Übung 8 – Persistenz (Serialisierung)

In dieser Übungen sollen endlich die Kommandos *File->Save* und *File->Open* implementiert werden damit Grafiken auf der Disk gespeichert werden können.

Die Figuren sollen mit einem *ObjectInputStream* gelesen und einem *ObjectOutputStream* geschrieben werden. Das Interface *Figure* ist bereits von *java.io.Serializable* (Markerinterface) abgeleitet und daher für Serialisierung vorbereitet.

Schreiben

Als erstes müssen Sie den Benutzer fragen wie die neue Datei heissen soll. Dazu kann die Klasse *JFileChooser* (Swing) oder *FileDialog* (AWT) verwendet werden. Beim *JFileChooser* kann man direkt mit *getFile()* auf das ausgewählte File zugreifen, beim *FileDialog* muss man dieses File konstruieren mit

```
new File(dlg.getDirectory() + dlg.getFile())
```

wobei *getDirectory* und *getFile* Methoden von *FileDialog* sind.

Auf diesem File kann ein *FileOutputStream* und dann ein *ObjectOutputStream* aufgesetzt werden.

Um eine Grafik abzuspeichern
a) kann das Modell abgespeichert werden (dieses muss dann serialisierbar sein)
b) können die Figuren einzeln in einer Schleife geschrieben werden.

Nach dem Schreiben muss der Stream mit *close* geschlossen werden.

Lesen

Beim Lesen muss ebenfalls zuerst der Filenamen mit einem *JFileChooser* bzw. *FileDialog* abgefragt werden. Danach kann analog zum Schreiben ein *FileInputStream* und ein *ObjectInputStream* erzeugt werden.

Die Grafik muss danach so gelesen werden wie sie geschrieben wurde,
a) lesen (und setzen) des Modells
b) lesen der Figuren in einer Schleife die dann in das Modell eingefügt werden.

Die alten Figuren der Grafik sollen gelöscht werden.

Übung 8 – Persistenz (Serialisierung)

In dieser Übungen sollen endlich die Kommandos *File->Save* und *File->Open* implementiert werden damit Grafiken auf der Disk gespeichert werden können.

Die Figuren sollen mit einem *ObjectInputStream* gelesen und einem *ObjectOutputStream* geschrieben werden. Das Interface *Figure* ist bereits von *java.io.Serializable* (Markerinterface) abgeleitet und daher für Serialisierung vorbereitet.

Schreiben

Als erstes müssen Sie den Benutzer fragen wie die neue Datei heissen soll. Dazu kann die Klasse *JFileChooser* (Swing) oder *FileDialog* (AWT) verwendet werden. Beim *JFileChooser* kann man direkt mit *getFile()* auf das ausgewählte File zugreifen, beim *FileDialog* muss man dieses File konstruieren mit

```
new File(dlg.getDirectory() + dlg.getFile())
```

wobei *getDirectory* und *getFile* Methoden von *FileDialog* sind.

Auf diesem File kann ein *FileOutputStream* und dann ein *ObjectOutputStream* aufgesetzt werden.

Um eine Grafik abzuspeichern
a) kann das Modell abgespeichert werden (dieses muss dann serialisierbar sein)
b) können die Figuren einzeln in einer Schleife geschrieben werden.

Nach dem Schreiben muss der Stream mit *close* geschlossen werden.

Lesen

Beim Lesen muss ebenfalls zuerst der Filenamen mit einem *JFileChooser* bzw. *FileDialog* abgefragt werden. Danach kann analog zum Schreiben ein *FileInputStream* und ein *ObjectInputStream* erzeugt werden.

Die Grafik muss danach so gelesen werden wie sie geschrieben wurde,
a) lesen (und setzen) des Modells
b) lesen der Figuren in einer Schleife die dann in das Modell eingefügt werden.

Die alten Figuren der Grafik sollen gelöscht werden.