Übungen zum Kurs Programmieren in Java

Aufgabe 1 Arbeiten mit der Java-API-Dokumentation

Suchen Sie die in Kapitel 8 vorgekommenen Klassen und ihre Methoden der Java-API[1]. Warum wird die Methode `add(Component)` (vgl. Skript) nicht als Methode von `java.awt.Frame` aufgeführt ?

Aufgabe 2 (P) Einfaches GUI-Programm

Kopieren Sie sich bitte das Beispielprogramm `/usr/proj/javakurs/Tag4/Minimal.skelett` in ein geeignetes Arbeitsverzeichnis unter dem Namen `Minimal.java` ! Versuchen Sie es zu übersetzen und auszuführen !

Nun soll die Funktionalität des Programms geändert werden. Beim Betätigen des Buttons soll keine Beendigung des Programms erfolgen, sondern ein weiterer Button am Bildschirm erscheinen. Auch bei dessen Betätigung soll ein Button am Bildschirm erscheinen. Die maximale Anzahl der Buttons soll auf neun beschränkt sein. Die Beschriftung der einzelnen Buttons soll anzeigen, um den wievielten es sich jeweils handelt ("der 1.te","der 2.te",..). Bitte ordnen Sie die Buttons mit Hilfe des LayoutManagers `GridLayout` an.

Aufgabe 3 (P) Einlesen aus Datei

Implementieren Sie ein Java-Programm, das den Inhalt einer Datei auf der Konsole ausgibt. Der Benutzer gibt den Namen der Datei als Argument beim Programmaufruf an.

Hinweis: Verwenden Sie die Klasse `FileReader`. `FileReader` erbt von `Reader` die Methode `int Reader.read()`, durch die einzelne Zeichen als `int` vom Eingabestrom gelesen werden können. Es liegt also nahe, in einer Schleife die Methode `read()` solange aufzurufen, bis das Ende der Datei erreicht ist. Da `read()` ein `int` liefert, ist ein Cast auf `char` notwendig. Beim Dateiende liefert `read()` den Wert `-1`.

Aufgabe 4 (P) Arbeiten mit `java.awt.List`

Kopieren Sie sich bitte das Beispielprogramm

`/usr/proj/javakurs/Tag4/MitarbeiterListfenster.skelett`

in ein geeignetes Arbeitsverzeichnis unter dem Namen `MitarbeiterListfenster.java`. Versuchen Sie es zu übersetzen und auszuführen. Fügen Sie dem Beispielprogramm Mitarbeiterfenster zwei neue Buttons hinzu . Beim Betätigen des einen Buttons soll ein neuer Eintrag in die Liste erfolgen. Der neue Eintrag kann durch den Benutzer eingegeben werden. Dazu ist ein Interaktionsobjekt der Klasse `TextField` (siehe auch API-Dokumentation) notwendig. Der andere Button soll den aktuell selektierten Eintrag aus der Liste löschen.

[1] siehe z.B. `http://medoc.informatik.tu-muenchen.de/Java/java11/api/packages.html`

Hinweis: Objekte von `TextField` können analog zu `java.awt.Button` instantiiert und angeordnet werden. Mittels der Methode `getText()` kann die aktuell im Textfeld angezeigte/eingegebene Zeichenreiche (Objekt vom Typ `String`) abgefragt werden. Es ist möglich, beide Buttons an die Methode `MitarbeiterListfenster.actionPerformed(ActionEvent e)` zu knüpfen. Zur Fallunterscheidung, welcher Button gedrückt wurde, ist die Methode `ActionEvent.getSource()` nützlich, die eine Referenz auf den auslösenden Button liefert.

Aufgabe 5 Spezifikation des GUI-Programms für Abteilungen

Überlegen Sie in Ihrer Tutorgruppe, inwieweit `MitarbeiterListfenster` aus Aufgabe 4 erweitert werden muß, um eine Abteilung zu verwalten. Versuchen Sie, Ihre Überlegungen geeignet auf Papier festzuhalten. Folgende Aspekte gilt es zu berücksichtigen:

a) Welche Daten werden auf dem Bildschirm angezeigt ?

b) Wie werden die Daten eingegeben ?

c) Wie reagiert das GUI-Programm auf Benutzereingaben ?

d) Aus welchen Klassen besteht das GUI-Programm ?

e) Wie werden die Daten persistent abgespeichert ?

Aufgabe 6 (P) Implementierung der GUI für Abteilungen

Kopieren Sie sich bitte die Dateien

`/usr/proj/javakurs/Tag4/Abteilungsfenster.skelett` und

`/usr/proj/javakurs/Tag4/Mitarbeiterfenster.java`,

sowie sämtliche Dateien von `/usr/proj/javakurs/Tag4/vonfirma` in ein geeignetes Arbeitsverzeichnis. Versuchen Sie die Dateien zu übersetzen und `Abteilungsfenster` auszuführen.

Vergleichen Sie die Ausführung mit den in Aufgabe 5 gemachten Überlegungen und ergänzen Sie bitte folgende noch fehlende Teile:

a) Zum Speichern und Einlesen von Abteilungen in bzw. aus Dateien müssen die Methoden `Abteilungsfenster.oeffnen()` und `Abteilungsfenster.speichern()` ergänzt werden.

b) Die Methode `Abteilungsfenster.getAbteilung()` muß implementiert werden, die aus dem Inhalt der Interaktionsobjekte ein neues Objekt der Klasse Abteilung erzeugt.

   Hinweis: Beachten Sie die Methode

   `Mitarbeiter Abteilungsfenster.getListMitarbeiter(int index)`.

c) Abhängig vom Zustand des Fensters (z.B. Listenselektion) ist es sinnvoll, manche Buttons und Menuitems für den Benutzer zu sperren. Suchen Sie dafür notwendige Methoden der API und ergänzen Sie die Klasse `Abteilung` entsprechend.