

Praxis der Softwareentwicklung



Organisatorisches
Professionelle IDEs
IntelliJ
Aufgaben

AT THE END OF A LONG CODING DAY

OR

THE CANARY IN THE CODE MINE



Organisatorisches

- ▶ wöchentlich 4 Stunden
 - ▶ 2x2 Stunden Theorie/Praxis
- ▶ Was tun wenn die Übung nicht fertig wird?
 - ▶ Ab einer Wassertiefe von 1,20m nimmt der Soldat selbstständig Schwimmbewegungen auf. Die Grußpflicht entfällt hierbei.
 - ▶ Bei Erreichen des Baumwipfels hat der Soldat die Kletterbewegungen selbstständig einzustellen.
- ▶ Lehrunterlagen
 - ▶ Folien werden verteilt
 - ▶ Notizen zu den Folien sind sinnvoll
 - ▶ Ein Buch lesen soll schlau machen
 - ▶ Es gibt eine große Java Community. Vieles findest man auch in Tutorials. Aber Achtung: nicht alles was man findet ist gut.

- ▶ Wir sind pünktlich
- ▶ Fragen sind immer willkommen
- ▶ Es wird nicht ge-
 - ▶ googelt
 - ▶ mailt
 - ▶ gessen
 - ▶ iphonet
 - ▶ daddelt
- ▶ In den Übungen:
 - ▶ IntelliJ Installation auf dem eigenen Rechner ist lauffähig
 - ▶ Screen Sharing aktivierbar
 - ▶ Probleme sofort berichten

- ▶ 3. Semester: Fortgeschrittene Programmierung mit Java
- ▶ 4. Semester: Kleines zusammenhängendes Projekt
- ▶ Andere Veranstaltungen bauen auf Java Kenntnissen auf
 - ▶ Algorithmen & Datenstruktur
 - ▶ Softwareproduktion
 - ▶ Datenbanken
 - ▶ Internet Anwendungsarchitekturen
 - ▶ Wahlpflichtfach ...
- ▶ Ratschlag: Sofort einsteigen, späteres Nachholen ist kaum möglich, Java lernt man nicht an einem Wochenende

- ▶ Ziel der Veranstaltung ist die Vermittlung von Konzepten, welche in der professionellen Softwareentwicklung (mit Java) angewandt werden. Hierzu gehören:
 - ▶ Fortgeschrittene Programmierung mit Java
 - ▶ Programmierung im Team
 - ▶ Softwarequalitätssicherung
 - ▶ Modellierung und Konzeptionierung nach Anforderungen aus dem Fachbereich
- ▶ Die Veranstaltung vermittelt Konzepte. Erfahrungen in Großprojekten sollen in der Praxis gesammelt werden.
- ▶ Sie haben generell zu wenig Übungszeit in den Veranstaltungen und sitzen zu zweit am Rechner, es folgt, dass eigenständige Nacharbeit notwendig ist und vorausgesetzt wird.

- ▶ Hausarbeit im 5. Semester (Q4 dieses Jahr)
- ▶ 4 Personen je Gruppe
- ▶ Gruppenbildung ca. 14 Tage vor Start des 5. Semesters
- ▶ Aufgabenstellung in der ersten Woche
- ▶ Bearbeitungszeit: 3 Wochen
- ▶ Nachprüfung im 6. Semester (Q3 nächstes Jahr)
- ▶ Auslandsstudierende können in Q4 an der Prüfungsleistung online mitarbeiten oder den Termin der Nachprüfung wahrnehmen

- ▶ Bei der Bewertung kommt es auf die Qualität der Programme an, nicht auf die Quantität. Sind die Konzepte der Programmiersprache richtig angewendet worden:
 - ▶ Wie gut werden OO-Prinzipien umgesetzt?
 - ▶ Wie gut sind die Programme lesbar?
 - ▶ Wie gut sind die Programme erweiterbar?
 - ▶ Wie gut sind die Programme wartbar?
 - ▶ Wie gut sind die Programme getestet?

- ▶ Einführung in die Entwicklungsumgebung IntelliJ
 - ▶ Programmierung im Team mit Versionsverwaltungstools wie Git
 - ▶ Buildtools
 - ▶ Ausnahmebehandlung
 - ▶ Generics, File Handling
 - ▶ Klasse Object und Collections
 - ▶ Multi Threading
 - ▶ Lambdas & Stream
 - ▶ Graphische Oberflächen
-
- ▶ Themen des 4. Sem: CRC Karten, Klassendiagramme, Test First, Mocking , GUI, ...

- ▶ Java Sprachdokumentation:
<http://docs.oracle.com/javase/specs/>
- ▶ Goll, Heinisch; Java als erste Programmiersprache: Ein professioneller Einstieg in die Objektorientierung mit Java
- ▶ Krüger, Hansen; Java-Programmierung: Das Handbuch zu Java 8 (O'Reilly)
- ▶ Ullenboom; Java ist auch eine Insel, Galileo Press (Online Version verfügbar: <http://openbook.rheinwerk-verlag.de/javainsel/>)
- ▶ Sierra, Bates; Java von Kopf bis Fuß, O'Reilly
- ▶ Inden, Michael: Der Weg zum Java Profi
- ▶ Hilfe in der Not: <http://stackoverflow.com/>

Professionelle IDEs



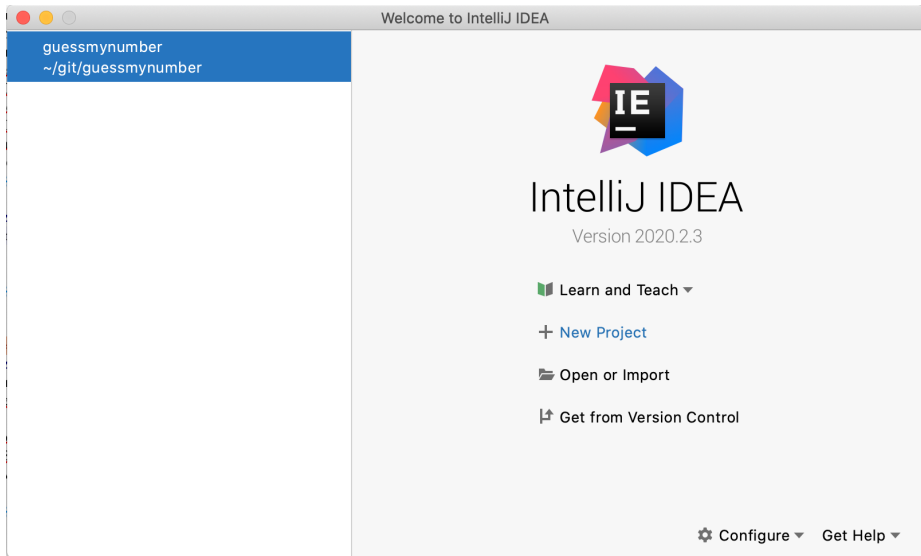
Es gibt eine Reihe von auf Java spezialisierten IDEs. Zu den bekanntesten gehören:

- ▶ Eclipse - www.eclipse.org [verfügbar]
- ▶ IntelliJ - <https://www.jetbrains.com/idea/> [verfügbar]
- ▶ NetBeans - <https://netbeans.org>

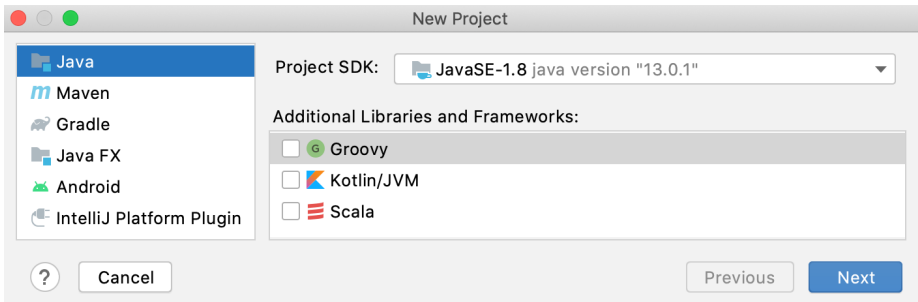
Wir verwenden in dieser Lehrveranstaltung IntelliJ.

1. Java-Programmierung in BlueJ und IntelliJ erfordert ein Projekt
2. BlueJ und IntelliJ öffnen ein separates Fenster pro Projekt
3. BlueJ und IntelliJ erlauben ein dialogbasiertes Hinzufügen von Klassen, Schnittstellen, etc.
4. In IntelliJ gibt es (im Default) keine grafische Übersicht über die Klassen im Projekt
5. In IntelliJ können Java Klassen nicht instanziiert und dynamisch getestet werden
6. In IntelliJ benötigen wir eine explizite Methode zum Starten unserer Anwendungen: die `main` Methode.
7. IntelliJ ist ein sehr mächtiges Werkzeug, das sehr viel mehr an Funktionalität bietet - davon werden wir hier jedoch nur einen kleinen Anteil benötigen

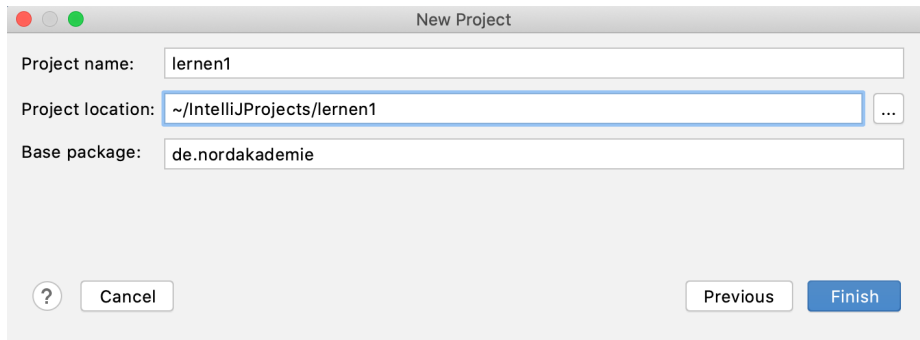
IntelliJ



- ▶ Create New Project
- ▶ Projekt SDK auswählen (1.8 oder neuer)
- ▶ Das Java SDK muss möglicherweise noch runtergeladen werden
- ▶ Next
- ▶ Template Command Line App auswählen
- ▶ Next



- ▶ Projektname vergeben
- ▶ Project location sollte auf den NAK Rechnern im Heimatverzeichnis liegen
- ▶ Sinnvolles base package eingeben (z.B. de.nordakademie.lernen1)
- ▶ Finish



New Project

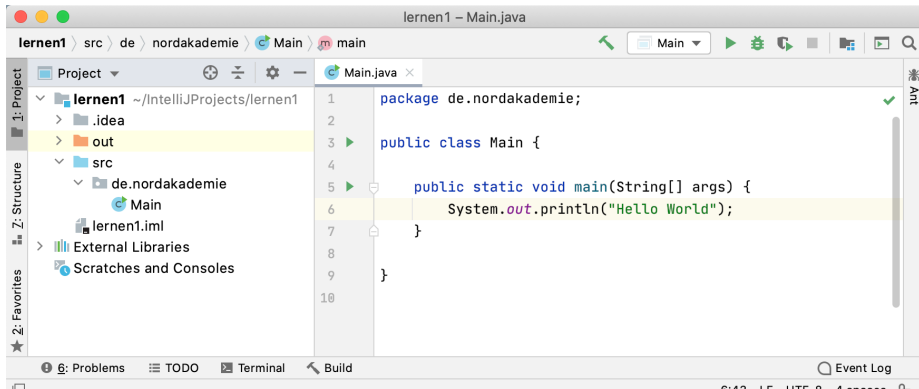
Project name:

Project location: ...

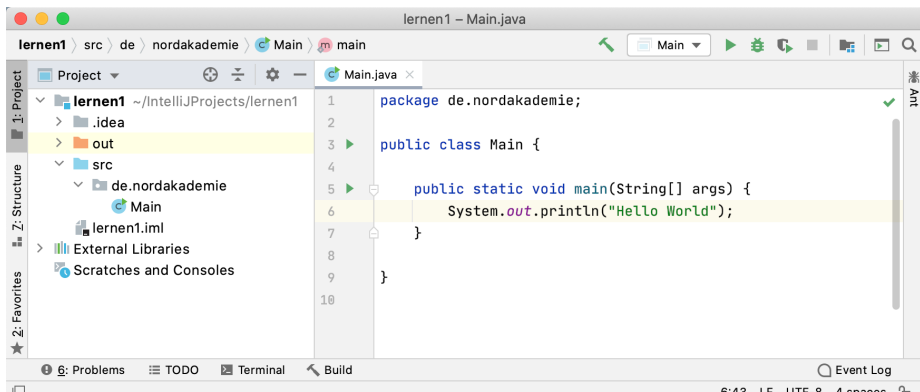
Base package:

? Cancel Previous Finish

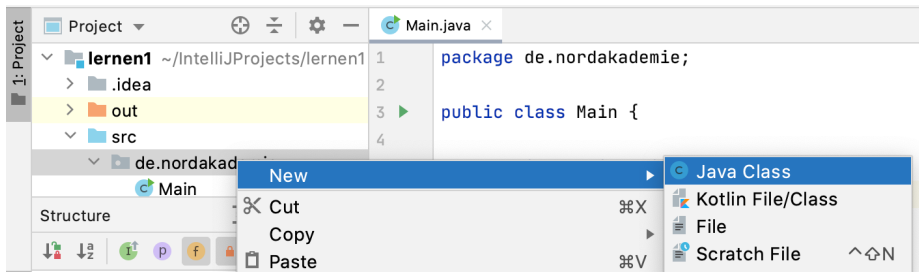
- ▶ Links wird die Projektstruktur angezeigt.
- ▶ Rechts wird der Editorbereich angezeigt zum Bearbeiten des Codes.
- ▶ Bei Bedarf können weitere Views eingeblendet werden.
- ▶ IntelliJ zeigt kontextsensitiv weitere Fenster an.



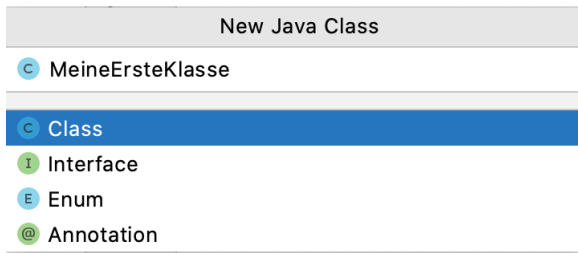
- ▶ Links kann beispielsweise die Ansicht Struktur eingeblendet werden.
- ▶ Beim Ausführen des Programms zeigt IntelliJ die Programmausgabe an.



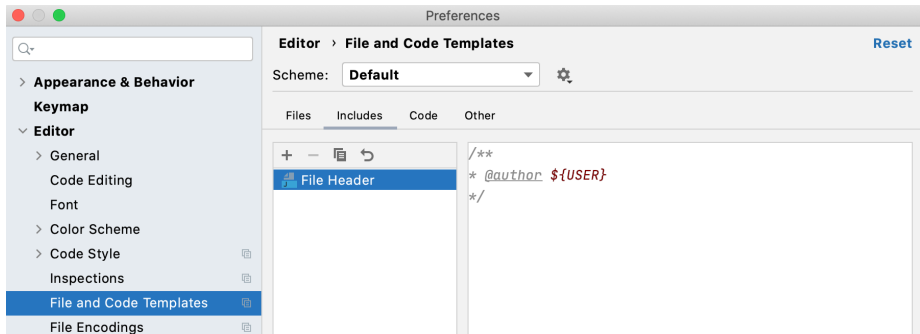
- ▶ Zum Anlegen einer neuen Klasse wählt man das Kontextmenü des Projekts oder des src Verzeichnisses in dem Projekt und wählt New → Java Class .
- ▶ Alternativ geht man über den Menüpunkt File → New → Java Class



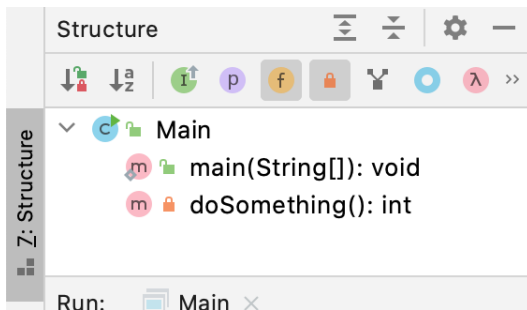
- ▶ Geben Sie einen Klassennamen an. Klassennamen sind groß und in CamelCase-Notation zu schreiben.



- ▶ Die Vorlagen für generierte Klassen können angepasst werden.
- ▶ Sinnvoll ist beispielsweise für jede Klasse eine Autorengabe.
- ▶ Unter Settings (Strg+Alt+S) → Editor → File and Code Templates im Tab Includes sollte File Header minimal sinnvoll befüllt werden.

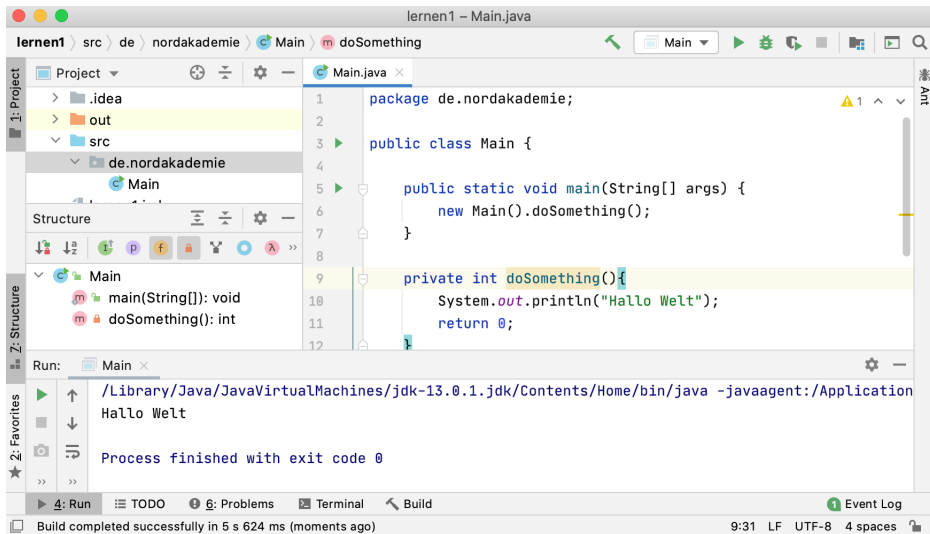


1. In der Structure Übersicht wird die Struktur der offenen Datei angezeigt.
2. Durch einen Klick springt der Cursor im Editor Fenster zur Methode.
3. Ob mithilfe dieser Ansicht oder direkt im Programmcode navigiert wird, kann frei gewählt werden.



- ▶ Eine Klasse kann mit einer der folgenden Methoden ausgeführt werden.
 - ▶ Klicken der grünen Pfeile am linken Rand des Editorfensters.
 - ▶ Klicken auf den grünen Pfeil rechts oben.
 - ▶ Shift + F10
- ▶ IntelliJ öffnet beim Start des Programms ein Ausgabeterminal.

Klasse ausführen



1. Fehler werden an der linken Seite des Editorfensters, im Scrollbalken und durch rotes Unterstreichen der fehlerhaften Stelle angezeigt.
2. Die Maus über dem Fehlermarker zeigt die Fehlerbeschreibung.
3. Nach Änderungen muss mit Strg+S gespeichert werden.

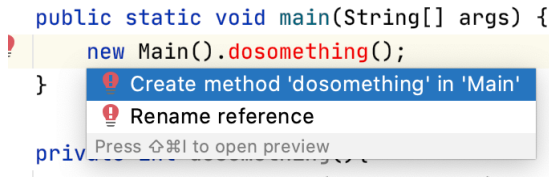
```
public static void main(String[] args) {  
    new Main().dosomething();  
}
```

Cannot resolve method 'dosomething' in 'Main'

Create method 'dosomething' in 'Main' More actions...

```
|  
private int doSomething(){  
    System.out.println("Hallo Welt");  
    return 0;  
}
```

1. Mit Alt+Enter oder durch Klick auf die Glühbirne können Fehlerbehebungsvorschläge angezeigt werden.
2. Diese sind mit Vorsicht zu genießen und nicht wahllos zu benutzen.



Aufgaben

- ▶ Erstellen Sie ein neues IntelliJ Projekt.
- ▶ Erstellen Sie in dem IntelliJ Projekt eine Klasse `App`, inkl. `main`-Methode und geben Sie auf der Konsole mit `System.out.println "Hello World"` aus.
- ▶ Führen Sie die `main`-Methode der Klasse `App` aus.
- ▶ Erstellen Sie in dem IntelliJ Projekt eine Klasse mit dem Namen `Circle` inkl. passendem Konstruktor.
- ▶ Erstellen Sie in dem IntelliJ Projekt eine weitere Klasse `Rectangle` inkl. passendem Konstruktor.
- ▶ Implementieren Sie in beiden Klassen die Methode `calculateArea`, welche den Flächeninhalt der jeweiligen Form ausrechnet.
- ▶ Geben Sie den Flächeninhalt von beliebigen Kreisen und Rechtecken auf der Konsole aus. Erweitern Sie dafür die `main`-Methode der Klasse `App`.

Machen Sie sich weiter mit der IntelliJ IDE vertraut.

- ▶ Bauen Sie Fehler in Ihr Projekt ein und schauen Sie, was IntelliJ Ihnen dann mitteilt und welche Optionen Ihnen nach der Auswahl des Fehlers und betätigen der Tastenkombination `Alt+Enter` angeboten werden.
- ▶ Schauen Sie sich ihre Ausführungskonfiguration zur Klasse `App` an. (Rechts oben vom Dropdown die Option `Edit Configurations...` auswählen)

Machen Sie sich mit dem Konzept der Packages in Java vertraut.
Erarbeiten Sie sich dafür folgende Artikel im Selbststudium:

- ▶ [Java ist auch eine Insel, Kapitel 3.6](#)
- ▶ [Packages in Java and how to use them](#)

Verschieben Sie auf Basis Ihrer Erkenntnisse die bestehenden Klassen in passende Packages.

Importieren Sie noch vorhandene Java - Projekte in IntelliJ

- ▶ Schauen Sie nach, ob IntelliJ Ihnen Warnungen anzeigt
- ▶ Bringen Sie mindestens eins der alten Projekte dazu auch in IntelliJ ausführbar zu werden.
- ▶ Verschieben Sie bestehende Klassen in passende Packages.
- ▶ Debuggen Sie eines der Projekte mittels IntelliJ (Klick auf den Käfer oder Shift+F9)
- ▶ Führen Sie ggf. existierende Unit Tests in IntelliJ aus und betrachten Sie die Ausgabe. Bauen Sie einen Fehler ein und wiederholen Sie die Tests.

Geben Sie auf die Konsole eine Figur (Sanduhr) aus. Ihr Code soll dabei folgende Anforderungen erfüllen:

- ▶ Anzahl der Zeilen modulo 2 = 1
- ▶ Anzahl der Zeilen > 4
- ▶ Bei nicht Einhaltung dieser Bedingungen soll jeweils eine Fehlermeldung ausgegeben werden.
- ▶ Die Eingabe der Anzahl der Zeilen soll über ein Programmargument (args-Array) erfolgen. Sehen Sie sich dazu noch einmal die Ausführungskonfiguration von IntelliJ für Ihre Klasse an.

Beispielausgabe (Raute = Leerzeichen):

```
*****
#***
##*
#***
*****
```

Falls Sie in Aufgabe 5/6 eine iterative Lösung haben, versuchen Sie das Problem rekursiv zu lösen. Falls Sie in Aufgabe 5/6 eine rekursive Lösung erstellt haben, versuchen Sie es nun iterativ.