

CPL: Getting started

1 Programe folosite

Dacă optați pentru folosirea mașinii virtuale, o descărcați de la <https://curs.upb.ro/mod/url/view.php?id=34783>, o importați în VirtualBox sau VMWare Player și treceți la Secțiunea 2.4 sau 3.3.

Username și parolă pentru mașina virtuală: student/student.

Pentru a porni Eclipse sau IntelliJ, folosiți path-urile specificate în fișierul `Readme.txt` aflat pe Desktop.

1.1 JDK

Recomandam folosirea versiunilor de JDK 11 LTS, 13 sau 14.

Pentru Windows: Descărcați și instalați pachetul de aici: <https://jdk.java.net/14/>

Pentru Ubuntu: Rulați comanda:

```
sudo apt-get install openjdk-14-jdk
```

1.2 Spim

Pentru a rula binare pentru platforma MIPS vom folosi emulatoarele Spim și QtSpim. QtSpim folosește același cod de emulator ca Spim, dar are o interfață grafică.

Pentru Windows: Descărcați și instalați binarul de QtSpim de Windows de aici: <https://sourceforge.net/projects/spimsimulator/files/>.

Dacă apare eroarea "The Program can't start because MSVCR110.dll is missing from your computer.", trebuie să instalați Visual C++ Redistributable pack for Visual Studio 2012, de aici: <https://www.microsoft.com/en-us/>

`download/details.aspx?id=30679`.

Dacă apare eroarea "The Program can't start because MSVCR120.dll is missing from your computer.", trebuie să instalați Visual C++ Redistributable pack for Visual Studio 2013, de aici: <https://www.microsoft.com/en-us/download/details.aspx?id=40784>.

Pentru Ubuntu: Pentru a instala Spim din linie de comandă:

```
sudo apt-get install spim
```

Pentru a instala QtSpim, descărcați pachetul *.deb de aici: <https://sourceforge.net/projects/spimsimulator/files/> și instalați-l:

```
sudo dpkg -i qtspim_*.deb
```

Dacă qtspim nu pornește, înseamnă că trebuie să instalați încă un pachet:

```
sudo apt-get install libxkbcommon-x11-0
```

1.3 Interpretorul pentru limbajul COOL

Trebuie să descărcați interpretorul de COOL de pe moodle.

Pe Linux este posibil să trebuiască marcat binarul ca executabil. Rulați:

```
chmod +x cool
```

2 Eclipse

2.1 Instalare Eclipse

Vom folosi Eclipse IDE for Java Developers, versiunea 2020-09. Descărcați și instalați pachetul pentru platforma voastră de aici: <https://www.eclipse.org/downloads/packages/release/2020-09/r/eclipse-ide-enterprise-java-developers>

2.2 ANTLR4 Tool

Vom folosi ANTLR versiunea 4.8. Descărcați JAR-ul de aici: <http://www.antlr.org/download/antlr-4.8-complete.jar>.

2.3 Configurare plugin-uri de Eclipse

Va urma configurarea plugin-urilor necesare pentru extensia ANTLR4 IDE¹. Instalarea extensiei XText:

- Deschideți Eclipse, intrați în Help -> Install New Software.

¹<https://github.com/jknack/antlr4ide/>

- La câmpul "Work With" completați cu <http://download.eclipse.org/releases/2020-09>, apăsați Enter.
- După ce se încarcă lista, expandați nodul **General Purpose Tools** și alegeți **Xtext Complete SDK**.
- Acceptați "terms of license agreement" și porniți instalarea.
- Când extensia s-a instalat cu succes, o să fiți întrebați dacă doriți să restartați programul. Acceptați și așteptați să se restarteze.

Instalarea extensiei "Faceted Project Framework":

- Deschideți Eclipse, intrați în **Help -> Install New Software**.
- La câmpul "Work With" completați cu <http://download.eclipse.org/releases/2020-09>, apăsați Enter.
- După ce se încarcă lista, expandați nodul **Web, XML, Java EE and OSGi Enterprise Development**.
- Selectați următoarele extensii:
 - **Eclipse Faceted Project Framework**
 - **Eclipse Faceted Project Framework JDT Enablement**
- Acceptați "terms of license agreement" și porniți instalarea.
- Când extensia s-a instalat cu succes, o să fiți întrebați dacă doriți să restartați programul. Acceptați și așteptați să se restarteze.

Instalarea extensiei "ANTLR 4 IDE":

- Deschideți Eclipse, intrați în **Help -> Eclipse Marketplace**.
- Căutați după **antlr**.
- Alegeți **ANTLR 4 IDE** (asigurați-vă că nu ați ales **ANTLR IDE**).
- Când extensia s-a instalat cu succes, o să fiți întrebați dacă doriți să restartați programul. Acceptați și așteptați să se restarteze.

Dacă apar probleme la instalarea de extensii, folosiți următoarele comenzi pentru instalarea manuală a extensiilor:

```
> eclipse-ide -application org.eclipse.equinox.p2.
    director -repository http://download.eclipse.org/
    releases/2020-09/ -installIU org.eclipse.xtext.sdk.
    feature.group
```

```

> eclipse-ide -application org.eclipse.equinox.p2.
  director -repository http://download.eclipse.org/
  releases/2020-09/ -installIU org.eclipse.wst.common.
  fproj.feature.group

> eclipse-ide -application org.eclipse.equinox.p2.
  director -repository http://download.eclipse.org/
  releases/2020-09/ -installIU org.eclipse.jst.common.
  fproj.enablement.jdt.feature.group

> eclipse-ide -application org.eclipse.equinox.p2.
  director -repository https://raw.githubusercontent.com
  /antlr4ide/antlr4ide-eclipse-release/master/ -
  installIU antlr4ide.sdk.feature.group

```

2.4 Configurarea unui proiect de Eclipse

- Deschideți Eclipse, intrați în **File -> New Project...**
- Selectați **General -> ANTLR 4 Project -> Next**.
- Atribuiți un nume proiectului, apăsați **Finish**.
- Va apărea noul proiect în Eclipse.
- Intrați în **File -> Properties -> Project Facets**.
- Selectați **"Convert to faceted form..."**.
- Selectați **Project Facet: Java 14** (sau cât este cazul), apăsați **Apply**.
- Ieșiți din fereastra de **Properties** și intrați din nou.
- În **Java Build Path**, tab-ul de **Libraries**, apăsați **"Classpath"** și **"Add External JARs..."**.
- Adăugați JAR-ul de ANTLR downloadat anterior, apăsați **Apply**.
- În fereastra de **properties**, alegeți secțiunea **ANTLR 4 -> Tool**.
- Apăsați **"Configure workspace settings..."**.
- Alegeți **"Add"** și adăugați JAR-ul cu ANTLR 4.8.
- Asigurați-vă că JAR-ul cu ANTLR 4.8 este bifat.
- În secțiunea **"Options"** din aceeași fereastră completați câmpul **"Directory"** cu **"src"**. Acesta va indica generatorului de cod ANTLR unde să plaseze clasele de cod sursa Java.

- Tot în secțiunea "Options" asigurați-vă că sunt bifate "Generate a parse tree listener (-listener)" și "Generate parse tree visitors (-visitor)".
- Apăsați "Apply and Close".

După ce fișierul "Hello.g4" a fost procesat, în folderul de build veți avea clasele compilate. Puteți rula clasele folosind alias-ul "grun" sau comenzile manuale:

```
> cd eclipse-workspace/lab0/build/classes
> java -cp "/path/to/antlr-4.8-complete.jar:." org.antlr.v4.gui.TestRig Hello r -tree
```

Parametri folosiți:

- `org.antlr.v4.gui.TestRig` - cerem rularea clasei "TestRig"
- `Hello` - numele gramaticii care va fi rulate
- `r` - regula de start
- `tree` - cerem generarea unui arbore de parsare

Dați un input care se potrivește gramaticii și închideți interpretorul

hello world

<Ctrl+D pe Linux> sau <Ctrl+Z pe Windows>

Se va afișa arborele de parsare: (`r hello world`).

Known issue:

În consola din Eclipse, caracterul EOF (End of File) nu este transmis corect la Ctrl+D. Workaround: schimbați focus-ul mutând cursorul în altă fereastră, apoi reveniți cu cursorul în fereastra de consolă.

3 IntelliJ IDEA

3.1 Instalare IntelliJ IDEA

O alternativă va fi folosirea IntelliJ IDEA, pe care îl puteți activa gratuit folosind contul de student. În același timp, puteți folosi gratuit oricare dintre IDE-urile lor pentru alte proiecte.

Pentru a crea un cont de student, completați următorul formular:

<https://www.jetbrains.com/shop/eform/students>

Pentru a instala IntelliJ (sau alte IDE-uri), folosiți Toolbox App:

<https://www.jetbrains.com/toolbox-app/>

Puteți în continuare să urmați pași din îndrumar din secțiunile următoare sau să vizionați acest videoclip pentru a configura rapid un proiect simplu.

https://www.youtube.com/watch?v=svEZtRjVBTY&t=3s&ab_channel=JohnKtejik

3.2 ANTLR4 Plugin

Ca în secțiunea Eclipse, vom folosi plugin-ul ANTLR 4.8. Acesta se instalează simplu, direct din IDE.

Pentru a instala plugin-ul, este nevoie să:

- Pornim IntelliJ fără a deschide un proiect.
- În partea de jos, accesăm **Configure -> Plugins**.
- Căutați plugin-ul "ANTLR v4 grammar plugin" și instalați-l.

3.3 Rularea unui proiect de IntelliJ

- Creăm un nou proiect. Folosiți opțiunile default și asigurați-vă că aveți JDK instalat și selectat..
- În cazul în care nu ați instalat ANTLR4, o puteți face și din fereastra proiectului. Accesați **File -> Settings -> Plugins**.
- Creați un fișier `g.g4` care va conține gramatica voastră de test.
- Completați fișierul astfel încât gramatica să recunoască textul "Hello World". Numiți regula de start simplu `r`.
- Click-dreapta pe fișierul `g.g4 -> Configure ANTLR` și setați ca **Output directory** folderul `src` din proiect. Acest pas va fi important în cazul laboratoarelor și temelor.
- Click-dreapta pe fișierul `g.g4 -> Generate ANTLR Recognizer`. În cazul în care totul este ok, un folder `gen` va apărea în proiect, fără vreun mesaj de eroare.
- Click-dreapta pe regula de start `r -> Test rule r`. Introduceți textul Hello World în căsuța de input, având arborele generat în dreapta dacă totul este ok.
- TODO Bonus: Puteți modifica regula astfel încât să recunoască "Hello " {Nume}.