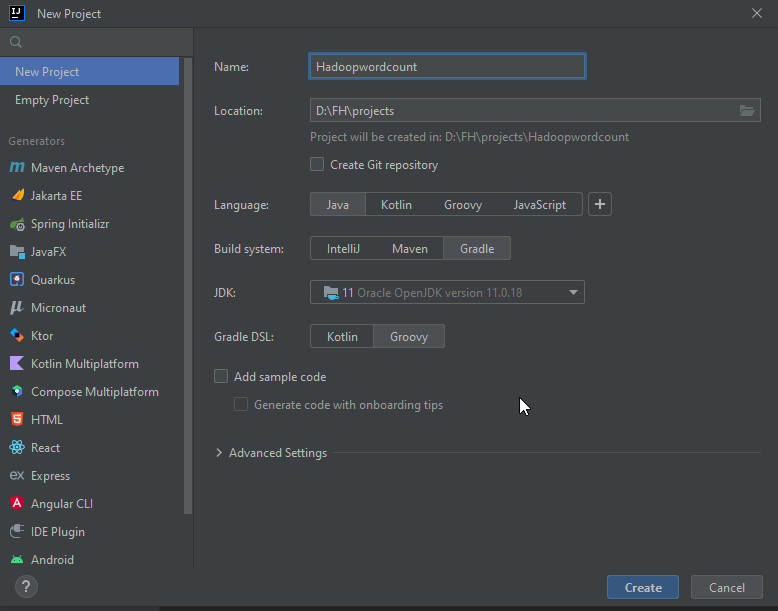
# MapReduce job mit Java in IntelliJ IDE

IntelliJ community Edition (lizenzfrei) von <https://www.jetbrains.com/idea/download/?section=windows> herunterladen

Nach Installation ein neues Java-Projekt anlegen, passende JDK-Version auswählen.

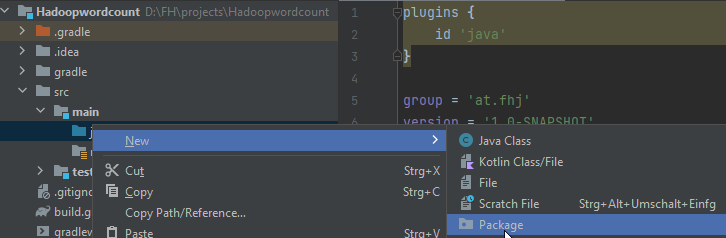
Am besten gradle oder maven als Buildsystem auswählen, damit dies leichter wiederverwendbar ist.



das automatisch generierte build.gradle sollte wie folgt aussehen, nur die "group" ist zu ändern:

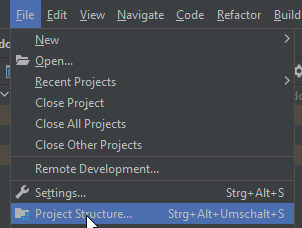


ein neues Package at.fhj anlegen, damit die Struktur der Defaultstruktur enspricht. Dort die 3 Sourcefiles ablegen

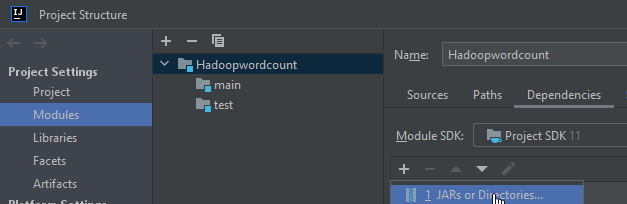


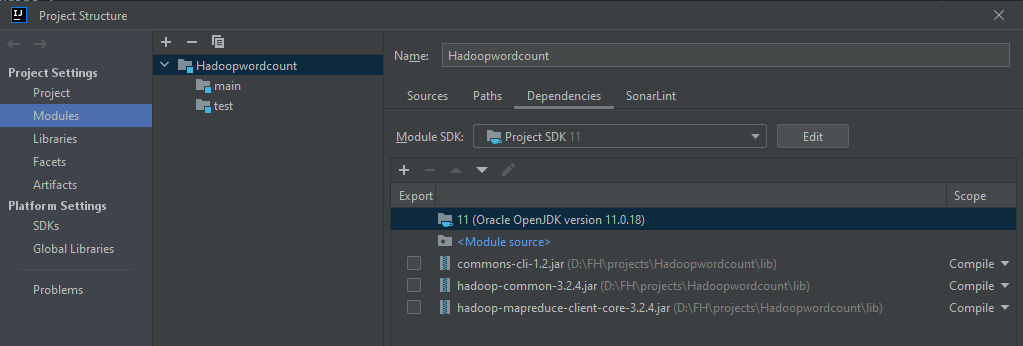


Projektstruktur anpassen:

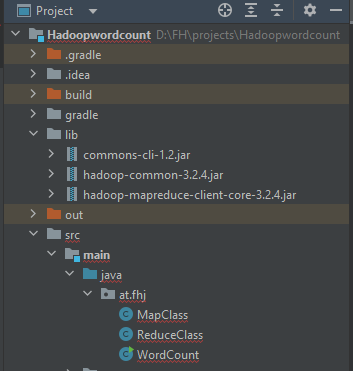


Jar-Dateien zum Projekt hinzufügen (die Dateien entweder aus der E-Learning-Plattform oder noch besser vom installierten hadoop in der virtuellen Maschine – ACHTUNG: Versionen müssen zusammenpassen):

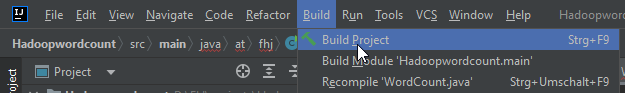




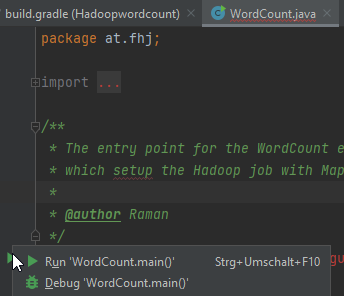
Projektstruktur, nachdem die Sourcedateien dort abgelegt wurden



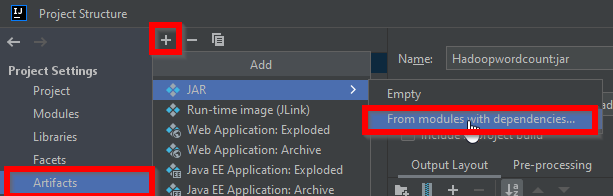
Projekt bauen, d.h. kompilieren

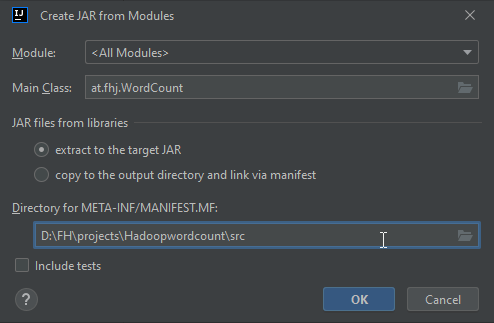


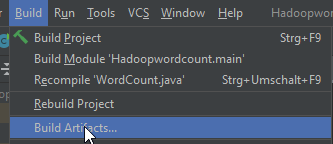
Ausführen oder Debuggen direkt bei den jeweiligen Methoden (müsste man remote debuggen auf der Zielplattform verwenden, wo DFS installiert ist)



Artifakt anlegen, d.h. über "Project settings" das Erstellen des jar-File konfigurieren und dem Jar-File mitteilen, wo der Einsprungspunkt, d.h. die Main-Klasse, zu finden ist.







Je nach verwendetem Buildsystem und deren Settings findet man das generierte Jar-File unter z.B. out/artifacts oder build/libs und beinhaltet der Dateiname eine Versionsnr. oder nicht.

Hadoop starten und Test aufrufen mit dem zuvor generierten und auf die Zielplattform kopierten Jar-File.

Job ausführen

1. Input Ordner und Beispiel Textdatei anlegen (besser wäre direkt von hdfs zu laden)
2. Job ausführen (Vorsicht Pfade sind ggf. anzupassen!)

su – hduser

start-dfs.sh

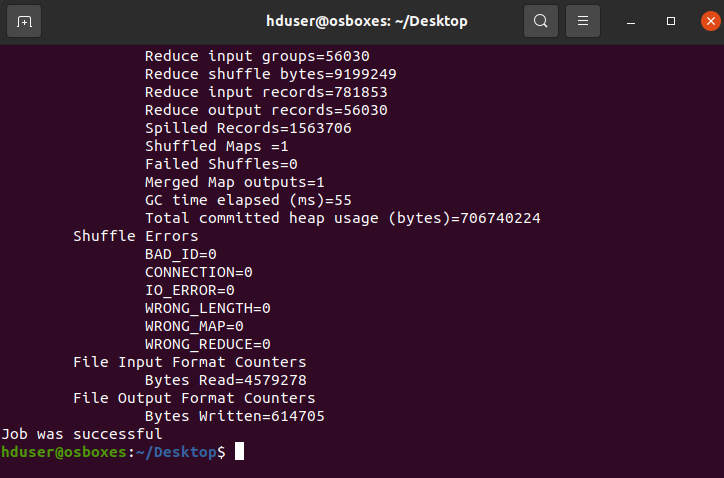
start-yarn.sh

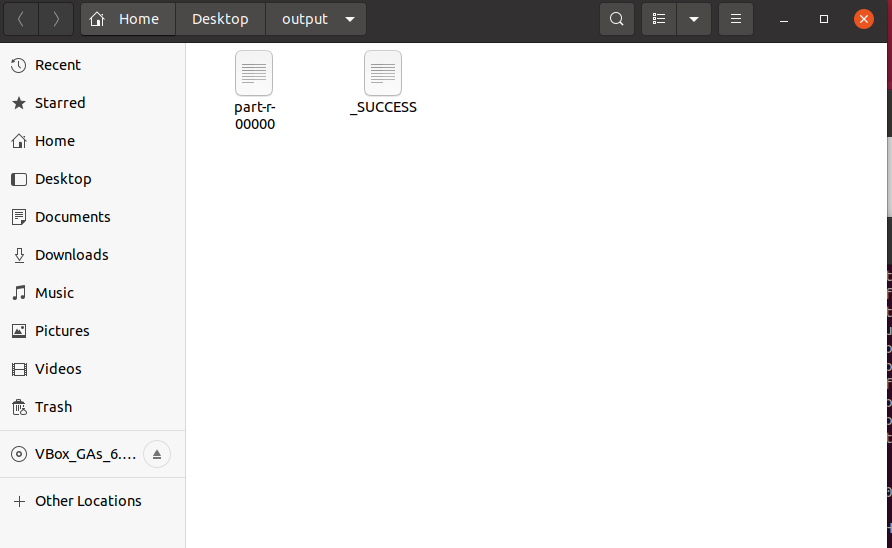
OutputDir=~/Desktop/output

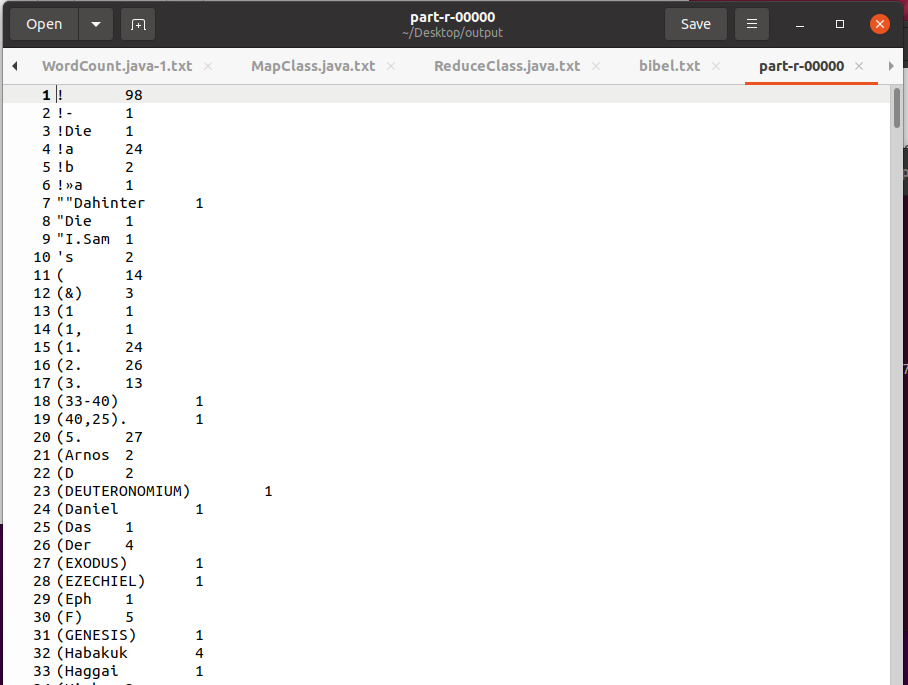
rm -R $OutputDir 2>/dev/null # otherwise following calls fail

hadoop jar HadoopWordCount.jar file:/tmp/Bibel.txt file:$OutputDir

Erwarteter Output wie folgt:







Wichtig: wenn der Job erneut gestartet werden soll, muss zuvor das Output-Verzeichnis gelöscht werden!