

Technological foundations of software development

Master your development environment and software dependencies

ICM – Computer Science Major – Course unit on Technological foundations of computer science

M1 Cyber Physical and Social Systems – Course unit on CPS2 engineering and development, Part 2: Technological foundations of software development

Maxime Lefrançois <https://maxime-lefrancois.info>

online: <https://ci.mines-stetienne.fr/cps2/course/tfsd/>

Objectives of the session

Ensure that you are familiar with integrated development environments, as well as the use of dependencies on external versioned libraries.

Technological foundations of software development

Master your development environment and software dependencies

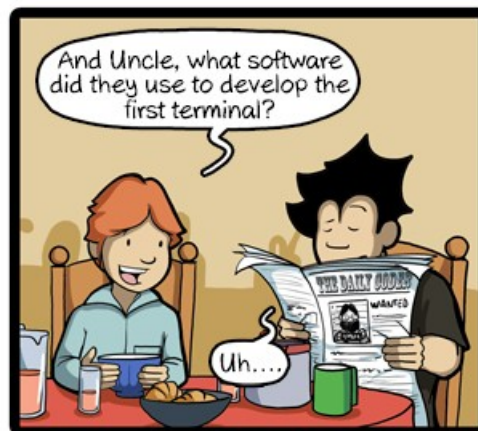
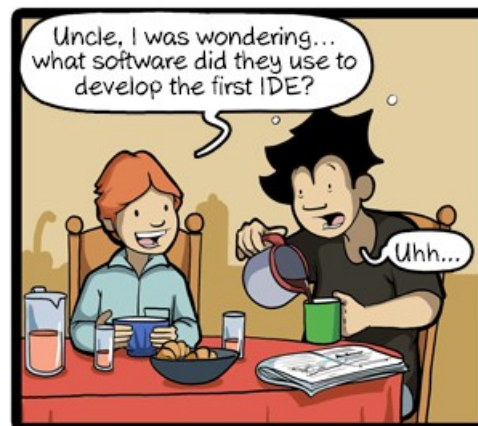
Part 1: Integrated Development Environments (IDEs)

ICM – Computer Science Major – Course unit on Technological foundations of computer scienceopment

M1 Cyber Physical and Social Systems – Course unit on CPS2 engineering and development, Part 2: Technological foundations of software development

Maxime Lefrançois <https://maxime-lefrancois.info>

online: <https://ci.mines-stetienne.fr/cps2/course/tfsd/>



IntelliJ IDEA



JetBrains, 2001,
one of the best Java IDEs starting 2010
written in **Java** and **Kotlin**
Community ed. Apache License 2.0 - Open source
Paid version - Trialware

NetBeans



Student (1996),
then Sun Microsystems (1999),
then Oracle Corporation (2010),
then Apache Software Foundation (2016)
written in **Java**
Apache License 2.0 - Open source

Eclipse



IBM (2001),
then Eclipse Foundation (2004)
written in **Java, C**
Eclipse Public License - Open source

Eclipse Che and Codenvy



Eclipse Che

Eclipse Foundation (2016),
Cloud-based IDE
written in **Java, Google Web Toolkit, Docker**
Eclipse Public License - Open source

<https://www.slant.co/topics/607/~best-java-ides-or-editors>

Slant

What are the best...

Log in · Sign up

ADD QUESTION

DEVELOPMENT JAVA IDE

FOLLOW

...

What are the best Java IDEs or editors?

15 OPTIONS
CONSIDERED

711 USER
RECS.

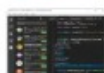
Sep 9, 2021 LAST
UPDATED

Visual Studio Code

Microsoft (2015)
written in **TypeScript, JavaScript,**
HTML, CSS
MIT License - Open source



THE BEST TOP 12 OPTIONS **WHY?**

BEST JAVA IDES OR EDITORS		PRICE	PLATFORMS	AUTO COMPLETE
93	 IntelliJ IDEA	FREE / PAID	Windows, Linux, Mac	-
84	 NetBeans	FREE	Windows, macOS, Linux	Yes
77	 Visual Studio Code	FREE	Windows, macOS, Linux	Yes
64	 Eclipse Che	FREE	Windows, Linux, Mac	-
--	 Codenvy	-	Browser-Based	Yes

IntelliJ IDEA

JetBrains, 2001,
one of the best Java IDEs starting 2010
written in **Java** and **Kotlin**
Community ed. Apache License 2.0 - Open source
Paid version - Trialware



NetBeans

Student (1996),
then Sun Microsystems (1999),
then Oracle Corporation (2010),
then Apache Software Foundation (2016)
written in **Java**
Apache License 2.0 - Open source



Android Studio

Google, JetBrains, 2013
Based on IntelliJ IDEA
freeware



Slant

What are the best...

Log in • Sign up

ADD QUESTION

DEVELOPMENT ANDROID

FOLLOW

...

What are the best IDEs for Android development?

15 OPTIONS
CONSIDERED

884 USER
RECS.

7 mins LAST
UPDATED

Visual Studio Code

Microsoft (2015)
written in **TypeScript, JavaScript,**
HTML, CSS
MIT License - Open source



THE BEST 2 OF 14 OPTIONS **WHY?**

BEST IDES FOR ANDROID DEVELOPMENT

PRICE

PLATFORMS

LATEST VERSION

93



IntelliJ IDEA

FREE / PAID

Windows, Linux, Mac

-

90



Android Studio

FREE

-

3.6 (Stable), 3.6 RC 3 (Preview)

82



NetBeans IDE

-

Windows, Linux, Mac, BSD

-

--



Visual Studio Code

FREE

Windows, macOS, Linux

-

--



JetBrains Rider

-

-

-

WebStorm

JetBrains, 2001,
written in **Java** and **Kotlin**
Community ed. Apache License 2.0 - Open source
Paid version – Trialware
WebStorm – For web, JavaScript and TypeScript
development. Many of JetBrains's other IDEs
include the feature set of WebStorm via plugins.



Slant



Log in • Sign up

ADD QUESTION

DEVELOPMENT

PRODUCTIVITY

FOLLOW



What are the best IDEs for Node.js?

23 OPTIONS
CONSIDERED

1K USER
RECS.

43 mins LAST
UPDATED

Vim

Bram Moolenaar (1991)
written in **C**
Vim License - Open source
<https://blog.talanlabs.com/vim-cest-bien-mais-vscode/>



Visual Studio Code

Microsoft (2015)
written in **TypeScript, JavaScript,**
HTML, CSS
MIT License - Open source



Atom

GitHub (2014)
written in **CoffeeScript,**
JavaScript, Less, HTML
MIT License - Open source

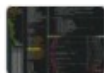


Sublime Text

Jon Skinner (2008)
written in **C++, Python**
Proprietary License



THE BEST TOP 19 OPTIONS **WHY?**

BEST IDES FOR NODE.JS		PRICE	PLATFORMS	MULTI LANGUAGE SUPPORT
90	 WebStorm	-	Windows; macOS; Linux	Yes
89	 Visual Studio Code	FREE	Windows, macOS, Linux	Yes
87	 Vim	-	Linux, macOS, Windows	-
76	 Atom	-	macOS, Linux	Yes
--	 Sublime Text	\$70	Windows; macOS; Linux	Yes

I RECOMMEND...



PyCharm

JetBrains, 2001,
written in **Java** and **Python**
Community ed. Apache License 2.0 - Open source
Paid version – Trialware
WebStorm – For web, JavaScript and TypeScript
development. Many of JetBrains's other IDEs
include the feature set of WebStorm via plugins.



Vim

Bram Moolenaar (1991)
written in **C**
Vim License - Open source
<https://blog.talanlabs.com/vim-cest-bien-mais-vscode/>



Spyder

Puierre Raybaut (2009)
written in **Python**
MIT License - Open source



Sublime Text

Jon Skinner (2008)
written in **C++**, **Python**
Proprietary License



Slant

What are the best...

Log in • Sign up

ADD QUESTION

DEVELOPMENT PRODUCTIVITY PYTHON

FOLLOW



What are the best free IDEs for Python?

23 OPTIONS
CONSIDERED

552 USER
RECS.

Sep 24, 2021 LAST
UPDATED

Visual Studio Code

Microsoft (2015)
written in **TypeScript**, **JavaScript**,
HTML, **CSS**
MIT License - Open source



THE BEST FOR 23 OPTIONS **WHY?**

BEST FREE IDES FOR PYTHON

PRICE

PLATFORMS

AUTO COMPLETE

91



PyCharm Community E...

-

-

-

88



Vim

FREE

Linux, macOS, Windows

-

--



Visual Studio Code

FREE

Windows, macOS, Linux

Yes

79



Spyder

-

-

Yes

--



Spacemacs with Pytho...

-

Windows, Unix, Mac

Yes

What are the best Arduino IDE alternatives?

8 OPTIONS
CONSIDERED

317 USER
RECS.

Sep 30, 2021 LAST
UPDATED

Arduino IDE



Arduino Software, 2005,
Written in **C and C++**
GNU General Public License 2 - Open source

PlatformIO



Ivan Kravets, 2014
Not an IDE – just a CLI + extensions
Written in **Python**
Useable as a plugin for:

- VSCode
- Eclipse
- Netbeans
- Atom
- Sublime Text
- ...

THE BEST 1 OF 7 OPTIONS WHY?

BEST ARDUINO IDE ALTERNATIVES		PRICE	PLATFORMS	LICENSE
91	CodeBlocks	-	-	-
--	Geany	FREE	Windows; OSX; *nix	GPL
62	platformio	FREE	-	-
--	Netbeans with Arduino ...	-	-	-
--	Emacs	-	Unix-like, Windows	GPL

A large number of programming
languages,
A large number of IDEs

Text editor



IDE

Slant ide for Log in • Sign up ADD QUESTION

DI  What are the best IDEs for C++ on Linux?

 What are the best IDEs for C++ on Mac OS X?

 What are the best IDEs for Node.js?

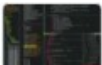
 What are the best IDEs for Android development?

 What are the best IDEs for C++ on Windows?

 What are the best IDEs for the Go programming language?

Can't find what you're looking for?

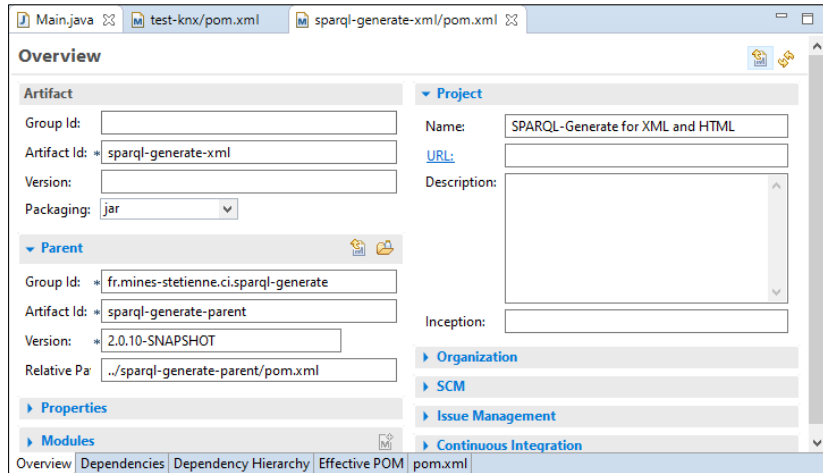
ADD QUESTION

90		WebStorm	-	Windows; macOS; Linux	Yes
89		Visual Studio Code	FREE	Windows, macOS, Linux	Yes
87		Vim	-	Linux, macOS, Windows	-
76		Atom	-	macOS, Linux	Yes
--		Sublime Text	\$70	Windows; macOS; Linux	Yes

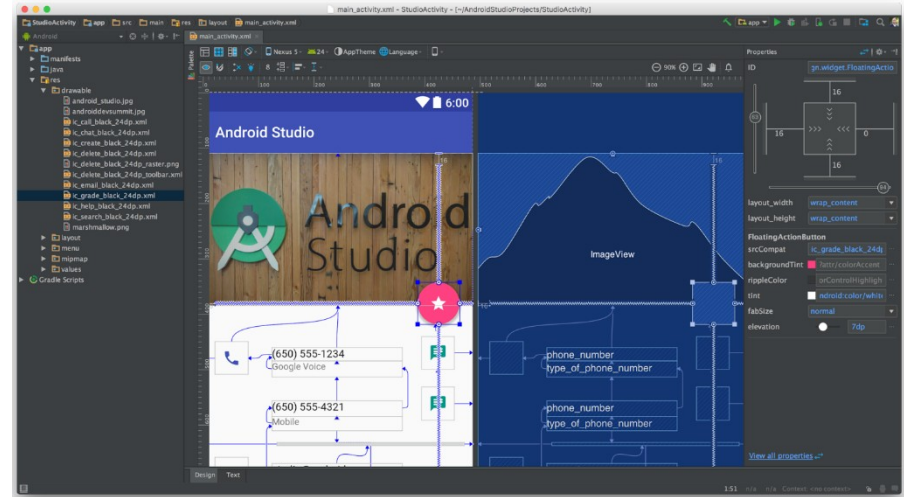
I RECOMMEND... ▼

Types of IDEs

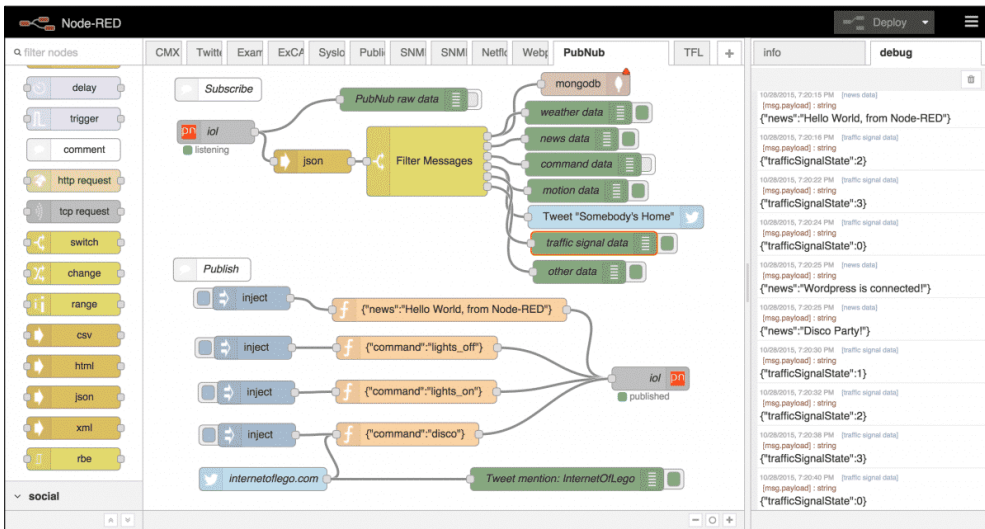
Visual programming



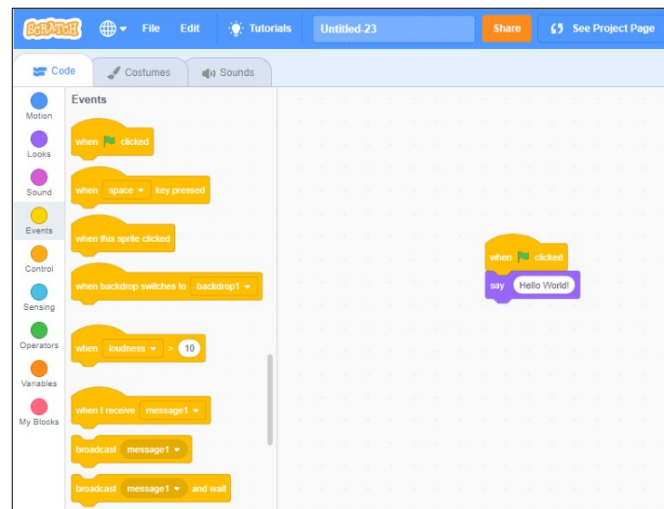
Formulaire to edit an XML document in Eclipse



Interface programming with Android Studio



Visual programming with Node-RED



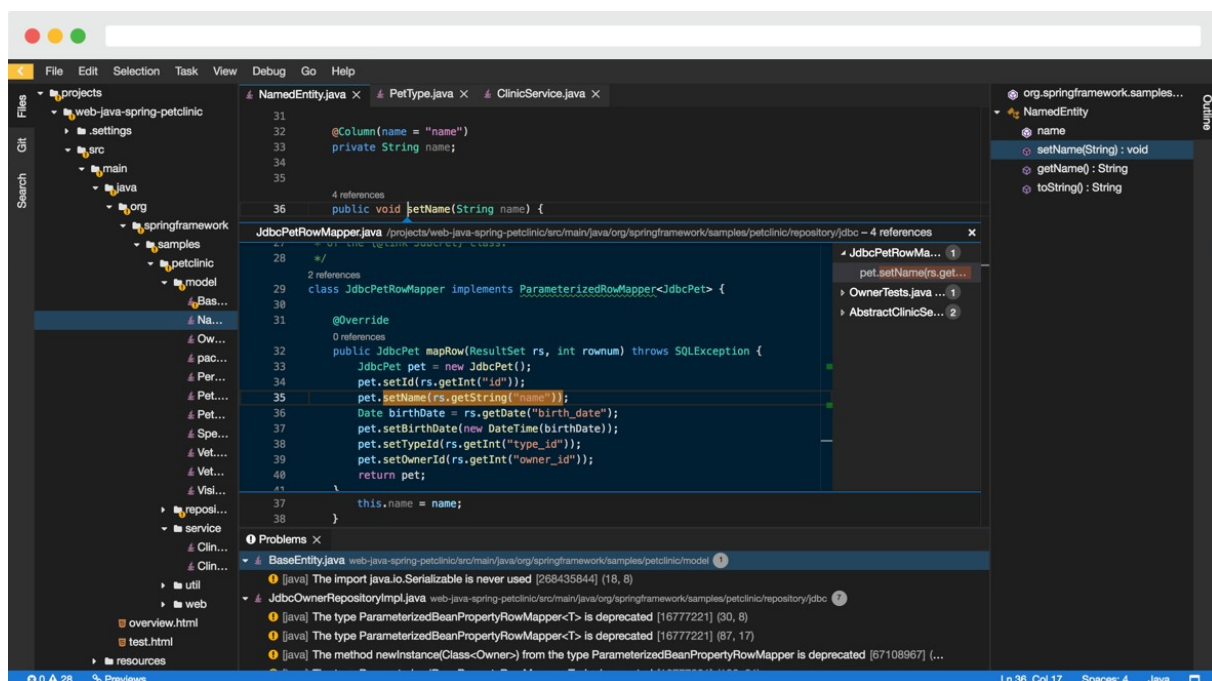
Visual programming for learning to program with Scratch

Types of IDEs

Web IDE

Actually to produce code

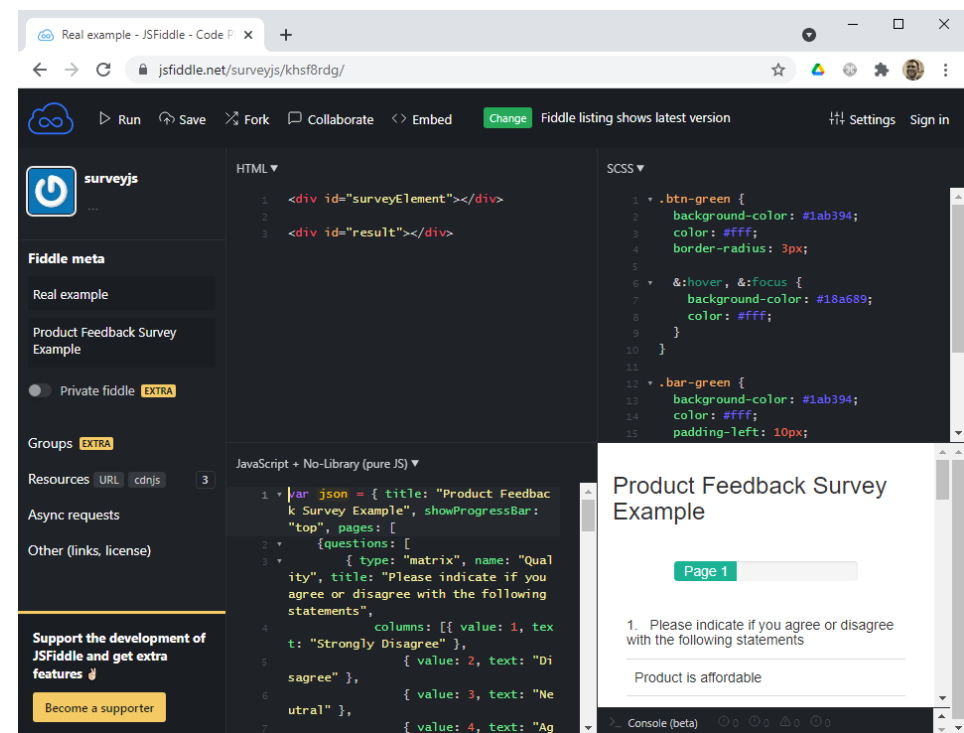
- Eclipse Che
- CodeAnywhere
- Replit
- Cloud9 IDE



Eclipse Che

Playgrounds and snippet sharing

- codesandbox.io
- codechef.com
- jsfiddle.com
- ideone.com

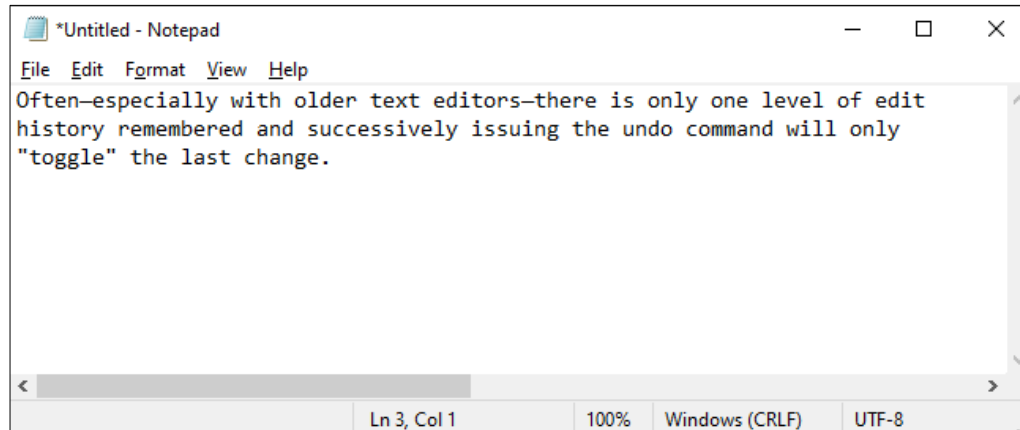


jsfiddle

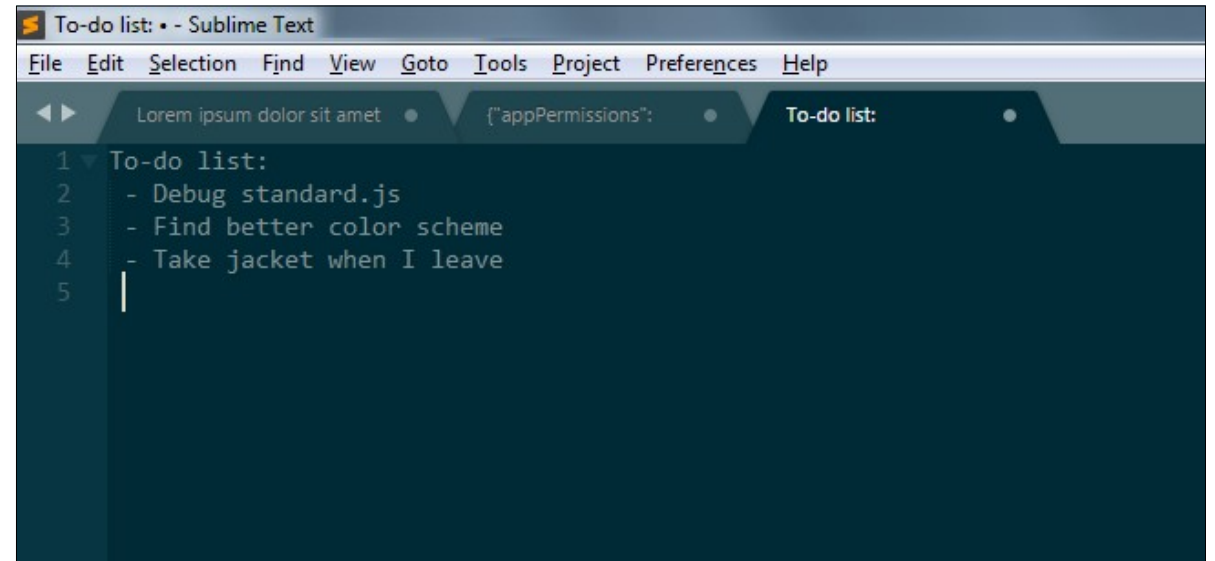
Text editors features

Undo- redo

- Stack of edits
- Unsaved changes



Stupid Undo – Redo with Notepad

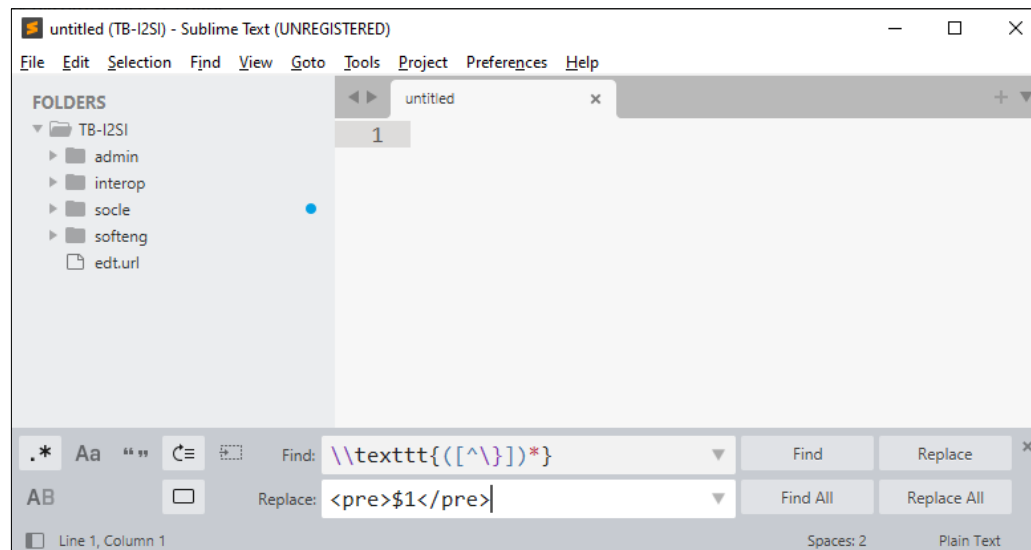


Unsaved files are available when you reopen Sublime Text

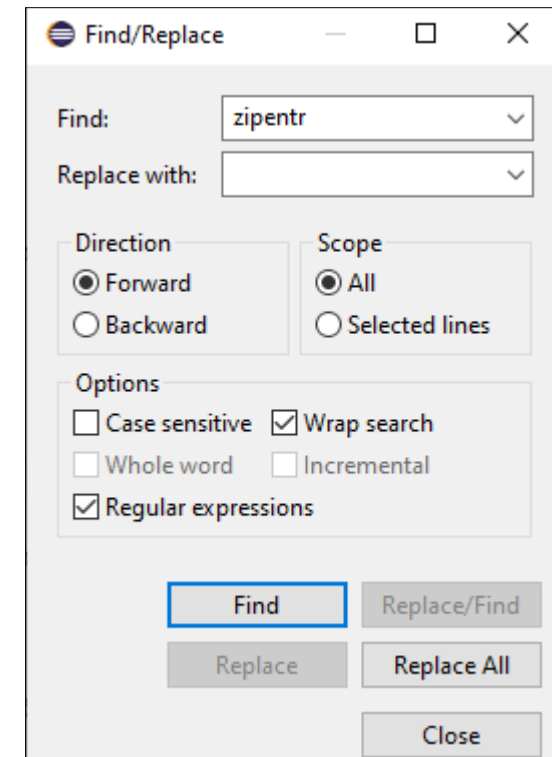
Text editors features

Find- Replace

- In file or set of files
- With REGEX - Regular expressions



Find – Replace with Sublime Text



Find – Replace with Eclipse



Find – Replace with IntelliJ IDEA

regex cheatsheet

Character classes

.	Any character, except newline
\w	Word
\d	Digit
\s	Whitespace
\W	Not word
\D	Not digit
\S	Not whitespace
[abc]	Any of a, b, or c
[a-e]	Characters between a and e
[1-9]	Digit between 1 and 9
[:print:]	Any printable character including spaces
[^abc]	Any character except a, b or c

Anchors

\G	Start of match
^	Start of string
\$	End of string
\A	Start of string
\Z	End of string
\z	Absolute end of string
\b	A word boundary
\B	Non-word boundary
^abc	Start with abc
abc\$	End with abc

Lookahead & Lookbehind

a(?=b)	Match a in baby but not in bay
a(?!b)	Match a in Stan but not in Stab
(?<=a)b	Match b in crabs but not in cribs
(?<!a)b	Match b in fib but not in fab

Escaped characters

\. * \\\	Escape special character used by regex
\t	Tab
\n	Newline
\r	Carriage return

Groups

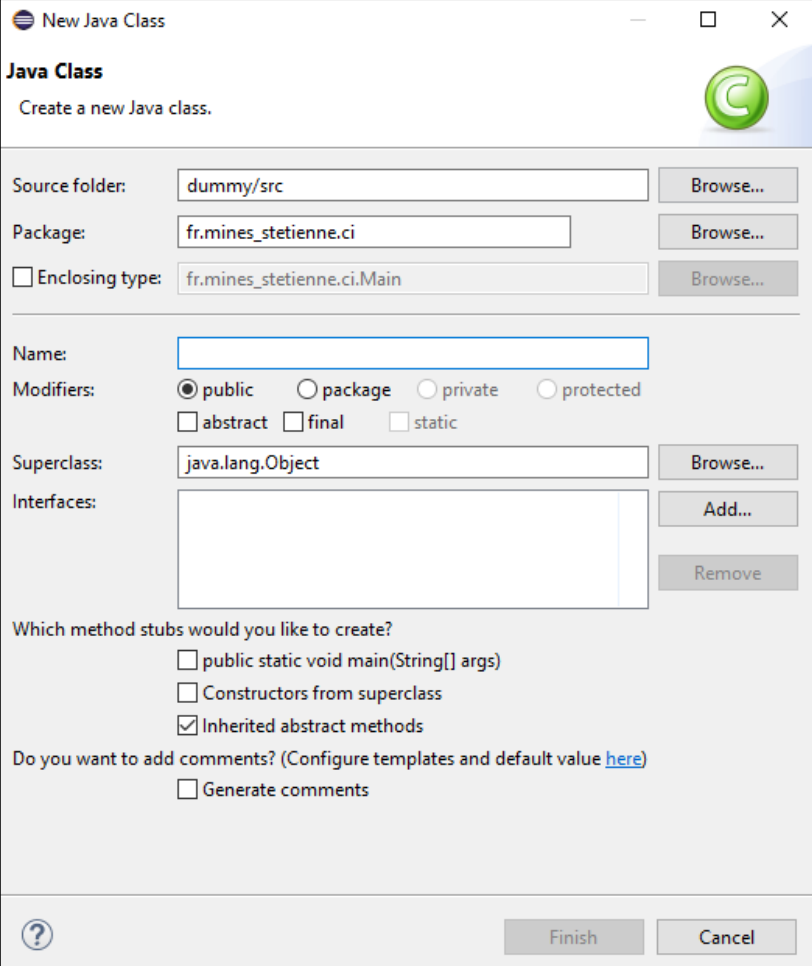
(abc)	Capture group
(a b)	Match a or b
(?:abc)	Match abc, but don't capture

Quantifiers

a*	Match 0 or more
a+	Match 1 or more
a?	Match 0 or 1
a{5}	Match exactly 5
a{, 3}	Match up to 3
a{3, }	Match 3 or more
a{1, 3}	Match between 1 and 3

Text editors features

Source code generation snippets



New Java Class

Java Class

Create a new Java class.

Source folder: Browse...

Package: Browse...

☐ Enclosing type: Browse...

Name:

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass: Browse...

Interfaces: Add... Remove

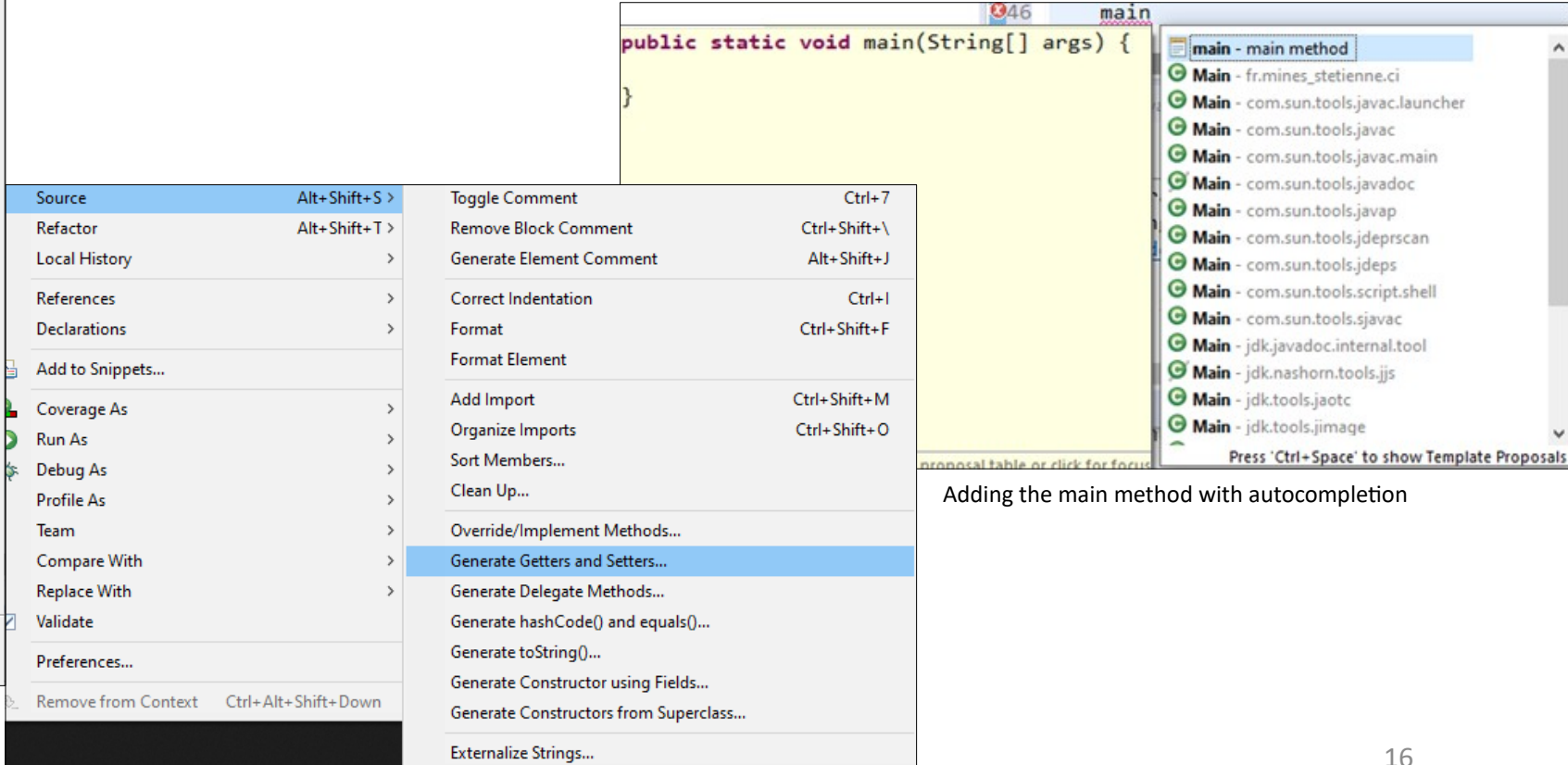
Which method stubs would you like to create?

☐ public static void main(String[] args)
☐ Constructors from superclass
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))
☐ Generate comments

Finish Cancel

File creation form



```
public static void main(String[] args) {  
}
```

main - main method

- Main - fr.mines_stetienne.ci
- Main - com.sun.tools.javac.launcher
- Main - com.sun.tools.javac
- Main - com.sun.tools.javac.main
- Main - com.sun.tools.javac.doc
- Main - com.sun.tools.javap
- Main - com.sun.tools.jdepscan
- Main - com.sun.tools.jdeps
- Main - com.sun.tools.script.shell
- Main - com.sun.tools.sjavac
- Main - jdk.javadoc.internal.tool
- Main - jdk.nashorn.tools.jjs
- Main - jdk.tools.jaotc
- Main - jdk.tools.jimage

Press 'Ctrl+Space' to show Template Proposals

Source Alt+Shift+S >
Refactor Alt+Shift+T >
Local History >
References >
Declarations >
Add to Snippets...
Coverage As >
Run As >
Debug As >
Profile As >
Team >
Compare With >
Replace With >
Validate >
Preferences...
Remove from Context Ctrl+Alt+Shift+Down

Toggle Comment Ctrl+7
Remove Block Comment Ctrl+Shift+\
Generate Element Comment Alt+Shift+J
Correct Indentation Ctrl+I
Format Ctrl+Shift+F
Format Element
Add Import Ctrl+Shift+M
Organize Imports Ctrl+Shift+O
Sort Members...
Clean Up...
Override/Implement Methods...
Generate Getters and Setters...
Generate Delegate Methods...
Generate hashCode() and equals()...
Generate toString()...
Generate Constructor using Fields...
Generate Constructors from Superclass...
Externalize Strings...

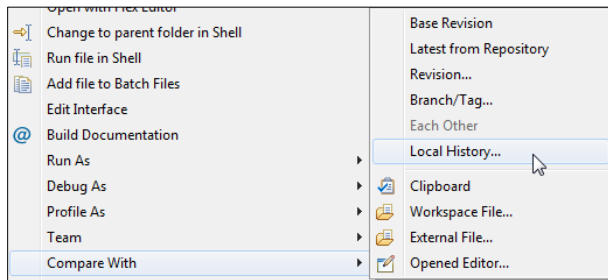
Adding the main method with autocompletion

Generation of frequent methods getters, setters, hashCode, equals, etc.

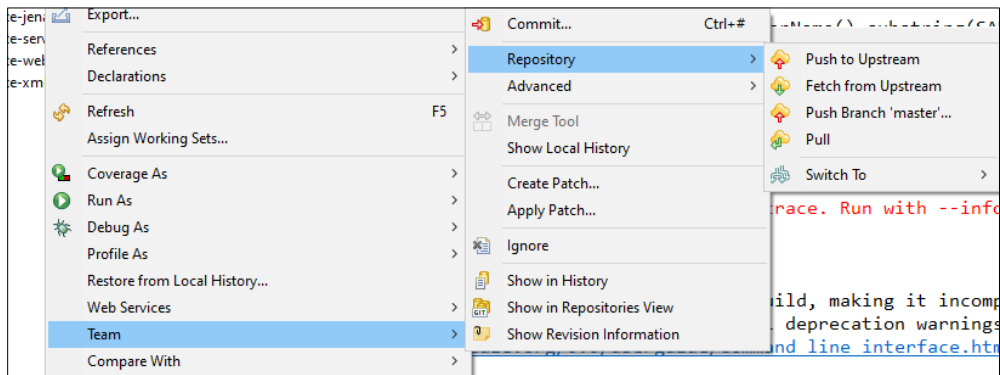
IDE features

Version control and file comparison

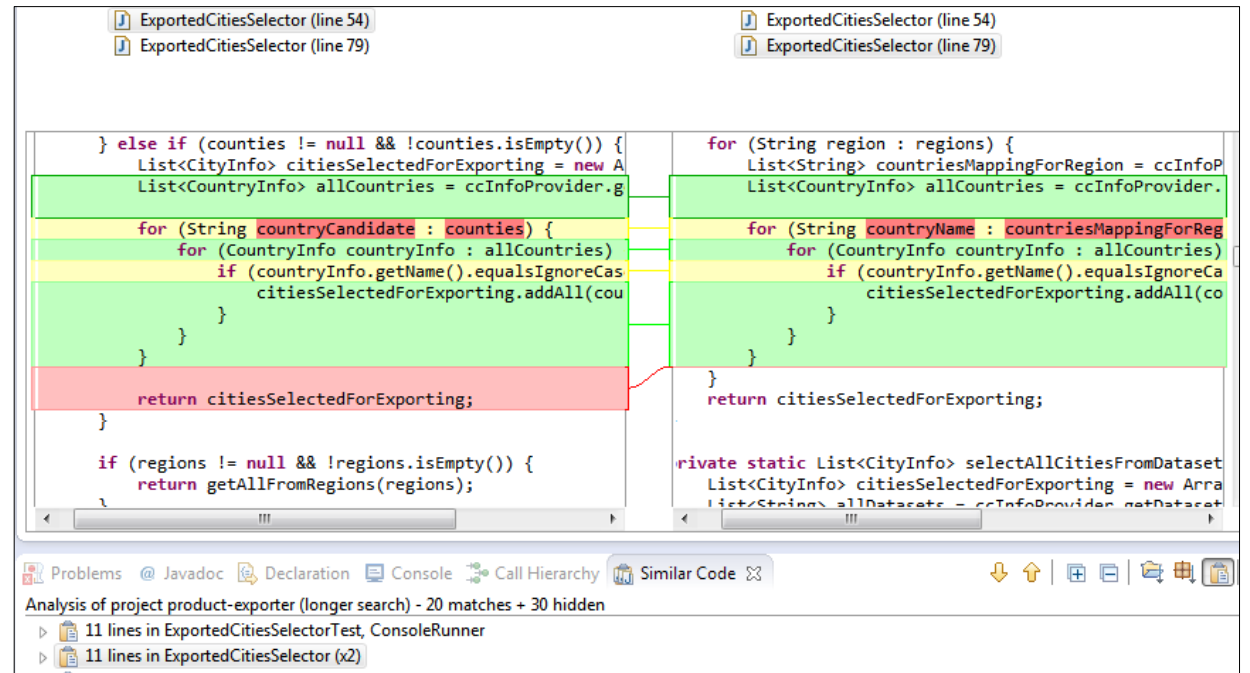
- Local history,
- Version Control System (VCS) integration
- File comparison utilities



Historique local avec Eclipse



Integration of git in Eclipse

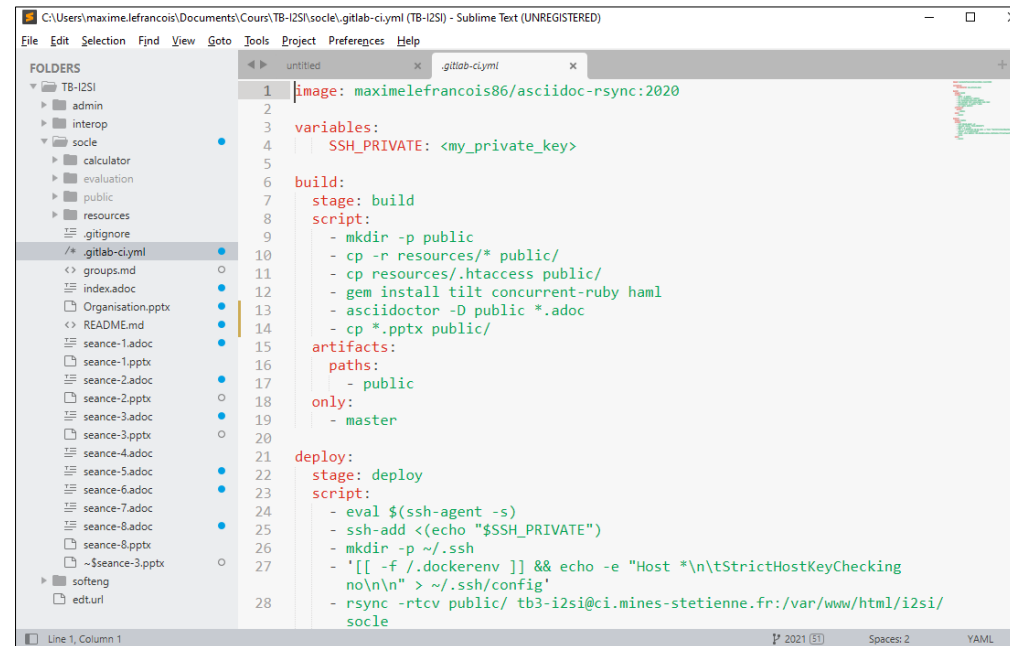


Comparing files in Eclipse

IDE features

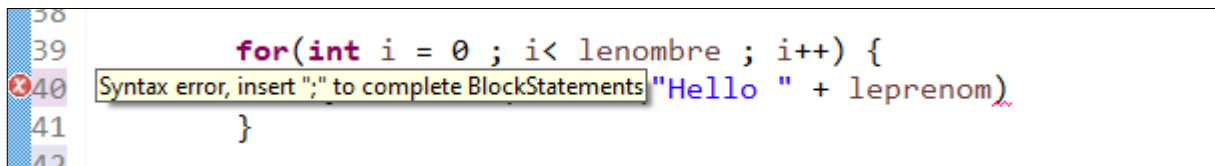
Syntax highlighting and validation

- Warnings and errors
- Linter
- Quick fixes



```
1 image: maximelefrancois86/asciidoc-rsync:2020
2
3 variables:
4   SSH_PRIVATE: <my_private_key>
5
6 build:
7   stage: build
8   script:
9     - mkdir -p public
10    - cp -r resources/* public/
11    - cp resources/.htaccess public/
12    - gem install tilt concurrent-ruby haml
13    - asciidoctor -D public *.adoc
14    - cp *.pptx public/
15
16 artifacts:
17   paths:
18     - public
19   only:
20     - master
21
22 deploy:
23   stage: deploy
24   script:
25     - eval $(ssh-agent -s)
26     - ssh-add <(echo "$SSH_PRIVATE")
27     - mkdir -p ~/.ssh
28     - '[[ -f /.dockerenv ]] && echo -e "Host *\n\tStrictHostKeyChecking\n\n" > ~/.ssh/config'
```

syntax highlighting with Sublime Text



```
38
39     for(int i = 0 ; i< lenombre ; i++) {
40         Syntax error, insert ";" to complete BlockStatements"Hello " + leprenom)
41     }
42
```

Syntax validation with Eclipse

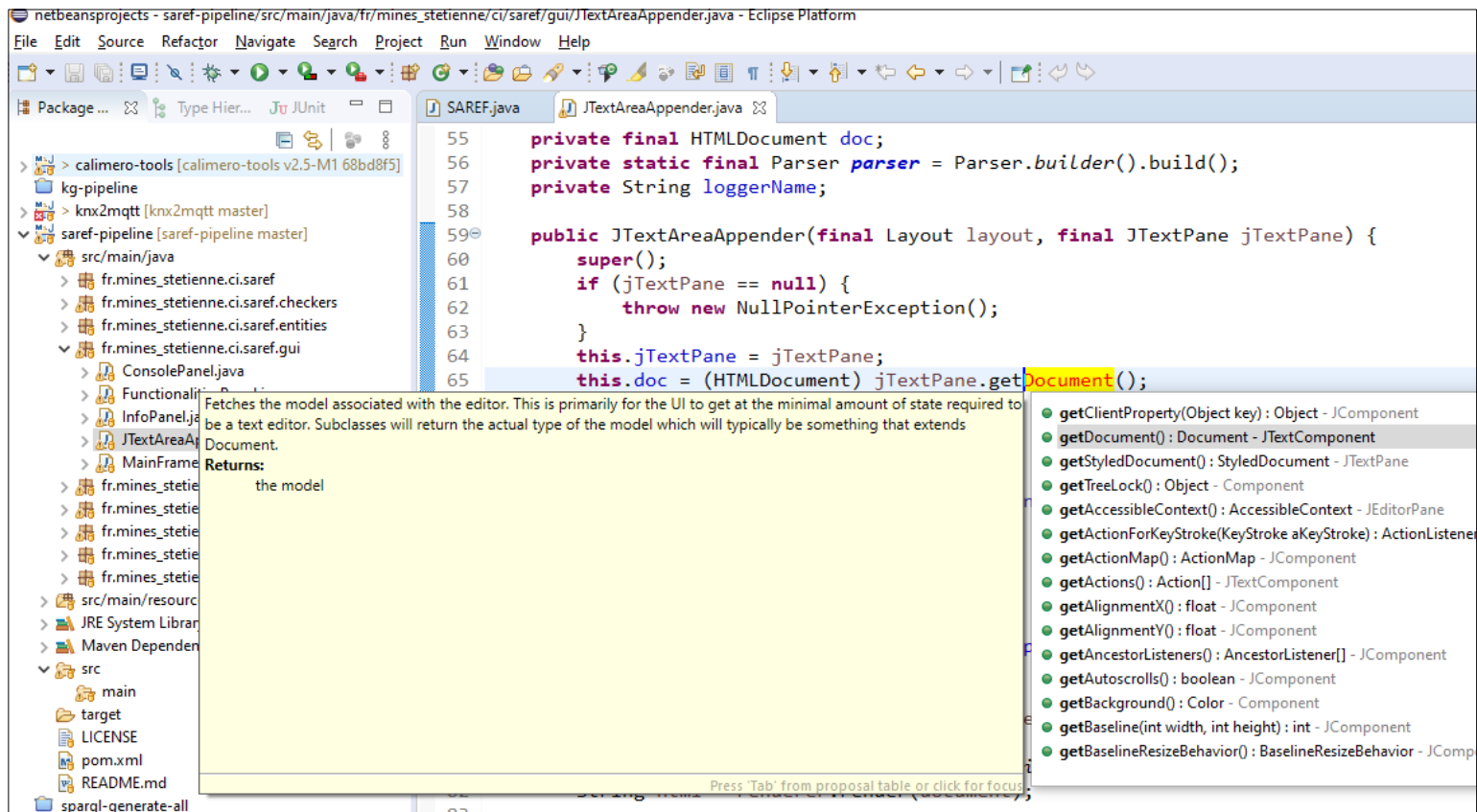


list of available grammars

IDE features

Code completion

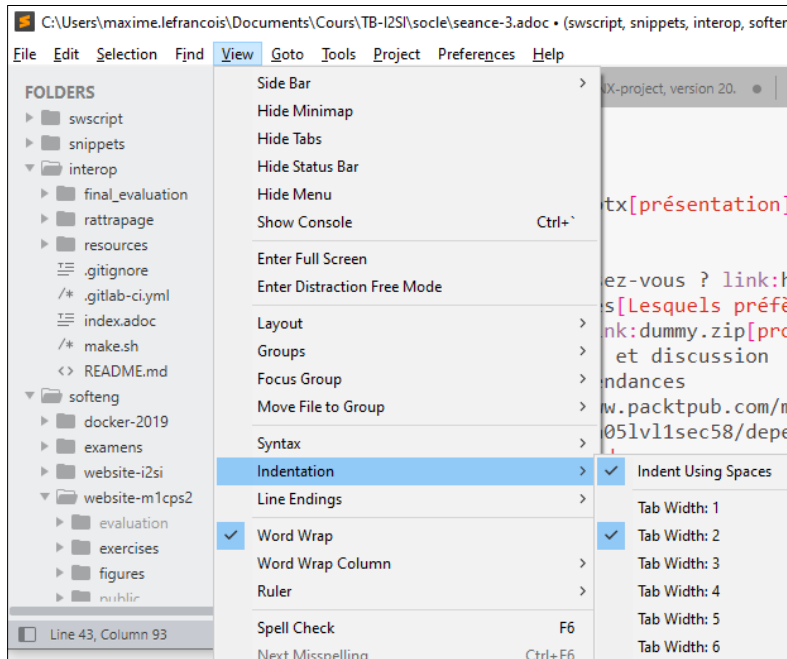
- More or less intelligent
- Based on current project or external libraries



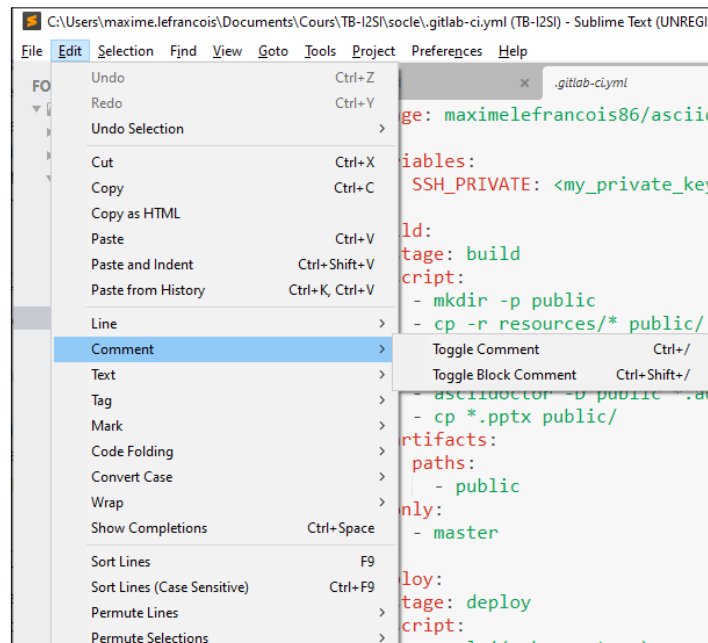
IDE features

Text formatting

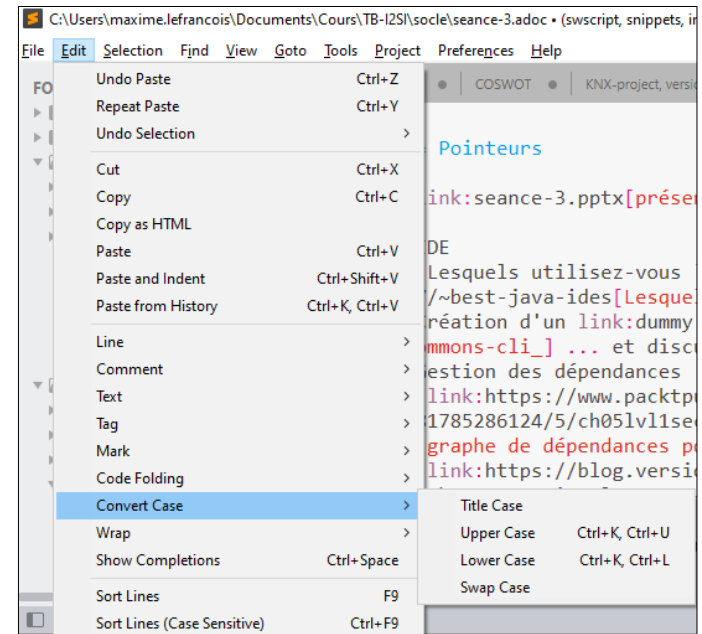
- Automatic line feed
- Auto-indentation tab/space
- Comment/uncomment



Tab vs space indentation



Comment/uncomment with Sublime Text

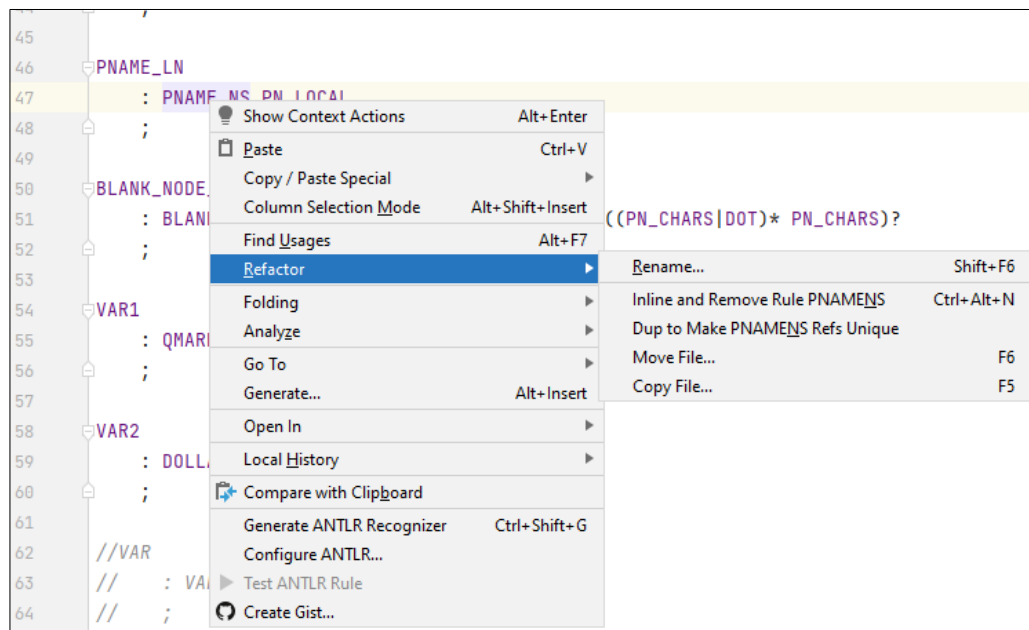


Case conversion

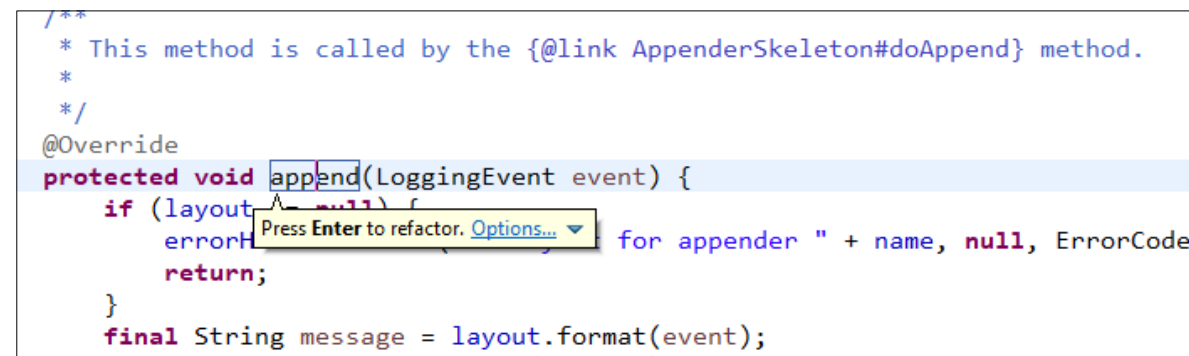
IDE features

Refactoring

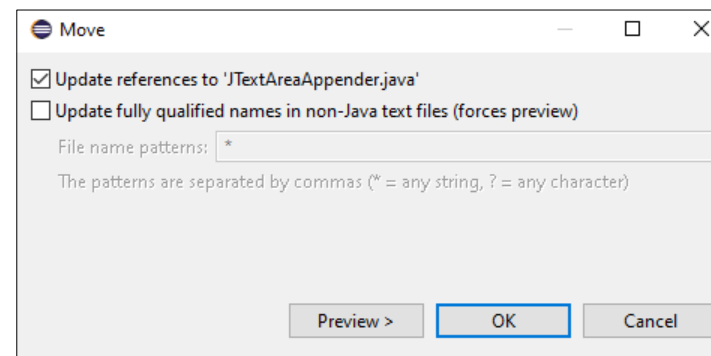
- Replacement of an identifier in the whole project
- identifier of variable, method, class, ...
- Change package of a java class, ...



Refactoring with IntelliJ: SHIFT+F6



in line refactoring with Eclipse: ALT+SHIFT+R

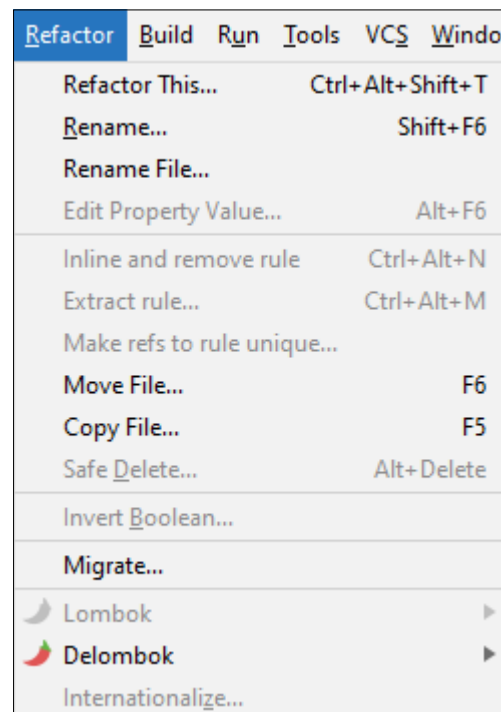


moving a Java class with Eclipse

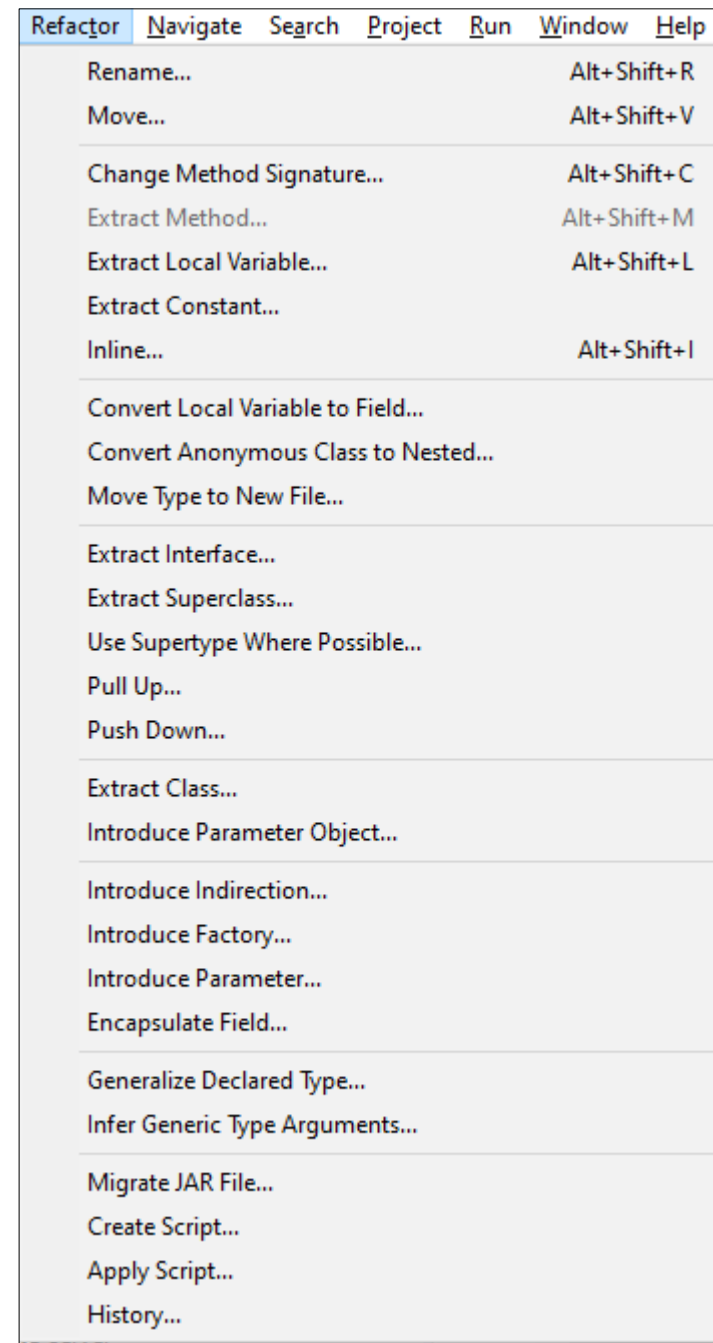
IDE features

Refactoring

- Replacement of an identifier in the whole project
- identifier of variable, method, class, ...
- Change package of a java class, ...



Refactor menu in IntelliJ

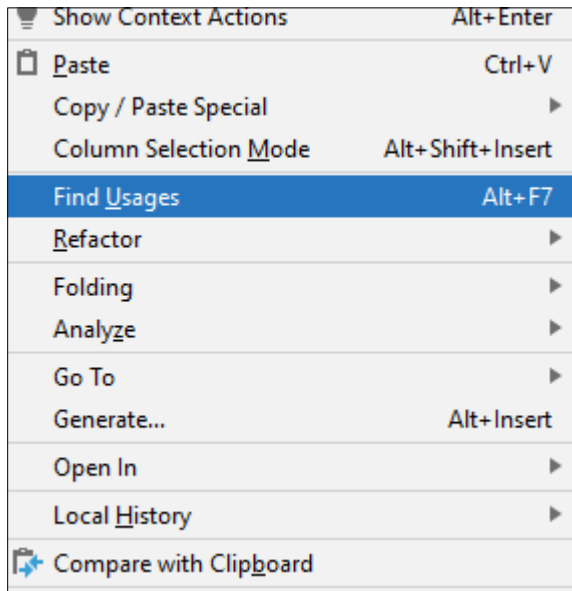


Refactor menu in Eclipse

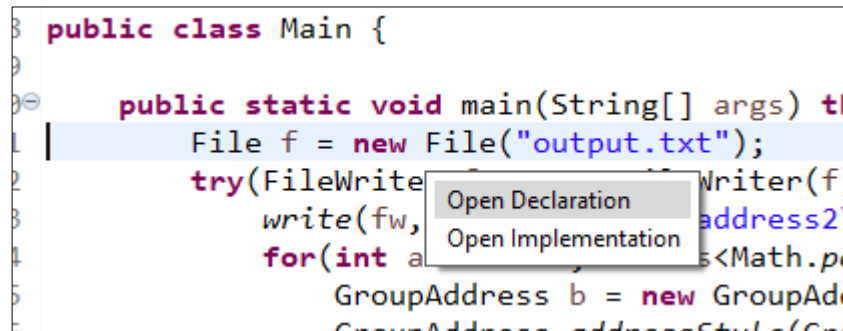
IDE features

Analysis and Navigation

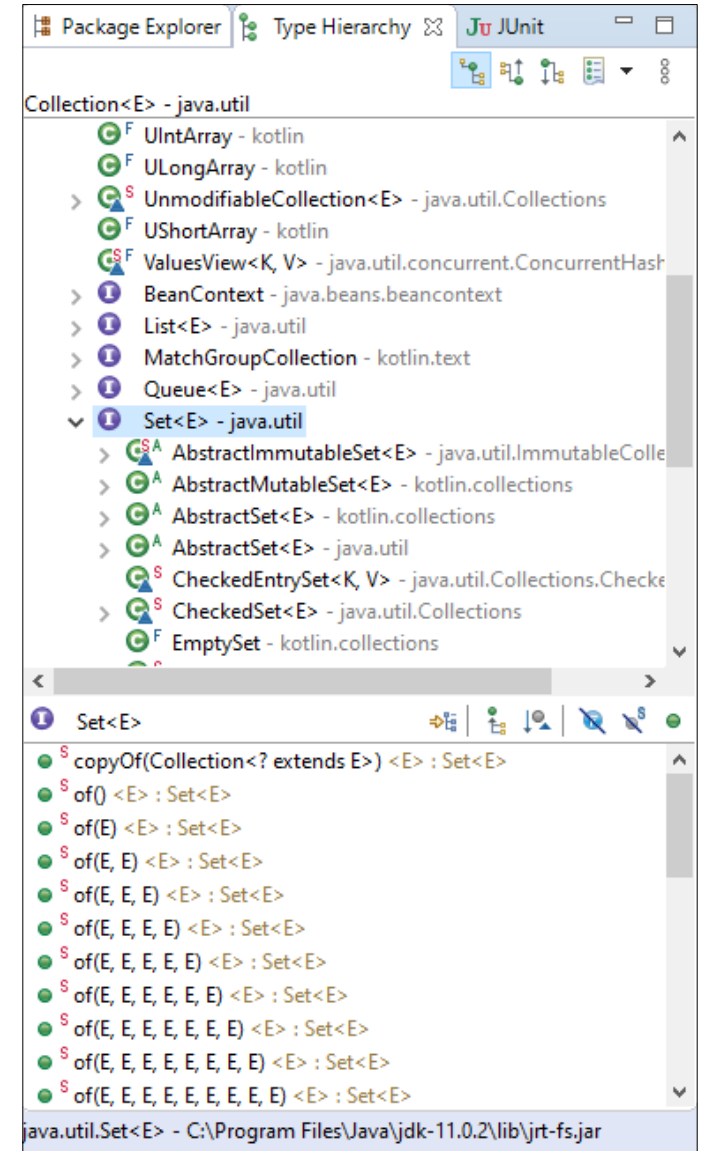
- CTRL+click to navigate to the declaration
- Viewing the hierarchy, viewing the members



IntelliJ – Find usages – ALT+F7



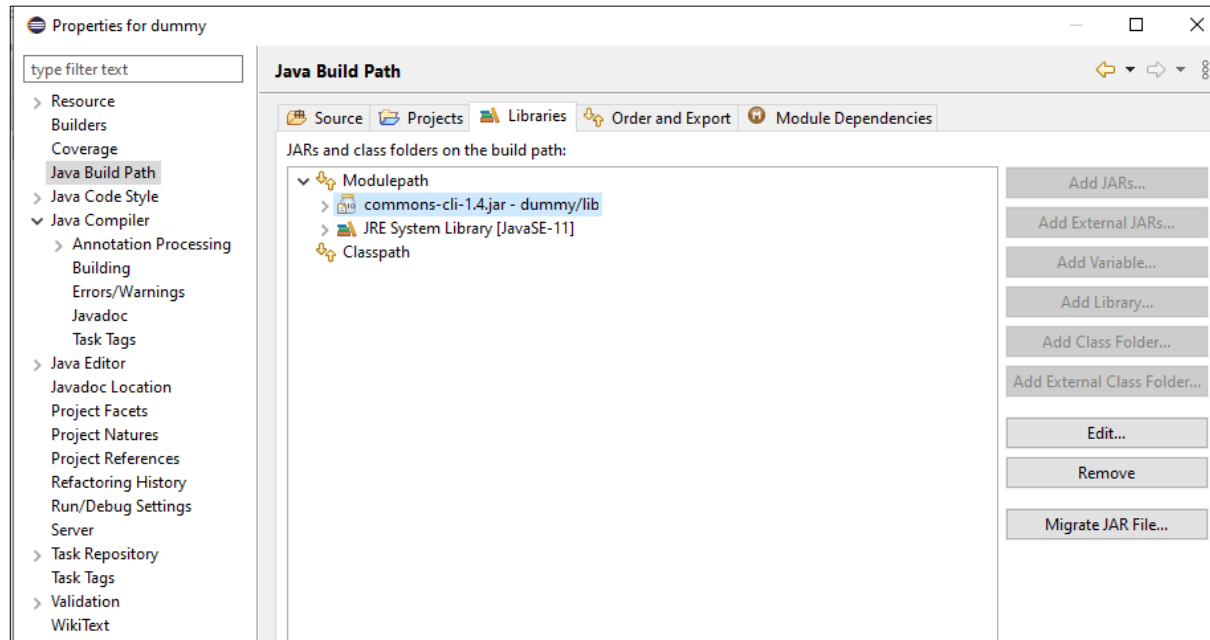
CTRL to display the navigation options



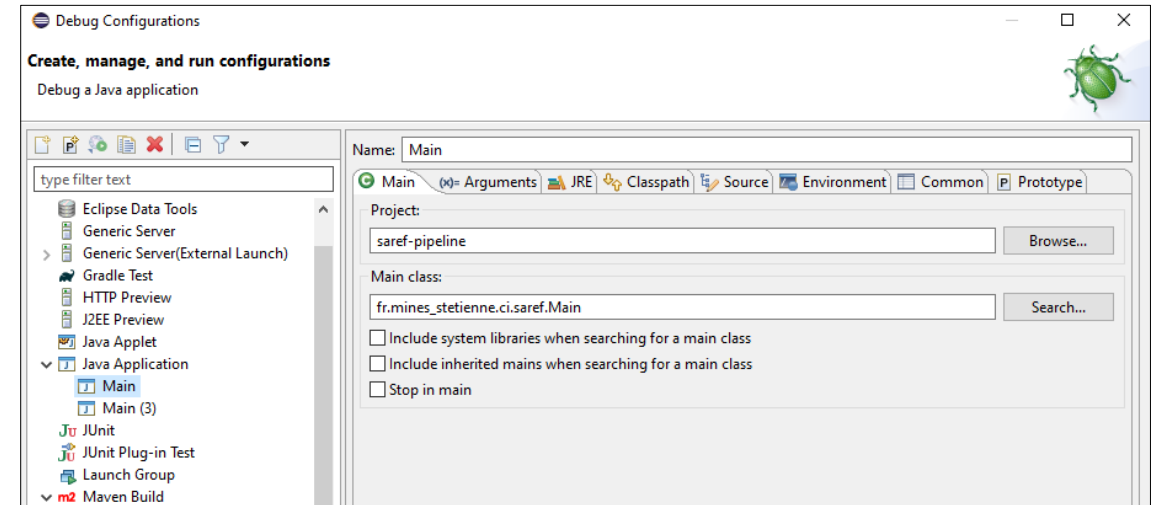
Eclipse Type Hierarchy and navigator pane - open for a class with F4

IDE features

Project configuration



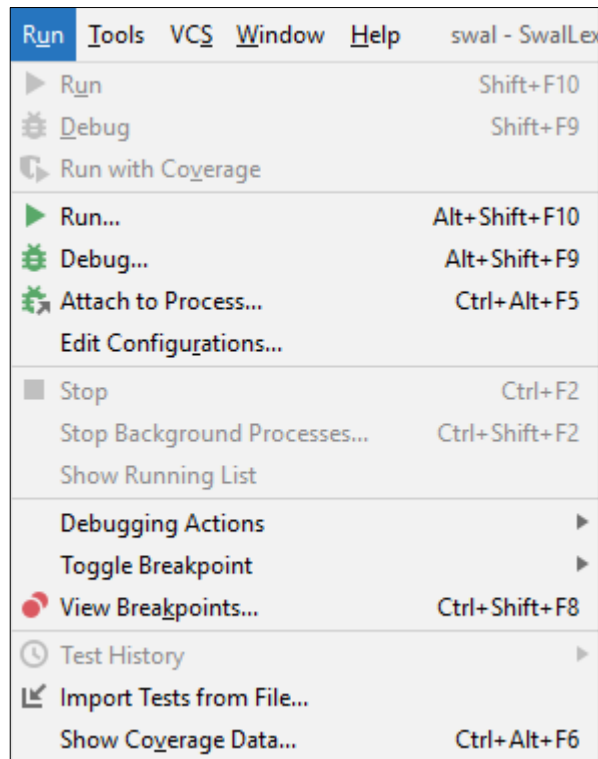
Add an external library to a java project in Eclipse



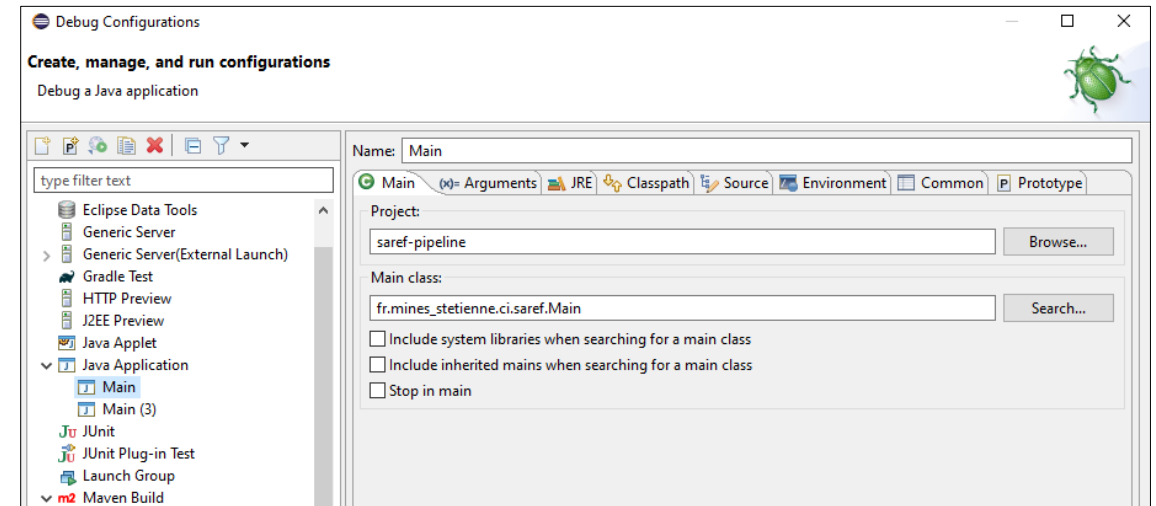
Execution configuration with Eclipse

IDE features

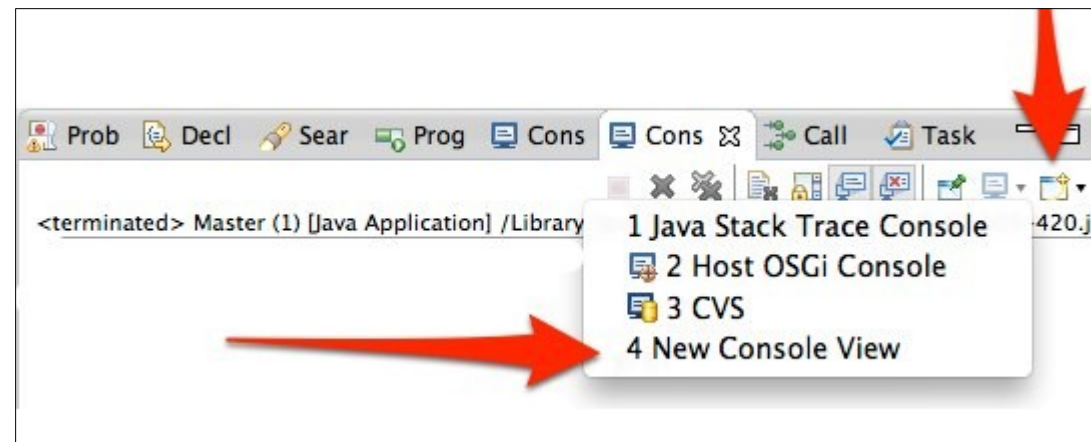
Run



IntelliJ Run menu



Execution configuration with Eclipse

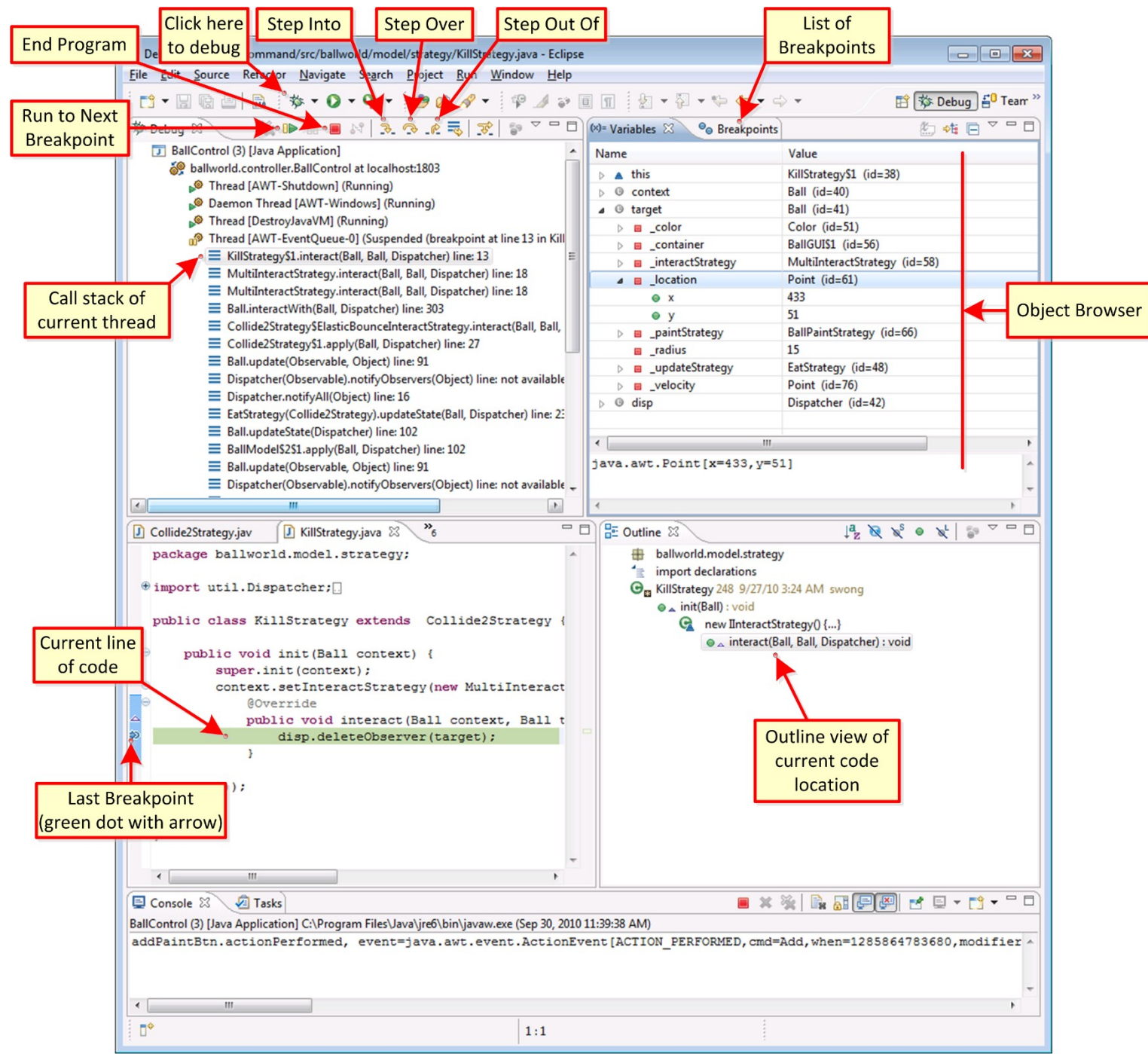


Eclipse console

IDE features

Debug

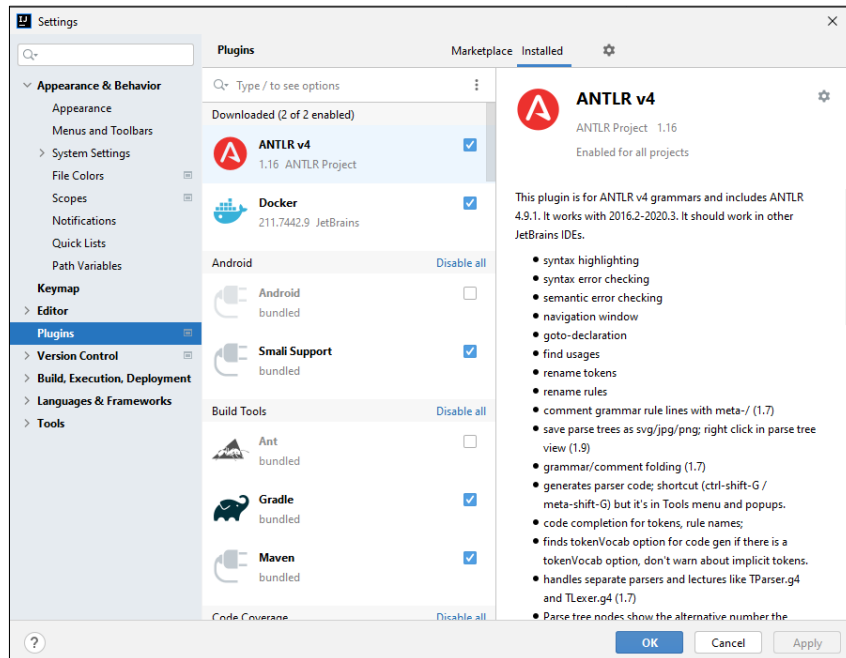
- Controlled execution of the program
- Monitoring and/or modification of the content of variables during execution
- Allows to trace bugs



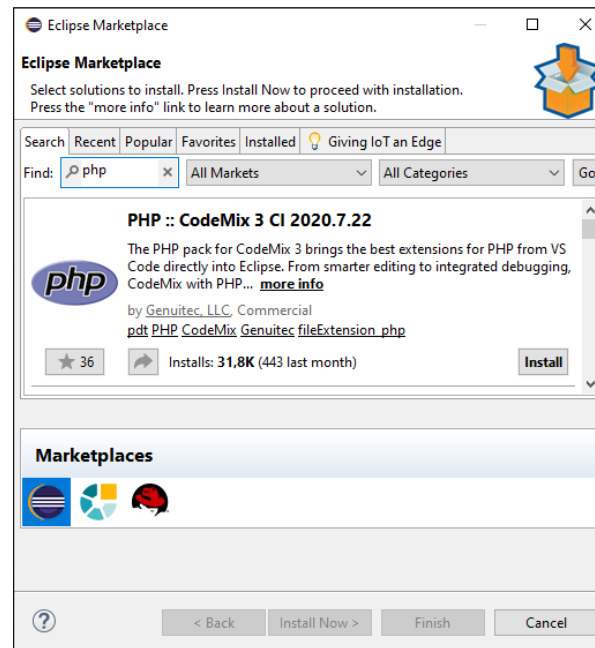
IDE features

Extensibility

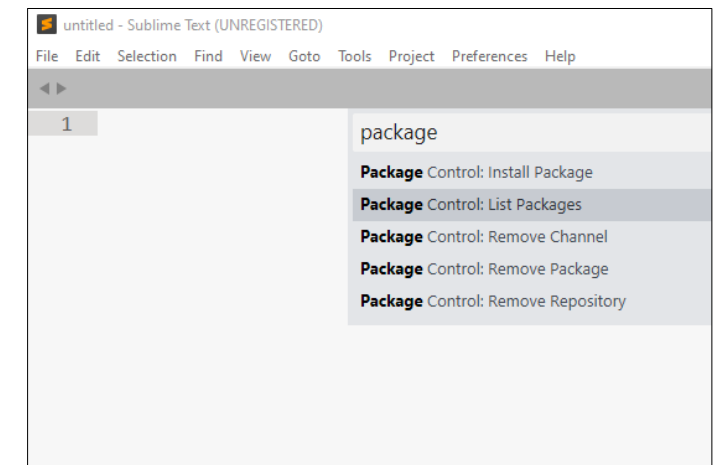
- Extension system, plugins, packages, ...



IntelliJ IDEA plugins



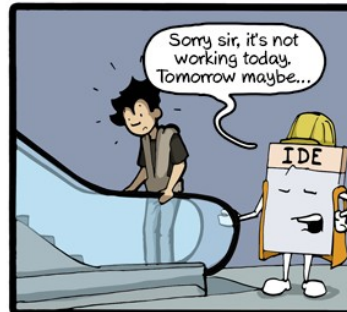
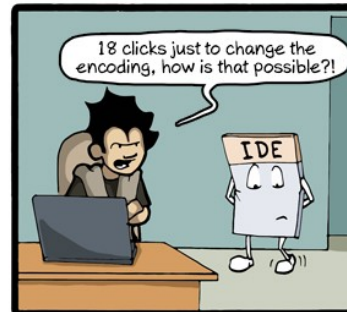
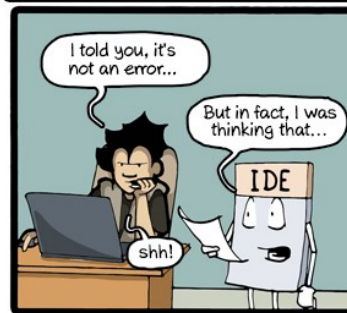
Eclipse Marketplace



Packages control in Sublime Text -> tool -> command palette



What's wrong with IDEs

When there are way too
many featuresWhen the IDE crashes because
of files changed externallyWhen the autocomplete
suddenly stops workingWhen a basic setting is buried
very very deep in the menusWhen the IDE underlines an
error which is not an errorWhen the IDE decides to
freeze at a critical moment

Technological foundations of software development

Master your development environment and software dependencies

Part 2: Versioned external libraries

ICM – Computer Science Major – Course unit on Technological foundations of computer scienceopment

M1 Cyber Physical and Social Systems – Course unit on CPS2 engineering and development, Part 2: Technological foundations of software development

Maxime Lefrançois <https://maxime-lefrancois.info>

online: <https://ci.mines-stetienne.fr/cps2/course/tfsd/>

External libraries

- Set of resources that can be reused
 - sources
 - compiled code
 - documentation

External libraries

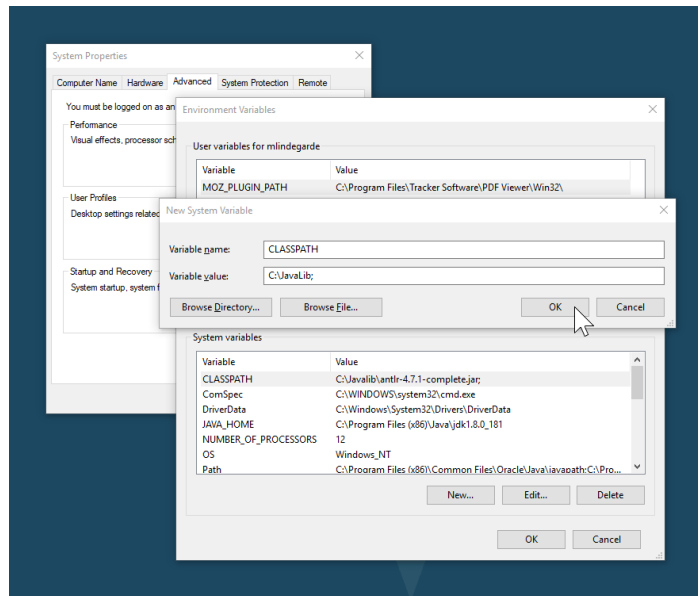
- Set of resources that can be reused
 - sources
 - compiled code
 - documentation
- Used in different phases of the software life cycle
 - For compilation (static library, *.lib, *.a)
 - For testing
 - For execution (dynamic library, *.dll, *.so)
 - At runtime (OSGi example)

Linking

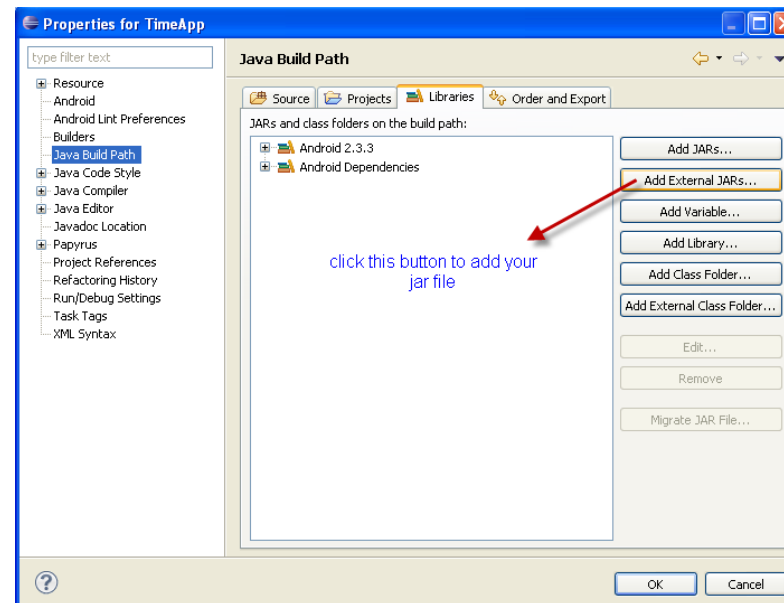
- Resolution phase of references to external libraries
- Static linking
 - copy the *.lib or *.a library during the creation of the executable
 - pro: more portable, no version conflicts
 - con: copy ***the whole library*** -> not optimized
- Dynamic linking
 - resolution and loading of the library at runtime
 - pro: save disk and memory space,
 - pro: in case of bug, just replace the *.dll or *.so
 - con: *DLL hell*: evolution and backward compatibility problems

With Java – class libraries

- The *Class Loader* resolves references with the Classpath
 - Compiled classes *.class, or archives *.jar
 - CLASSPATH environment variable
 - jvm option -classpath



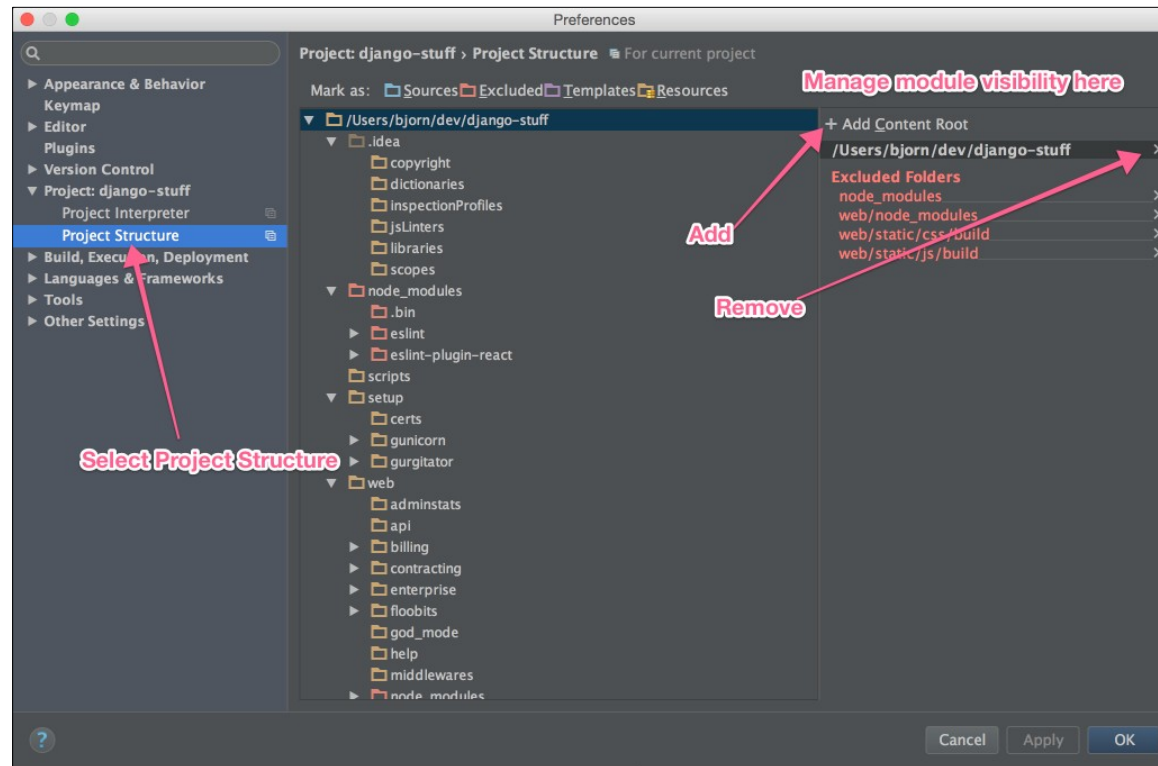
Classpath environment variable



Set the classpath of java projects in Eclipse

With Python

- *Python module path* resolves references to modules
 - Environnement variable PYTHONPATH



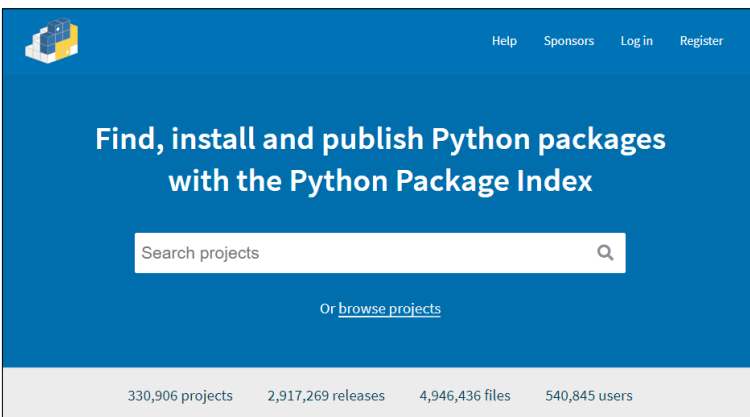
Set the pythonpath of python projects in PyCharm

Library repository

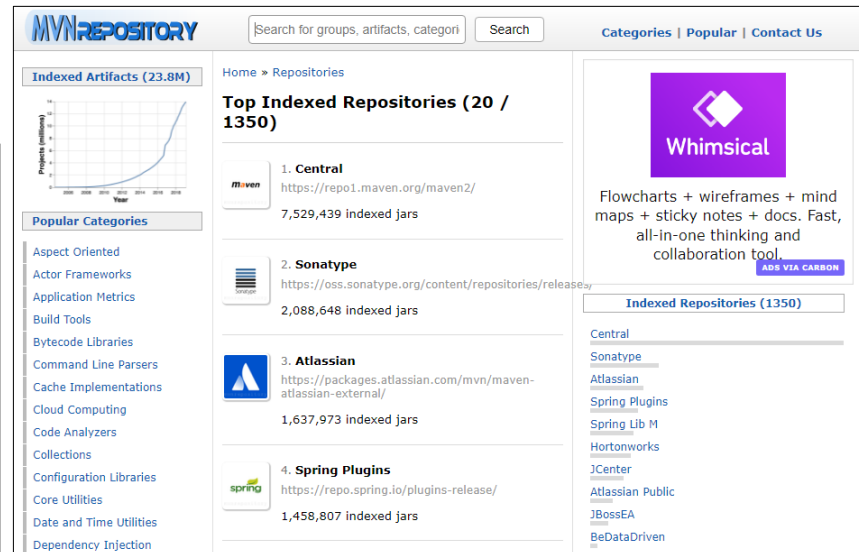
- Place where libraries and their metadata are stored
 - example for Linux:
<https://wiki.debian.org/DebianRepository/>
<https://help.ubuntu.com/community/Repositories/Ubuntu>

Library repository

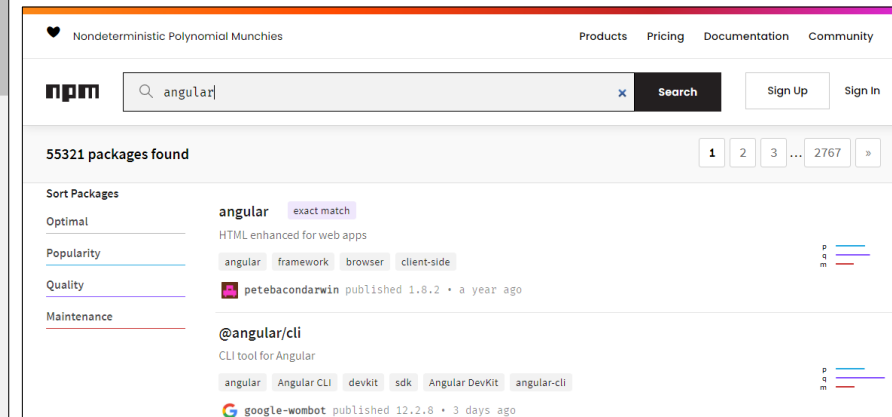
- Place where libraries and their metadata are stored
 - Java: Maven Central
 - .NET: NuGet
 - Node.js: npm
 - Python: PyPI
 - ...



PyPI central repository
<https://pypi.org/>



Maven distributed repositories
<https://mvnrepository.com/> - <https://search.maven.org/> - ...



npm central repository
<https://npmjs.com/>

Library manager

- Software that allows to download or update libraries and their dependencies (transitive)

Package Manager	Description
npm	A package manager for Node.js ^[21]
pip	A package installer for Python ^[22]
apt	For managing Debian Packages ^[23]
Homebrew	A package installer for MacOS that allows you to install packages Apple didn't ^[24]
vcpkg	A package manager for C and C++ ^{[25][26]}
yum and dnf	Package manager for Fedora and Red Hat Enterprise Linux ^[27]
pacman	Package manager for Arch Linux ^[28]

Typical library artifacts

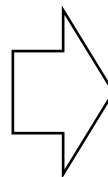
- Binaries, sources, documentation
- Metadata
 - Name, description
 - Publisher, developer
 - version number
 - Links to sources, issue tracker
 - Checksum
 - Necessary dependencies
- Example for Java: com.google.code.gson:gson:2.8.8
<https://mvnrepository.com/artifact/com.google.code.gson/gson/2.8.8> --> link « **view all** »
- Example for Ubuntu: asciidoc:8.6.10-2
<https://packages.ubuntu.com/source/bionic/asciidoc> --> download dsc

Java maven central: G-A-V coordinates

- **G-A-V = Group, Artefact, Version**

```
▼<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>com.google.code.gson</groupId>
    <artifactId>gson-parent</artifactId>
    <version>2.8.8</version>
  </parent>
  <artifactId>gson</artifactId>
  <name>Gson</name>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>com.github.wvengen</groupId>
      <artifactId>proguard-maven-plugin</artifactId>
      <version>2.4.0</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
  <build>
```

<https://repo1.maven.org/maven2/com/google/code/gson/gson/2.8.8/gson-2.8.8.pom>



com/google/code/gson/gson			
..			
1.1/	2008-08-10 14:29	-	
1.4/	2010-03-24 19:29	-	
1.5/	2011-03-14 12:57	-	
1.6/	2011-03-15 05:23	-	
1.7/	2011-04-12 22:19	-	
1.7.1/	2011-04-13 20:00	-	
1.7.2/	2011-09-30 18:58	-	
2.0/	2011-11-13 21:23	-	
2.1/	2011-12-31 07:20	-	
2.2/	2012-05-06 04:21	-	
2.2.1/	2012-05-11 18:45	-	
2.2.2/	2012-07-03 01:01	-	
2.2.3/	2013-04-12 21:37	-	
2.2.4/	2013-05-13 21:56	-	
2.3/	2014-08-11 23:41	-	
2.3.1/	2014-11-20 09:42	-	
2.4/	2015-10-04 00:47	-	
2.5/	2015-11-24 19:15	-	
2.6/	2016-02-11 18:04	-	
2.6.1/	2016-02-12 04:00	-	
2.6.2/	2016-02-27 02:32	-	
2.7/	2016-06-14 23:55	-	
2.8.0/	2016-10-27 06:02	-	
2.8.1/	2017-05-31 01:39	-	
2.8.2/	2017-09-20 01:42	-	
2.8.3/	2018-04-27 22:36	-	
2.8.4/	2018-05-01 18:10	-	
2.8.5/	2018-05-22 03:08	-	
2.8.6/	2019-10-04 18:55	-	
2.8.7/	2021-05-24 23:23	-	
2.8.8/	2021-08-20 16:08	-	
maven-metadata.xml	2021-08-20 16:14	1256	
maven-metadata.xml.md5	2021-08-20 16:14	32	
maven-metadata.xml.sha1	2021-08-20 16:14	40	
maven-metadata.xml.sha256	2021-08-20 16:14	64	
maven-metadata.xml.sha512	2021-08-20 16:14	128	

<https://repo1.maven.org/maven2/com/google/code/gson/gson/>

Packaging flow for python

- Configuration file pyproject.toml

- Build

 - \$ python3 -m build .

 - **source distribution**, (or **sdists**) ex., pip-23.3.1.tar.gz

 - Sources + PKG-INFO file with [metadata](#)

 - **binary distribution(s)**, (or **wheels**) ex., pip-23.3.1-py3-none-any.whl

 - extension modules, written in compiled languages like C, C++ and Rust, need to be compiled into platform-dependent machine code.

 - Wheels contain compiled, executable code (.so/.dll files)

- Upload to PyPI (or other)

 - \$ twine upload --repository testpypi dist/*

- Download and install

 - \$ python3 -m pip install package-name

Software versions

- Version name unique to a software state
 - internal names (build number or timestamp)
 - names for publication
- A wide variety of naming choices
 - When should you change this or that part of the version name?

Software versions

- Version name unique to a software state
 - internal names (build number or timestamp)
 - names for publication
- A wide variety of naming choices
 - When should you change this or that part of the version name?
 - Marketing and superstition sometimes get in the way
 - Photoshop 5 -> 5.5
 - Office 2007 (v12) ->Office 2010 (v14)
 - Sometimes specified by language
 - Python: PEP 440 – Version Identification and Dependency Specification



CommitStrip.com

<https://www.commitstrip.com/en/2013/08/20/numeros-de-version/>

The convention is sometimes singular

TeX 3 ; 3.1 ; 3.14 ; 3.141 ; 3.1415 ; ...

Software versions

- Calendar Versioning (CalVer)
 - Office 2007, 2010
 - Ubuntu 18.04 = April 2018
 - Wine 20040505 = 2004-05-05
- Standardisation attempt: <https://calver.org/>

2020.1.4018-pre



The diagram shows the version string '2020.1.4018-pre' with four colored curly braces underneath it. The first brace is purple and spans '2020', the second is teal and spans '.1', the third is orange and spans '.4018', and the fourth is grey and spans '-pre'. Below each brace is a label: 'year' (purple), 'release' (teal), 'build' (orange), and 'status' (grey).

year release build status

Software versions

- Semantic Versioning (SemVer)
 - Standardisation attempt: <https://semver.org>
 - **MAJOR: increment = change not retro-compatible**
 - **Minor: increment = feature added**
 - **Patch: increment = bug corrected**
- x.y.z-a1 = first alpha version
- x.y.z-b3 = third beta version
- x.y.z-rc1 = first release candidate
- 0.y.z = work in progress

4 . 2 . 1
MAJOR *Minor* patch



4.2.1
MAJOR Minor patch

Technological foundations of software development

Master your development environment and software dependencies

ICM – Computer Science Major – Course unit on Technological foundations of computer scienceopment

M1 Cyber Physical and Social Systems – Course unit on CPS2 engineering and development, Part 2: Technological foundations of software development

Maxime Lefrançois <https://maxime-lefrancois.info>

online: <https://ci.mines-stetienne.fr/cps2/course/tfsd/>

... Your turn

Complete the TODO section:

https://ci.mines-stetienne.fr/cps2/course/tfsd/course-2.html#_todos