

# FOCUS task evaluation: a user guide

This is an end-user guide useful to complete two tasks with two different Java libraries.

## Setup installation

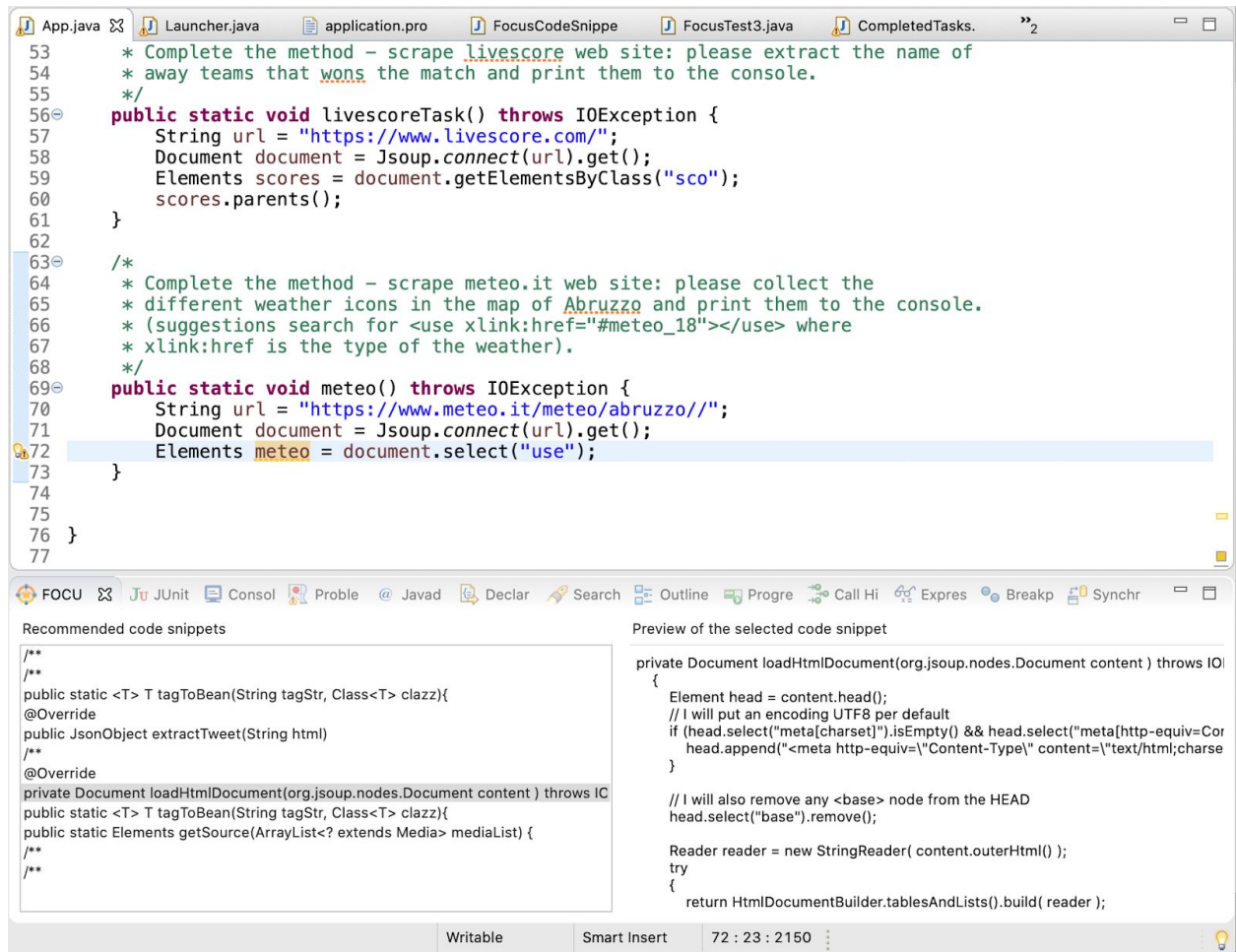
To install the Eclipse IDE, please follow these instructions:

1. Download and install Eclipse (<https://www.eclipse.org/downloads/> and choose the **JEE distribution**)
2. Be sure to have JRE and JDK software on your machine
3. Open Eclipse IDE and choose **File -> Import**
4. Choose **Maven-> Existing Maven project** as import wizard
5. Select the folder in which you downloaded the Github project (the links are available for each task)

## Focus plugin installation

To install the SOrec plugin, please follow these instructions:

1. Go to Help->Install new software
2. Choose Add and put this site in Location:  
<http://ci5.castalia.camp:8080/job/scava-ide/job/dev/lastSuccessfulBuild/artifact/eclipse-based-ide/org.eclipse.scava.root/releng/org.eclipse.scava.update/target/repository/>
3. Important: Be sure to avoid the usage of University connection for all these operations
4. Check CROSSMINER Eclipse-based IDE and choose Next->Install
5. Restart Eclipse
6. Go to Window->Preferences->CROSSMINER->Remote settings and choose the following configuration:
  - a. KnowledgeBase server address->http://83.212.75.210
  - b. KnowledgeBase server port -> 8080
  - c. WebDashboard base path -> https://crossminer.biterg.io
7. After you select the code, you can do Right click->**CROSSMINER->Request FOCUS Code Snippet recommendation** and you got something like that:



```
53  * Complete the method - scrape livescore web site: please extract the name of
54  * away teams that wons the match and print them to the console.
55  */
56  public static void livescoreTask() throws IOException {
57      String url = "https://www.livescore.com/";
58      Document document = Jsoup.connect(url).get();
59      Elements scores = document.getElementsByClass("sco");
60      scores.parents();
61  }
62
63  /*
64  * Complete the method - scrape meteo.it web site: please collect the
65  * different weather icons in the map of Abruzzo and print them to the console.
66  * (suggestions search for <use xlink:href="#meteo_18"></use> where
67  * xlink:href is the type of the weather).
68  */
69  public static void meteo() throws IOException {
70      String url = "https://www.meteo.it/meteo/abruzzo/";
71      Document document = Jsoup.connect(url).get();
72      Elements meteo = document.select("use");
73  }
74
75
76 }
77
```

Recommended code snippets

```
/**
/**
public static <T> T tagToBean(String tagStr, Class<T> clazz){
@Override
public JsonObject extractTweet(String html)
/**
@Override
private Document loadHtmlDocument(org.jsoup.nodes.Document content ) throws IO
public static <T> T tagToBean(String tagStr, Class<T> clazz){
public static Elements getSource(ArrayList<? extends Media> mediaList) {
/**
/**
```

Preview of the selected code snippet

```
private Document loadHtmlDocument(org.jsoup.nodes.Document content ) throws IO
{
    Element head = content.head();
    // I will put an encoding UTF8 per default
    if (head.select("meta[charset]").isEmpty() && head.select("meta[http-equiv=Cor
head.append("<meta http-equiv='Content-Type' content='text/html;charse
}

// I will also remove any <base> node from the HEAD
head.select("base").remove();

Reader reader = new StringReader( content.outerHtml() );
try
{
    return HtmlDocumentBuilder.tablesAndLists().build( reader );
}
```

## Task 1: Parsing command line parameters by apache-cli library - Maintenance Task (with FOCUS)

The Apache Commons CLI library provides an API for parsing command-line options passed to programs. It's also able to print help messages detailing the options available for a command-line tool.

[SQLDump](#) is a simple java project that exports SQL data to CSV files. In particular, it is a command-line utility to execute SQL queries and export results to a CSV file. Apache-cli is used for taking parameters from the command line.

Adapt the method of Launcher class to complete the following tasks with the support of apache-cli library:

1. SQLDump shall require the:
  - a. DB protocol,
  - b. address,
  - c. user,
  - d. username,

- e. password, and
  - f. Query
- as mandatory parameters;
2. SQLDump shall use a CSV file path as an optional parameter;
  3. SQLDump shall parse all required parameter values;
  4. SQLDump shall parse the optional parameter value if exists, a default value should be return otherwise;
  5. SQLDump shall print the API usage if the provided parameters are incomplete or incorrect.

The java program has to be run as the following shell command:

```
java -jar SQLDump-0.4.jar -url [jdbc:oracle:thin:@hostname:port:sid] -user [username]
-pass [password] -sql [query]
```

## Task 2: Scraping HTML pages by Gsoup - Development Task (without FOCUS)

[jsoup](#) is a Java library for working with real-world HTML. It provides a very convenient API for extracting and manipulating data, using the best of DOM, CSS, and jquery-like methods.

The jsoup-example project has been generated from the example methods provided in the <https://jsoup.org/> web site.

Please finalize the methods in App class to complete the following tasks:

1. scrape tuttojuve mobile website and select all thumbnails from "CALIOMERCATO" news. Print the results on the console;
2. scrape and modify vocegiallorossa mobile website: add a 2 fake news: one on top of the list and one to the bottom and store the new document as a file;
3. scrape livescore web site: please extract the name of away teams that won the match and print them to the console;
4. scrape meteo.it web site: collect the different weather icons in the map of Abruzzo and print them to the console.

The link for the corresponding questions is available here:

[https://docs.google.com/forms/d/e/1FAIpQLSdiMGDdq4RWP4aW1ftga\\_vlOyqC02HUCfD00Nht aJTgSpxpHA/viewform?usp=sf\\_link](https://docs.google.com/forms/d/e/1FAIpQLSdiMGDdq4RWP4aW1ftga_vlOyqC02HUCfD00Nht aJTgSpxpHA/viewform?usp=sf_link)