

Bitte packt diesmal alle Dateien Eurer Lösung in eine JAR-Datei ein. In dieser Datei sollen sowohl die Quelltexte als auch die kompilierten Klassen liegen, so dass er Eure Programme einfach mit `java -classpath <EureDatei>.jar <Klassenname>` ausführen kann.

In dieser Übung geht es darum, die Entwicklung von GUIs mit Swing zu trainieren. Dafür legen wir Euch den GUI-Editor von Netbeans ans Herz.

## 1 Tic-Tac-Toe, Teil 4 (3 Punkte)

Im vierten und letzten Teil der Tic-Tac-Toe-Aufgaben geht es darum, einen Client zu schreiben, der das Spielbrett in einem Fenster darstellt und Züge einfach per Mausklick annimmt. Verwende dazu z.B. ein 3x3-Array von `JButtons`, deren Beschriftungen die aktuelle Belegung des Feldes darstellen.

Man kann die Beschriftung eines `JButton`-Objekts mit der Methode `setText()` ändern. Der Button wird dann automatisch neu gezeichnet.

## 2 Formular (3+2 Punkte)

Schreibe ein GUI, das ein Formular mit Personaldaten einliest. Das Formular soll folgende Daten enthalten:

- Vor- und Nachname, Telefonnummer (Strings)
- Personalstatus (Professor, Wiss. Mitarbeiter, Sekretärin)
- Gruppe (Pinkal, Uszkoreit, Barry, Crocker)
- Promoviert? (ja/nein)

Wähle geeignete GUI-Komponenten zur Eingabe der einzelnen Daten aus und beschrifte sie mit Labels. Auf Buttonklick soll das Programm die Einträge auf der Konsole ausgeben.

Für die Bonusaufgabe erweitere Dein Programm so, dass es ein Objekt mit den Adressdaten auf die Festplatte schreiben und von der Festplatte einlesen kann. Du kannst Object-Streams verwenden, wenn Deine Klasse das Interface `java.io.Serializable` implementiert. Der Dateiname für die zu ladende oder speichernde Datei könnte z.B. mit einem `JFileChooser` eingelesen werden.

### 3 Dateiverwaltung (3+2 Punkte)

Schreibe ein Programm, das alle Dateien im aktuellen Verzeichnis und allen seinen Unterverzeichnissen darstellen kann.

Das Programm soll eine `JSplitPane` verwenden, die auf der linken Seite die Verzeichnisstruktur in einem `JTree` darstellt und auf der rechten Seite den Inhalt einer Datei in einem großen `JLabel`. Die inneren Knoten des Baums repräsentieren die Verzeichnisse; die Kinder eines Knotens sind die Dateien und Verzeichnisse, die in diesem Verzeichnis liegen. Die Blätter des Baumes entsprechen normalen Dateien, und wenn der Benutzer auf ein solches Blatt klickt, soll der Inhalt der Datei auf der rechten Seite (als Text) angezeigt werden.

Wie man `JTree` verwendet, ist im Netz ausführlich dokumentiert, z.B. auf der Seite <http://java.sun.com/docs/books/tutorial/uiswing/components/tree.html>. Alle nötigen Methoden zur Berechnung des Verzeichnisbaums findest Du in der Klasse `java.io.File`.

Für die Bonuspunkte statte die rechte Seite mit einer `JTabbedPane` aus, die zwei verschiedene Ansichten der Datei unterstützt: erstens als Text (wie in der normalen Aufgabe) und zweitens als Hexdump. Du kannst dazu Deinen Hexdump-Code aus der 8. Übung verwenden.