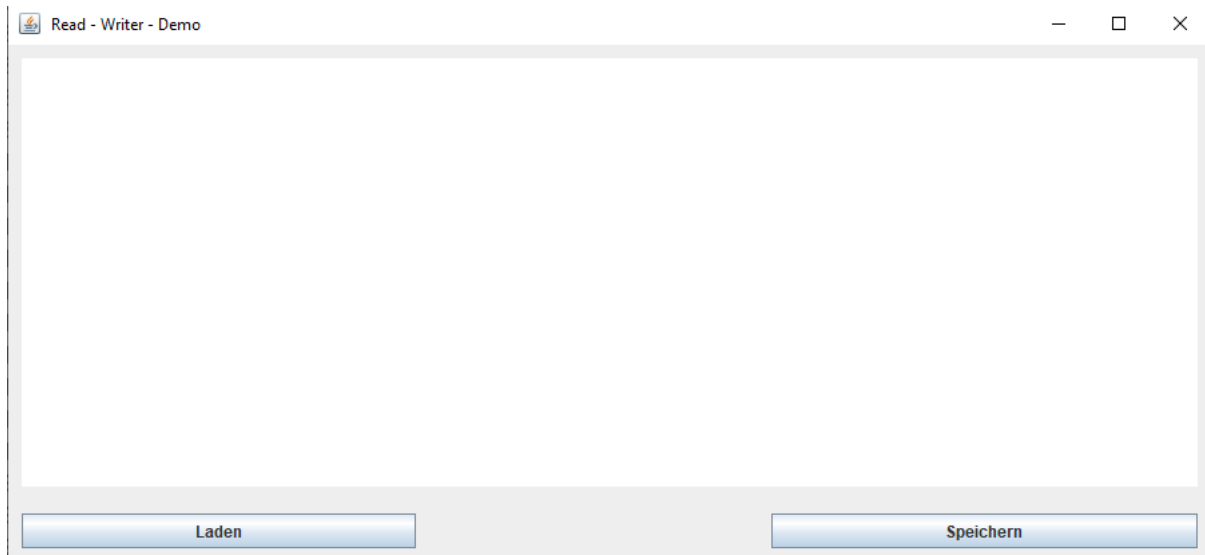


## Übung - Demo: Arbeiten mit Dateien

Im Rahmen dieser Übung bzw. Demo wollen wir uns mit dem Lesen und Schreiben von Dateien vertraut machen.

Wenn Sie die Vorlage öffnen erscheint bereits das folgende Frame



Das Frame besteht aus einem JTextArea zum Anzeigen und Schreiben von Text. Wenn der Button Laden betätigt wird, soll sich ein File-Dialog zur Auswahl der Datei öffnen. Genauso soll mit dem Button Speichern ein Dialog zum Abspeichern in eine Datei erscheinen. Nach dem Laden der Datei soll der Text im TextArea angezeigt werden. Beim Speichern soll der Text entsprechend in die Datei geschrieben werden

Dieses Verhalten werden wir wie gewohnt schrittweise realisieren.

### Aufgabe 1:

Beginnen Sie mit dem Lesen Text File aus einer vorgegebenen Datei Kurztest.txt. Öffnen Sie mit `eingabeDatei = new BufferedReader(new FileReader(dateiName));` den InputStream und lesen Sie die Datei Zeilenweise an und hängen Sie die Zeilen an das TextArea an. Schliessen des Streams am Ende nicht vergessen!

### Aufgabe 2:

Als nächstes sorgen wir dafür, dass der Text im TextArea in einer Datei abgespeichert wird. Dazu erzeugen wir den FileWriter mit dem Dateinamen der Zieldatei.

`ausgabeDatei = new PrintWriter(new FileWriter(dateiName, false));`

Dann kann mit `print()` der Text in die Datei geschrieben werden.

### Aufgabe 3:

Nachdem das prinzipielle Lesen und Schreiben der Datei funktioniert, machen Sie die Dateiauswahl mit dem JFileChooser komfortabler. Schliessen des Streams am Ende nicht vergessen!

#### Aufgabe 4:

In der Vorlage finden Sie noch zwei leere Methoden

```
private void schreibeDatei(String[] zeilen, String dateiName) {  
}  
  
private String[] leseDatei(String dateiName) {  
  
    return null;  
}
```

Die einzulesende Textdatei soll jetzt in ein String Array eingelesen werden. Damit stehen die Daten auch für weitere Bearbeitung zur Verfügung. Passen Sie den bisherigen Code dahingehend an, dass zum Lesen der Datei die Methode aufgerufen wird und dann das retournierte String Array auf dem TextArea erscheint.

Beim Schreiben in die Datei wird ebenso ein String Array übergeben. Sie müssen also das TextArea in dieses Array umkopieren.

**Tipp:** Um den bestehenden Code aus Aufgabe 1-3 nicht zu verändern, können Sie einfach in Eclipse die Klasse kopieren und umbenennen.

#### Aufgabe 5:

Was fällt auf, wenn Sie einen längeren Text laden? Ergänzen Sie eine JScrollPane als Container für das TextArea, so dass man durch längere Texte scrollen kann.

