

Java-Editor und SQLite-Datenbanken

Um ein Java-Programm zu erstellen, das eine SQLite-Datenbank¹ verwendet, müssen in der Entwicklungsumgebung ein paar Vorbereitungen getroffen werden. Die notwendigen Schritte werden für den Java-Editor² im Folgenden erklärt.

Schritt 1:

Zunächst muss der aktuelle SQLite JDBC Treiber für die Kommunikation mit einer SQLite-Datenbank heruntergeladen werden. Zum Zeitpunkt der Erstellung dieser Anleitung bestand der Treiber aus zwei jar-Dateien (*sqlite-jdbc-X.X.X.X.jar* und *slf4j-api-X.X.X.X.jar*), die z. B. über GitHub³ erhältlich sind. Die X im Dateinamen stehen für die aktuelle Versionsnummer. Die Dateien können an einem beliebigen Ort abgespeichert werden. Wichtig ist, dass man sich den Speicherort merkt, um den Dateipfad im nächsten Schritt im JavaEditor zu hinterlegen. Im Beispiel in Abbildung 2 wurden die Dateien in einem Ordner *sqlite* auf Laufwerk C gespeichert.

Schritt 2:

Der Speicherort der Dateien *sqlite-jdbc-X.X.X.X.jar* und *slf4j-api-X.X.X.X.jar* muss im Java-Editor als Klassenpfad hinterlegt werden. Dazu wird im Menü unter *Fenster* → *Konfiguration* das *Konfigurationsfenster* geöffnet.

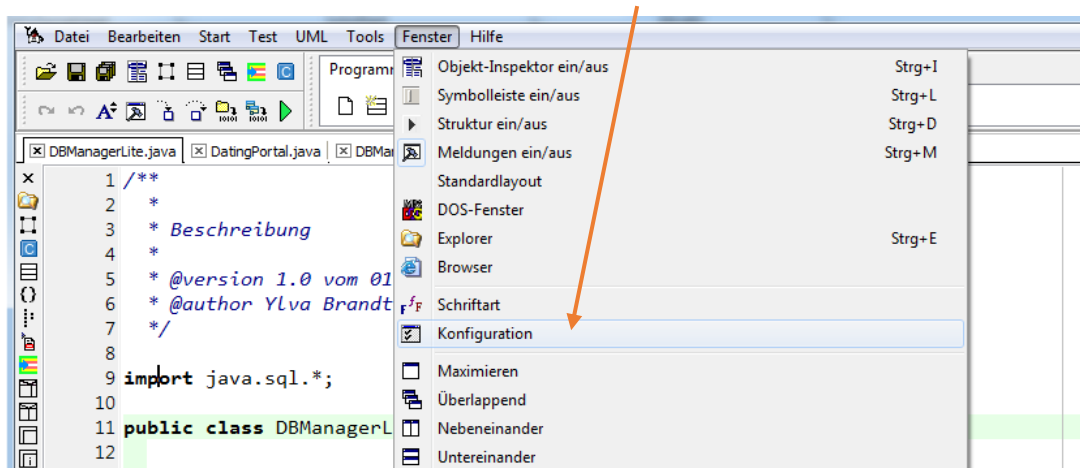


Abbildung 1: Öffnen des Fensters zur Konfiguration

¹ <https://www.sqlite.org/> [Datum des Zugriffs: 02.02.2024]

² G. Röhner. Java-Editor. <https://javaeditor.org> [Datum des Zugriffs: 02.02.2024]

³ <https://github.com/xerial/sqlite-jdbc#usage> Datum des Zugriffs: 17.01.2024

Bei den Einstellungen für den Interpreter muss nun bei *Classpath-User* der Pfad zum Speicherort der jar-Dateien angegeben werden. Klickt man auf den Button *Bearbeiten* in dieser Zeile, kann der Pfad über einen Datei-Browser ausgewählt werden.

Anschließend kann das Fenster über den Button *Speichern* verlassen werden.

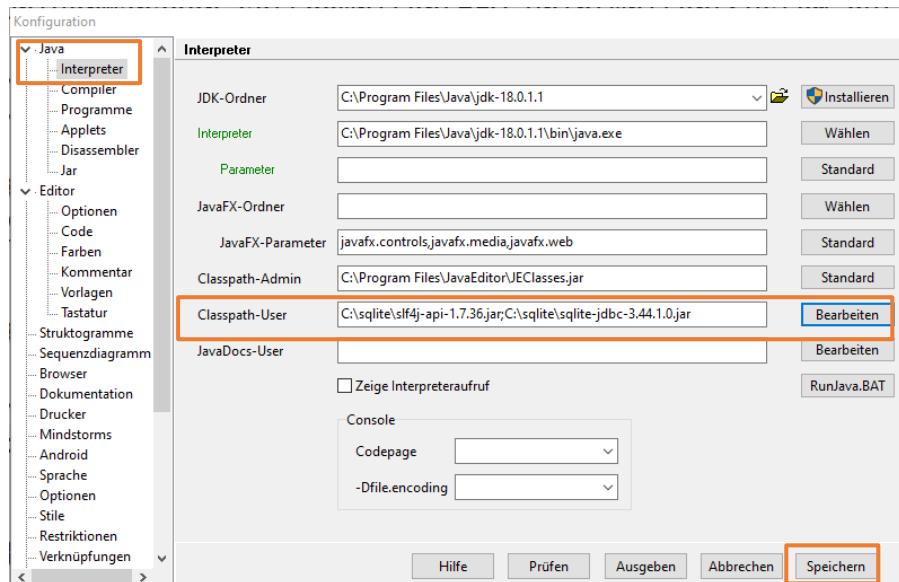


Abbildung 2: Hinterlegen des Klassenpfades

Schritt 3:

Um die Verwendung einer Datenbank in einem Java-Programm ein wenig zu vereinfachen, verwenden wir die Klasse *DBManagerSQLite*, welche die Kommunikation mit der Datenbank übernimmt. Um die Klasse verwenden zu können, muss die Datei *DBManagerSQLite.java* in dem Ordner gespeichert werden, in dem Sie auch Ihr Java-Programm speichern. Öffnen Sie die Datei einmal im *Java-Editor* und kompilieren Sie sie.

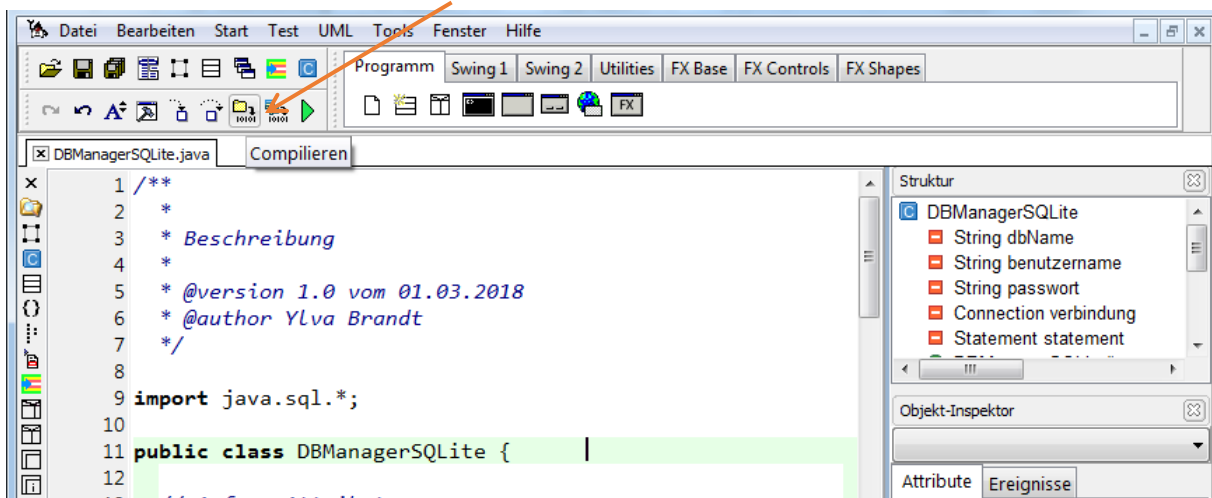


Abbildung 3: Kompilieren der Datei *DBManagerSQLite.java*

Weitere Informationen zur Verwendung der Klasse *DBManagerSQLite* finden Sie in der Datei *Anleitung_DBAnwendung_JavaEditor.pdf*.

Schritt 4:

Eine *SQLite Datenbank* ist eine Datei mit der Endung *.db*. Die Datenbank, die Sie verwenden möchten, muss ebenfalls in dem Ordner gespeichert werden, in dem Sie Ihr Java-Programm speichern.

Schritt 5:

Erzeugen Sie nun eine Java-Datei, welche die Klasse *JFrame* erweitert. Das ist über *Datei* → *Neu* → *JFrame* oder durch Klick auf das Icon zum Erzeugen eines neuen *JFrames* möglich (s. orange Umrahmung in Abbildung 4). Diese Datei muss im gleichen Ordner wie die Datenbank und die Klasse *DBManagerSQLite* gespeichert werden. Weitere Informationen zum Erstellen eines *JFrames* für eine Datenbankanwendung finden Sie in der Datei *Anleitung_DBANwendung_JavaEditor.pdf*.

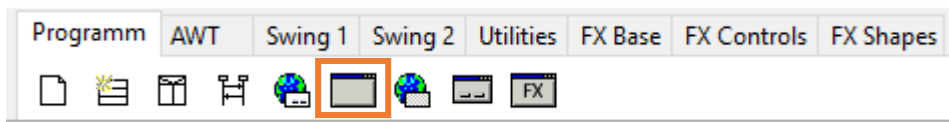


Abbildung 4: Icon zum Erzeugen eines *JFrames*

Dieses Werk ist lizenziert unter einer [Creative Commons Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz](#). Von der Lizenz ausgenommen ist das InfSII-Logo.

Für die korrekte Ausführbarkeit der Quelltexte wird keine Garantie übernommen. Auch für Folgeschäden, die sich aus der Anwendung der Quelltexte oder durch eventuelle fehlerhafte Angaben ergeben, wird keine Haftung oder juristische Verantwortung übernommen.