren spezifizieren das Verhalten dieser Methoden (nicht die Implementierung).

Erweitern Sie `LinearMap` so, dass die Klasse zur Schnittstelle `IMap` passt und geeignet ist, Objekte unter einem Suchbegriff abzuspeichern und wiederzufinden (Testen)! Als Suchbegriff können grundsätzlich beliebige Objekte verwendet werden (meist nimmt man Strings). Die Fehler sind nicht besonders markiert. Was zu tun ist, ergibt sich aus dem Test und insbesondere auch aus den Kommentaren in dem Interface `IMap`! Die Tests sind auf drei Klassen verteilt um einmal die nötigen Fehlermeldungen zu testen, um die grundlegenden Operationen zu testen und um die Löschoption genauer zu testen.

Implementierungshinweise:

Die Klasse `LinearMap` soll genau das tun, was in der Schnittstelle angegeben ist, nicht weniger aber auch nicht mehr! Insbesondere soll die Klasse keinerlei Ein- oder Ausgabe enthalten!

Das Feld **data** soll keine „Lücken“ enthalten. Wenn z.B. das Feld 4 Werte fasst und 2 tatsächlich benutzt, sind die Elemente **data[0]** und **data[1]** mit Daten belegt und **data[2]** und **data[3]** sind auf **null** gesetzt.

Verwenden Sie unbedingt **int indexOf(K key)** ! Dieses sollte den Arrayindex zu dem gesuchten Schlüssel zurückgeben. Gleichzeitig prüft es, ob **null** übergeben wurde. Jegliche Suche in dem Feld **data** kann auf **indexOf** zurückgeführt werden..**Beachten Sie auch die Hinweise in der Vorlesung!**

Wenn Sie `indexOf` richtig verwenden, brauchen Sie Exceptions nicht zu beachten!

Lernziele: Genaue Implementierung einer Schnittstelle, systematisches Testen, lineare Suche, Umgang mit Java-Dokumentation.

Aufgabe 3.

Nachdem **LinearMap** den Test besteht, sollen Sie in **FinancialHistory** folgendes machen: Ersetzen Sie in den Variablendeklarationen **java.util.Map** durch **IMap** und **java.util.HashMap** durch **LinearMap**. Wenn Sie bisher alles richtig gemacht haben, sollten sowohl **FinancialHistoryTest** wie auch **Application** korrekt laufen.

Aufgabe 4.

Verteilen Sie die Klassen auf unterschiedliche Pakete: **finance** für Klassen, die anwendungsbezogen sind, **util** für die Klassen der Datenhaltung und **gui** für Klassen, die die graphische Darstellung unterstützen.

Sie sollen erklären können, welche Java-Anweisungen mit dem Paketkonzept zu tun haben und was „Defaultsichtbarkeit“ bedeutet!

Lernziele: Verständnis der Klassenstruktur. Paketkonzept.