

## Übung 8: Save & Load: Speichern und Laden von Grafiken

In dieser Übung sollen 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 (diese Klassen haben Sie im Zusammenhang beim Cloning mit Serialisierung kennengelernt). Alle Objekte, die dabei serialisiert werden, müssen das Interface `java.io.Serializable` (Markerinterface) implementieren. Ihre Figuren erfüllen diese Bedingung, da das Interface `Figure` bereits von `Serializable` abgeleitet ist.

Ein `ObjectInputStream` bzw. `ObjectOutputStream` kann wie folgt erzeugt werden:

```
ObjectInputStream ois = new ObjectInputStream(new FileInputStream(...));  
ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(...));
```

### Schreiben

Als erstes muss der Benutzer gefragt werden, wie die neue Datei heissen soll. Ein entsprechender Dialog ist in der Klasse `JFileChooser` implementiert. Die Methode `getFile` erlaubt direkt auf das ausgewählte File zuzugreifen. Auf diesem File kann ein `FileOutputStream` und dann ein `ObjectOutputStream` aufgesetzt werden.

Um eine Grafik abzuspeichern, müssen alle im Modell enthaltenen Figuren separat nacheinander in den `ObjectOutputStream` geschrieben werden. Dass bei jeder Figur die registrierten Listener (und damit indirekt das Modell) mit gespeichert werden, lässt sich verhindern, wenn von jeder Figur vor dem Kopieren mit der Methode `clone` eine Kopie erzeugt wird.

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

### Lesen

Beim Lesen muss ebenfalls zuerst der Filename mit einem `JFileChooser` abgefragt werden. Danach kann analog zum Schreiben ein `FileInputStream` und ein `ObjectInputStream` erzeugt werden.

Danach können in einer Schleife die Figuren einzeln eingelesen und in das Modell eingefügt werden. Vor dem Einlesen der Figuren können die im Modell vorhandenen Figuren gelöscht, oder alternativ die Figuren aus der Datei zusätzlich in die Szene eingefügt werden.

Wenn die Figuren über den `ObjectInputStream` gelesen werden ist unklar, wann alle in der Datei gespeicherten Figuren ausgelesen worden sind. Das Ende des Stroms kann dazu beim Schreiben explizit z.B. mit einer "Figur" als null-Referenz markiert werden. Alternativ können auch einfach die Figuren gelesen werden, bis eine `EOFException` geworfen wird, oder es wird zusätzlich am Anfang die Anzahl der zu speichernden Figuren in der Datei gespeichert.

Vielleicht haben Sie im Modul OOP2 noch weitere Methoden kennengelernt wie eine Objektstruktur serialisiert werden kann (Stichwort: XML oder JSON). Sie können auch die dort gelernten Techniken umsetzen.