

Inhaltsverzeichnis

1.1	Installation	2
1.1.1	JDK Installation	2
1.1.2	Gradle Installation	6
1.1.3	Java Installation (JRE)	6
1.1.4	Installation der Software	6
1.1.5	Programmausführung	7

1.1 Installation

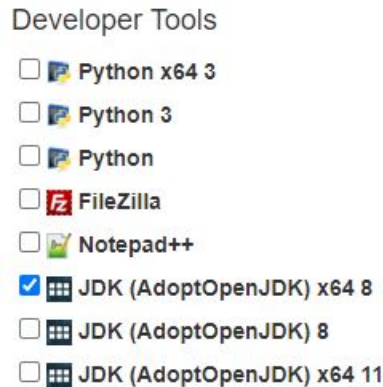
Um das Programm zu installieren, benötigen sie zunächst sowohl eine valide JDK installation, als auch eine Gradle installation. Wenn sie eine der beiden nicht haben, verfolgen sie bitte folgenden zwei Installationsanleitungen. Sollten sie sowohl eine JDK als auch eine Gradle Installation schon haben, so können sie bei Punkt 0.0.3 weiterlesen.

1.1.1 JDK Installation

Sie benötigen zunächst eine valide JDK Installation. JDK steht für Java Development Kit und wird verwendet um Java Programme zu entwickeln. Bei unserem Programm können sie eine beliebige JDK mit einer Version von 8+ herunterladen. Wir werden die Vorgehensweise einer Installation von JDK 8 als Beispiel durchführen.

Windows JDK Installation

Navigieren sie bitte auf die Webseite <https://www.ninite.com> . Ninite ist eine Webseite mit welcher man mehrere Programme gleichzeitig automatisiert herunterladen kann. Dort werden sie unter "Developer Tools" eine JDK Version mit dem Namen "JDK (AdoptOpenJDK) x64 8" sehen. Kreuzen sie die box daneben mit einem Mausclick an. Scrollen sie nun ganz herunter und klicken sie auf den Knopf "Get Your Ninite". Führen sie die heruntergeladene Datei aus, welche dann automatisch JDK 8 installieren wird.

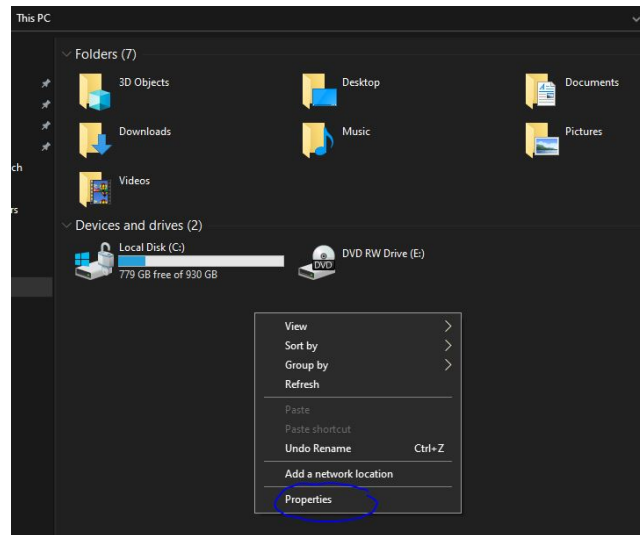


Um zu verifizieren das JDK korrekt installiert wurde, öffnen sie ein CMD oder Powershell Fenster, entweder durch ihre Apps oder mit Strg + R, tippen sie dann CMD oder Powershell und drücken sie Enter. Geben sie im geöffneten Fenster das Kommando Javac ein und drücken sie Enter. Sollte nichts passieren, und sie einen Fehler bekommen, so wurde das JDK nicht korrekt installiert und sie müssen das folgende Subkapitel auch noch durchgehen. Wird das Kommando

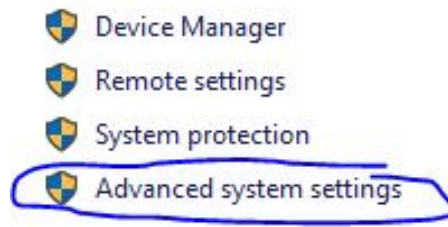
wie im Bild korrekt ausgeführt, so können sie beim Kapitel "Gradle Installation" weiterlesen.

```
PS C:\Users\Leonard> javac
options> <source files>
where possible options include:
  @<filename>           Read options and filenames from file
  -Akey[=value]          Options to pass to annotation processors
  --add-modules <module>(<modules>)*
                        Root modules to resolve in addition to the initial modules, or all modules
                        on the module path if <module> is ALL-MODULE-PATH.
  --boot-class-path <path>, -bootclasspath <path>
                        Override location of bootstrap class files
  --class-path <path>, -classpath <path>, -cp <path>
                        Specify where to find user class files and annotation processors
  -d <directory>         Specify where to place generated class files
  -deprecation           Output source locations where deprecated APIs are used
  --enable-preview       Enable preview language features. To be used in conjunction with either -source or --release.
  -encoding <encoding>   Specify character encoding used by source files
  -endorseddirs <dirs>   Override location of endorsed standards path
  -extdirs <dirs>        Override location of installed extensions
  -g                    Generate all debugging info
  -g:[lines,vars,source] Generate only some debugging info
  -g:none               Generate no debugging info
  -h <directory>         Specify where to place generated native header files
  --help, -help, -?      Print this help message
  --help-extra, -X       Print help on extra options
  -implicit:{none,class} Specify whether or not to generate class files for implicitly referenced files
  -J<flag>               Pass <flag> directly to the runtime system
  --limit-modules <module>(<modules>)*
                        Limit the universe of observable modules
  --module <module>(<modules>)*, -m <module>(<modules>)*
                        Compile only the specified module(s), check timestamps
  --module-path <path>, -p <path>
                        Specify where to find application modules
  --module-source-path <module-source-path>
                        Specify where to find input source files for multiple modules
  --module-version <version>
                        Specify version of modules that are being compiled
  -nowarn                Generate no warnings
  -parameters            Generate metadata for reflection on method parameters
  -proc:{none,only}      Control whether annotation processing and/or compilation is done.
  -processor <class1>[,<class2>,<class3>...]
                        Names of the annotation processors to run; bypasses default discovery process
  --processor-module-path <path>
```

PATH und JAVA_HOME Sollte das Kommando im letzten Schritt ein Fehler angezeigt haben, so ist JDK nicht korrekt konfiguriert. Um dieses zu ändern, navigieren sie bitte zu dem Installationspfad ihrer Jdk, welcher einen Namen wie "C:/Program Files/Java/jdk-version haben wird". In diesem Ordner werden sie einen weiteren Ordner "bin" finden und erst einmal öffnen. Kopieren sie nun den Pfad dieses Ordners (z.B. "C:/Program Files/Java/jdk-12.0.2/bin" für JDK 12.0.2) und öffnen sie unter Windows XP/7 My PC, oder unter Windows 8+/10 This PC. Klicken sie die rechte Maustaste und wählen sie "Properties" aus.



In dem geöffneten Fenster auf der linken Seite werden sie einen Knopf "Advanced System Settings" sehen, welchen sie nun anklicken.



In dem nun geöffneten Fenster, ganz unten Rechts, klicken sie auf den Knopf "Environment Variables".



Hier sind alle Variablen von Programmen die Windows durch die Konsole erkennt vorhanden. Wir müssen nun unsere JDK Installation dazufügen. Um dieses zu tun, klicken sie bei den "System Variables" auf den "New..." Knopf und geben der Variable den Namen "JAVA_HOME", beachten sie das der Name korrekt und groß geschrieben ist, und als Pfad geben sie den kopierten Pfad ohne die ".bin" endung an, also z.B. "C:/Program Files/Java/jdk-12.0.2" und klicken sie dann auf Ok.

Variable name:

Variable value:

(*) Jetzt müssen sie nur noch ihre System PATH Variable bearbeiten um Java als System Variable einzufügen. Um dieses zu tun, scrollen sie unter "System Variables" runter bis sie eine Variable mit dem Namen "Path" sehen, und dann diese doppelclicken. In dem jetzt angezeigten Fenster sind sämtliche Programme die von Windows als System Variablen erkannt werden drin. Clicken sie rechts auf "New" und fügen sie den gesamten kopierten Pfad ein und drücken sie Enter. Sie können nun alle Fenster schließen.

System variables	
Variable	Value
OS	Windows_NT
Path	C:\Program Files (x86)\Intel\iCLS Client\;C:\Program Files\Intel\iCLS...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64
PROCESSOR_IDENTIFIER	Intel64 Family 6 Model 60 Stepping 3, GenuineIntel
PROCESSOR_LEVEL	6
PROCESSOR_REVISION	3c03

(*) Für die Installation auf anderen Betriebssystemen, lesen sie bitte die Dokumentation auf der JDK Oracle Webseite.

1.1.2 Gradle Installation

Nun werden wir Gradle installieren. Gradle ist ein Tool mit welchem man Java Applikationen bauen kann.

Windows Gradle Installation

Navigieren sie auf die Webseite <https://gradle.org/install/> und scrollen sie runter zu "Installing manually". Klicken sie dort auf "Binary-only" welches Gradle herunterladen sollte.

Installing manually

Step 1. **Download** the latest Gradle distribution

The current Gradle release is version 6.5.1, released on 30 J
comes in two flavors:

- Binary-only**
- Complete, with docs and sources

Extrahieren sie die heruntergeladene ZIP datei und kopieren sie den Pfad der bin datei in der Gradle datei, z.B. "PFAD/gradle-6.4.1/bin" und erstellen für diesen einen neuen Eintrag in der System Path Variable, siehe (*) bei der JDK Installation. Sie sollten nun die Möglichkeit haben in einem CMD oder Powershell fenster das Kommando "gradle -version" ausführen zu können.

(*) Für die Installation auf anderen Betriebssystemen, lesen sie bitte die Dokumentation auf der Gradle Webseite.

1.1.3 Java Installation (JRE)

Um das Programm ausführen zu können, werden sie eine normale Java Installation benötigen. Der Unterschied zwischen der JRE (Java Runtime Edition) und der JDK, ist das die JDK zur Entwicklung von Java Programmen gedacht ist, die JRE aber zur Ausführung dieser Programme benutzt wird. Um die neuste JRE zu installieren, navigieren sie einfach in ihrem Browser auf die Webseite <https://www.java.com/download> und laden sie sich Java herunter.(Klicken sie einfach immer auf weiter in der Installation)

(*) Wenn sie möchten dass das Programm schneller läuft, so können sie dem mehr Arbeitsspeicher zur Verfügung stellen. Hierbei ist die 32 bit JRE allerdings auf 2GB beschränkt, also könnten sie sich gleich eine 64 bit JRE herunterladen.

1.1.4 Installation der Software

Nachdem sie nun alle obigen Programme installiert haben, können sie nun endlich die Applikation selbst installieren. Navigieren sie hierzu bitte in den Ordner der Software (galaxytrucker in der Regel), öffnen sie eine Konsole ihrer Wahl und führen sie das Kommando "gradle clean desktop:dist" aus. Sollte ihre Gradle Version hiermit Probleme haben, so können sie auch den Projekt begleitendem

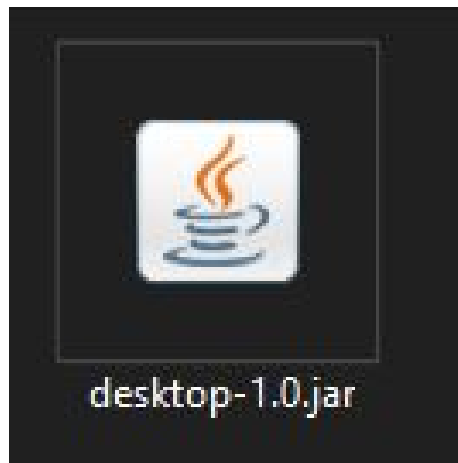
gradlew File benutzen, indem sie stattdessen das Kommando ".\gradlew clean desktop:dist" ausführen. (Unter Windows funktioniert letzteres nur in Powershell und nicht in CMD) **Der Gradle Build wird zwar ganz viele Warnun-**

```
./gradlew clean desktop:dist
```

gen geben, dieses liegt allerdings da dran das Gradle nicht versteht wozu die @NonNull tags gedacht sind, also ignorieren sie diese.

Nachdem sie die Nachricht "BUILD SUCCESSFUL" bekommen, können sie die Programmdatei im Unterordner "desktop→build→libs" finden. Es wird eine Datei vom Format JAR sein und wahrscheinlich den Namen "desktop-1.0.jar" haben.

```
BUILD SUCCESSFUL in 36s  
8 actionable tasks: 8 executed
```



1.1.5 Programmausführung

Um das Programm nun auszuführen, können sie entweder die JAR durch doppelklick direkt ausführen, oder, wenn sie dem Programm mehr Arbeitsspeicher zur Verfügung stellen wollen (damit es schneller läuft), dieses mit dem Kommando "java -Xms1G -Xmx4G -jar .\desktop-1.0.jar" starten. Letzteres wird dem Programm mindestens 1 GB Arbeitsspeicher (Xms steht für Minimum) und bis zu maximal 4 GB Arbeitsspeicher zur Verfügung stellen (Xmx steht für Maximum).

```
java -Xms1G -Xmx4G -jar .\desktop-1.0.jar
```