Ex.No – 05 **Date –** 12/08/2024

Installation and Configuration of CloudSim in Eclipse IDE

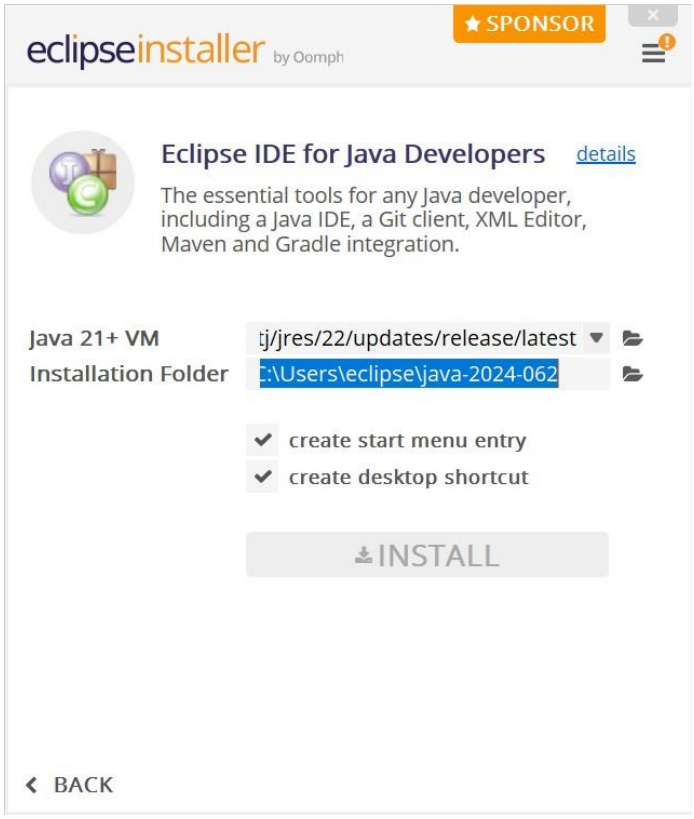
AIM:

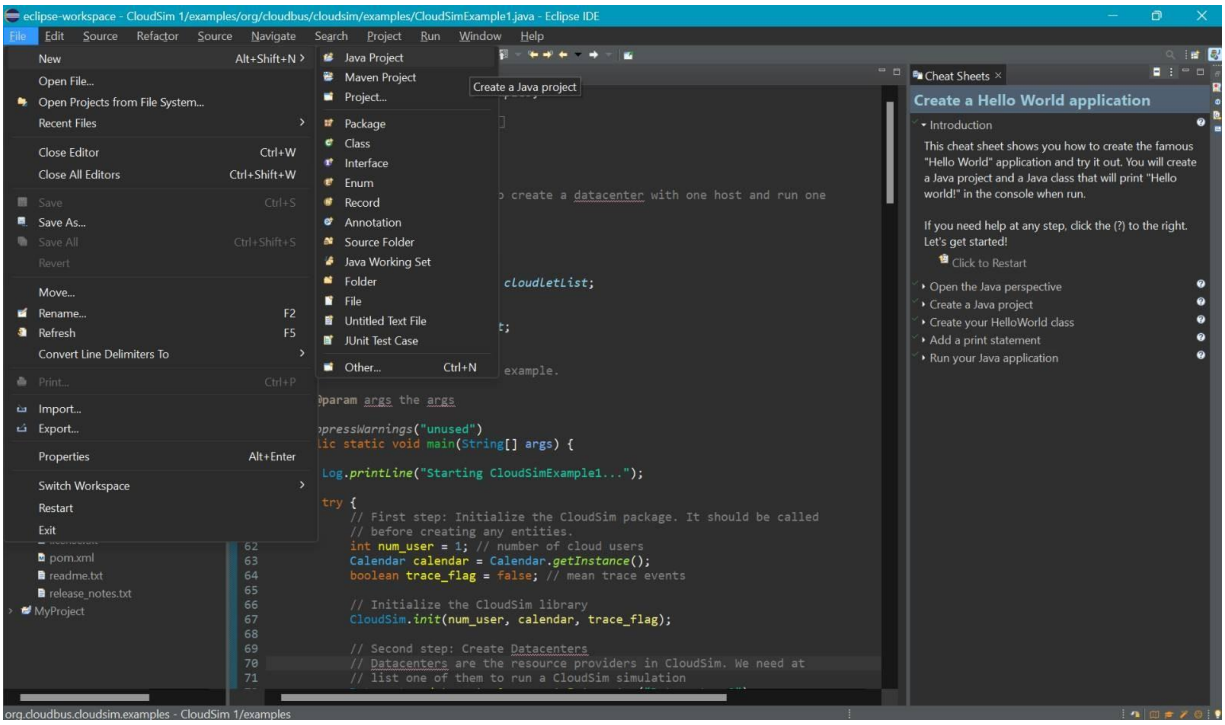
To install and configure the CloudSim in Eclipse IDE and run a java program in it.

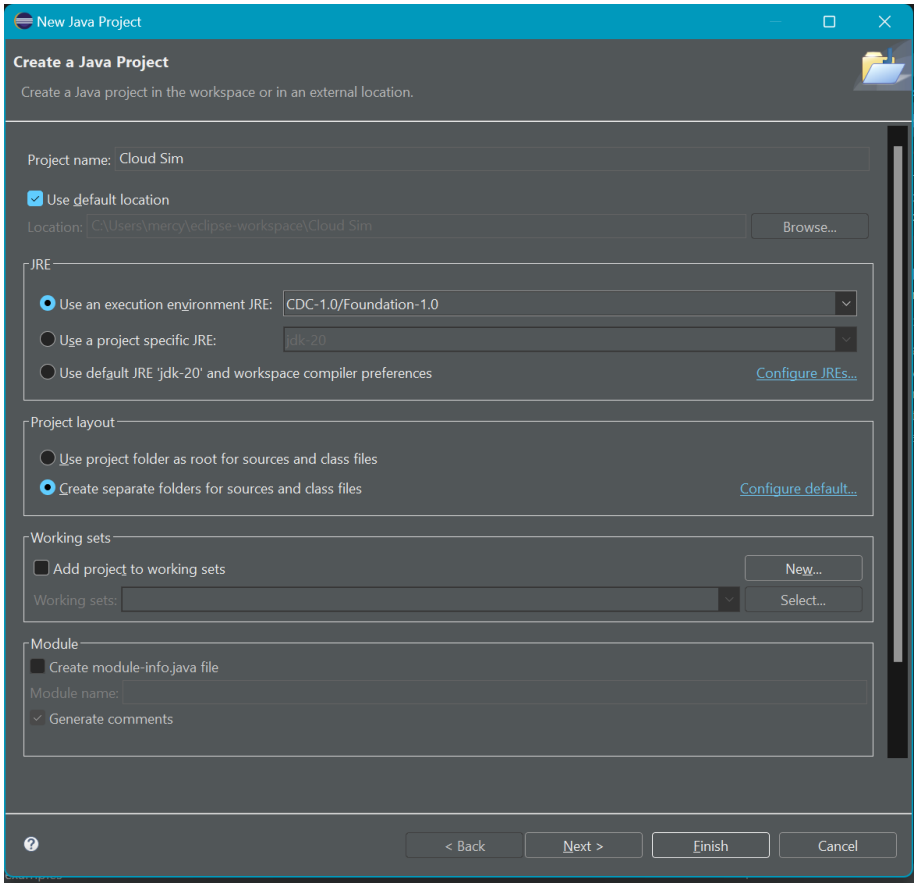
PROCEDURE:

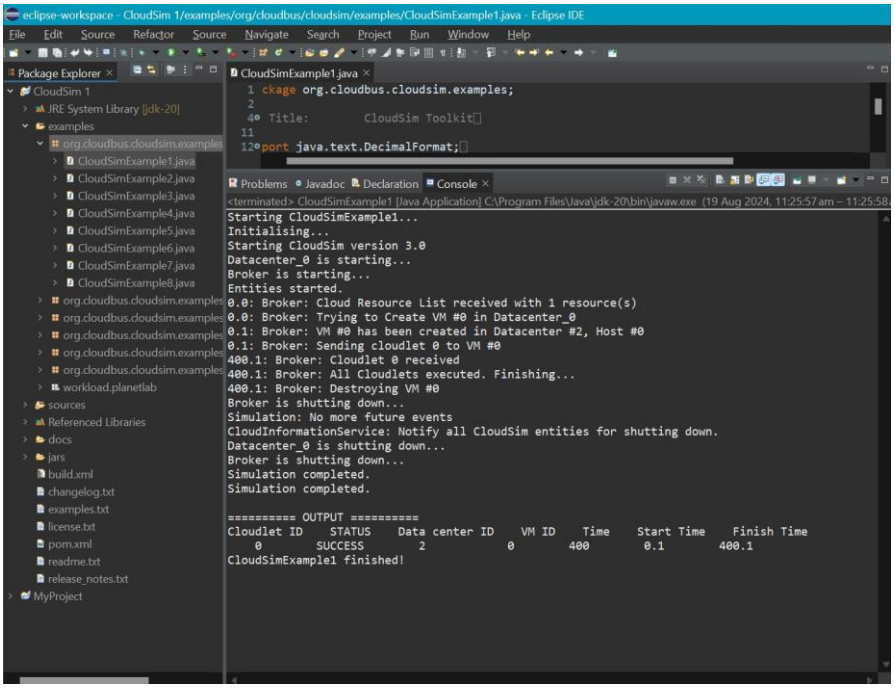
1. Java Installation:
    a. Check Java in your system.
    b. If Java not installed then download Java.
    c. Install Java setup.
    d. Set the path for Java in Environment Variables.
2. Download Cloud Sim and Additional JAR file:
    a. Download CloudSim 3.0.3
    b. Download common math 3 JAR file
3. Eclipse IDE Installation:
    a. Download the correct version of Eclipse IDE for your system.
    b. Install Eclipse IDE.
4. Run Cloud Sim in Eclipse:
    a. Put the common math 3 JAR file in the JAR folder of CloudSim.
    b. Build a new java project with CloudSim folder.

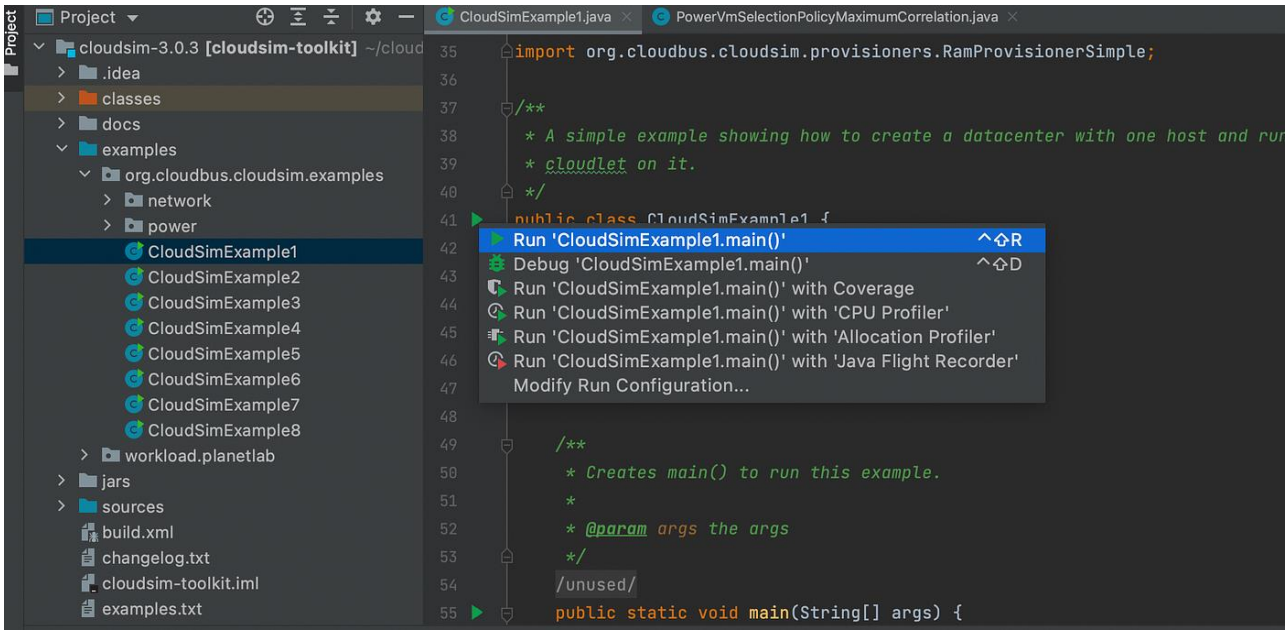
OUTPUT:

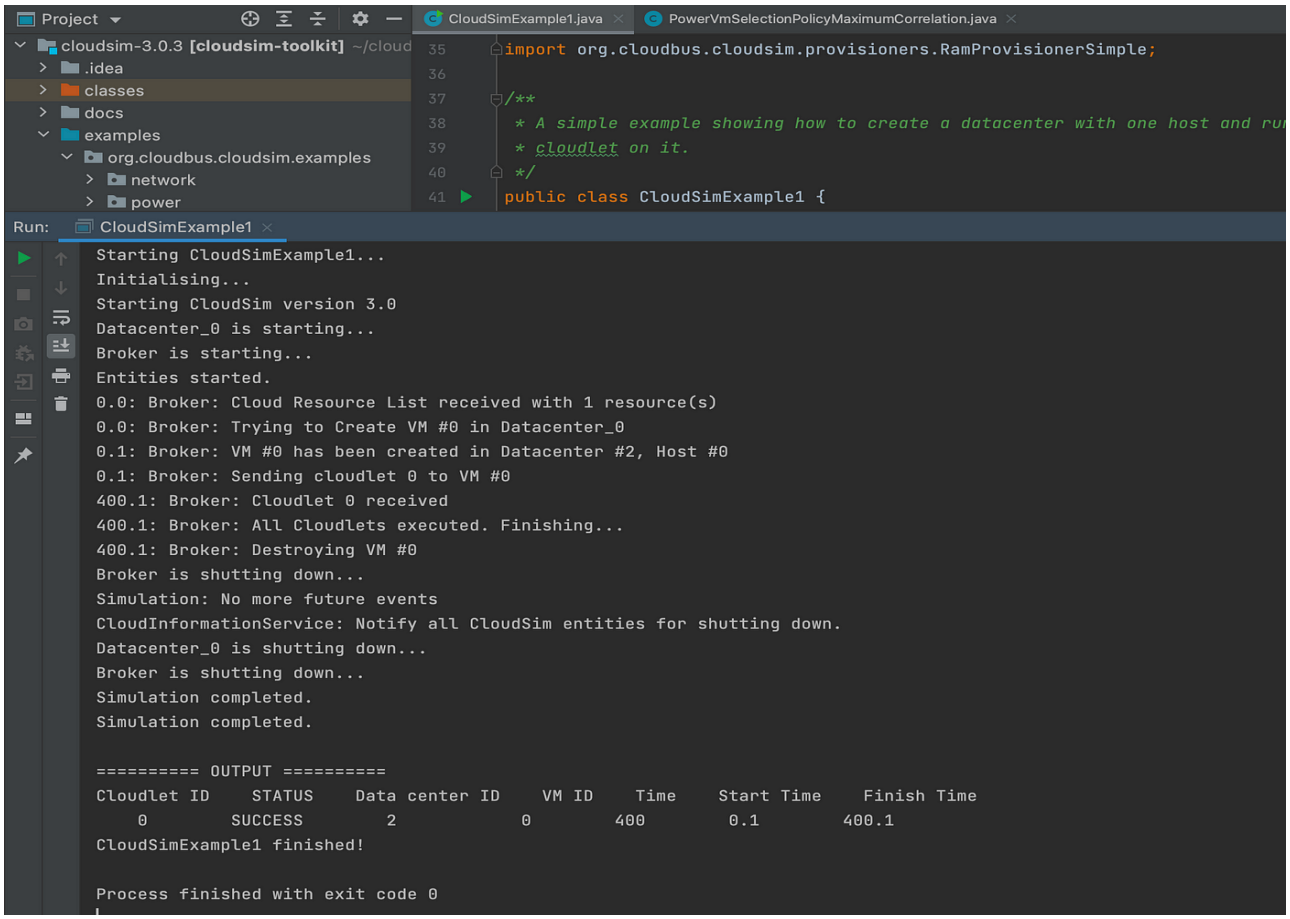

eclipseinstaller by Oomph
SPONSOR
Eclipse IDE for Java Developers
details
The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Maven and Gradle integration.
Java 21+ VM
tj/jres/22/updates/release/latest
Installation Folder
C:\Users\eclipse\java-2024-062
create start menu entry
create desktop shortcut
INSTALL
BACK

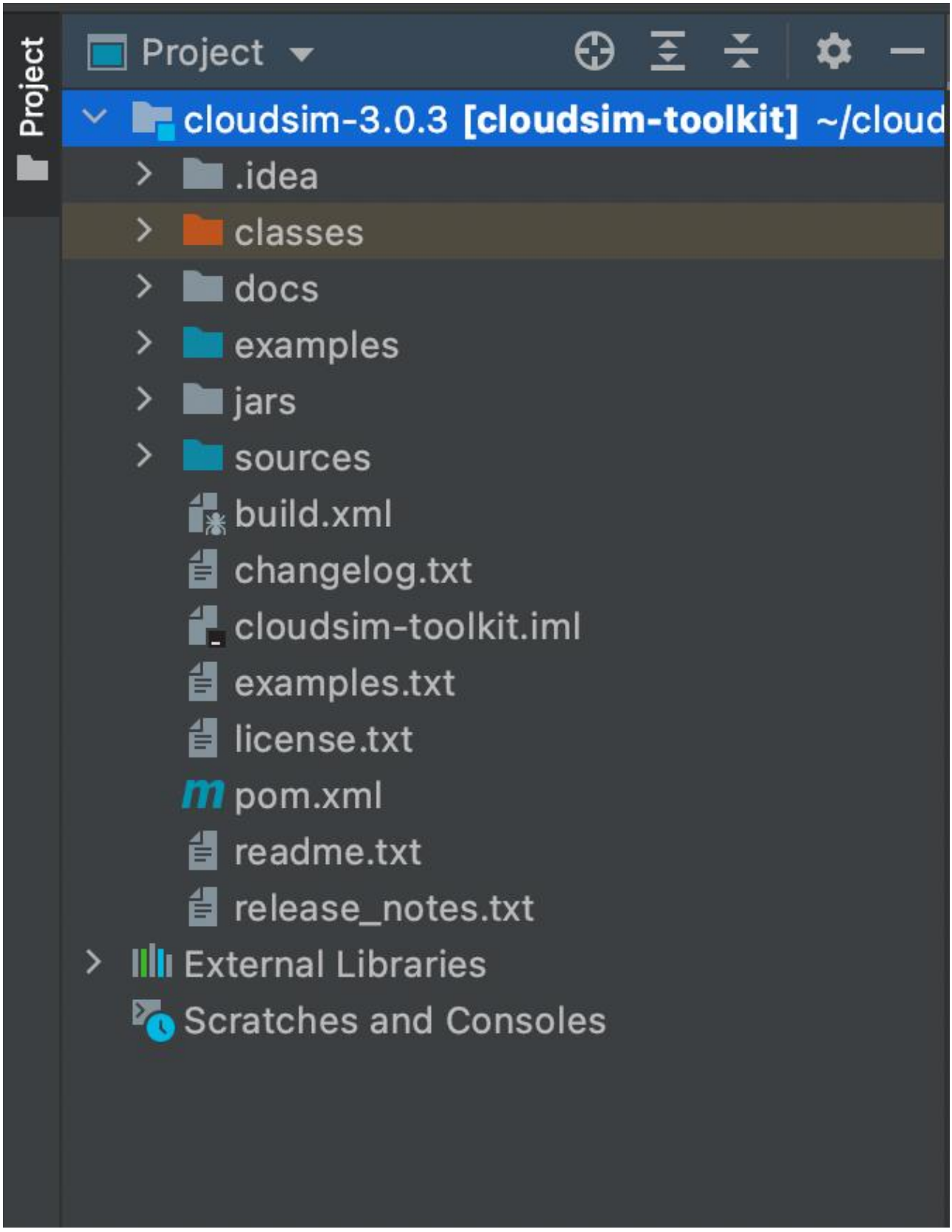

eclipse-workspace - CloudSim 1/examples/org/cloudbus/cloudsim/examples/CloudSimExample1.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
New Alt+Shift+N
Open File...
Open Projects from File System...
Recent Files
Close Editor Ctrl+W
Close All Editors Ctrl+Shift+W
Save Ctrl+S
Save As...
Save All Ctrl+Shift+S
Revert
Move...
Rename... F2
Refresh F5
Convert Line Delimiters To
Print... Ctrl+P
Import...
Export...
Properties Alt+Enter
Switch Workspace
Restart
Exit
Java Project
Maven Project
Project...
Create a Java project
Package
Class
Interface
Enum
Record
Annotation
Source Folder
Java Working Set
Folder
File
Untitled Text File
JUnit Test Case
Other... Ctrl+N
Cheat Sheets
Create a Hello World application
Introduction
This cheat sheet shows you how to create the famous "Hello World" application and try it out. You will create a Java project and a Java class that will print "Hello world!" in the console when run.
If you need help at any step, click the (?) to the right. Let's get started!
Click to Restart
Open the Java perspective
Create a Java project
Create your HelloWorld class
Add a print statement
Run your Java application
pom.xml
readme.txt
release_notes.txt
MyProject
Log.printLine("Starting CloudSimExample1...");
try {
// First step: Initialize the CloudSim package. It should be called
// before creating any entities.
int num_user = 1; // number of cloud users
Calendar calendar = Calendar.getInstance();
boolean trace_flag = false; // mean trace events
// Initialize the CloudSim library
CloudSim.init(num_user, calendar, trace_flag);
// Second step: Create Datacenters
// Datacenters are the resource providers in CloudSim. We need at
// list one of them to run a CloudSim simulation
org.cloudbus.cloudsim.examples - CloudSim 1/examples


New Java Project
Create a Java Project
Create a Java project in the workspace or in an external location.
Project name: Cloud Sim
Use default location
Location: C:\Users\mercy\eclipse-workspace\Cloud Sim
Browse...
JRE
Use an execution environment JRE: CDC-1.0/Foundation-1.0
Use a project specific JRE: jdk-20
Use default JRE 'jdk-20' and workspace compiler preferences
Configure JREs...
Project layout
Use project folder as root for sources and class files
Create separate folders for sources and class files
Configure default...
Working sets
Add project to working sets
New...
Working sets:
Select...
Module
Create module-info.java file
Module name:
Generate comments
< Back
Next >
Finish
Cancel


eclipse-workspace - CloudSim 1/examples/org/cloudbus/cloudsim/examples/CloudSimExample1.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Package Explorer
CloudSim 1
JRE System Library [jdk-20]
examples
org.cloudbus.cloudsim.examples
CloudSimExample1.java
CloudSimExample2.java
CloudSimExample3.java
CloudSimExample4.java
CloudSimExample5.java
CloudSimExample6.java
CloudSimExample7.java
CloudSimExample8.java
org.cloudbus.cloudsim.examples
org.cloudbus.cloudsim.examples
org.cloudbus.cloudsim.examples
org.cloudbus.cloudsim.examples
org.cloudbus.cloudsim.examples
workload.planetlab
sources
Referenced Libraries
docs
jars
build.xml
changelog.txt
examples.txt
license.txt
pom.xml
readme.txt
release_notes.txt
MyProject
CloudSimExample1.java
1 ckage org.cloudbus.cloudsim.examples;
2
4* Title: CloudSim Toolkit
11
12*port java.text.DecimalFormat;
Problems Javadoc Declaration Console
<terminated> CloudSimExample1 [Java Application] C:\Program Files\Java\jdk-20\bin\javaw.exe (19 Aug 2024, 11:25:57 am – 11:25:58
Starting CloudSimExample1...
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: Sending cloudlet 0 to VM #0
400.1: Broker: Cloudlet 0 received
400.1: Broker: All Cloudlets executed. Finishing...
400.1: Broker: Destroying VM #0
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.
========== OUTPUT ==========
Cloudlet ID STATUS Data center ID VM ID Time Start Time Finish Time
0 SUCCESS 2 0 400 0.1 400.1
CloudSimExample1 finished!


Project
cloudsim-3.0.3 [cloudsim-toolkit] ~/cloud
.idea
classes
docs
examples
org.cloudbus.cloudsim.examples
network
power
CloudSimExample1
CloudSimExample2
CloudSimExample3
CloudSimExample4
CloudSimExample5
CloudSimExample6
CloudSimExample7
CloudSimExample8
workload.planetlab
jars
sources
build.xml
changelog.txt
cloudsim-toolkit.iml
examples.txt
CloudSimExample1.java
PowerVmSelectionPolicyMaximumCorrelation.java
import org.cloudbus.cloudsim.provisioners.RamProvisionerSimple;
/**
* A simple example showing how to create a datacenter with one host and ru
* cloudlet on it.
*/
public class CloudSimExample1 {
Run 'CloudSimExample1.main()' ^⇧R
Debug 'CloudSimExample1.main()' ^⇧D
Run 'CloudSimExample1.main()' with Coverage
Run 'CloudSimExample1.main()' with 'CPU Profiler'
Run 'CloudSimExample1.main()' with 'Allocation Profiler'
Run 'CloudSimExample1.main()' with 'Java Flight Recorder'
Modify Run Configuration...
/**
* Creates main() to run this example.
*
* @param args the args
*/
/unused/
public static void main(String[] args) {


Project
cloudsim-3.0.3 [cloudsim-toolkit] ~/cloud
.idea
classes
docs
examples
org.cloudbus.cloudsim.examples
network
power
CloudSimExample1.java
PowerVmSelectionPolicyMaximumCorrelation.java
import org.cloudbus.cloudsim.provisioners.RamProvisionerSimple;
/**
* A simple example showing how to create a datacenter with one host and ru
* cloudlet on it.
*/
public class CloudSimExample1 {
Run: CloudSimExample1
Starting CloudSimExample1...
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: Sending cloudlet 0 to VM #0
400.1: Broker: Cloudlet 0 received
400.1: Broker: All Cloudlets executed. Finishing...
400.1: Broker: Destroying VM #0
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.
========== OUTPUT ==========
Cloudlet ID STATUS Data center ID VM ID Time Start Time Finish Time
0 SUCCESS 2 0 400 0.1 400.1
CloudSimExample1 finished!
Process finished with exit code 0


Project
Project
cloudsim-3.0.3 [cloudsim-toolkit] ~/cloud
.idea
classes
docs
examples
jars
sources
build.xml
changelog.txt
cloudsim-toolkit.iml
examples.txt
license.txt
pom.xml
readme.txt
release_notes.txt
External Libraries
Scratches and Consoles

RESULT:

Thus, the installation and configuration of CloudSim in Eclipse IDE has been successfully completed.