

Nutzung KleukersSEU



HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

Thema:	Umgang mit KleukersSEU
Autor:	Prof. Dr. Stephan Kleuker
Version/Datum:	1.0.3 / 19.1.2022
Empfänger:	Teilnehmende Personen der Lehrveranstaltungen von Prof. Dr. Stephan Kleuker

KleukersSEU steht für ein Zip-Verzeichnis und eine Installation auf den Rechnern der Hochschule, die wesentliche Entwicklungswerkzeuge (SEU = Software-Entwicklungsumgebung) umfasst, die in Lehrveranstaltungen von Prof. Kleuker genutzt werden. Diese Sammlung einzelner Werkzeuge ist entstanden, da viele Informatik-Veranstaltungen bestimmte Werkzeuge in speziellen Konfigurationen benötigen, die nicht immer kompatibel sind, wobei das Finden von Fehlern sehr aufwändig sein kann. Fehler in KleukersSEU sind leider auch möglich, sind dann direkt an den Autoren zu schicken. Generell sind die Werkzeuge aber so konfiguriert, dass sie unabhängig von anderen Installationen sind. Es wird bewusst Wert auf eine stabile Installation gelegt, so dass eine kontinuierliche Aktualisierung nur bei größeren Änderungen stattfindet.

Neben der Nutzung in der Hochschule steht KleukersSEU für Teilnehmer der Lehrveranstaltungen von Prof. Kleuker zur Verfügung. Es wird Windows 10 in der 64-Bit-Version benötigt. Die Nutzung erfolgt vollständig auf eigene Verantwortung, die Lizenzen der genutzten Werkzeuge sind zu beachten. Die Nutzung erfolgt vollständig auf eigene Gefahr.

Die Installation erfolgt durch einfaches Auspacken der Zip-Datei. Einzige, aber elementare Randbedingung ist, dass der *Installationspfad keine Leerzeichen enthalten* darf. Ein Auspacken unter „C:\Program Files“, auch irritierend als „C:\Programme“ ausgegeben, ist nicht möglich.

Man beachte, dass es sich hier nicht um eine vollständig portable Lösung handelt. Zwar ist KleukersSEU auch auf USB-Sticks auspack- und vollständig nutzbar, allerdings besitzt jedes Werkzeug Konfigurationsdateien, die im User-Ordner gespeichert werden. Dies ermöglicht die identische Nutzung von jedem Hochschulrechner aus, bedeutet aber, dass sich die Einstellungen auf privaten Rechnern von denen in der Hochschule unterscheiden können. Für eine vollständige Portabilität müssten diese Konfigurationsdateien ebenfalls lokal gespeichert werden, was durch Aufrufparameter und Einstellungsdateien jeder sich bei Bedarf konfigurieren kann.

Grundsätzlich erfolgt der Start der gegebenen Programme *ausschließlich* über die gegebenen „Start...bat“-Dateien.

 PROGRAMMSTART_NUR_MIT_STARTDATEIEN.txt	13.12.2019 12:01	TXT-Datei	0 KB
 StartBlueJ.bat	05.01.2020 15:20	Windows-Batchd...	1 KB
 StartDerby.bat	05.01.2020 15:22	Windows-Batchd...	1 KB
 StartDerbyJ.bat	16.01.2020 10:42	Windows-Batchd...	1 KB
 StartEclipse.bat	05.01.2020 15:25	Windows-Batchd...	1 KB
 StartKonsole.bat	16.01.2020 14:23	Windows-Batchd...	1 KB
 StartLiteIDE.bat	16.01.2020 15:15	Windows-Batchd...	1 KB
 StartSpin.bat	16.01.2020 10:40	Windows-Batchd...	1 KB
 StartSQLWorkbench.bat	05.01.2020 15:28	Windows-Batchd...	1 KB
 StartUMLet.bat	05.01.2020 15:29	Windows-Batchd...	1 KB
 StartVSCoDeGo.bat	16.01.2020 14:26	Windows-Batchd...	1 KB
 StopDerby.bat	05.01.2020 15:31	Windows-Batchd...	1 KB

Details zu den Werkzeugen können Links auf den jeweiligen Veranstaltungsseiten entnommen werden.

Veranstaltung	Werkzeuge
Programmierung 1	StartBlueJ: Erlernen von Java StartEclipse: professionelle Umgebung (falls Zeit dazu in der Veranstaltung)
Datenbanken	StartDerby: Start der genutzten Datenbank StopDerby: Stopp der genutzten Datenbank StartSQLWorkbench: Bearbeitung von Datenbanken StartUMLet: ER-Modellierung StartEclipse: Java-Entwicklung
Software-Qualität	StartEclipse: Java-Entwicklung
Logik	StartEclipse: Prolog-Entwicklung (SWI-Prolog eingebunden), Java-Entwicklung; es wird weiterhin Drools genutzt
Objektorientierte Analyse und Design	StartEclipse: Java-Entwicklung Start UMLet: UML-Modellierung
Software Quality Management	StartSpin: Nutzung des Model Checkers SPIN StartEclipse: Java-Entwicklung StartLiteIDE, StartVSCoDeGo: Go-Entwicklung

In der Eclipse-Version ist die Möglichkeit zur Entwicklung von C/C++-Programmen mit Hilfe von MinGW enthalten.

Unverändert gültige Lizenzen der genutzten Werkzeuge:

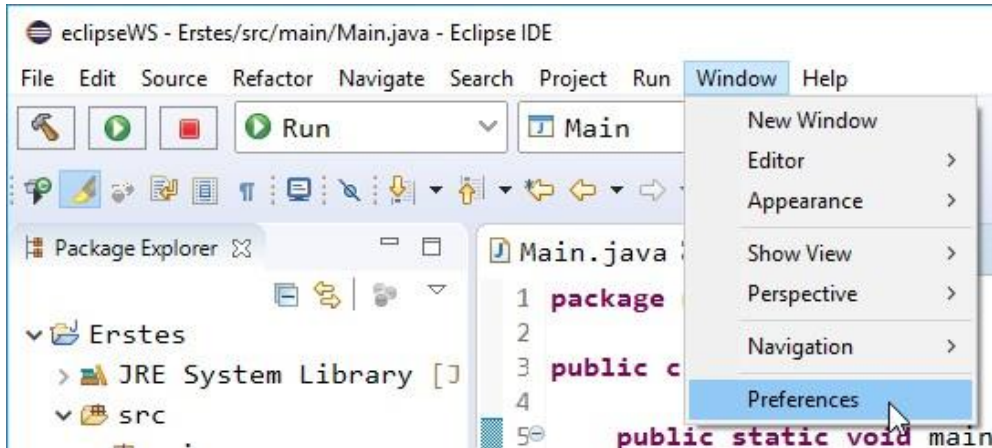
ActiveState ActiveTcl Community Edition	https://www.activestate.com/products/activetcl/downloads/ ActiveState Community Edition License Agreement https://www.activestate.com/wp-content/uploads/2018/10/activestate_community_edition_software_license_agreement.pdf
ANSI Escape in Console	http://marketplace.eclipse.org/content/ansi-escape-console Apache 2.0 https://www.apache.org/licenses/LICENSE-2.0
Apache Derby	https://db.apache.org/derby/ Apache 2.0 https://www.apache.org/licenses/LICENSE-2.0
BlueJ	https://www.bluej.org/ GNU GENERAL PUBLIC LICENSE Version 2 https://www.bluej.org/about/LICENSE.txt
ByteBuddy	https://bytebuddy.net/#/ Apache 2.0 licence / 3-clause BSD license https://www.apache.org/licenses/LICENSE-2.0.html https://asm.ow2.io/license.html
Checkstyle	https://checkstyle.org/eclipse-cs/#!/ GNU Lesser General Public License Version 2.1, February 1999 https://www.gnu.de/documents/lgpl-2.1.de.html
Drools	https://www.drools.org/ Apache License 2.0 https://www.drools.org/code/license.html
Eclipse	https://www.eclipse.org/ Eclipse Public License – v 2.0 https://www.eclipse.org/legal/epl-2.0/
Eclipse Metrics plugin continued	http://metrics2.sourceforge.net/ Eclipse Public License - v 2.0 https://www.eclipse.org/legal/epl-2.0/
Git	https://git-scm.com/ GNU General Public License version 2.0 http://opensource.org/licenses/GPL-2.0
Go	https://golang.org/ GO License https://golang.org/LICENSE

Javassist	https://www.javassist.org/ MOZILLA PUBLIC LICENSE Version 1.1 https://raw.githubusercontent.com/jboss-javassist/javassist/master/License.html
LiteIDE	http://liteide.org/en/ GNU Lesser General Public License v2.1, February 1999 https://github.com/visualfc/liteide/blob/master/LICENSE.LGPL
MinGW	http://www.mingw.org/ http://www.mingw.org/license https://sourceforge.net/p/mingw/mingw-org-wsl/ci/21762bb4a1bd0c88c38eead03f59e8d994349e83/tree/LICENSE
Prolog Development Tools	https://sewiki.iai.uni-bonn.de/research/pdt/docs/start Eclipse Public License v1.0 https://www.eclipse.org/org/documents/epl-v10.html
Spin	http://spinroot.com/spin/Src/index.html standard BSD 3-Clause open source license
SQLWorkbench/J	https://www.sql-workbench.eu/ modified version of the Apache License, Version 2.0 https://www.sql-workbench.eu/manual/license.html
SWI Prolog	https://www.swi-prolog.org/ Simplified BSD license https://www.swi-prolog.org/license.html
UMLet	https://www.umlet.com/ GNU General Public License http://www.gnu.de/documents/gpl.de.html
Visual Studio Code	https://code.visualstudio.com/ MICROSOFT-SOFTWARE-LIZENZBESTIMMUNGEN für MICROSOFT VISUAL STUDIO CODE (The source code is free and open source and released under the permissive MIT License. The compiled binaries are freeware and free for private or commercial use) https://code.visualstudio.com/license?lang=de
Zulu Community OpenJDK	https://www.azul.com/downloads/zulu-community/# All Zulu Community™ builds of OpenJDK are TCK-tested and free to download and use without restrictions https://www.azul.com/products/zulu-and-zulu-enterprise/zulu-terms-of-use/

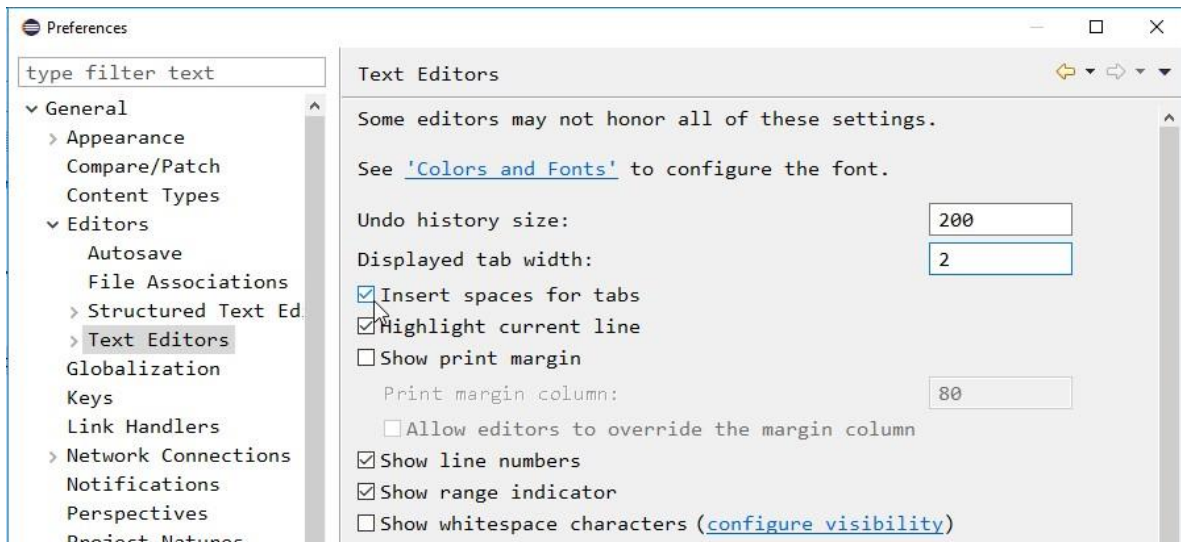
Typische Eclipse-Einstellungen

Wird Eclipse auf einem neuen Rechner genutzt oder nur der Workspace gewechselt, ist es sinnvoll einige einheitliche Einstellungen vorzunehmen, die hier kurz vorgestellt werden.

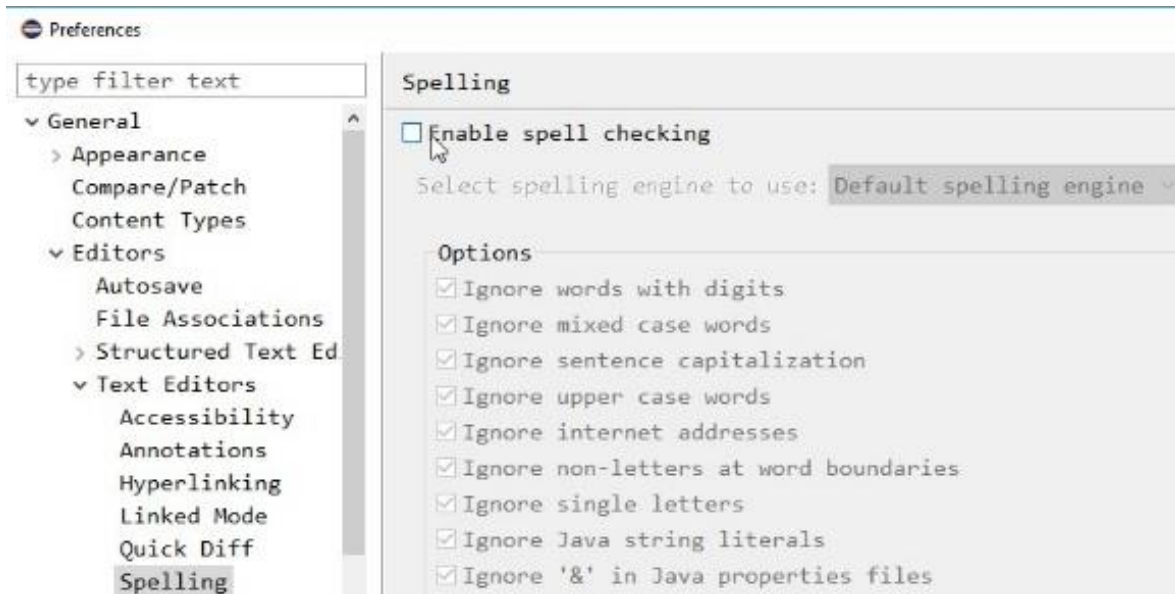
Generelle Einstellungen für alle Projekte werden unter „Window > Preferences“ vorgenommen.



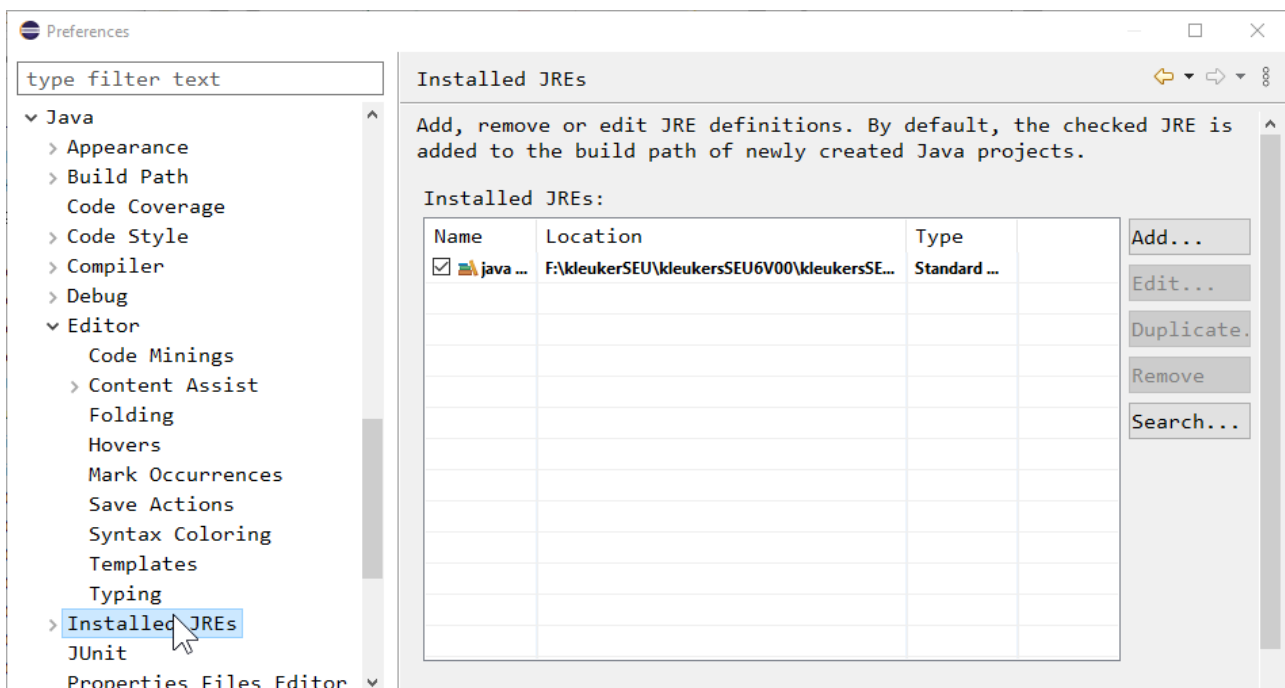
Unter „General > Editors > Text Editors“ ist einstellbar, dass keine Tabulatoren im Code verwendet werden („Insert spaces for tabs“). Weiterhin ist die Anzahl an Leerzeichen für Einrückungen über „Displayed tab with:“ einstellbar. Übliche Werte sind 2 oder 4. Ein Haken bei „Show print margin“ ist sinnvoll, um einen visuellen Hinweis für zu lange und damit unübersichtliche Zeilen zu bekommen. Die Zeilennummern („Show line numbers“) sollten immer eingestellt sein.



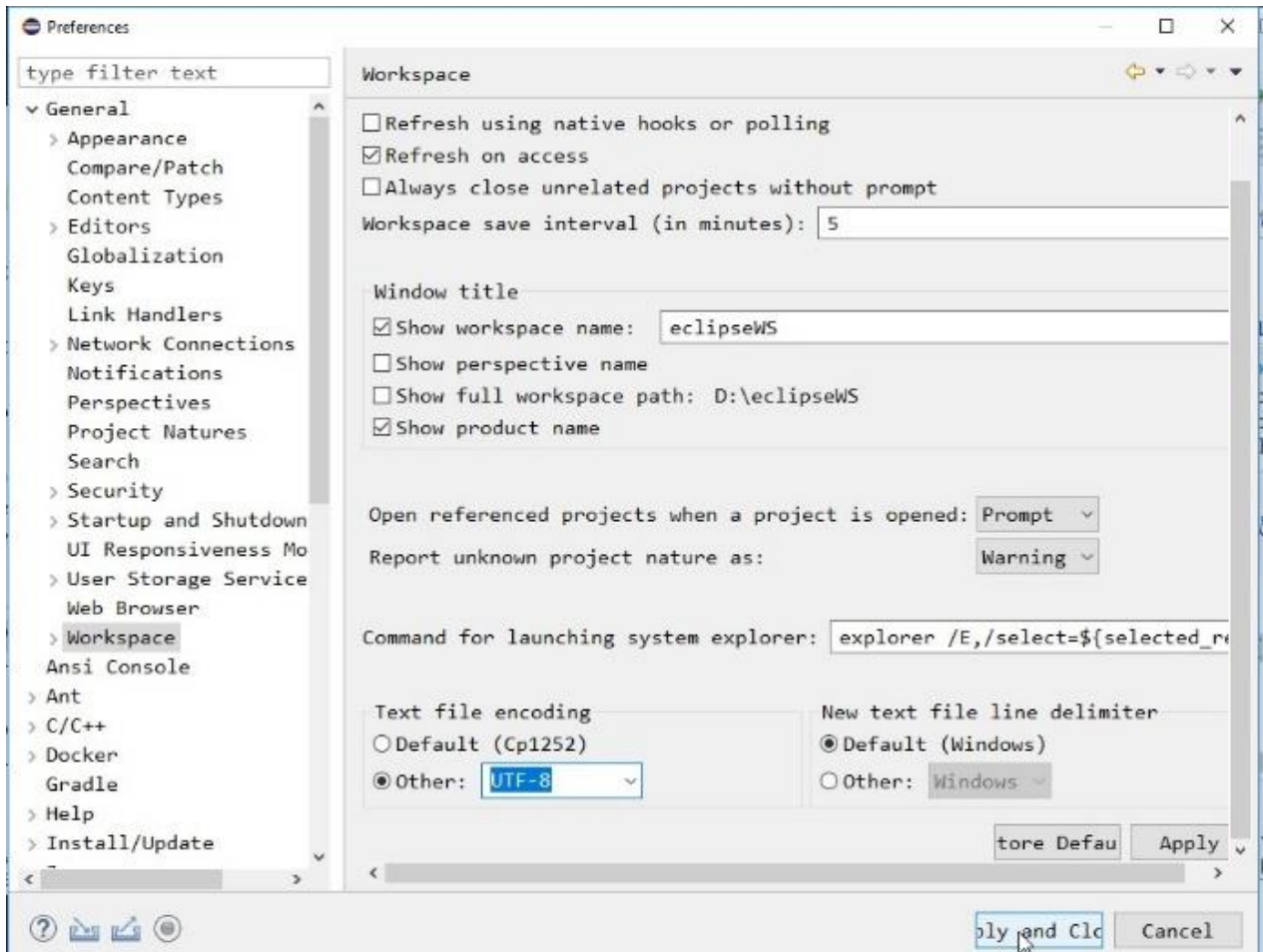
Der Haken für „Enable spell checking“ unter General > Editors > Text Editors > Spelling sollte nur gesetzt sein, wenn ein passendes Wörterbuch installiert ist. Da die Kontrolle Zeit benötigt kann generell darauf verzichtet werden.



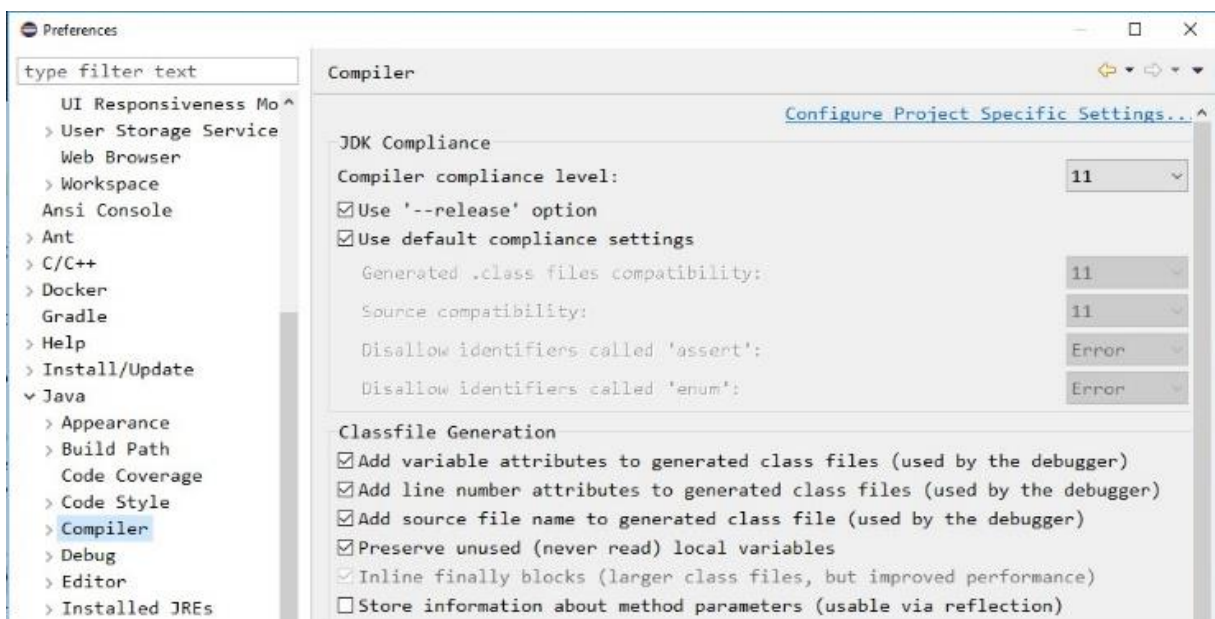
Unter Java > „Installed JREs“ muss die aktuell von KleukersSEU verwendete JRE eingestellt sein. Sollten mehrere JREs benötigt werden, ist der Default-Auswahlhaken für neue Projekte bei der zur SEU gehörenden JRE zu setzen.



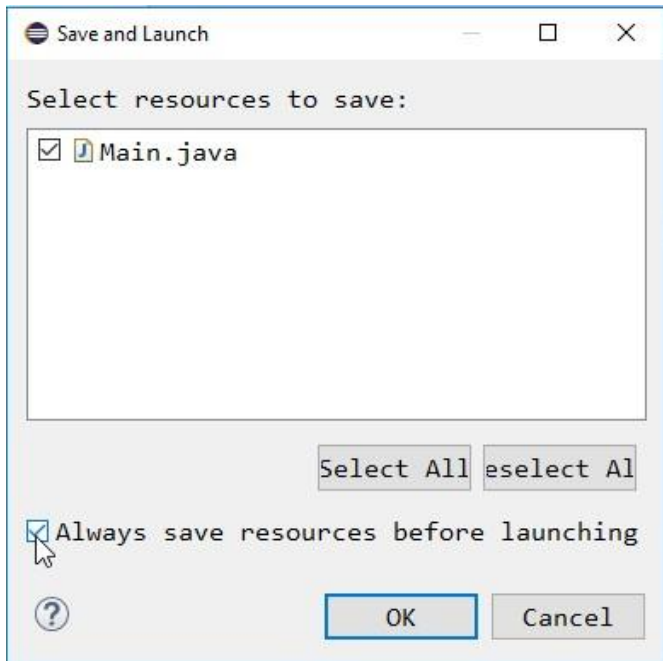
Eine sehr wichtige Einstellung ist die verwendete Zeichenkodierung mit der Textdateien, also auch Quellcode, abgespeichert werden. Um möglichst flexibel zu sein, sollte unter General > Workspaces links-unten unter „Text file encoding“ der Eintrag auf „Other“ und in der Auswahl auf „UTF-8“ zu setzen.



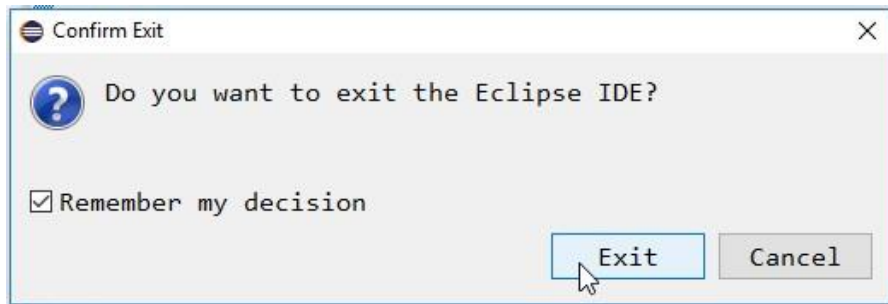
Unter Java > Compiler ist zu prüfen, dass der „Compiler compliance level:“ auf 11 oder höher steht.



Wird ein nicht gespeicherte Java-Klasse übersetzt, ist der Haken bei „Always ...“ sinnvoll.

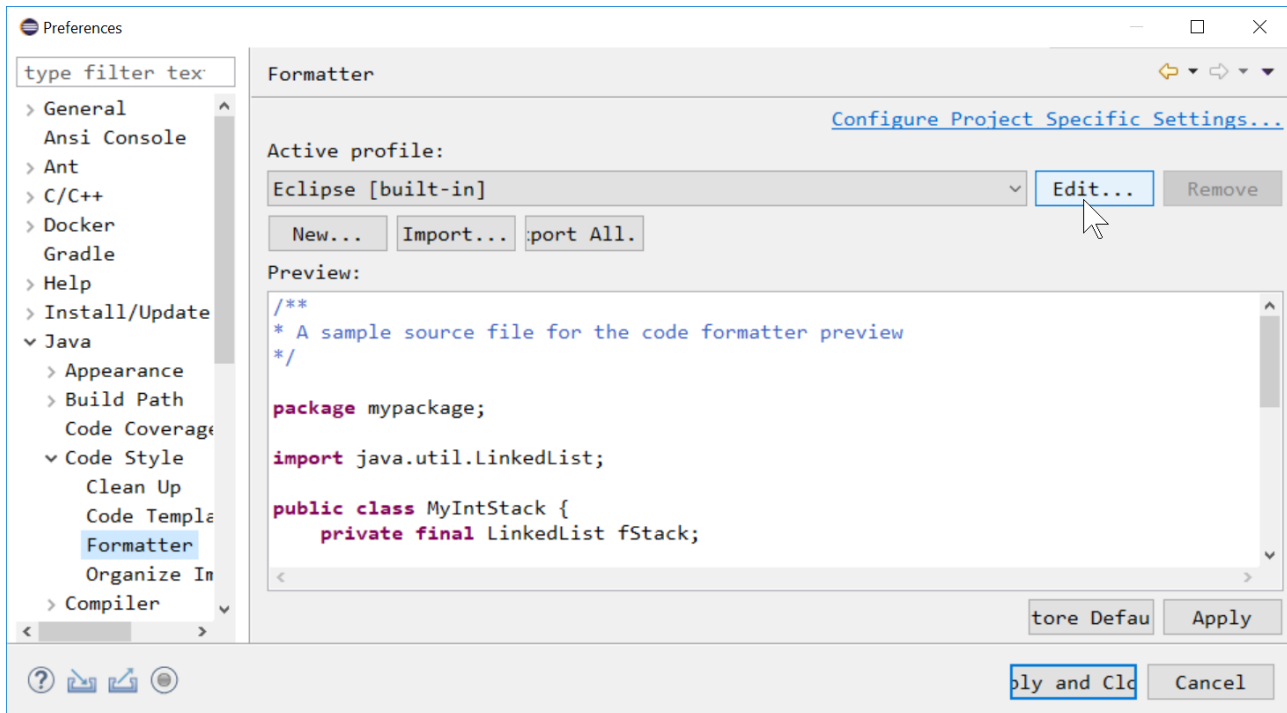


Beim Schließen von Eclipse ist der Haken bei „Remember my decision“ sinnvoll.“



Weitere Java-Formatierungseinstellungen

Die folgenden Einstellungen können sinnvoll sein, generell sollten in größeren Projekten hier gemeinsame Einstellungen vorgegeben werden. Weitere Details werden unter „Window > Preferences > Java > Code Style > Formatter > Edit...“ eingestellt.



Links sind Einstellungen auswählbar, rechts werden Beispiele gezeigt. Hier wird die „Tab policy“ auf „Spaces Only“, „Indentation size“ auf 2, „Tab size“ auf 2, unter „Line Wrapping“ die „Maximum line width“ auf 80 und der Haken bei „Never join already wrapped lines“ gesetzt. Weiterhin muss oben ein neuer „Profile Name“ vergeben werden, dabei besteht rechts die Möglichkeit die Einstellungen zu exportieren um sie in ein anderes Eclipse zu importieren. Die Schritte werden mit „OK“ und „Apply and Close“ abgeschlossen.

