

## 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 müssen Sie den Benutzer fragen wie die neue Datei heissen soll. Dazu kann die Klasse `JFileChooser` verwendet werden. Beim `JFileChooser` kann man direkt mit `getFile()` auf das ausgewählte File zugreifen. Auf diesem File kann ein `FileOutputStream` und dann ein `ObjectOutputStream` aufgesetzt werden.

Um eine Grafik abzuspeichern iterieren Sie über die Figuren und speichern jede Figur separat ab. Damit nicht bei jeder Figur die registrierten Listeners (und damit indirekt das Modell) abgespeichert werden erzeugen Sie von jeder Figur vor dem Kopieren mit der Methode `clone` eine Kopie.

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 lesen Sie die Figuren in einer Schleife ein und fügen jede Figur in das Modell ein. Vor dem Einlesen der Figuren können sie die im Modell vorhandenen Figuren löschen. Sie können aber auch die Figuren aus der Datei zusätzlich in ihre Szene einfügen.

Wenn Sie die Figuren lesen wissen Sie nicht, wie viele Figuren abgespeichert worden sind. Sie können das Ende jedoch explizit markieren indem Sie z.B. als letzte "Figur" eine `null`-Referenz abspeichern. Alternativ können Sie auch einfach die Figuren lesen bis eine `EOFException` geworfen wird, oder sie speichern zusätzlich am Anfang die Anzahl der zu speichernden Figuren.

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

Abgabe: 21. November 2017