

Aufgabe 2

Android-Studio-Projekt "LaborDjinni"

In dieser Aufgabe soll der JNI Code durch Djinni generiert werden. Wie in der Aufgabe 1a, soll nur ein String aus C++ in der `TextView` angezeigt werden.

Projekt erstellen

Erstelle in dem Ordner "android" ein neues Android Projekt mit dem Namen "**Greeter**" und Company Domain "**hsfl.com**". Bei der Erstellung des Projekts muss der *C++ Support* aktiviert werden. Bitte darauf achten, dass der *C++11 Standard* mit den *Exceptions* (`-fexceptions`) und *Runtime Type Information Support* (`-frtti`) benutzt werden.

Djinni IDL erstellen

Erstelle in dem Ordner `djinni` eine Datei "`greeter.djinni`". Die Datei definiert die Cross-Platform Interfaces:

```
greeter = interface +c {
    static create_greeter() : greeter;
    welcomeFromCpp() : string;
}
```

JNI generieren

In der Konsole das Script `run_djinni.sh` ausführen. Dieses Script fügt einige neue Ordner und Dateien dem "Greeter" Projekt hinzu. Als nächstes muss die Projektstruktur im Android Studio aktualisiert werden, dazu auf `Build > Refresh Linked C++ Projects` klicken.

CMake

Der Inhalt der `CMakeLists.txt` in dem Android Projekt, muss durch folgende Zeilen ersetzt werden:

```
cmake_minimum_required(VERSION 3.4.1)
set(LIB_DIR ../../../../)
add_subdirectory(${LIB_DIR} deps/libgreeter)
```

Implementierung C++

Erstelle eine neue C++ Klasse, welche die Funktionalität von der abstrakten `Greeter` (`greeter.hpp`) Klasse implementiert. Per `Build > Refresh Linked C++ Projects` Projektstruktur aktualisieren. Die Funktion `welcomeFromCpp` soll nur ein String zurückgeben.

Beispiel `create_greeter()`:

```
std::shared_ptr<Greeter> Greeter::create_greeter() {
    return std::make_shared<GreeterImpl>();
}
```

Java

In der `MainActivity` muss die C++ Library "GreeterLib" geladen werden.

```
static {  
    System.loadLibrary("GreeterLib");  
}
```

Benutzt die `Greeter` (`Greeter.java`) Klasse um den String aus der `welcomeFromCpp()` Funktion abzurufen und in der `TextView` anzuzeigen.