Knowledge Querying (I)

一、RDF4J配置

示例代码：https://github.com/eclipse/rdf4j/tree/main/examples

注意须知：

若电脑上已有java的同学，可跳过“安装JDK”与“环境变量配置步骤”。

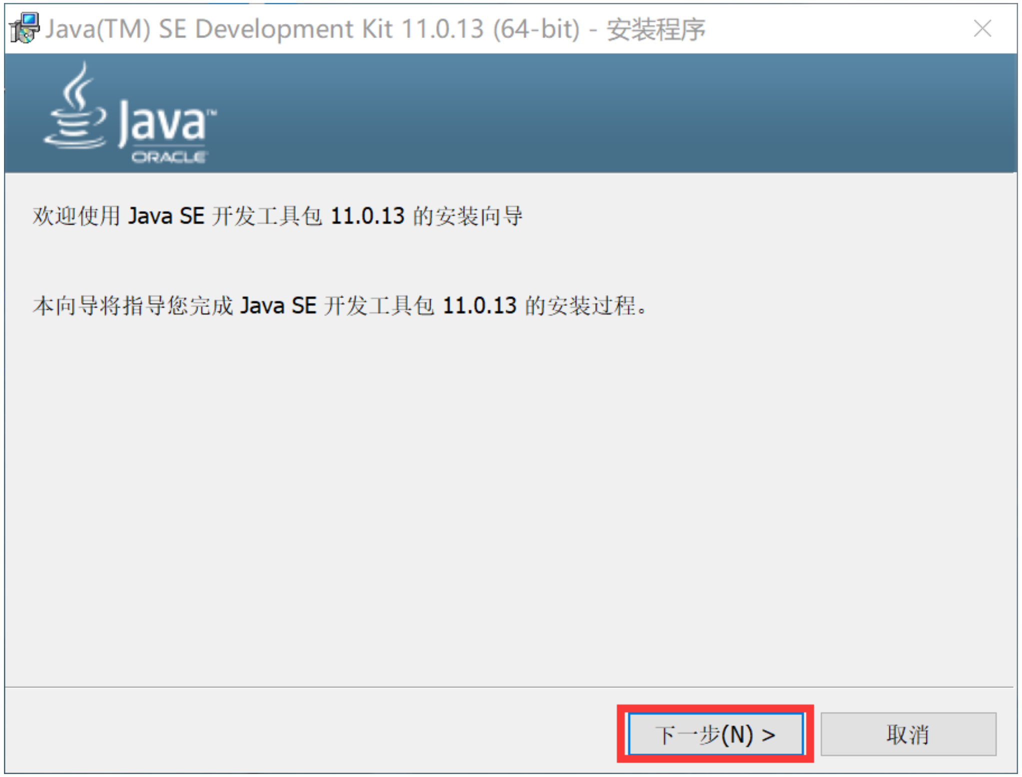
【命令页面输入java -version查询java版本】

以下操作以java 11为例，其他版本的java操作类似

Java JDK 11.0.13：https://www.oracle.com/java/technologies/downloads/#java11

环境配置—安装JDK【涉及到的安装包均已提供】

1. 双击安装包（Windows系统选择.exe后缀，Mac系统选择.tar.gz后缀，本示例将以Windows系统为例），点击“下一步”开始安装。



Java(TM) SE Development Kit 11.0.13 (64-bit) - 安装程序

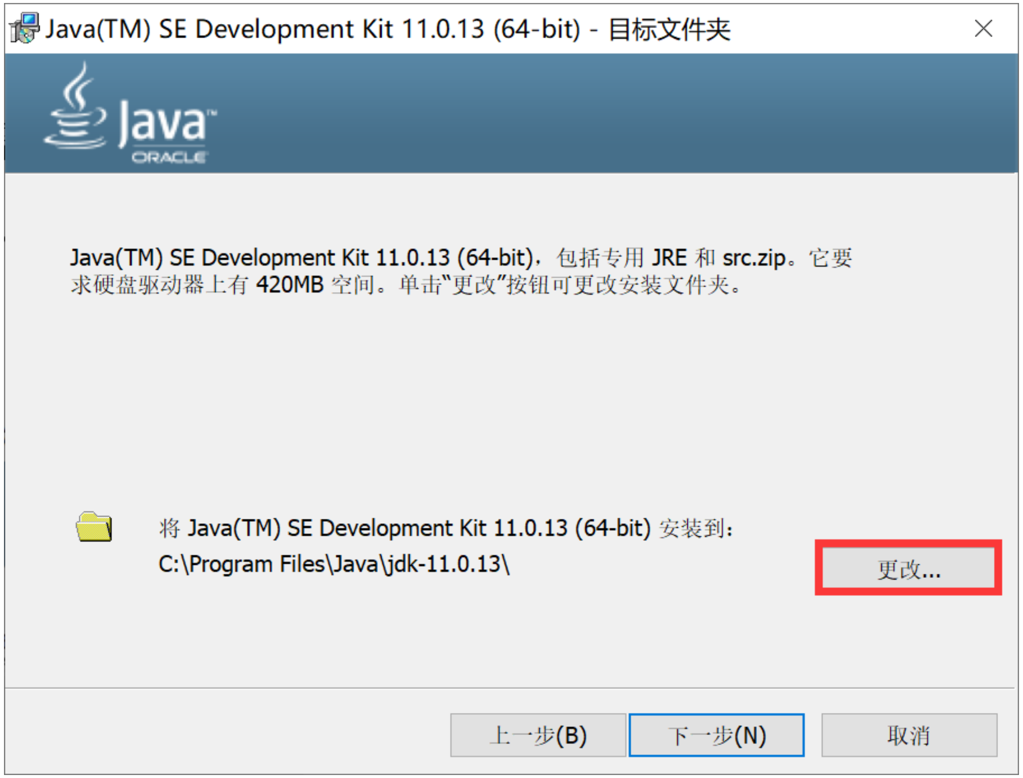
欢迎使用 Java SE 开发工具包 11.0.13 的安装向导

本向导将指导您完成 Java SE 开发工具包 11.0.13 的安装过程。

环境配置—安装JDK

2. 选择安装路径（请记录该安装路径，后续会用到），本示例采用的路径为 G:\java\JDK\jdk-11.0.13\

3. 点击下一步 继续安装



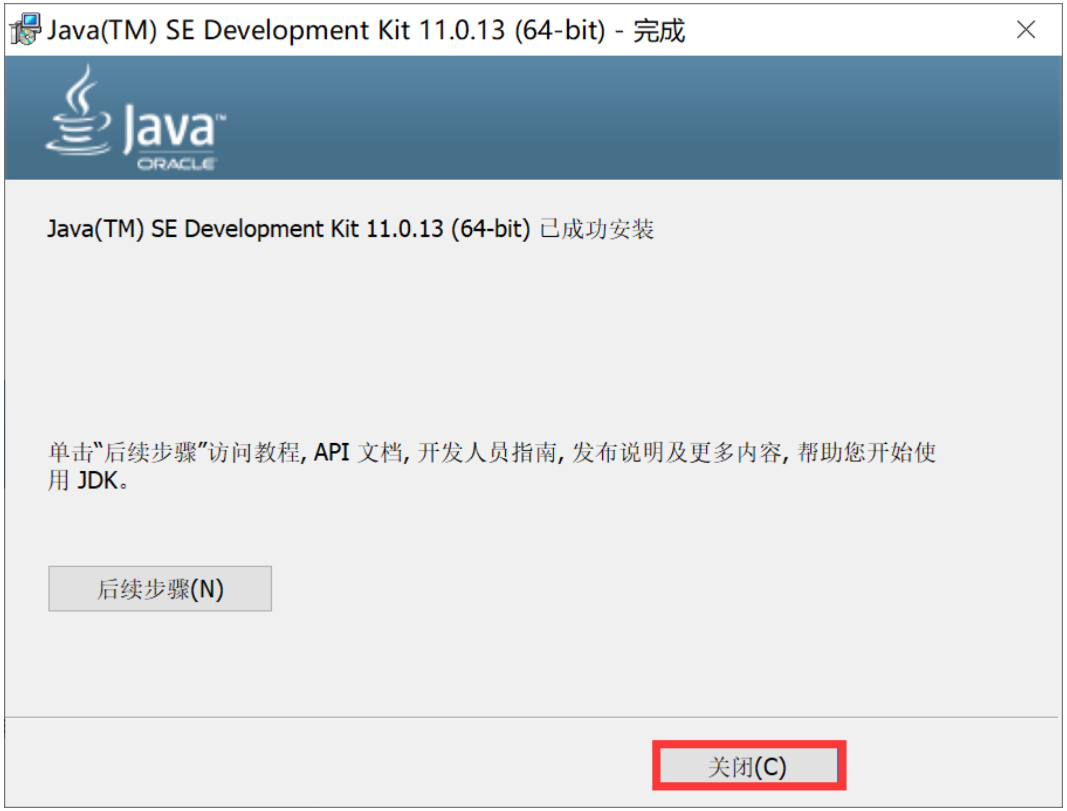
Java(TM) SE Development Kit 11.0.13 (64-bit) - 目标文件夹

Java(TM) SE Development Kit 11.0.13 (64-bit)，包括专用 JRE 和 src.zip。它要求硬盘驱动器上有 420MB 空间。单击"更改"按钮可更改安装文件夹。

将 Java(TM) SE Development Kit 11.0.13 (64-bit) 安装到：
C:\Program Files\Java\jdk-11.0.13\

环境配置—安装JDK

4. 等待JDK安装，点击“关闭”完成安装。



Java(TM) SE Development Kit 11.0.13 (64-bit) - 完成

Java(TM) SE Development Kit 11.0.13 (64-bit) 已成功安装

单击"后续步骤"访问教程, API 文档, 开发人员指南, 发布说明及更多内容, 帮助您开始使用 JDK。

环境配置—环境变量配置

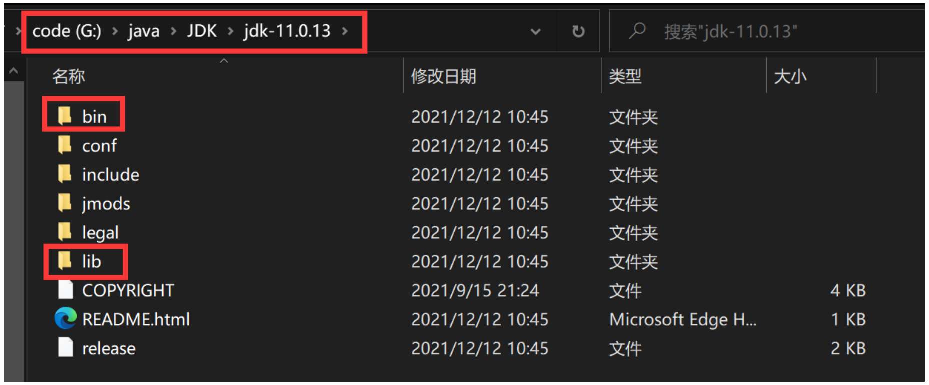
1. 找到对应路径。

记录3个路径：

1) 安装JDK时的安装路径，对应后续步骤中的 JAVA_HOME
   - 本示例为 G:\java\JDK\jdk-11.0.13\

2) JDK安装路径下的lib文件，对应后续步骤中的 CLASSPATH
   - 本示例为 G:\java\JDK\jdk-11.0.13\lib

3) JDK安装路径下的bin文件，对应后续步骤中的 Path
   - 本示例为 G:\java\JDK\jdk-11.0.13\bin



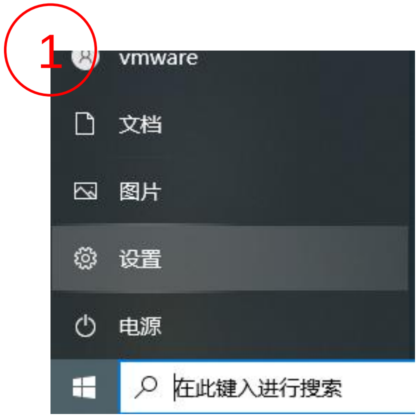
code (G:) › java › JDK › jdk-11.0.13 ›

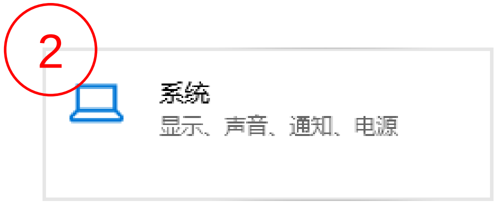
搜索"jdk-11.0.13"

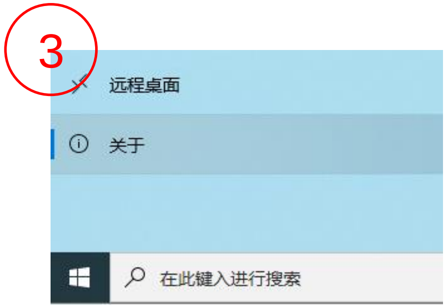
名称	修改日期	类型	大小
bin	2021/12/12 10:45	文件夹	
conf	2021/12/12 10:45	文件夹	
include	2021/12/12 10:45	文件夹	
jmods	2021/12/12 10:45	文件夹	
legal	2021/12/12 10:45	文件夹	
lib	2021/12/12 10:45	文件夹	
COPYRIGHT	2021/9/15 21:24	文件	4 KB
README.html	2021/12/12 10:45	Microsoft Edge H...	1 KB
release	2021/12/12 10:45	文件	2 KB

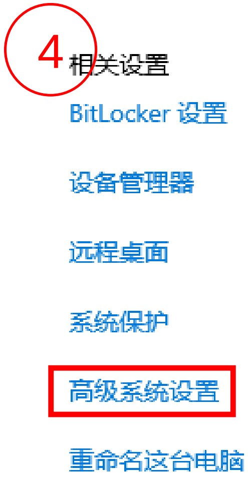
环境配置—环境变量配置

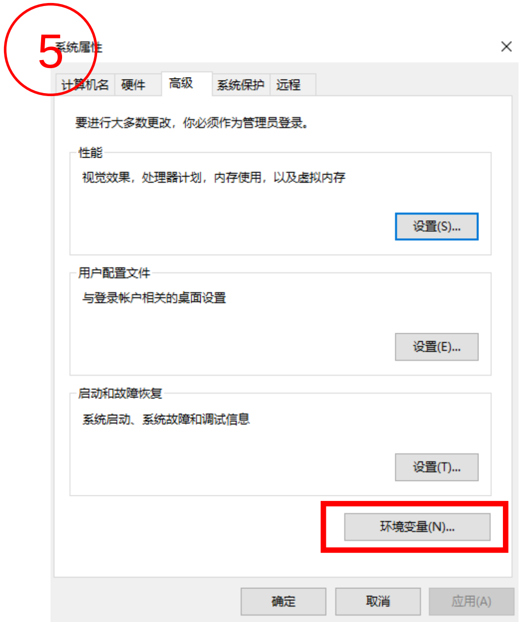
2. 打开 设置 --> 系统 --> 关于 --> 高级系统设置 --> 环境变量











要进行大多数更改，你必须作为管理员登录。

性能

视觉效果，处理器计划，内存使用，以及虚拟内存

用户配置文件

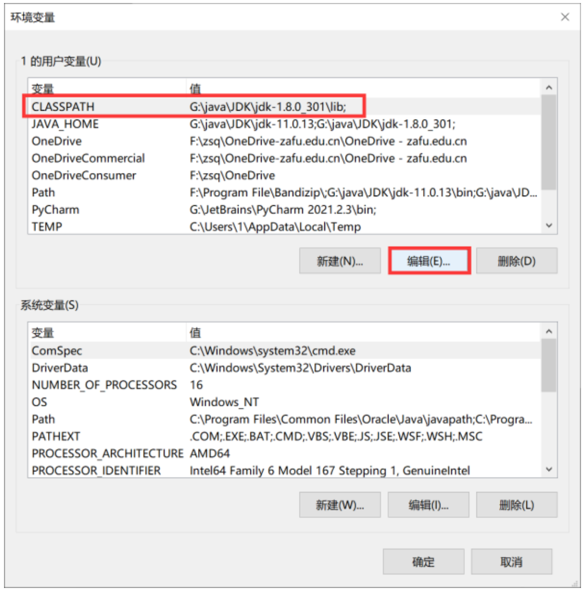
与登录帐户相关的桌面设置

启动和故障恢复

系统启动、系统故障和调试信息

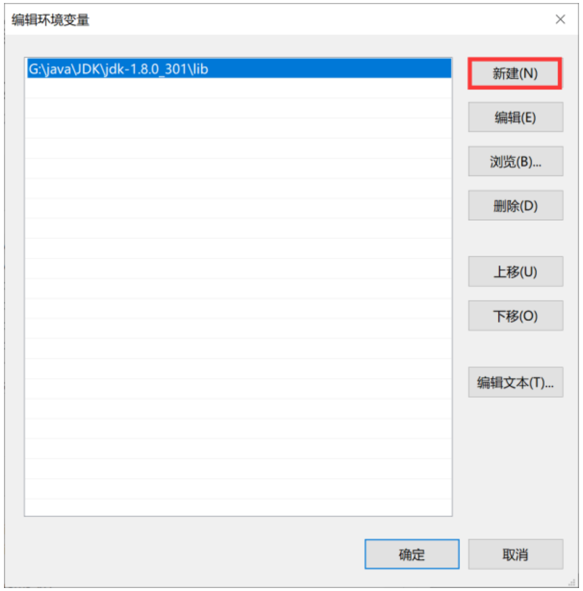
环境配置—环境变量配置

3. 编辑“CLASSPATH”环境变量，选中“CLASSPATH”，点击“编辑”，在弹出的窗口选择“新建”，然后输入步骤1中的记录值点“上移”，将新建的路径调整到最顶端。

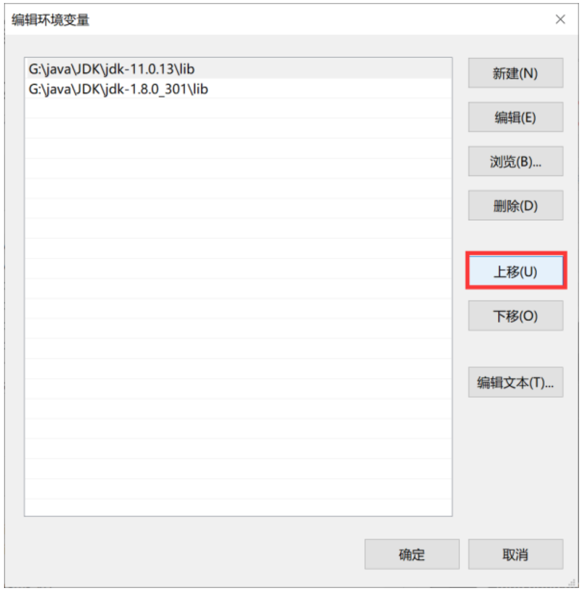


变量	值
CLASSPATH	G:\java\JDK\jdk-1.8.0_301\lib;
JAVA_HOME	G:\java\JDK\jdk-11.0.13;G:\java\JDK\jdk-1.8.0_301;
OneDrive	F:\zsq\OneDrive-zafu.edu.cn\OneDrive - zafu.edu.cn
OneDriveCommercial	F:\zsq\OneDrive-zafu.edu.cn\OneDrive - zafu.edu.cn
OneDriveConsumer	F:\zsq\OneDrive
Path	F:\Program File\Bandizip\;G:\java\JDK\jdk-11.0.13\bin;G:\java\JD...
PyCharm	G:\JetBrains\PyCharm 2021.2.3\bin;
TEMP	C:\Users\1\AppData\Local\Temp

变量	值
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	16
OS	Windows_NT
Path	C:\Program Files\Common Files\Oracle\Java\javapath;C:\Progra...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64
PROCESSOR_IDENTIFIER	Intel64 Family 6 Model 167 Stepping 1, GenuineIntel



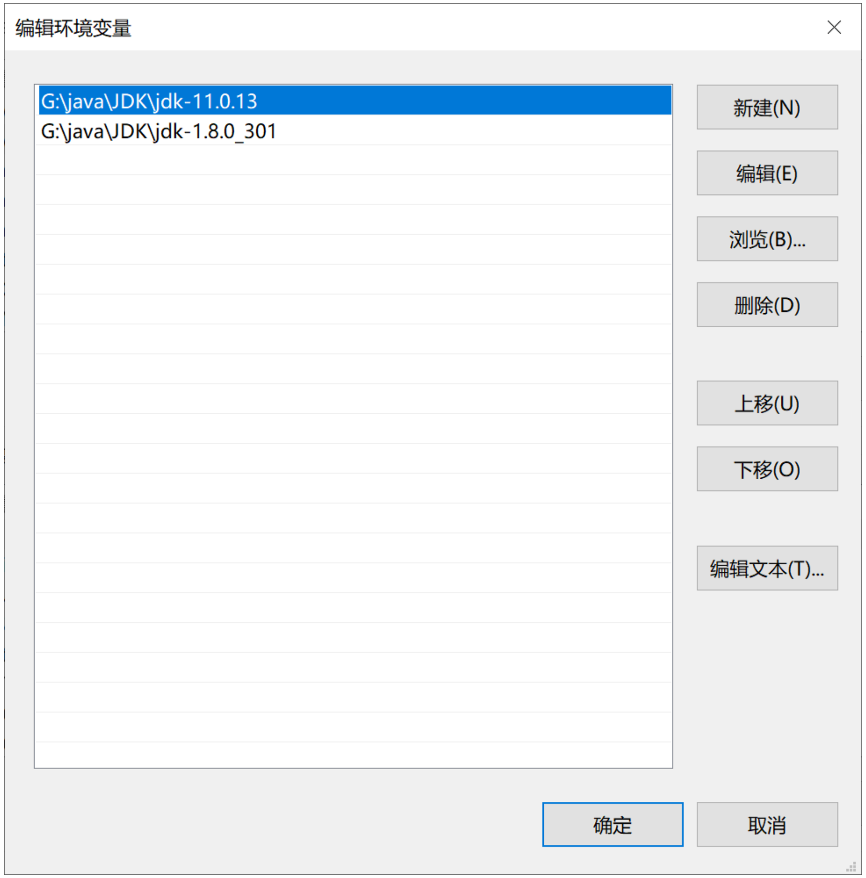
G:\java\JDK\jdk-1.8.0_301\lib



G:\java\JDK\jdk-11.0.13\lib
G:\java\JDK\jdk-1.8.0_301\lib

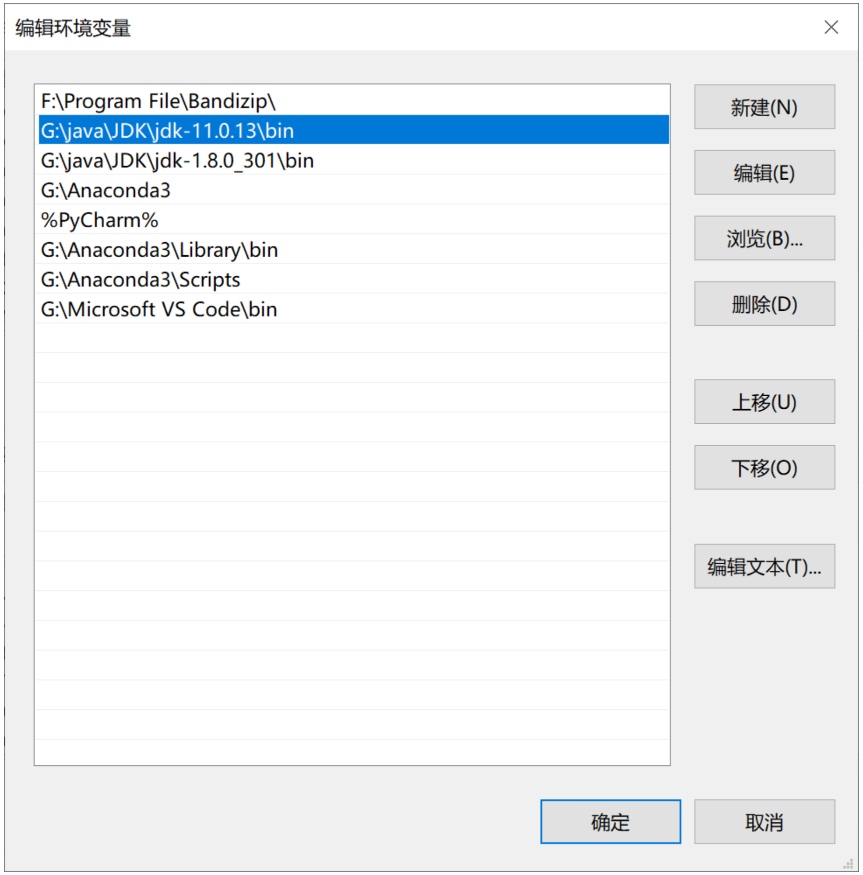
环境配置—环境变量配置

4. 依次编辑 “JAVA_HOME”和“Path”环境变量。



编辑环境变量

G:\java\JDK\jdk-11.0.13
G:\java\JDK\jdk-1.8.0_301

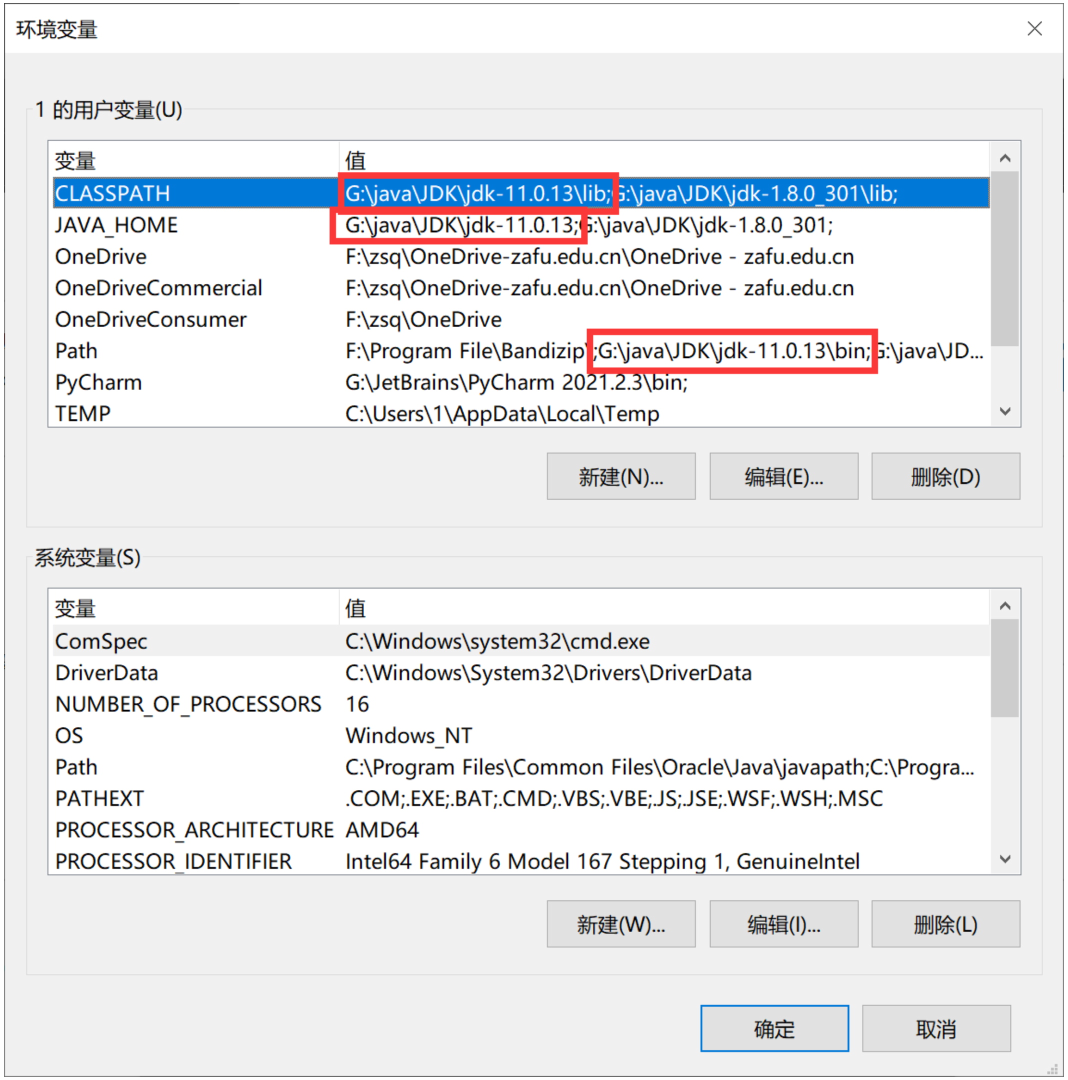


编辑环境变量

F:\Program File\Bandizip\
G:\java\JDK\jdk-11.0.13\bin
G:\java\JDK\jdk-1.8.0_301\bin
G:\Anaconda3
%PyCharm%
G:\Anaconda3\Library\bin
G:\Anaconda3\Scripts
G:\Microsoft VS Code\bin

环境配置—环境变量配置

5. 最终完成界面如右图所示，点击“确定”



1 的用户变量(U)

变量	值
CLASSPATH	G:\java\JDK\jdk-11.0.13\lib;G:\java\JDK\jdk-1.8.0_301\lib;
JAVA_HOME	G:\java\JDK\jdk-11.0.13;G:\java\JDK\jdk-1.8.0_301;
OneDrive	F:\zsq\OneDrive-zafu.edu.cn\OneDrive - zafu.edu.cn
OneDriveCommercial	F:\zsq\OneDrive-zafu.edu.cn\OneDrive - zafu.edu.cn
OneDriveConsumer	F:\zsq\OneDrive
Path	F:\Program File\Bandizip\;G:\java\JDK\jdk-11.0.13\bin;G:\java\JD...
PyCharm	G:\JetBrains\PyCharm 2021.2.3\bin;
TEMP	C:\Users\1\AppData\Local\Temp

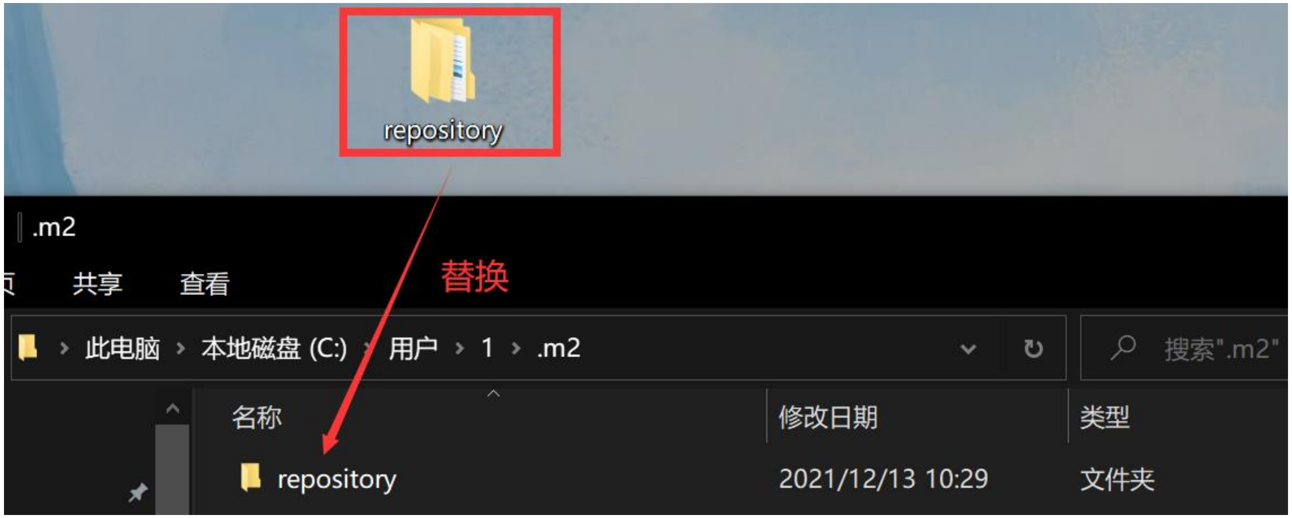
系统变量(S)

变量	值
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	16
OS	Windows_NT
Path	C:\Program Files\Common Files\Oracle\Java\javapath;C:\Progra...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64
PROCESSOR_IDENTIFIER	Intel64 Family 6 Model 167 Stepping 1, GenuineIntel

环境配置—打开项目文件

1. 直接替换本地Maven仓库

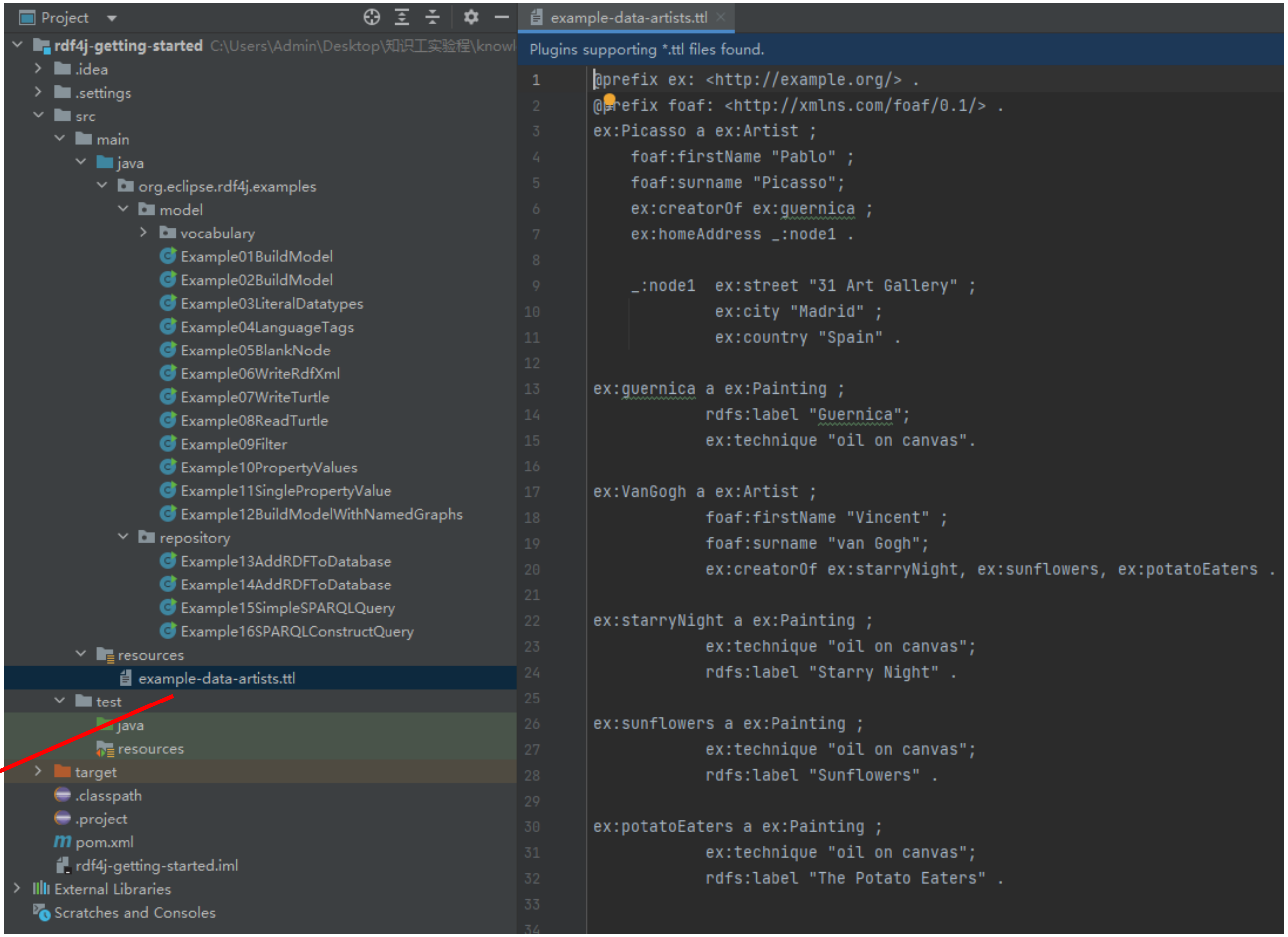
Maven项目将自动下载项目依赖的jar包，这个过程较慢。可将下发的“repository.zip”解压后替换本地Maven仓库中的repository。本地Maven仓库默认位于C:\Users\用户名\.m2 文件夹下。



名称	修改日期	类型
repository	2021/12/13 10:29	文件夹

环境配置—打开项目文件

2. 使用IDEA成功打开的项目界面如右图所示。



Plugins supporting *.ttl files found.

```
@prefix ex: <http://example.org/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
ex:Picasso a ex:Artist ;
    foaf:firstName "Pablo" ;
    foaf:surname "Picasso";
    ex:creatorOf ex:guernica ;
    ex:homeAddress _:node1 .

    _:node1  ex:street "31 Art Gallery" ;
             ex:city "Madrid" ;
             ex:country "Spain" .

ex:guernica a ex:Painting ;
            rdfs:label "Guernica";
            ex:technique "oil on canvas".

ex:VanGogh a ex:Artist ;
            foaf:firstName "Vincent" ;
            foaf:surname "van Gogh";
            ex:creatorOf ex:starryNight, ex:sunflowers, ex:potatoEaters .

ex:starryNight a ex:Painting ;
            ex:technique "oil on canvas";
            rdfs:label "Starry Night" .

ex:sunflowers a ex:Painting ;
            ex:technique "oil on canvas";
            rdfs:label "Sunflowers" .

ex:potatoEaters a ex:Painting ;
            ex:technique "oil on canvas";
            rdfs:label "The Potato Eaters" .
```

example-data-artists.ttl
用于存放turtle数据

根据下面的链接配置环境
https://blog.csdn.net/weixin_53573372/article/details/118875386

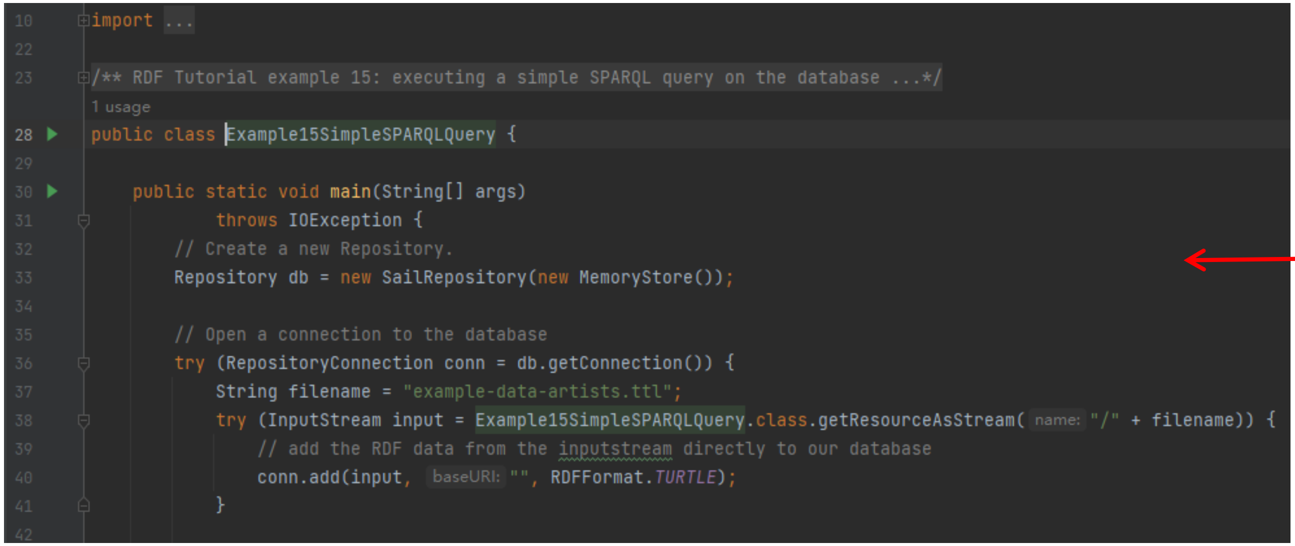
SPARQL SELECT查询

SPARQL查询语句：

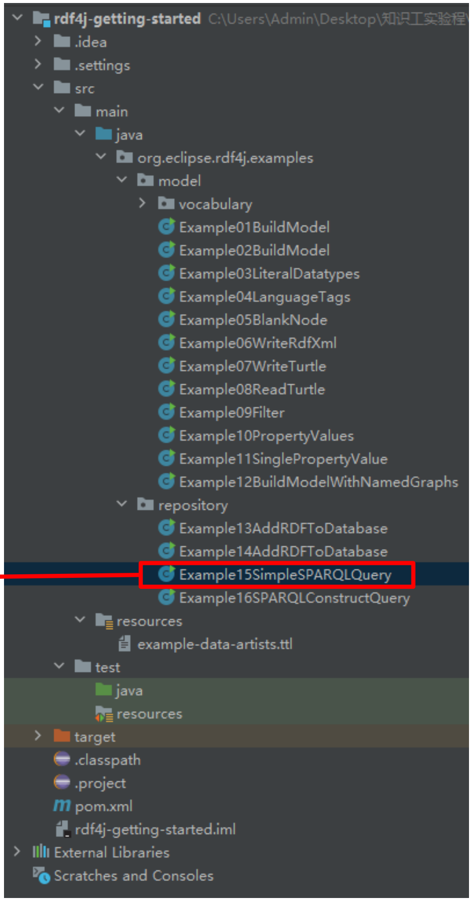
```
PREFIX ex: <http://example.org/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?s ?n
WHERE {
        ?s a ex:Artist;
                foaf:firstName ?n .
}
```

查询有名字的艺术家以及他们的名字

SPARQL SELECT查询



```
10   import ...
22
23   /** RDF Tutorial example 15: executing a simple SPARQL query on the database ...*/
     1 usage
28   public class Example15SimpleSPARQLQuery {
29
30       public static void main(String[] args)
31               throws IOException {
32           // Create a new Repository.
33           Repository db = new SailRepository(new MemoryStore());
34
35           // Open a connection to the database
36           try (RepositoryConnection conn = db.getConnection()) {
37               String filename = "example-data-artists.ttl";
38               try (InputStream input = Example15SimpleSPARQLQuery.class.getResourceAsStream( name: "/" + filename)) {
39                   // add the RDF data from the inputstream directly to our database
40                   conn.add(input,  baseURI: "", RDFFormat.TURTLE);
41               }
42
```



```
rdf4j-getting-started  C:\Users\Admin\Desktop\知识工实验程
  .idea
  .settings
  src
    main
      java
        org.eclipse.rdf4j.examples
          model
            vocabulary
            Example01BuildModel
            Example02BuildModel
            Example03LiteralDatatypes
            Example04LanguageTags
            Example05BlankNode
            Example06WriteRdfXml
            Example07WriteTurtle
            Example08ReadTurtle
            Example09Filter
            Example10PropertyValues
            Example11SinglePropertyValue
            Example12BuildModelWithNamedGraphs
          repository
            Example13AddRDFToDatabase
            Example14AddRDFToDatabase
            Example15SimpleSPARQLQuery
            Example16SPARQLConstructQuery
      resources
        example-data-artists.ttl
    test
      java
      resources
  target
  .classpath
  .project
  pom.xml
  rdf4j-getting-started.iml
External Libraries
Scratches and Consoles
```

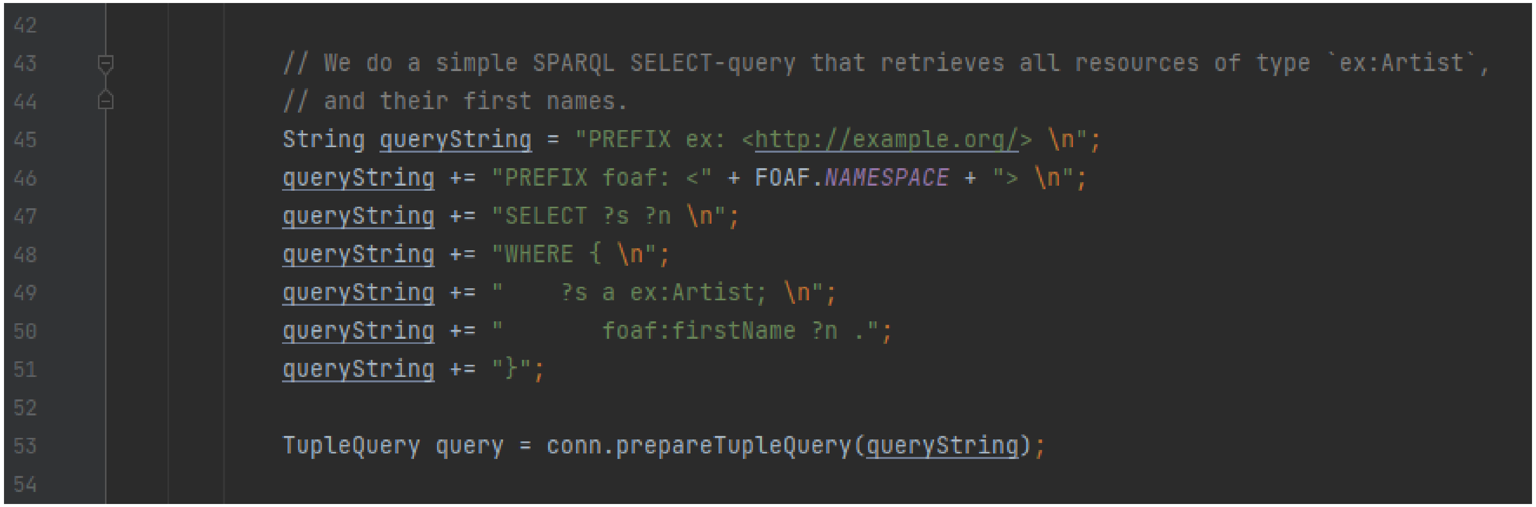
line 33　首先建立一个内存仓库db存取数据

line 36　建立一个仓库链接conn连接到内存仓库db

line 38　打开数据文件作为输入流input

line 40　将输入流input通过数据库连接conn加入到内存仓库db中

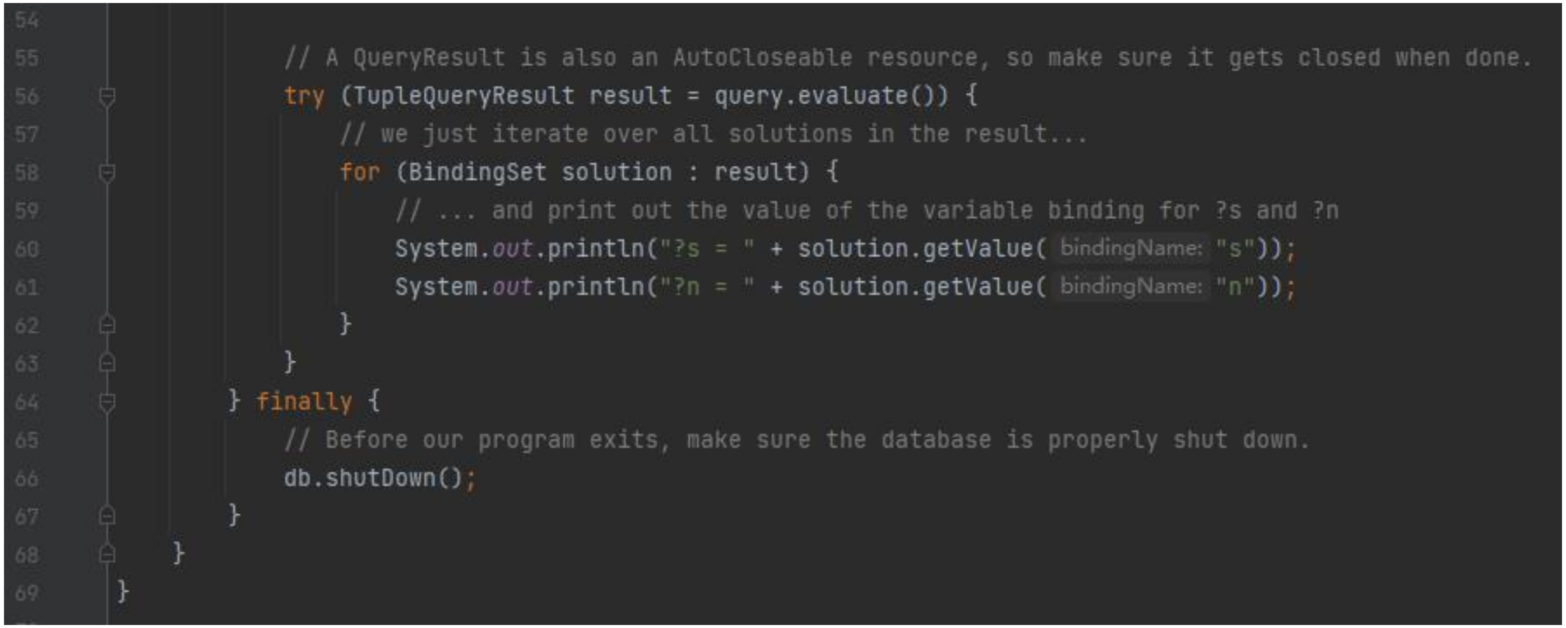
SPARQL SELECT查询



```
42
43      // We do a simple SPARQL SELECT-query that retrieves all resources of type `ex:Artist`,
44      // and their first names.
45      String queryString = "PREFIX ex: <http://example.org/> \n";
46      queryString += "PREFIX foaf: <" + FOAF.NAMESPACE + "> \n";
47      queryString += "SELECT ?s ?n \n";
48      queryString += "WHERE { \n";
49      queryString += "    ?s a ex:Artist; \n";
50      queryString += "       foaf:firstName ?n .";
51      queryString += "}";
52
53      TupleQuery query = conn.prepareTupleQuery(queryString);
54
```

line 45 - 51　　将SPARQL语句存入java字符串queryString中
line 53　　在conn中用SPARQL查询，并将三元组返回到query中

SPARQL SELECT查询



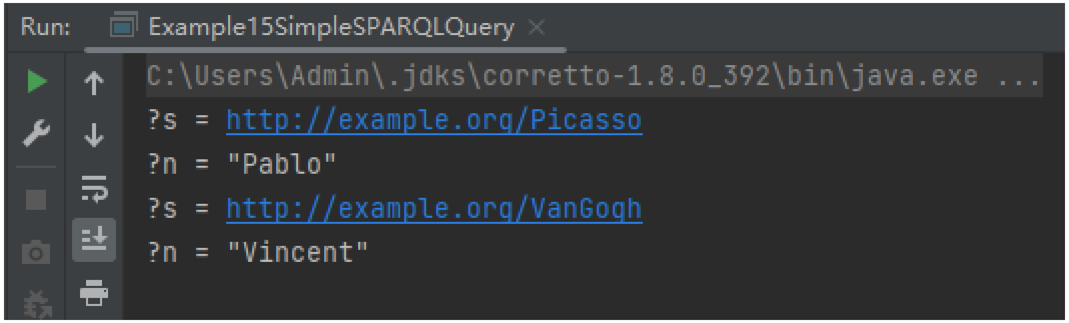
```
54
55              // A QueryResult is also an AutoCloseable resource, so make sure it gets closed when done.
56              try (TupleQueryResult result = query.evaluate()) {
57                  // we just iterate over all solutions in the result...
58                  for (BindingSet solution : result) {
59                      // ... and print out the value of the variable binding for ?s and ?n
60                      System.out.println("?s = " + solution.getValue( bindingName: "s"));
61                      System.out.println("?n = " + solution.getValue( bindingName: "n"));
62                  }
63              }
64          } finally {
65              // Before our program exits, make sure the database is properly shut down.
66              db.shutDown();
67          }
68      }
69  }
```

line 58-62　　输出查询到的结果
line 64-67　　关闭内存仓库db

SPARQL SELECT查询

右击Example15SimpleSPARQLQuery -> Run 运行

查询返回结果：



```
C:\Users\Admin\.jdks\corretto-1.8.0_392\bin\java.exe ...
?s = http://example.org/Picasso
?n = "Pablo"
?s = http://example.org/VanGogh
?n = "Vincent"
```

三、作业

作业

1.针对下列自然语言，使用SPARQL查询语句返回结果：

(1) List the creators (including paintings) of Guernica and Sunflowers, respectively.

(2) List all the artists (including living places) who live in Spain or other places.

(3) List all paintings, their names, and the corresponding techniques.