

Informatik II (SS 2003)

1. Übungsblatt

1. Aufgabe

Auf der Übungswebseite finden Sie den Quelltext eines Programms, mit dem Internettarife verglichen werden können. Ihre Aufgabe ist die Programmierung der folgenden Erweiterungen des Programms. Abzugeben sind per Email alle Quelltexte und in Papierform die Quelltexte der veränderten und neuen Klassen sowie eine Dokumentation, in der Sie die vorgenommenen Änderungen begründen. (Außer den üblichen Kommentaren im Quelltext ist keine weitere Dokumentation erforderlich.)

- (a) Anzeige der mittleren monatlichen Kosten** 5 Punkte
Unter dem Label für die Anzahl der Tarife soll ein weiteres Label das arithmetische Mittel der monatlichen Kosten aller Tarif anzeigen. Erstellen sie dazu nach dem Vorbild von `AnzahlLabel` eine neue Klasse `MittelLabel`, die die Anzeige bei jeder Veränderung der Liste aktualisiert.
- (b) Neue Tarifart „gestaffelter Zeittarif“** 7 Punkte
Neben (einfachen) Zeittarifen und Volumentarifen soll das Programm auch gestaffelte Zeittarife verwalten. Bei gestaffelten Zeittarifen sinkt der Stundenpreis, wenn die Online-Zeit eine Staffलगrenze übersteigt. Zu jedem gestaffelten Zeittarif soll der Nutzer eingeben: Name des Tarifs, monatliche Fixkosten (in Euro), Staffलगrenze (in Stunden), Kosten pro Stunde bis Staffलगrenze (in Euro), Kosten pro Stunde oberhalb Staffलगrenze (in Euro). Das Einführen der neuen Tarifart darf keine Änderung der Klasse `MittelLabel` erfordern.
- (c) Sortieren nach monatlichen Kosten** 6 Punkte
Mit einem Button im Hauptfenster soll der Nutzer veranlassen können, dass die Tarifliste aufsteigend nach monatlichen Kosten sortiert wird. Erstellen Sie dazu eine Klasse `SortListModel`. Diese Klasse soll Unterklasse der Klasse `DefaultListModel` sein und zum Sortieren eine Methode `void sort(Comparator c)` enthalten. Verwenden Sie ein geeignetes Sortierverfahren, zum Beispiel Bubble Sort. Erstellen Sie eine neue Klasse `TarifComparator`, welche die Schnittstelle `Comparator` implementiert. Übergeben Sie ein Objekt dieser Klasse der Methode `sort` als Parameter, und verwenden Sie es in dieser Methode für Vergleiche von Tarifen.
- (d) Begründung des Comparator** 1 Punkt
In Teilaufgabe (c) soll der Methode `SortListModel.sort` ein `TarifComparator`-Objekt übergeben werden, um direkte Vergleiche von Monatskosten (etwa mit `t1.getMonatsKosten() < t2.getMonatsKosten()`) in dieser Methode zu vermeiden. Nennen Sie mindestens einen Vorteil dieses Vorgehens!
- (e) Menü** 3 Punkte
Alle Funktionen des Programms, die bisher durch Buttons ausgelöst werden, sollen zusätzlich in zwei Menüs *Nutzung* und *Tarif* auswählbar sein.

Hinweise

- Bitte beachten Sie die „Hinweise zu den Übungsblättern“ auf der Webseite zu Informatik 2. Dort finden Sie auch Links zu den im folgenden genannten Dokumenten.
- Folgen Sie soweit wie möglich den Java Code Conventions oder der Java Stilbibel.
- Alle benötigten Klassen aus der Java-Klassenbibliothek sind in der Java 2 Platform API Specification dokumentiert. Es ist beabsichtigt und gehört zu den Zielen dieser Aufgabe, dass nicht alle benötigten Klassen und Methoden in der Vorlesung oder Übung besprochen werden, und Sie deshalb diese Spezifikation verwenden müssen.

Abgabetermin: Übung der 18. Kalenderwoche 2003