JavaFX: IntelliJ

Ein schneller, kurzer Überblick über IntelliJ. Dies ist meine bevorzugte Entwicklungsumgebung daher möchte ich diese kurz Vorstellen.

Dies ist ein kurzer Blog – das Hauptgewicht liegt hier auf dem YouTube Video.

## Teile der JavaFX Serie (aktualisiert)

* [JavaFX 01: Übersicht / Planung](https://blog.kneitzel.de/2020/06/29/javafx-uebersicht-planung/) ([YouTube](https://youtu.be/emCUpGFpf_Q))
* [JavaFX 02: Installation](https://blog.kneitzel.de/2020/06/24/javafx-uebersicht/) (YouTube)
* [JavaFX 03: Maven & Gradle](https://blog.kneitzel.de/2020/11/02/javafx-maven-und-gradle/) (YouTube)
* [JavaFX 04: IntelliJ](https://blog.kneitzel.de/2020/11/02/javafx-intellij/) (YouTube)

### Teile der JavaFX Serie (alt / geplant)

* [JavaFX 05: Scene Builder](https://blog.kneitzel.de/2020/06/26/javafx-scene-builder/)

* [JavaFX --: Erläuterung der Applikation](https://blog.kneitzel.de/2020/06/26/javafx-erlaeuterungen-der-applikation/)
* [JavaFX 06: Model / View / Controller](https://blog.kneitzel.de/2020/06/26/javafx-model-view-controller/)
* [JavaFX 07: MVVM](https://blog.kneitzel.de/2020/06/29/javafx-mvvm/)
* JavaFX: Komplette MVVM Applikation (geplant)
* JavaFX: Layouts (geplant)
* JavaFX: Formulare einfach erstellen (geplant)
* JavaFX: Multi Plattform (geplant)

## Code der JavaFX Serie

Der Source Code sowie alle Dokumente finden sich auf GitHub: <https://github.com/kneitzel/blog-javafx-series>

## Versionen

IntelliJ IDEA, oft einfach IntelliJ genannt, wird in zwei Versionen angeboten: Der freien Community Edition und der kostenpflichtigen Ultimate Edition.

### Community Edition

Die Community Edition ist komplett frei verwendbar und eine voll funktionstüchtige Entwicklungsumgebung. Lediglich gewisse Enterprise Funktionen sind nicht integriert, was in meinen Augen für Anfänger eher gut ist, da die Entwicklungsumgebung dadurch weniger Elemente bietet, die für den Benutzer uninteressant sind.

Ebenso gefällt mir die Aufgliederung in mehrere Entwicklungsumgebungen. Anders als Eclipse muss IntelliJ nicht alles abdecken. Für C/C++, PHP, JavaScript, … gibt es eigene, spezialisierte Entwicklungsumgebungen. Der Focus liegt bei IntelliJ auf Java – aber ohne darauf komplett beschränkt zu sein! So lässt sich ebenso auch HTML, JavaScript und co komfortabel in IntelliJ bearbeiten…

Die Nutzung beschränkt sich nicht auf nicht gewerbliche Nutzung. Die Community Edition ist auch kommerziell einsetzbar.

Und natürlich lassen sich auch Enterprise Applikationen mit IntelliJ schreiben. Lediglich gewisse Dinge sind dann nicht komfortabel in die IDE integriert.

### Ultimate

Die Ultimate Edition wurde um einige Enterprise Features erweitert. So werden viele Elemente aus dem Enterprise Umfeld erkannt und komfortabel in die Oberfläche integriert. Einige Aufgaben lassen sich so mit wenigen Klicks erledigen und die IDE bietet eine erweiterte Unterstützung in diesem Bereich. Daher kann ich diese Version, die ich auch im Einsatz habe, nur empfehlen.

JetBrains bietet die Tools nur im „Abo“ an, wobei nach einem Jahr auch eine dauerhafte Lizenz für die dann aktuelle Version erworben wird und das Abo nicht wie bei Anderen Anbietern wie Adobe, dauerhaft behalten werden muss. Da die Kosten vom ersten bis zum dritten Jahr sinken, kann ich nur empfehlen, das Abo beizubehalten, so man die Version dauerhaft nutzen möchte.

Die Preise für eine individuelle Lizenz sind aus meiner Sicht auch voll gerechtfertigt – für eine individuelle Lizenz für IntelliJ zahlt man derzeit 149€/Jahr und der Preis sinkt bis zum dritten Jahr auf 89€/Jahr. (Bzw. wenn man alle Tools von Jetbrains nutzen möchte: 249€ im ersten Jahr und der Preis sind bis zu 149€ im dritten Jahr.) Verglichen zu anderen Anbietern von (kommerziellen) Entwicklungsumgebungen ist das aus meiner Sicht ein geringer Preis.

|  |  |
| --- | --- |
| Wichtig! | Die Ultimate Edition ist nicht notwendig! Es muss kein Geld ausgegeben werden, um in Java zu entwickeln! Die Community Edition ist frei verfügbar und sehr gut. Und sollte diese nicht den Ansprüchen genügen, so kann Eclipse oder Netbeans vielleicht überzeugen. |

## Installation

### Manuelle Installation

Beim Hersteller kann ein Installer von IntelliJ Community Edition heruntergeladen werden. Ein Setup führt einen dann mit wenigen Schritten durch die Installation so dass am Ende IntelliJ installiert ist.

Und der Start ist dann nach der Installation wie gewohnt möglich, d.h. das Setup trägt die Applikation entsprechend ein, so dass es auf Windows, gnome, KDE, … im Startmenü aufgeführt wird.

Wichtig ist: Updates müssen dann auch manuell installiert werden, also erneutes herunterladen, ggf. eine Deinstallation der alten Version und dann de manuelle Installation der neuen Version.

### Toolbox

JetBrains bietet ein kleines Programm an, dass sich Toolbox nennt. Dieses sitzt im Systray und prüft regelmäßig, was für neue Versionen verfübar sind.

Es bietet einen einfachen Weg, die Jetbrains Produkte als auch das Android Studio zu installieren, upzudaten oder zu deinstallieren.

Des Weiteren können mehrere Versionen parallel installiert sein. Dies kann im Hintergrund passieren, d.h. bei einem Update wird per Default die alte Version noch nicht deinstalliert. Somit ist bei Problemen ein schneller Wechsel zur alten Version möglich. (Dieses Verhalten lässt sich natürlich konfigurieren, z.B. wenn Speicherplatz knapp sein sollte.)

Die Toolbox kann genutzt werden, die installierte Software zu starten. In den Entwicklungsumgebungen geöffnete Projekte (die dann dort auch beim Start aufgelistet werden, so nicht automatisch das letzte, geöffnete Projekt erneut geöffnet werden soll) ist ebenfalls über den Reiter Projekte direkt startbar.

## Erster Start

### Konfiguration aus alter Version übernehmen

Wenn die Software das erste mal gestartet wird (es wird für diese Version noch keine Konfiguration gefunden), dann sucht IntelliJ nach früheren Versionen, dessen Konfiguration übernommen werden könnte.

Dann öffnet IntelliJ ein Fenster, in dem es fragt, von welcher Version eine Konfiguration übernommen werden soll. So eine frühere Version gefunden wurde, kann diese ausgewählt werden. Es kann aber auch eine Konfiguration selbst ausgesucht werden – z.B. um eine Vorgabe der Firma oder des Teams direkt zu übernehmen.

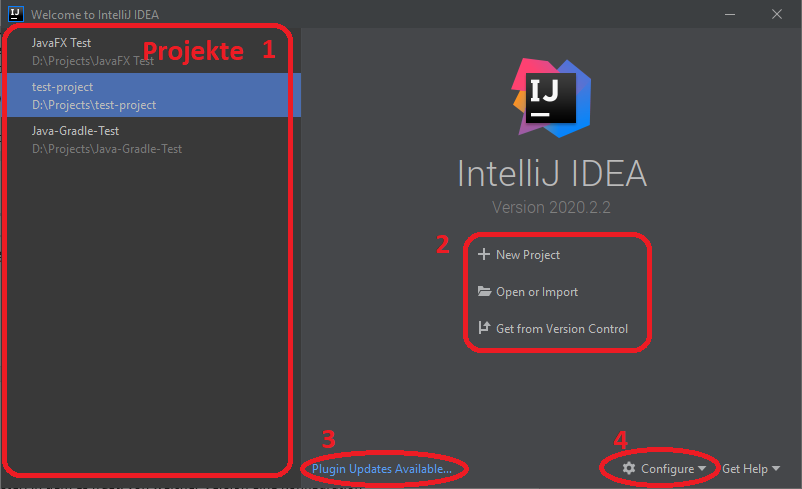
### Erstkonfiguration

So keine Konfiguration importiert wurde, wird man über einen Konfigurations-Wizard geleitet. Dieser fragt das gewünschte Aussehen, PlugIns und Addons ab, die genutzt werden sollen.

Die Auswahl ist unkritisch, da dies alles jederzeit verändert werden kann. Daher einfach das im Augenblick bevorzugte Aussehen wählen und die Standards bei den weiteren Seiten beibehalten, so dass wir zum eigentlichen Startfenster von IntelliJ kommen.

## Startfenster

|  |  |
| --- | --- |
| Wichtig! | Dieses Startfenster erscheint nur, wenn es kein zuletzt geöffnetes Projekt gibt (weil erster Start, der Bereich mit den zuletzt geöffneten Projekten ist dann auch nicht vorhanden) oder wenn die Standard Einstellung deaktiviert wurde, so dass nicht mehr das zuletzt geöffnete Projekt geöffnet wird. |
| Empfehlung | Ich selbst mag es nicht, wenn beim Start das letzte Projekt erneut geöffnet wird, denn ich arbeite zu oft an wechselnden Projekten. Daher ist meine Empfehlung diese Einstellung zu ändern. Diese findet sich in den Settings unter Appearance & Behaviour -> System Settings im Punkt Reopen Projects on Startup. |



Es finden sich mehrere Bereiche:

1. Zuletzt geöffnete Projekte  
   In der Liste der zuletzt geöffneten Projekte kann man schnell und einfach ein Projekt öffnen.  
   Wenn man mit der Maus über ein Projekt geht, dann erscheint am rechten Rand ein X, mit dem ein Projekt aus der Liste entfernt werden kann.
2. Es kann in dem Menü entweder ein komplett neues Projekt erstellt werden oder ein bestehendes Projekt geöffnet / importiert werden. Dabei kann sogar direkt auf eine Source Verwaltung zugegriffen werden, um das Projekt erst lokal zu holen (z.B. per git von GitHub).
3. Beim Start überprüft IntelliJ, ob Updates für Plugins verfügbar sind und gibt einen Link mit dem man direkt zu den Plugins kommt mit Sicht auf alle aktualisierbaren Plugins.
4. In dem Configure Menü kann man verschiedene Konfigurationen aufrufen. Neben den Settings finden sich dort vor allem auch die Plugins. Aber auch viele weitere Einstellungen, auf die ich nicht näher eingehen möchte.

## Projekt von SourceVerwaltung öffnen

Startet wir, indem wir einfach ein Projekt von der Sourceverwaltung öffnen. Dazu können wir die Sourcen der Blog Serie als Beispiel nehmen:

1. Wir wählen „Get from Version Control“ aus.
2. Im folgenden Fenster können wir die git URL eingeben. Wichtig: Dies ist nicht die Web-URL des Github Projekts, sondern die git Adresse, die man auf der Webseite angezeigt bekommt, wenn man auf den grünen „Code“ Knopf drückt. Für unser Projekt ist dies <https://github.com/kneitzel/blog-javafx-series.git>.  
   Und unter der Adresse können wir den Ordner wählen, in welches die Sourcen kopiert werden sollen, z.B. c:\Projects\javafx-blog
3. Sollte auf unserem Rechner kein git installiert sein, dann meldet IntelliJ dies und bietet an, git für uns zu installieren.
4. So wir alles eingegeben haben, können wir mit dem Knopf „clone“ das Herunterladen starten.
5. Sobald IntelliJ alles heruntergeladen hat, wird der Ordner als Projekt geöffnet. IntelliJ erkennt die Build-Files der Gradle und Maven Projekte und bietet den Import an, welchen wir annehmen sollten.
6. Nun sollten wir uns erst einmal in Geduld üben: IntelliJ arbeitet nun alle Projekte ab, um diese richtig zu integrieren. Dies dauert einige Minuten und wir sollten IntelliJ diese Zeit geben.

## Neues Projekt erstellen

Wenn wir ein neues Projekt erstellen wollen, dann bietet IntelliJ uns eine ganze Reihe an möglichen Projekten.

Wichtig ist, dass wir die Java Version, die open im Fenster eingeblendet ist, richtig setzen. So wir nur ein JDK installiert haben wird IntelliJ dieses korrekt erkannt und oben eingetragen haben. Aber wir können über das Dropdown nicht nur die richtige Version auswählen sondern wir können bereits installierte Versionen hinzufügen und sogar weitere JDKs installieren.

Bezüglich Auswahl des Projekt-Typs ist meine Empfehlung, dass man bei Gradle oder Maven bleibt. So bleibt man bei den Projekten unabhängig und ist nicht an eine spezielle Entwicklungsumgebung gebunden.

Die vielen Möglichkeiten der unterschiedlichen Projekte im Detail zu beschreiben würde den Rahmen des Blog Beitrages sprengen, so dass ich es bei diesem Ratschlag belasse.

## Bestehendes Projekt öffnen

Bestehende Projekte lassen sich problemlos öffnen. Dazu muss lediglich der Ordner mit dem Projekt ausgewählt werden und IntelliJ öffnet es für einen.

## Links

* IntelliJ IDEA: <https://www.jetbrains.com/idea/>
* JetBrains Toolbox: <https://www.jetbrains.com/toolbox-app/>
* Git Link für die Sourcen der JavaFX Serie: <https://github.com/kneitzel/blog-javafx-series.git>

## Änderungsnachweis

* 2020-10-30 Erste Veröffentlichung