

Grundlagen der Programmierung

Listen



Hinweise zur Bearbeitung:

Achte auf einen sauberen Quellcode, insbesondere Einrückungen sind wichtig!

Wichtige Materialien für heute:

Handbuch: Listen → Implementieren → Leere Liste anlegen

Handbuch: Listen → Implementieren → Liste mit Initialwerten anlegen

Handbuch: Listen → Implementieren → Größe einer Liste lesen

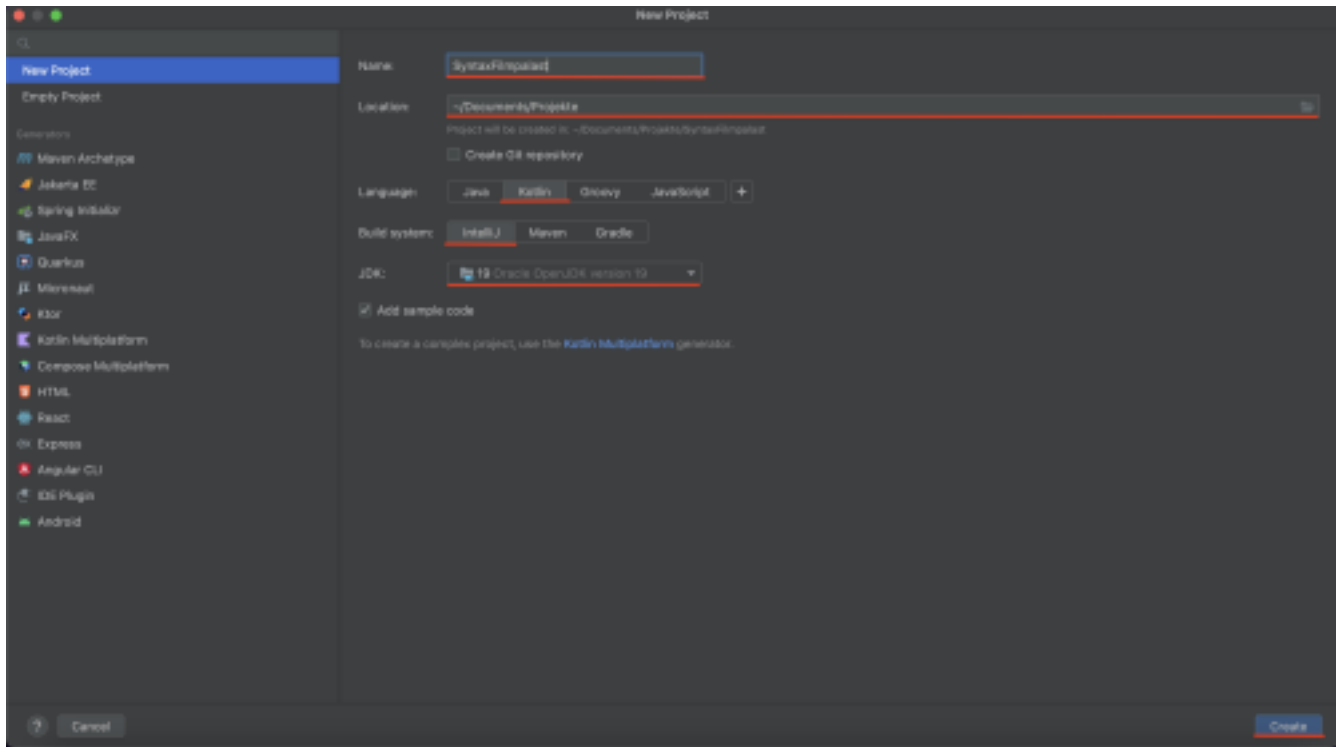
Handbuch: Listen → Implementieren → Wert einer Liste an Index auslesen

Handbuch: Variablen & Datentypen → Nachschlagen → Die wichtigsten Datentypen

Aufgabe 1 - Eure eigene Videothek

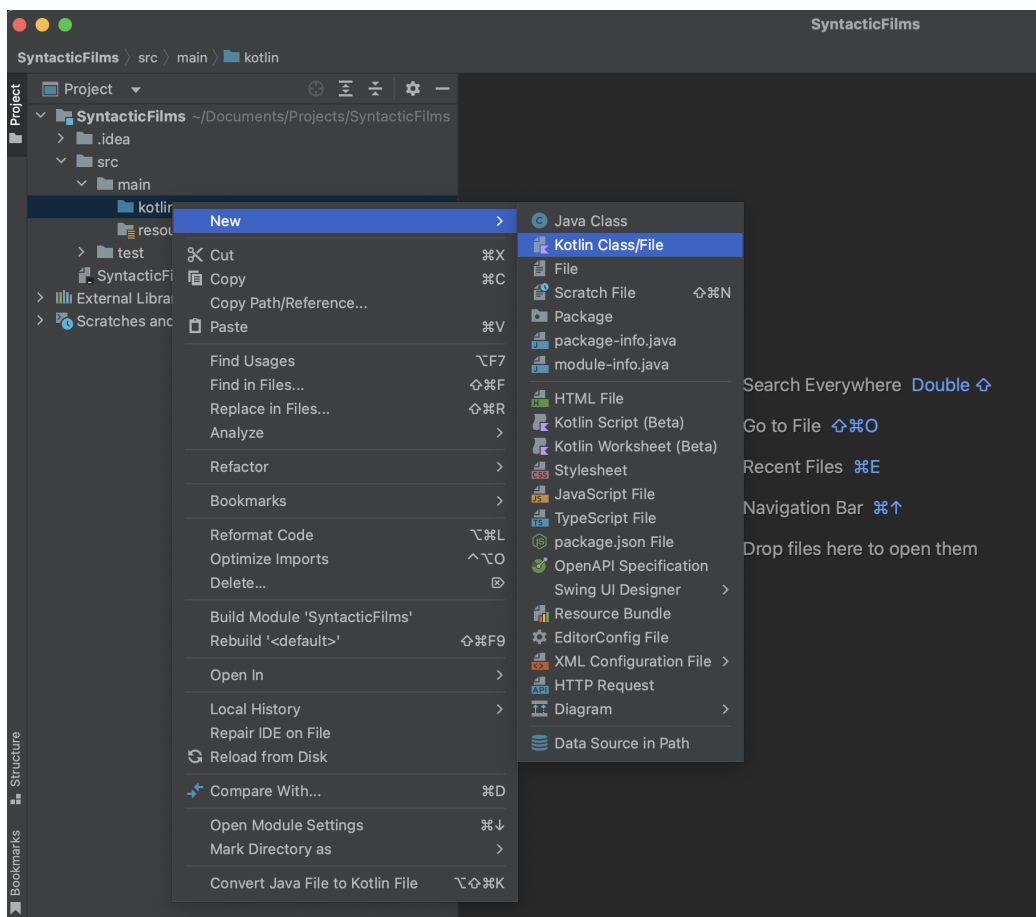
Ihr seid Inhaber einer digitalen Videothek und sollt passend zu dieser ein Programm entwickeln, welches euch beim Abwickeln eures Geschäfts unterstützt. Dazu müsst ihr zunächst ein neues Projekt entwerfen. Den Namen dieses Projekts könnt ihr so wählen, wie ihr eure Videothek taufen würdet (Bsp. Syntax Filmpalast). *Ihr könnt natürlich auch statt Filmen mit anderen Produkten arbeiten.*

Achtet beim Erstellen darauf, dass ihr das Projekt in einen für euch einfach zu findenden Dateipfad abspeichert (hierfür bietet sich bspw. der Speicherort eurer anderen Projekte). Das folgende Bild zeigt euch nochmal, wie ihr ein neues Projekt erstellen könnt.

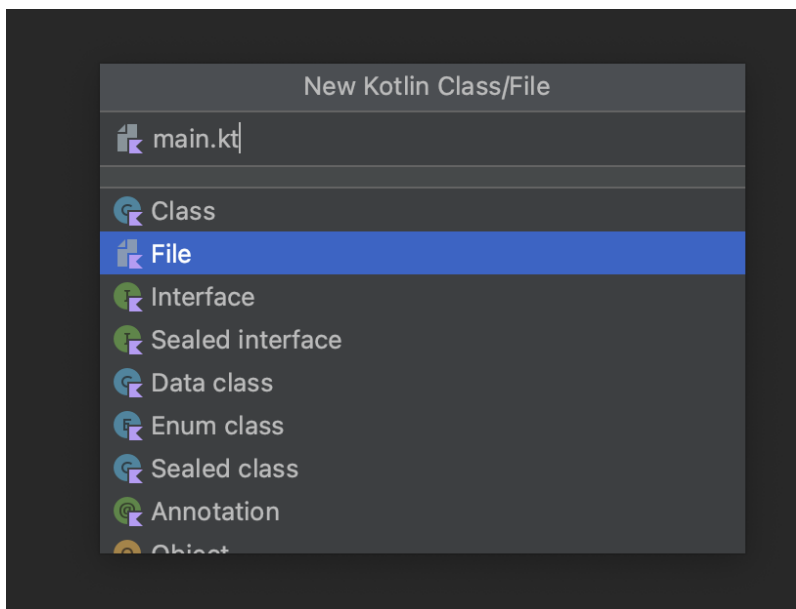


Nachdem ihr erfolgreich das Projekt erstellt habt, müsste das Programm bereits eine Main Datei zur Verfügung stellen, in welcher wir in den folgenden Aufgaben weiterarbeiten werden.

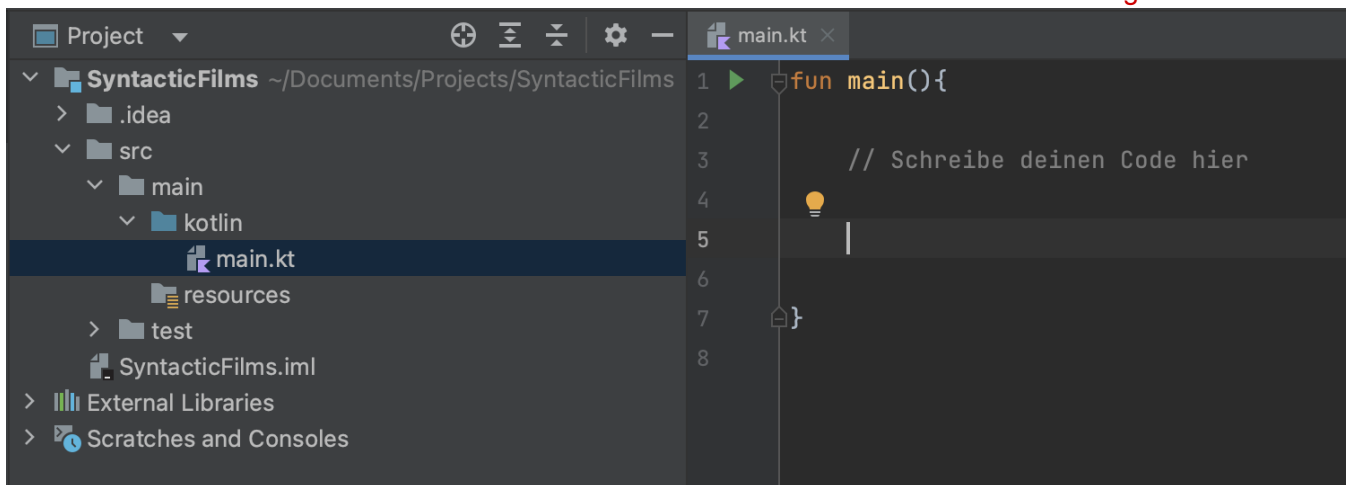
Falls das nicht der Fall ist, erstellt zunächst eine neue Kotlin Datei und nennt diese "main.kt". In dieser Datei solltet ihr nun eine Main-Funktion erstellen, in welcher der gesamte Code der restlichen Aufgaben geschrieben wird. Die folgenden Bilder zeigen euch nochmal, wie ihr eine neue Datei erstellt.



Schritt 1: rechtsklick auf den Ordner, in dem ihr die Datei erstellen wollt (z.B. src/main/kotlin)



Schritt 2: Dateinamen eingeben



Schritt 3: Erstellen der main-Funktion

Aufgabe 2 - Den Bestand erfassen.

Wie soll man eine Videothek betreiben, ohne ein vernünftiges Sortiment zur Verfügung zu stellen...? Richtig, gar nicht. Deswegen sollten wir uns schnell darum kümmern, ein paar Filme zum Verkauf bereitzustellen. Erstelle hierfür eine MutableList, die Werte des Datentyps String enthält. Befüllen kannst du diese mit 3 deiner liebsten Filme.

Allerdings wollen wir die Filme ja auch verkaufen. Dazu muss jedem Film erst einmal ein Preisschild umgehängt werden. Dies können wir einfach umsetzen, indem wir eine weitere MutableList erstellen, welche aus Double Werten besteht. Auch hier könnt ihr kreativ sein und für eure 3 Filme, 3 verschiedene Preise wählen (bleibt hierbei z.B. zwischen 5,00€ und 20,00€).

Aufgabe 3 - Den Kunden begrüßen.

Nun besitzt unser Shop bereits ein richtiges Sortiment, worauf wir nicht nur zugreifen können, sondern das auch anpassbar ist. Allerdings wollen wir auch unsere Kunden beim Kauf unterstützen. Dafür wollen wir eine freundliche Begrüßung ausgegeben haben (via println).

Nach der Begrüßung soll dem Kunden unser gesamtes Sortiment (also zuerst die Liste der Namen

und dann die Liste der dazugehörigen Preise unserer Filme) vorgestellt werden.

Aufgabe 4 - Den Kunden bedienen.

Unser Ziel ist es, dass wir unsere Kunden bestmöglich zufriedenstellen. Deshalb geben wir für den Kunden zuerst einmal den günstigsten Film mit dem entsprechenden Preis aus (wir wollen den Kauf ja schmackhaft machen).

Nun soll der Kunde gefragt werden, welcher der Filme (aus dem ganzen Sortiment) ihn zum Kauf interessiert (erinnert euch hierbei an die `.readln()` - Funktion). Mithilfe einer Eingabe (z.B. von einem Index) soll der Kunde so seinen gewünschten Film wählen können. Seine Eingabe speichert ihr in einer von euch neu erstellten Variable ab.

Nachdem der Kunde seinen Film gewählt hat, soll eine Bestätigung ausgegeben werden, damit der Kunde weiß, ob er alles richtig durchgeführt hat.

Aufgabe 5 - Den Kauf abwickeln.

Jeder Kunde besitzt anfangs ein Budget von 50,00€. Legt dazu eine Variable an, welche ihr entsprechend benennt.

Um den Kauf nun endgültig abzuwickeln, muss zunächst der ausstehende Betrag vom Budget des Kunden abgezogen werden.

Nachdem der gewählte Film jetzt allerdings verkauft ist, müsst ihr dafür sorgen, dass dieser aus dem Sortiment genommen wird (achtet dabei darauf, euch nicht nur um den Namen, sondern auch um den Preis zu kümmern). Damit der Kunde einen Überblick hat, was für ein Budget ihm noch zur Verfügung steht, gebe ihm das aktualisierte Budget zusammen mit einer Kaufbestätigung des Films aus.

Aufgabe 6 - Erweiterung des Sortiments.

Seit neuestem ist "The Batman" auf Blu-Ray erhältlich. Du willst natürlich dein Angebot aktuell halten und entscheidest dich dementsprechend dazu, den Film in dein Sortiment aufzunehmen. Du gehst davon aus, dass du diesen für 19,99€ verkaufen kannst.

Nun möchte ein zweiter Kunde einen Kauf tätigen. Behandle diesen ähnlich wie den ersten nur jetzt mit dem aktualisierten Angebot. (Begrüßung → Sortiment zeigen → Kunden Auswahl einlesen → Bestätigung ausgeben → Budget anpassen → Film aus Sortiment nehmen → Kaufbestätigung ausgeben)

Aufgabe 7 - System überarbeiten.

Euer Kollege hat euch letztes von Maps in Kotlin erzählt und ihr wart davon direkt überzeugt, dies auch in eurer Videothek umzusetzen. Euch fällt auf, dass sich eure Filmliste perfekt dazu eignet, diese in eine Map umzuwandeln (erstellt bitte eine `MutableMap`). Dabei soll jeweils der Name des Films den *key* dieser Map einnehmen und der Preis des Films den dazugehörigen *value*. Orientiert euch dabei an die Listen, die ihr in

Aufgabe 2 erstellt hattet und ergänzt diese beim Erstellen wieder mit "The Batman"!

Aufgabe 8 - Ist der Film im Sortiment?

Eines Tages kommt ein Kunde direkt auf euch zu und fragt, ob ihr denn "The Batman" auf Lager habt. Ihr seid euch unsicher und entscheidet euch deswegen dazu eine Abfrage an die Map zu schicken, ob der Film noch vorhanden ist. Gebt dem Gast eine Rückmeldung, zusammen mit dem fälligen Preis aus.

Jetzt könnt ihr den Kunden wieder wie die ersten beiden weiterbehandeln. (Budget anpassen → Film aus Sortiment nehmen → Kaufbestätigung ausgeben)

Wichtig: Wenn ihr beim Anpassen des Budgets über den key der Map auf den Preis des Films zugreift, kann ein nullable-Wert zurückgegeben werden. Lösen könnt ihr das Problem ganz leicht, indem ihr hinter der Subtraktion 2 Ausrufezeichen (bsp. `10.00 - filme["Titanic"]!!`) schreibt.

Aufgabe 9 - Überblick verschaffen.

Damit euch nicht nochmal passiert, dass ihr nicht wisst, was in eurem Bestand ist, wollt ihr eine Lösung finden. Dazu habt ihr euch überlegt, zuerst alle Filmtitel auszugeben und danach dann alle Preise (nutzt dafür die Map Funktionen).