



Software Engineering Übung 1

Programmverstehen, Nachdokumentation, Ablaufkonstrukte

1 Information

1.1 Daten

- Ausgabe Di 24.10.2006
- Abgabe So 5.11.2006
- Besprechung Fr 10.11.2006

1.2 Formales

Abgegeben wird ein Zip-File mit den Lösungen als PDF und dem Source Code in einem Unterverzeichnis. Abgabe an cramer@ifi.unizh.ch und smeier@ifi.unizh.ch. Verspätete Abgaben werden nicht bewertet!

1.3 Gruppen

Die Übung ist in 3er Gruppen zu lösen. Falls die Aufgaben aufgeteilt wurden, muss klar ersichtlich sein, wer welchen Teil bearbeitet hat. Alle Gruppenmitglieder müssen über alle Teile Auskunft geben können.

1.4 Hinweise

Der Source Code kann von der Übungsseite als Zip-File heruntergeladen werden. Zur Ausführung des Codes wird Java 1.4 oder höher benötigt. Zur Ausführung des Source Codes benötigen Sie zudem die im Paket enthaltene Bibliothek *Xanthine.jar*.

2 Aufgabenstellung

2.1 Teil A (2 Punkte)

Finden Sie im Code alle annehmenden Schleifen. Beurteilen Sie, ob diese Schleifen zu einem Problem führen könnten. Schreiben Sie zu jeder annehmenden Schleife eine abweisende Schleife, welche die gleiche Funktionalität bietet.

2.2 Teil B (2 Punkte)

Die Methode `readAddresses()` der Klasse `AddressList` enthält ein `break`-Statement. Diskutieren Sie, inwiefern dies gegen die Regel, geschlossene Ablaufkonstrukte zu verwenden, verstößt.

2.3 Teil C (4 Punkte)

Im Code gibt es eine rekursive Funktion. Erklären Sie, was diese Funktion macht. Programmieren Sie eine dazu äquivalente iterative Funktion. Diskutieren Sie, ob die iterative oder die rekursive Funktion an dieser Stelle besser geeignet ist. Erklären Sie allgemein, wann rekursive Funktionen Vorteile gegenüber iterativen Funktionen haben.

2.4 Teil D (6 Punkte)

Versuchen Sie herauszufinden wie die verschiedenen Klassen zusammenspielen und was sie machen. Dokumentieren Sie das Programm mittels Klassen- und Methodenkommentaren, sowie die Methodenkörper (wo es nötig ist) im Source Code nach. Achten Sie darauf dass Ihre Kommentare die Funktionalität der Klassen und Methoden erklären. Softwareverträge (Vor- und Nachbedingungen, Invariante, Verpflichtung) müssen Sie nicht speziell kommentieren, wie dies in der mitgelieferten Styleguide gefordert wird (diese werden Gegenstand einer späteren Übung sein). Schreiben Sie Ihre Kommentare nach den Richtlinien im beiliegenden Java Styleguide. Benennen Sie ausserdem die Klassen und Methoden so um, dass Ihre Aufgabe aus dem Klassen- bzw. Methodennamen ersichtlich wird.

2.5 Teil E (4 Punkte)

Analysieren Sie, wo der gegebene Code problematisch ist. Nutzen Sie dazu Kapitel 6 des Skriptes und Kapitel 2 des Java Styleguide. Gehen Sie dabei auf Probleme im Code selbst ein und nicht auf Probleme, welche schon in Teilaufgabe D betrachtet wurden. Geben Sie zu jedem Punkt, den Sie als problematisch erachten, an, warum dieser ein Problem darstellt und wie der Code verbessert werden kann.

2.6 Teil F (2 Punkte)

Mit der vorliegenden Bibliothek soll eine kleine Software entwickelt werden, welche folgenden Anforderungen genügt: Emailadressen von verschiedene Personen sind in zwei verschiedenen Listen gespeichert (`liste1.txt` und `liste2.txt`). Diese sollen zu einer Liste (`liste.txt`) zusammengefügt werden. Einige Personen sind in beiden Listen eingetragen, die zusammengefügte Liste soll aber keine doppelten Einträge enthalten. Emailadressen welche nicht dem definierten Format entsprechen, sollen aus der Liste gelöscht werden. Das richtige Format ist in der Klasse `EmailAddress` definiert. Es wäre ausserdem wünschenswert, wenn die neue Liste nach Namen sortiert wird. Programmieren und dokumentieren Sie (unter Benutzung der gegebenen Klassen) ein kleines Demonstrationsprogramm, welches diese Anforderungen erfüllt.