

Мультиагентно-орієнтоване програмування

3-й рівень навчання, осінь 2021

- Доц. Баклан І.В.
- Email: iaa@ukr.net
- Web: baklaniv.at.ua

Практичне заняття 1

Інсталяція платформи JaCaMo

У нас є різні варіанти початку використання JaCaMo: Eclipse, команди сценарію оболонки, Gradle або Docker.

Параметри сценаріїв Eclipse та оболонки вимагають, щоб ви завантажили та встановили JaCaMo на своєму комп'ютері. Оскільки сценарій оболонки, Gradle та Docker не містять IDE, ви можете або імпортувати проект у робочу область Eclipse (інструкції відображаються сценарієм створення), або скористатися якимось текстовим редактором (ми пропонуємо Atom, див. Нижче).

Eclipse Plugin

Вимоги до програмного забезпечення:

- Java \geq 8
- Eclipse Java-EE або Committers \geq Photon Release (4.8.0)

Для початку дотримуйтеся інструкцій плагіна JaCaMo eclipse до кроку 12. Кроки 13–17 ілюструють, як створити та запустити нову програму.

Крок 1

Завантажте останню версію JaCaMo за посиланням:

<http://sourceforge.net/projects/jacamo/files/>



P Найбільший вибір
Підйомників

[Открыть](#)

Advertisement

[Home](#) / [Browse](#) / [Development](#) / [Frameworks](#) / [jacamo](#) / [Files](#)

jacamo

Status: **Alpha**Brought to you by: [aricci303](#), [asanti](#), [bordini](#), [Jomifred](#), [obolssler](#)[Summary](#)[Files](#)[Reviews](#)[Support](#)[Mailing Lists](#)[Code](#)[Download Latest Version](#)

jacamo-0.9.zip (66.0 MB)

[Get Updates](#)

Крок 2

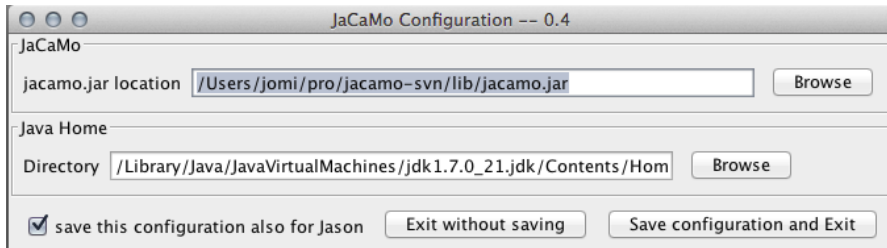
Після завантаження розпакуйте його в будь -якому каталозі вашого комп'ютера.

Крок 3

Навіть якщо раніше ви запускали JaCaMo на своєму комп'ютері, двічі клацніть на ньому файл “**lib/jacamo.jar**”. Ви також можете виконати цей файл за допомогою такої команди:

```
java -jar lib/jacamo.jar
```

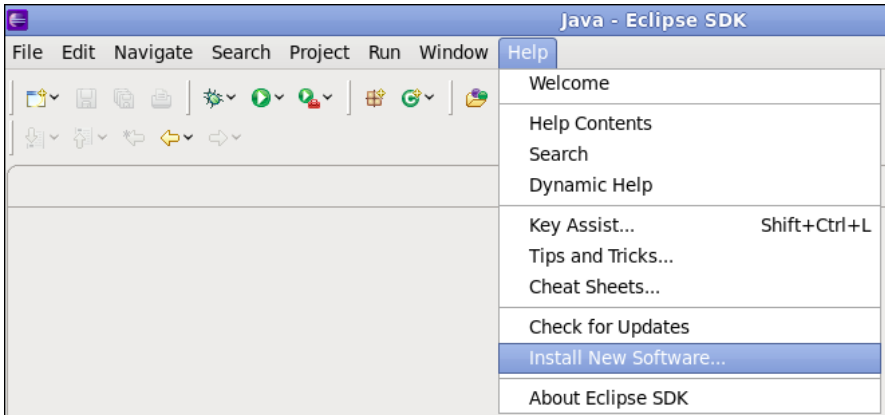
З'явиться наступне вікно. Ми пропонуємо вам змінити лише каталог "Java Home" (зверніть увагу, що він повинен вказувати на встановлення JDK).



*** Якщо ваш доступ до Інтернету проходить через проксі - сервер, натисніть тут <http://help.eclipse.org/luna/index.jsp?topic=%2Forg.eclipse.platform.doc.user%2Freference%2Fref-net-preferences.htm> , щоб дізнатися, як налаштувати проксі -сервер у вашому Eclipse, перш ніж продовжити встановлення плагіна.

Крок 4

Нарешті, ви можете встановити плагін JaCaMo для Eclipse, відкривши платформу Eclipse і перейшовши до опції "Встановити нове програмне забезпечення ..." у меню "Довідка":



Крок 5

Отже, з'явиться наступне вікно.

Install

Available Software

Select a site or enter the location of a site.

Work with: ▼

Find more software by working with the ["Available Software Sites"](#) preferences.

Name	Version
<input type="checkbox"/> ⓘ There is no site selected.	

Details

Show only the latest versions of available software

Group items by category

Show only software applicable to target environment

Contact all update sites during install to find required software

Hide items that are already installed

What is [already installed](#)?

Крок 6

Натисніть кнопку «Додати» та заповніть форму, як показано на наступному малюнку. Параметри такі

Name: **jacamoide**

Location (for **JaCaMo** **>= 0.4** and **Eclipse*** 2020-09 or newer):

<http://jacamo.sourceforge.net/eclipseplugin/20x>

Location (for **JaCaMo** \geq **0.4** and **Eclipse*** Juno/Kepler/Luna/Mars/Neon/Oxygen/Photon/2018-2020):

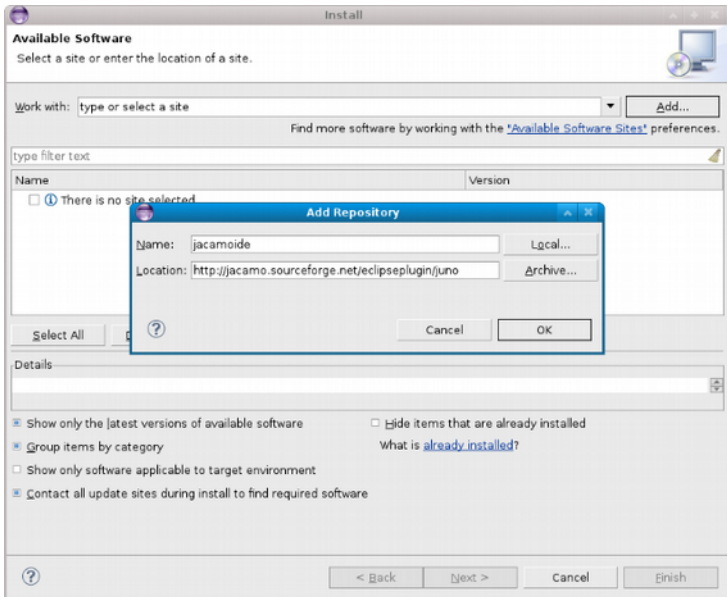
<http://jacamo.sourceforge.net/eclipseplugin/juno>

Location (for **JaCaMo** \leq **0.3a** and **Eclipse*** Juno/Kepler/Luna/Mars/Neon/Oxygen/Photon/2018-2020):

http://jacamo.sourceforge.net/eclipseplugin/juno_old

(*) Лише Java-EE або Committers.

Для завершення натисніть кнопку "ОК".



Крок 7

Позначте опцію "jasamoide", а потім натисніть кнопку "наступний". Отже, вам доведеться почекати хвилину, поки Eclipse шукатиме залежності.

Install



Available Software

Check the items that you wish to install.



Work with: jacamoide - <http://jacamo.sourceforge.net/eclipseplugin/juno>

Add...

Find more software by working with the ["Available Software Sites"](#) preferences.

type filter text

Name	Version
jacamoide	
<input type="checkbox"/> jacamoide_feature	1.0.0.201408152015

Select All

Deselect All

1 item selected

Details

jaCaMo is a framework for Multi-Agent Programming that combines three separate technologies (jason, CARtAgO, and [More...](#))

- Show only the latest versions of available software
 - Hide items that are already installed
 - Group items by category
 - Show only software applicable to target environment
 - Contact all update sites during install to find required software
- What is [already installed?](#)



< Back

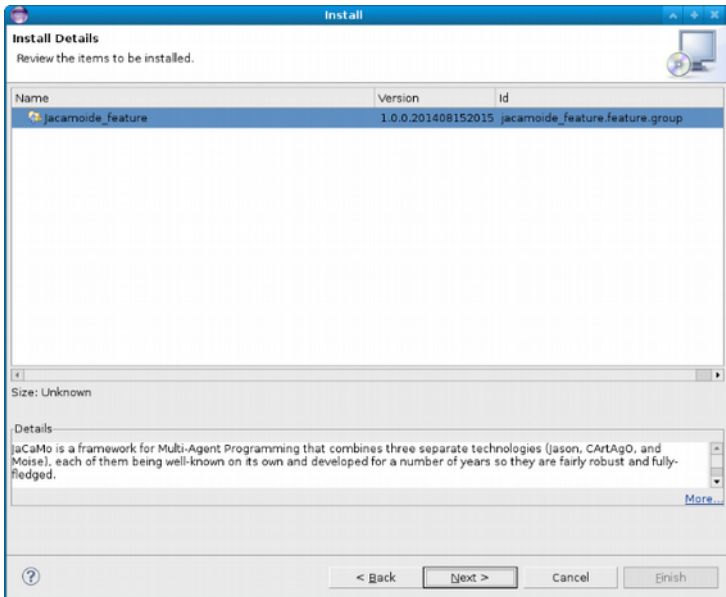
Next >

Cancel

Finish

Крок 8

У наступних вікнах просто знову натисніть кнопку «далі».



Крок 9

Останнє вікно, яке вам буде показано, стосується ліцензії. Поставте прапорець біля пункту "Я приймаю умови ліцензійних угод", а потім натисніть кнопку "закінчити". Потім інсталяція продовжується, це може зайняти кілька хвилин, тому зачекайте.

Review Licenses

Licenses must be reviewed before the software can be installed. This includes licenses for software required to complete the install.



Licenses:

- ▶ Copyright (C) 2014 Olivier Boissier, Rafael H. Bordini, Jomi F. Hübner
- ▶ Eclipse Foundation Software User Agreement

License text:

Copyright (C) 2014 Olivier Boissier, Rafael H. Bordini, Jomi F. Hübner, Alessandro Ricci, Andrea Santi, Maicon R. Zlatelli, et al. JaCaMo is distributed under LGPL. See file LGPL.txt in the JaCaMo directory.

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation: either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

To contact the authors:

Olivier Boissier - <http://www.emse.fr/~boissier>
Rafael H. Bordini - <http://www.inf.pucrs.br/rf.bordini>
Jomi F. Hübner - <http://www.das.ufsc.br/~jomi>
Alessandro Ricci - <http://alice.unibo.it/xwiki/bin/view/>

- I accept the terms of the license agreements
- I do not accept the terms of the license agreements



< Back

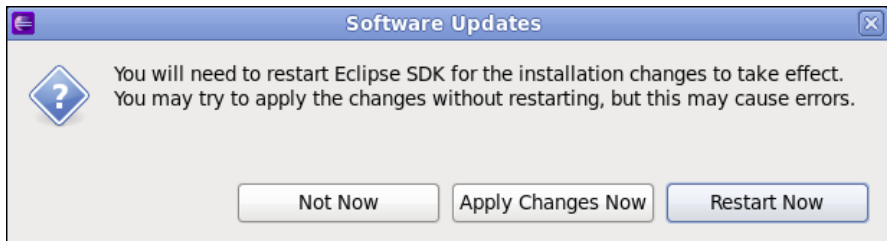
Next >

Cancel

Finish

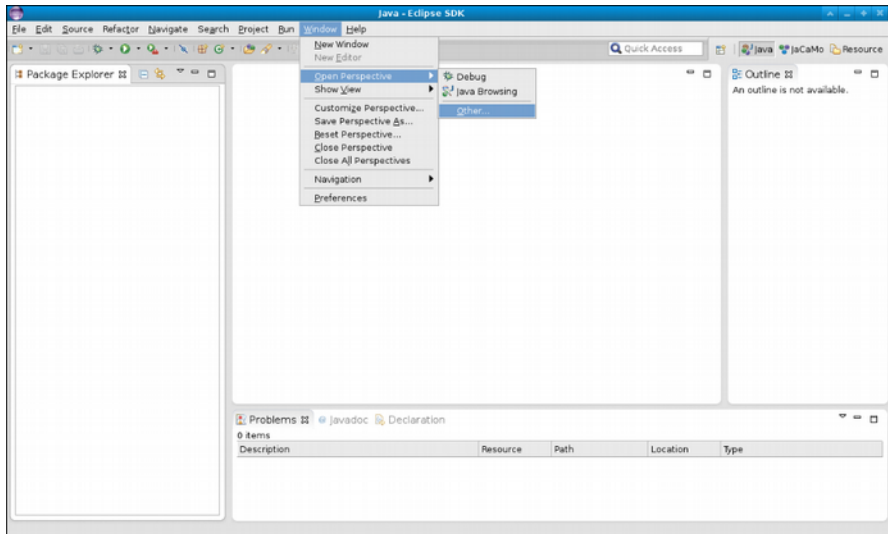
Крок 10

В кінці цього процесу буде показано вікно для завершення установки. Виберіть опцію «Перезапустити зараз».



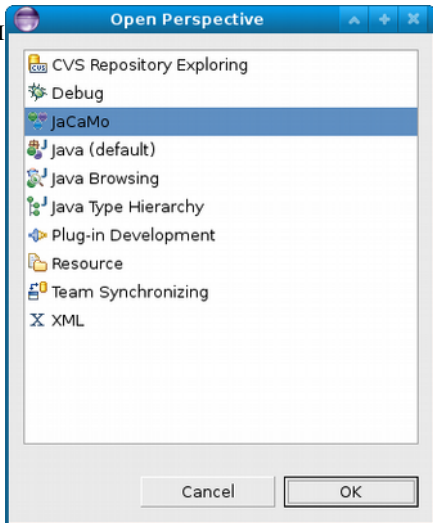
Крок 11

Тепер ви можете відкрити перспективу JaCaMo. Це можна зробити в меню (Вікно> Відкрити перспективу> Інше)



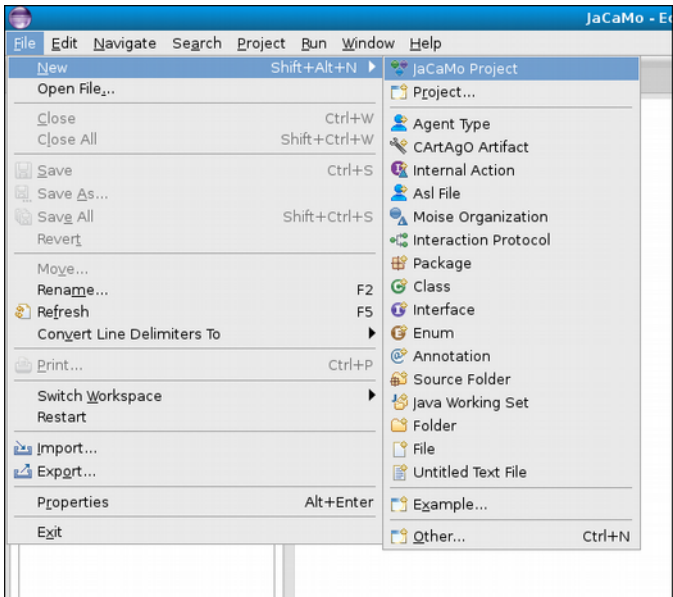
Крок 12

Потім ви можете вибрати JaCaMo і натиснути «ОК».



Крок 13

Нарешті, все готово. Для того, щоб перевірити встановлення плагіна, ми пропонуємо створити простий проект hello world. Це можна зробити в меню (Файл> Створити> Проект JaCaMo) або (Файл> Створити> Інше> JaCaMo> Проект JaCaMo).



Крок 14

Заповніть поле "Назва проекту" і натисніть кнопку "Готово".



New JaCaMo Project



New JaCaMo Project

This wizard creates a new JaCaMo Project

Project name:

Agent:

Environment:



Cancel

Finish

Крок 15

Якщо все в порядку, у вас буде створено ваш перший проект!



JaCaMo Navigator

- helloworld
 - src/agt
 - src/int
 - src/org
 - src/env
 - vector.jar - /home/maicon/sv
 - cartago.jar - /home/maicon/t
 - c4jason.jar - /home/maicon/t
 - moise.jar - /home/maicon/sv
 - jacamo.jar - /home/maicon/s
 - intmas.jar - /home/maicon/s
 - bin
 - doc
 - lib
 - log
 - src
 - helloworld.jcm**
 - logging.properties

helloworld.jcm

```

/*
    JaCaMo Project File

    This file defines the initial state of the MAS (initial
    (see below the documentation about what can be defined)

*/

mas helloworld {

  agent sample_agent

  /*
  agent definition

    <agent>      ::= agent <name> { <parameter>* }
    <parameter> ::= <id> : <value> [ (, | EOL) <value> ]

    e.g.
    agent bob {
      beliefs:    p("this is a condition",15000) //
                 friend(alice)
      goals:      start, go(home) //
      source:     participant.asl //
      ag-class:   tt.MyAgClass //
      ag-arch:    myp.myArch1
                 mypkg.MyCustomAgArch //

```

Крок 16

Тепер ви можете запусити програму, натиснувши кнопку Виконати.



JaCaMo Navigator

- helloworld
 - src/agt
 - src/int
 - src/org
 - src/env
 - vector.jar - /home/maicon/sv
 - cartago.jar - /home/maicon/t
 - c4jason.jar - /home/maicon/t
 - moise.jar - /home/maicon/sv
 - jacamo.jar - /home/maicon/s
 - intmas.jar - /home/maicon/s
 - bin
 - doc
 - lib
 - log
 - src
 - helloworld.jcm**
 - logging.properties

helloworld.jcm

```

/*
    JaCaMo Project File

    This file defines the initial state of the MAS (initial
    (see below the documentation about what can be defined)

*/
mas helloworld {
    agent sample_agent

    /*
    agent definition

        <agent>      ::= agent <name> { <parameter>* }
        <parameter> ::= <id> : <value> [ (, | EOL) <value> ]

    e.g.
    agent bob {
        beliefs:      p("this is a condition",15000) //
                    friend(alice)
        goals:        start, go(home) //
        source:       participant.asl //
        ag-class:     tt.MyAgClass //
        ag-arch:      myp.myArch1
                    mypkg.MyCustomAgArch //

```

Крок 17

Результатом буде повідомлення "привіт світу" на екрані.

JaCaMo - helloworld/helloworld.jcm - Eclipse

File Edit Navigate Search Project Run Window Help

Quick Access

JaCaMo Navigator

- helloworld
 - src/agt
 - src/int
 - src/org
 - src/env
 - jason.jar - /home/maicon/...
 - cartago.jar - /home/maicon/...
 - c4jason.jar - /home/maicon/...
 - moise.jar - /home/maicon/...
 - jacamo.jar - /home/maicon/...
 - intmas.jar - /home/maicon/...
 - bin
 - doc
 - lib
 - log
 - src
 - helloworld.jcm**
 - logging.properties

helloworld.jcm

```
/*
    JaCaMo Project File

    This file defines the initial state of the MAS (initial agents, environment, orgs)
    (see below the documentation about what can be defined in this file)

*/
mas helloworld {
    agent sample_agent
    /*
    agent definition
    agents {
        MAS agent names { parameters }
    }
}

```

MAS Console - helloworld

all	Jason Http Server running on http://127.0.0.1:3272
sample_agent	[sample_agent] hello world.

Clean Stop Pause Debug Sources New agent

Gradle

Вимоги:

Java \geq 8

Щоб створити новий додаток, виконайте наведені нижче дії.

Примітка. Перший запуск програми триває довше після завантаження JaCaMo.

Unix

```
curl -s -O http://jacamo.sourceforge.net/nps/np1.0.zip  
unzip np1.0.zip  
./gradlew --console=plain
```

Windows

1. Завантажте

<http://jacamo.sourceforge.net/nps/np1.0.zip>

2. Unzip

3. Запустіть **gradlew.bat**

Порада: Замість **np1.0.zip** ви можете використовувати **npss.zip** для створення програми JaCaMo на основі поточної версії знімка.

Вам буде запропоновано ввести ідентифікатор програми, після чого будуть показані інструкції щодо її запуску.

Приклад виводу:

```
> Task :run
```

```
JaCaMo 1.0 built on Thu Jun 25 21:12:43 BRT 2020
```

```
Enter the identification of the new application:
```

```
helloworld
```

```
Creating path /Users/jomi/tmp/helloworld
```

```
You can run your application with:
```

```
cd /Users/jomi/tmp/helloworld
```

```
./gradlew -q --console=plain
```

```
or (if you have JaCaMo scripts installed)
```

```
jacamo /Users/jomi/tmp/helloworld/helloworld.jcm
```

```
an Eclipse project can be created using
```

```
'Existing Gradle Project' from Eclipse menu
```

```
File/Import
```

or

```
./gradlew eclipse
```

Якщо ви хочете запустити програму JaCaMo, яка не була створена з файлом `build.gradle`, ви можете завантажити шаблон тут <https://raw.githubusercontent.com/jacamo-lang/jacamo/master/src/main/resources/templates/build.gradle> і замінити

<VERSION> by the required JaCaMo release (e.g. 1.0)

<PROJECT-FILE> by your .jcm file (e.g. hello.jcm).

Команди скрипту Shell

Вимоги:

Java \geq 8

Щоб використовувати сценарії для створення та запуску програм JaCaMo, ви можете завантажити JaCaMo звідси або клонувати з GitHub. Щоб налаштувати ці сценарії:

Переконайтеся, що у вас правильно встановлено JASAMO_HOME (ця змінна вказує на каталог із бібліотеками підкаталогів та сценаріями JaCaMo), а потім налаштуйте **PATH**.

Якщо ви завантажили та розпакували JaCaMo у локальну папку, скажімо `/home/bob/jacamo`, налаштування виконується за допомогою таких команд:

```
export JACAMO_HOME=/home/bob/jacamo
export PATH=$JACAMO_HOME/scripts:$PATH
```

у випадку, якщо ви вирішили клонувати JaCaMo з GitHub:

```
git clone https://github.com/jacamo-lang/jacamo.git
cd jacamo
./gradlew config
export JACAMO_HOME=`pwd`/build
export PATH=$JACAMO_HOME/scripts:$PATH
```

Переконайтеся, що у вас JAVA_HOME правильно встановлено (ця змінна вказує на каталог Java Development Kit (JDK)), наприклад:

```
export JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk-12.0.2.jdk/Contents/Home
```

Доступні сценарії:

1. Створіть новий додаток, ідентифікований `helloworld` за допомогою

```
jacamo-new-project ../somewhere/helloworld
```

Приклади агента, артефакту та організації розміщуються в папці `src`. Якщо ім'я програми не повідомляється, користувача просять надати його. Другий необов'язковий аргумент `--console` і створює додаток, де повідомлення друкуються в консолі замість відкриття графічного інтерфейсу консолі MAS.

2. Скомпілюйте та запустіть додаток за допомогою

```
jacamo ../somewhere/helloworld/helloworld.jcm
```

3. Запустіть додаток за допомогою

```
jacamo-run ../somewhere/helloworld/helloworld.jcm
```

4. Щоб створити єдиний файл `jar` з усіма ресурсами для запуску програми:

```
jacamo-jar helloworld.jcm
```

яким можна керувати

```
java -jar helloworld.jar
```

Script	Arguments	Examples
jacamo-new-project	<i>[application name]</i> <i>[, --console]</i>	<code>jacamo-new-project</code> <code>jacamo-new-project helloworld</code> <code>jacamo-new-project helloworld --console</code>
jacamo	<i>application name</i>	<code>jacamo helloworld.jcm</code>
jacamo-run	<i>application name</i>	<code>jacamo-run helloworld.jcm</code>
jacamo-jar	<i>application name</i>	<code>jacamo-jar helloworld.jcm</code>

Docker

Вимоги:

- Docker

Інструкції щодо створення та запуску програми JaCaMo за допомогою зображень Docker див. на сторінці JaCaMo-Docker - <https://github.com/jacamo-lang/docker> .

Atom

Для IDE, легшої за Eclipse, ми рекомендуємо Atom - <https://atom.io/> . Наступні пакети забезпечують підсвічування синтаксису для проектів AgentSpeak (мова Jason) та JCM: language-jcm <https://atom.io/packages/language-jcm> та language-agentspeak <https://atom.io/packages/language-agentspeak> .

writing_paper.jcm — ~/pro/jacamo-glt/examples/writing-paper

Project t1.jcm writing_paper.jcm ag2.asl ag3.asl

norm-board

- bin
- log
- src
 - .DS_Store
 - ag1.asl
 - ag2.asl
 - ag3.asl
 - ag4.asl
 - e1.npl
 - logging.properties
 - t1.jcm
 - t2.jcm
 - t3.jcm
 - t3.xml
 - t4.jcm

```
1 /*
2   JaCaMo Project File
3
4   This file defines the initial state of the MAS (initial agents,
5   environment, organisation, ...)
6 */
7 mas writing_paper {
8
9     agent bob
10    agent alice
11    agent carol
12
13    organisation opaper: wp-os.xml {
14        group paper_group: wpgroup {
15            responsible-for: s1
16            players: bob editor
17                   alice writer
18                   carol writer
19        }
20        scheme s1: writePaperSch
21    }
22
23    asl-path: src/agt, src/agt/inc
24
25 }
26
27
```

~/pro/jacamo-glt/examples/writing-paper/writing_paper.j 1 LF UTF-8 JaCaMo Project File master 0 files 2 updates

Project

t1.jcm

writing_paper.jcm

ag3.asl

ag2.asl

norm-board

bin

log

src

.DS_Store

ag1.asl

ag2.asl

ag3.asl

ag4.asl

e1.npl

logging.properties

t1.jcm

t2.jcm

t3.jcm

t3.xml

t4.jcm

```
1
2 !start.
3
4 +!start : true
5     <- makeArtifact(org,"ora4mas.nopl.OrgBoard",["t3.xml"],AId);
6     createGroup(g1,wpgroup,GId);
7     focus(GId);
8     adoptRole(editor);
9     adoptRole(writer);
10    debug(inspector_gui(on));
11
12    createScheme(s1,writePaperSch,SId);
13    addScheme(s1);
14    debug(inspector_gui(on))[artifact_name(s1)];
15    commitMission(mManager)[artifact_name(s1)];
16
17
18 /* application domain goals */
19 +!wtitle    <- .wait(500);.print("writing title...").
20 +!wabs      <- .print("writing abstract...").
21 +!wsectitles <- .print("writing section title...").
22 +!wsecs     <- .print("writing sections...").
23 +!wconc     <- .print("writing conclusion...").
24 +!wrefs     <- .print("doing refs...").
25 +!wp        <- .print("paper finished!").
26 +!submit    <- .print("submit").
27 +!present   <- .print("presented!").
28
29 { include("$jacamoJar/templates/common-cartago.asl") }
```

Завдання:

1. Встановити інструментарій для роботи з JaCaMo:
Eclipse,
команди сценарію Shell,
Gradle,
Docker,
Atom.
2. Підготуйте звіт з скріншотами про виконані кроки.

Література

Olivier Boissier, Rafael H. Bordini, Jomi F. Hübner, Alessandro Ricci, Andrea Santi. Multi-agent oriented programming with JaCaMo. - Science of Computer Programming 78 (2013) 747–761.

Rafael C. Cardoso and Angelo Ferrando. A Review of Agent-Based Programming for Multi-Agent Systems. - Computers 2021, 10, 16. - <https://doi.org/10.3390/computers10020016> .

Multi-Agent Programming: Languages, Platforms and Applications. - © 2005 by Springer. - ISBN-10: 0-387-24568-5

Engineering Multi-Agent Systems. 6th International Workshop, EMAS 2018. Stockholm, Sweden, July 14–15, 2018. - © Springer Nature Switzerland AG 2019. - <https://doi.org/10.1007/978-3-030-25693-7>

Rafael H. Bordini · Mehdi Dastani · Jürgen Dix · Amal El Fallah Seghrouchni, Ed. Multi-Agent Programming: Languages, Tools and Applications. - © Science+Business Media, LLC 2009. - DOI 10.1007/978-0-387-89299-3