



《大数据基础编程、实验和案例教程（第2版）》

教材官网：

<http://dmlab.xmu.edu.cn/post/bigdatappractice2/>

温馨提示：编辑幻灯片母版，可以修改每页PPT的厦大校徽和底部文字

第5章 HBase的安装和基础编程

（PPT版本号：2020年12月版本）



扫一扫访问教材官网

林子雨

厦门大学计算机科学系

E-mail: ziyulin@xmu.edu.cn ▶▶

主页: <http://dmlab.xmu.edu.cn/linziyu>





教材简介

本书是与《大数据技术原理与应用（第3版）》教材配套的唯一指定实验指导书

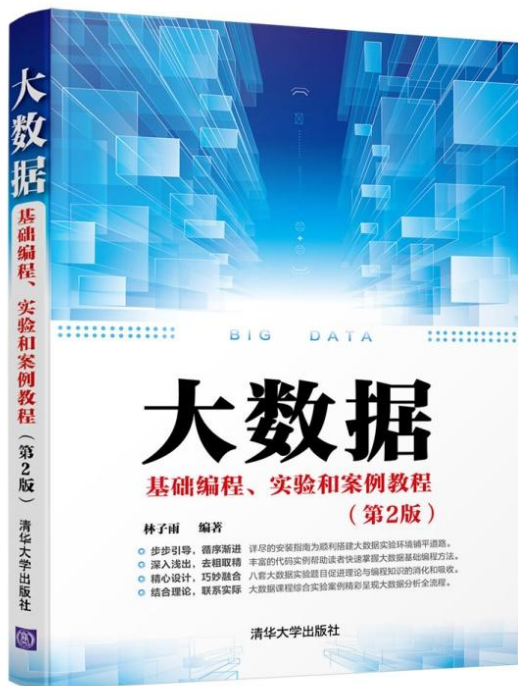
林子雨编著《大数据基础编程、实验和案例教程（第2版）》

清华大学出版社 ISBN:978-7-302-55977-1 定价：69元，2020年10月第2版

教材官网：<http://dbllab.xmu.edu.cn/post/bigdatapRACTICE2/>



扫一扫访问
教材官网



- 步步引导，循序渐进，详尽的安装指南为顺利搭建大数据实验环境铺平道路
- 深入浅出，去粗取精，丰富的代码实例帮助快速掌握大数据基础编程方法
- 精心设计，巧妙融合，八套大数据实验题目促进理论与编程知识的消化和吸收
- 结合理论，联系实际，大数据课程综合实验案例精彩呈现大数据分析全流程



提纲

- 5.1 安装HBase
- 5.2 HBase的配置
- 5.3 HBase常用Shell命令
- 5.4 HBase编程实践



高校大数据课程

公共服务平台

百度搜索厦门大学数据库实验室网站访问平台





5.1 安装HBase

- 5.1.1 下载安装文件
- 5.1.2 配置环境变量
- 5.1.3 添加用户权限
- 5.1.4 查看HBase版本信息



5.1.1 下载安装文件

访问HBase官网下载安装文件hbase-2.2.2-bin.tar.gz文件

下载完安装文件以后，需要对文件进行解压。按照Linux系统使用的默认规范，用户安装的软件一般都是存放在“/usr/local/”目录下。请使用hadoop用户登录Linux系统，打开一个终端，执行如下命令：

```
$ sudo tar -zxf ~/下载/hbase-2.2.2-bin.tar.gz -C /usr/local
```

将解压的文件名hbase-2.2.2改为hbase，以方便使用，命令如下：

```
$ sudo mv /usr/local/hbase-2.2.2 /usr/local/hbase
```



5.1.2 配置环境变量

将HBase安装目录下的bin目录（即/usr/local/hbase/bin）添加到系统的PATH环境变量中，这样，每次启动HBase时就不需要到“/usr/local/hbase”目录下执行启动命令，方便HBase的使用。请使用vim编辑器打开“~/.bashrc”文件，命令如下：

```
$ vim ~/.bashrc
```

打开.bashrc文件以后，可以看到，已经存在如下所示的PATH环境变量的配置信息，因为，之前在第3章安装配置Hadoop时，我们已经为Hadoop添加了PATH环境变量的配置信息：

```
export PATH=$PATH:/usr/local/hadoop/sbin:/usr/local/hadoop/bin
```



5.1.2 配置环境变量

这里，需要把HBase的bin目录“/usr/local/hbase/bin”追加到PATH中。当要在PATH中继续加入新的路径时，只要用英文冒号“:”隔开，把新的路径加到后面即可，追加后的结果如下：

```
export PATH=$PATH:/usr/local/hadoop/sbin:/usr/local/hadoop/bin:  
/usr/local/hbase/bin
```

添加后，执行如下命令使设置生效：

```
$ source ~/.bashrc
```




5.1.3 添加用户权限

需要为当前登录Linux系统的hadoop用户添加访问HBase目录的权限，将HBase安装目录下的所有文件的所有者改为hadoop，命令如下：

```
$ cd /usr/local  
$ sudo chown -R hadoop ./hbase
```




5.1.4 查看HBase版本信息

可以通过如下命令查看HBase版本信息，以确认HBase已经安装成功：

```
$ /usr/local/hbase/bin/hbase version
```

```
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase 2.2.2
Source code repository git://6ad68c41b902/opt/hbase-rm/output/hbase revision=e6513a76c91cceda95dad7af246ac81d46fa2589
Compiled by hbase-rm on Sat Oct 19 10:10:12 UTC 2019
From source with checksum 4d23f97701e395c5d34db1882ac5021b
```



5.2 HBase的配置

5.2.1 单机模式配置

5.2.2 伪分布式配置



5.2.1 单机模式配置

1. 配置hbase-env.sh文件

使用vim编辑器打开“/usr/local/hbase/conf/hbase-env.sh”，命令如下：

```
$ vim /usr/local/hbase/conf/hbase-env.sh
```

打开hbase-env.sh文件以后，需要在hbase-env.sh文件中配置JAVA_HOME、HBASE_CLASSPATH和HBASE_MANAGES_ZK。其中，HBASE_CLASSPATH设置为本机Hadoop安装目录下的conf目录（即/usr/local/hadoop/conf）。JAVA_HOME和HBASE_MANAGES_ZK的配置方法和上面单机模式的配置方法相同。修改后的hbase-env.sh文件应该包含如下三行信息：

```
export JAVA_HOME=/usr/lib/jvm/ jdk1.8.0_162
export HBASE_CLASSPATH=/usr/local/hbase/conf
export HBASE_MANAGES_ZK=true
```



5.2.1 单机模式配置

2. 配置hbase-site.xml文件

使用vim编辑器打开并编辑“/usr/local/hbase/conf/hbase-site.xml”文件，命令如下：

```
$ vim /usr/local/hbase/conf/hbase-site.xml
```



5.2.1 单机模式配置

```
<configuration>
  <property>
    <name>hbase.rootdir</name>
    <value>hdfs://localhost:9000/hbase</value>
  </property>
  <property>
    <name>hbase.cluster.distributed</name>
    <value>true</value>
  </property>
  <property>
    <name>hbase.unsafe.stream.capability.enforce</name>
    <value>>false</value>
  </property>
</configuration>
```



5.2.1 单机模式配置

3. 启动运行HBase

首先登陆SSH，由于之前在“第3章 Hadoop的安装和使用”中已经设置了无密码登录，因此这里不需要密码。然后，切换至“/usr/local/hadoop”，启动Hadoop，让HDFS进入运行状态，从而可以为HBase存储数据，具体命令如下：

```
$ ssh localhost  
$ cd /usr/local/hadoop  
$ ./sbin/start-dfs.sh
```

然后，启动HBase，命令如下：

```
$ cd /usr/local/hbase  
$ bin/start-hbase.sh
```



5.2.1 单机模式配置

输入命令jps，如果出现以下进程，则说明HBase启动成功：

```
$ jps
HMaster
HQuorumPeer
NameNode
HRegionServer
SecondaryNameNode
DataNode
```

现在就可以进入HBase Shell模式，命令如下：

```
$ bin/hbase shell #进入HBase shell命令行模式
```




5.2.1 单机模式配置

4. 停止运行HBase

最后，可以使用如下命令停止HBase运行：

```
$ bin/stop-hbase.sh
```

关闭HBase以后，如果不再使用Hadoop，就可以运行如下命令关闭Hadoop：

```
$ cd /usr/local/hadoop  
$ ./sbin/stop-dfs.sh
```



5.3 HBase常用Shell命令

在使用具体的Shell命令操作HBase数据之前，需要首先启动Hadoop，然后再启动HBase，并且启动HBase Shell，进入Shell命令提示符状态，具体命令如下：

```
$ cd /usr/local/hadoop
$ ./sbin/start-dfs.sh
$ cd /usr/local/hbase
$ ./bin/start-hbase.sh
$ ./bin/hbase shell
```

5.3.1 在HBase中创建表

5.3.2 添加数据

5.3.3 查看数据

5.3.4 删除数据

5.3.5 删除表

5.3.6 查询历史数据

5.3.7 退出HBase数据库



5.3.1 在HBase中创建表

假设这里要创建一个表student，该表包含Sname、Ssex、Sage、Sdept、course等字段。需要注意的是，在关系型数据库（比如MySQL）中，需要首先创建数据库，然后再创建表，但是，在HBase数据库中，不需要创建数据库，只要直接创建表就可以。在HBase中创建student表的Shell命令如下：

```
hbase> create 'student','Sname','Ssex','Sage','Sdept','course'
```



5.3.1 在HBase中创建表

创建完“student”表后，可通过describe命令查看“student”表的基本信息。

```
hbase(main):009:0> describe 'student'
DESCRIPTION                               ENABLED
'student', {NAME => 'Sage', DATA_BLOCK_ENCODING => 'NONE', BLOOMFILTER => 'NONE', REPLICATION_SCOPE => '0', VERSIONS => '3', COMPRESSION => 'NONE', MIN_VERSIONS => '0', TTL => '2147483647', KEEP_DELETED_CELLS => 'false', BLOCKSIZE => '65536', IN_MEMORY => 'false', ENCODE_ON_DISK => 'true', BLOCKCACHE => 'true'}, {NAME => 'Sdept', DATA_BLOCK_ENCODING => 'NONE', BLOOMFILTER => 'NONE', REPLICATION_SCOPE => '0', VERSIONS => '3', COMPRESSION => 'NONE', MIN_VERSIONS => '0', TTL => '2147483647', KEEP_DELETED_CELLS => 'false', BLOCKSIZE => '65536', IN_MEMORY => 'false', ENCODE_ON_DISK => 'true', BLOCKCACHE => 'true'}, {NAME => 'Sname', DATA_BLOCK_ENCODING => 'NONE', BLOOMFILTER => 'NONE', REPLICATION_SCOPE => '0', VERSIONS => '3', COMPRESSION => 'NONE', MIN_VERSIONS => '0', TTL => '2147483647', KEEP_DELETED_CELLS => 'false', BLOCKSIZE => '65536', IN_MEMORY => 'false', ENCODE_ON_DISK => 'true', BLOCKCACHE => 'true'}, {NAME => 'Ssex', DATA_BLOCK_ENCODING => 'NONE', BLOOMFILTER => 'NONE', REPLICATION_SCOPE => '0', VERSIONS => '3', COMPRESSION => 'NONE', MIN_VERSIONS => '0', TTL => '2147483647', KEEP_DELETED_CELLS => 'false', BLOCKSIZE => '65536', IN_MEMORY => 'false', ENCODE_ON_DISK => 'true', BLOCKCACHE => 'true'}, {NAME => 'course', DATA_BLOCK_ENCODING => 'NONE', BLOOMFILTER => 'NONE', REPLICATION_SCOPE => '0', VERSIONS => '3', COMPRESSION => 'NONE', MIN_VERSIONS => '0', TTL => '2147483647', KEEP_DELETED_CELLS => 'false', BLOCKSIZE => '65536', IN_MEMORY => 'false', ENCODE_ON_DISK => 'true', BLOCKCACHE => 'true'}
1 row(s) in 0.0380 seconds
```

可以使用list命令查看当前HBase数据库中已经创建了哪些表，命令如下：

```
hbase> list
```



5.3.2 添加数据

可以用Shell命令手工插入数据，命令如下：

```
hbase> put 'student','95001','Sname','LiYing'
```

下面继续添加4个单元格的数据，用来记录LiYing同学的相关信息，命令如下：

```
hbase> put 'student','95001','Ssex','male'  
hbase> put 'student','95001','Sage','22'  
hbase> put 'student','95001','Sdept','CS'  
hbase> put 'student','95001','course:math','80'
```



5.3.3 查看数据

HBase中有两个用于查看数据的命令：

- **get**命令：用于查看表的某一个单元格数据；
- **scan**命令：用于查看某个表的全部数据。

比如，可以使用如下命令返回student表中95001行的数据：

```
hbase(main):024:0> get 'student','95001'
COLUMN                                CELL
Sage:                                  timestamp=1442912525676, value=20
Sdept:                                  timestamp=1442912586483, value=CS
Sname:                                  timestamp=1442912495442, value=LiYing
Ssex:                                    timestamp=1442912510852, value=male
course:math                             timestamp=1442912802499, value=80
5 row(s) in 0.0080 seconds
```



下面使用scan命令查询student表的全部数据:

```
hbase(main):025:0> scan 'student'  
ROW                                COLUMN+CELL  
 95001                             column=Sage:, timestamp=1442912525676, value=20  
 95001                             column=Sdept:, timestamp=1442912586483, value=CS  
 95001                             column=Sname:, timestamp=1442912495442, value=LiYing  
 95001                             column=Ssex:, timestamp=1442912510852, value=male  
 95001                             column=course:math, timestamp=1442912802499, value=80  
1 row(s) in 0.0120 seconds
```




5.3.4 删除数据

首先，使用delete命令删除student表中95001这行中的Ssex列的所有数据

```
hbase(main):026:0> delete 'student','95001','Ssex'
0 row(s) in 0.0020 seconds

hbase(main):027:0> get 'student','95001'
COLUMN                                CELL
Sage:                                  timestamp=1442912525676, value=20
Sdept:                                  timestamp=1442912586483, value=CS
Sname:                                  timestamp=1442912495442, value=LiYing
course:math                             timestamp=1442912802499, value=80
4 row(s) in 0.0120 seconds
```

然后，使用deleteall命令删除student表中的95001行的全部数据，命令如下：

```
hbase> deleteall 'student','95001'
```



5.3.5 删除表

删除表需要分两步操作，第一步先让该表不可用，第二步删除表。比如，要删除student表，可以使用如下命令：

```
hbase> disable 'student'  
hbase> drop 'student'
```



5.3.6 查询历史数据

为了查询历史数据，这里创建一个teacher表，首先，在创建表的时候，需要指定保存的版本数（假设指定为5），命令如下：

```
hbase> create 'teacher',{NAME=>'username',VERSIONS=>5}
```

然后，插入数据，并更新数据，使其产生历史版本数据，需要注意的是，这里插入数据和更新数据都是使用put命令，具体如下：

```
hbase> put 'teacher','91001','username','Mary'  
hbase> put 'teacher','91001','username','Mary1'  
hbase> put 'teacher','91001','username','Mary2'  
hbase> put 'teacher','91001','username','Mary3'  
hbase> put 'teacher','91001','username','Mary4'  
hbase> put 'teacher','91001','username','Mary5'
```



5.3.6 查询历史数据

查询时，默认情况下回显示当前最新版本的数据，如果要查询历史数据，需要指定查询的历史版本数，由于上面设置了保存版本数为5，所以，在查询时制定的历史版本数的有效取值为1到5，具体命令如下：

```
hbase(main):020:0> get 'teacher','91001',{COLUMN=>'username',VERSIONS=>5}
COLUMN          CELL
username:       timestamp=1469451374420, value=Mary5
username:       timestamp=1469451369561, value=Mary4
username:       timestamp=1469451366448, value=Mary3
username:       timestamp=1469451363530, value=Mary2
username:       timestamp=1469451351102, value=Mary1
5 row(s) in 0.0290 seconds

hbase(main):021:0> get 'teacher','91001',{COLUMN=>'username',VERSIONS=>3}
COLUMN          CELL
username:       timestamp=1469451374420, value=Mary5
username:       timestamp=1469451369561, value=Mary4
username:       timestamp=1469451366448, value=Mary3
3 row(s) in 0.0310 seconds

hbase(main):022:0> █
```



5.3.7 退出HBase数据库

最后退出数据库操作，输入**exit**命令即可退出，命令如下：

```
hbase> exit
```

注意，这里退出HBase数据库是退出HBase Shell，而不是停止HBase数据库后台运行，执行**exit**后，HBase仍然在后台运行，如果要停止HBase运行，需要使用如下命令：

```
$ bin/stop-hbase.sh
```



5.4 HBase编程实践

5.4.1 在Eclipse中创建项目

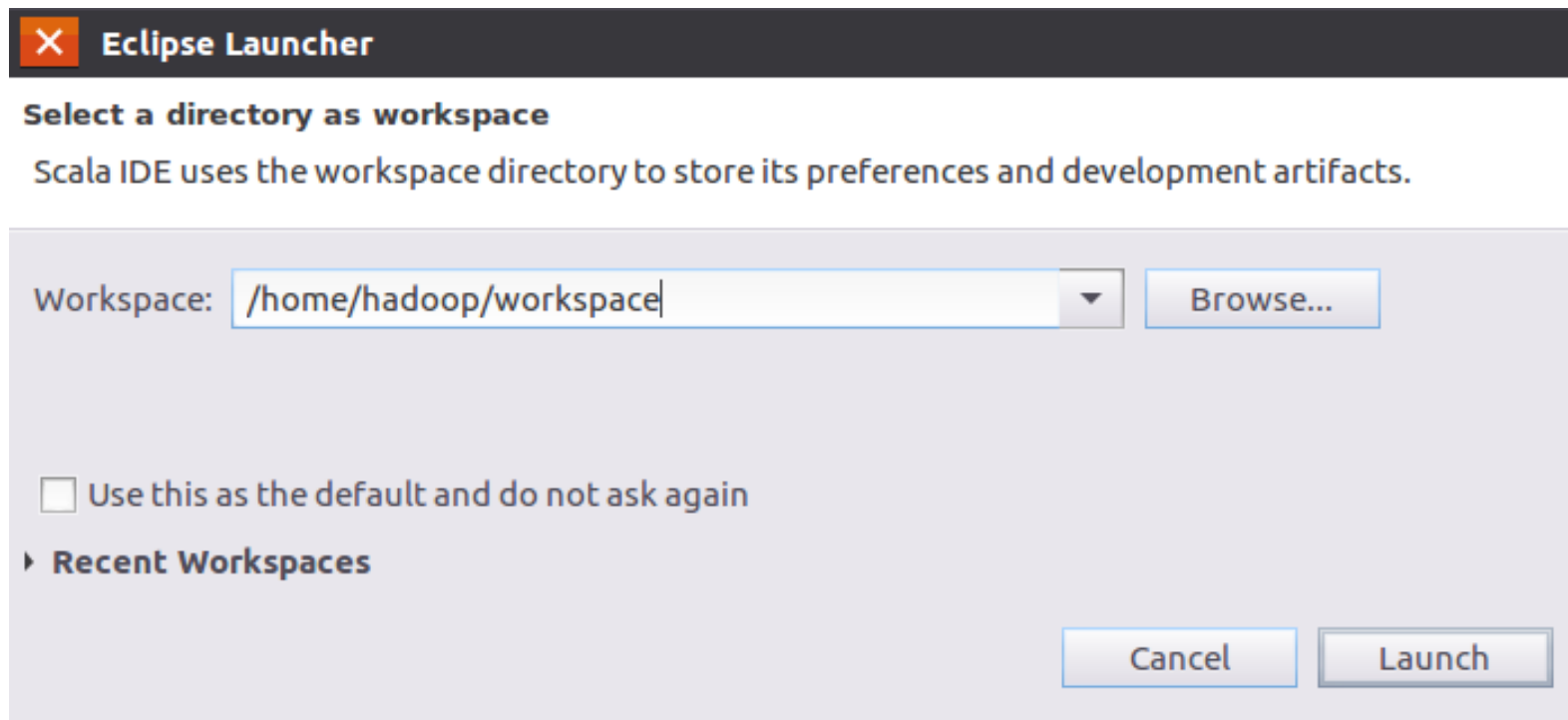
5.4.2 为项目添加需要用到的JAR包

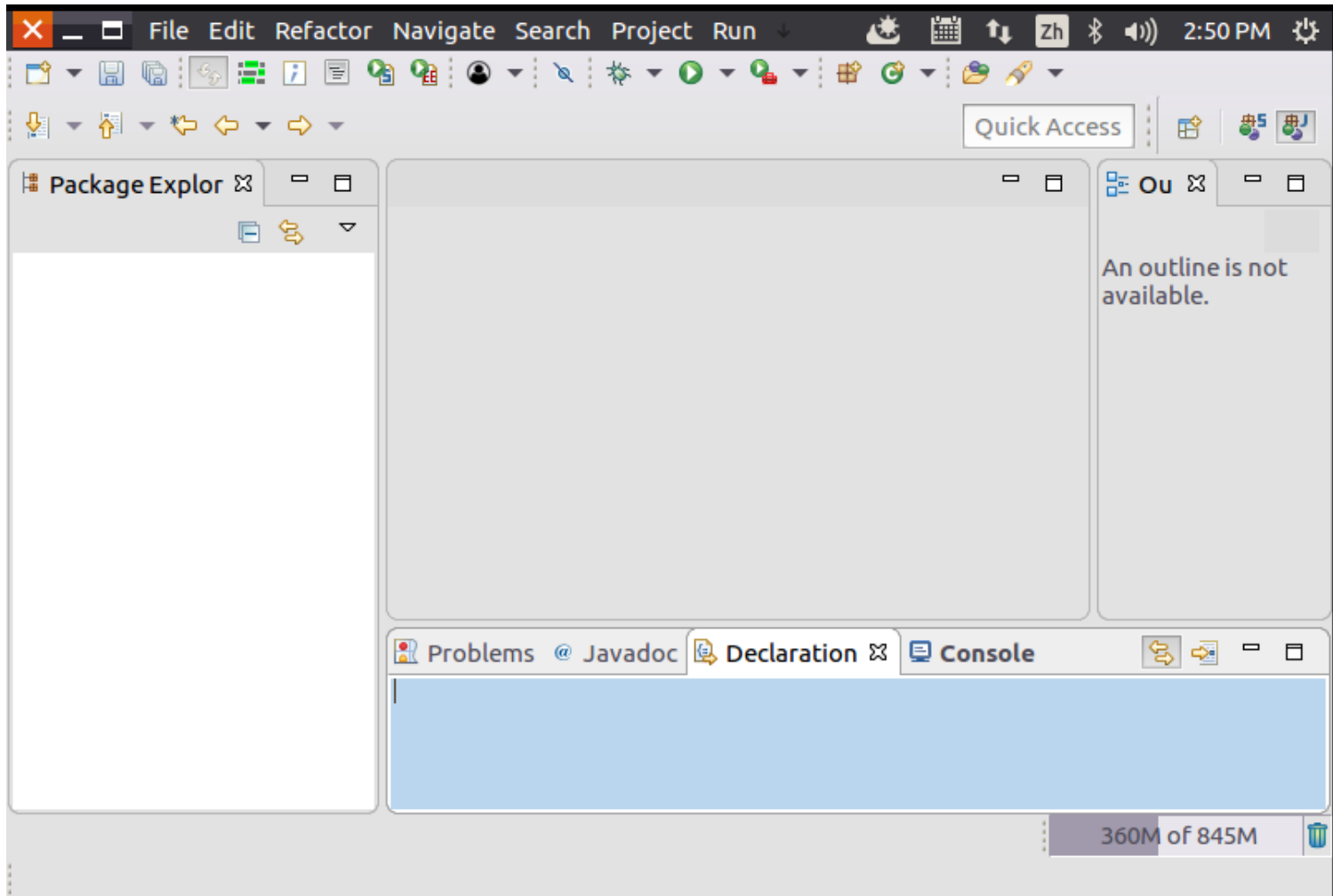
5.4.3 编写Java应用程序

5.4.4 编译运行程序



5.4.1 在Eclipse中创建项目







New Java Project

Create a Java Project
Create a Java project in the workspace or in an external location.

Project name:

Use default location

Location:

JRE

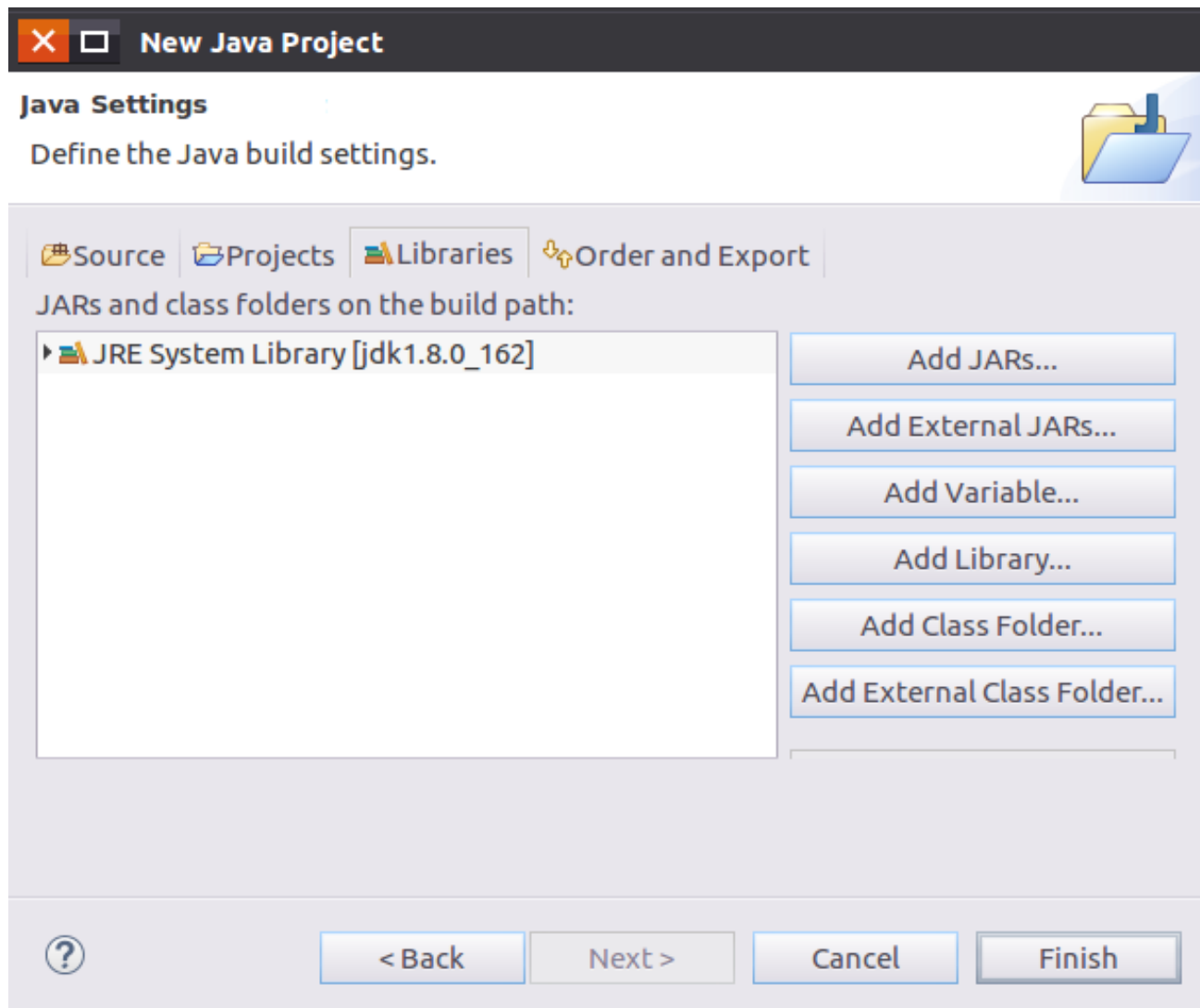
Use an execution environment JRE:

Use a project specific JRE:

Use default JRE (currently 'jdk1.8.0_162')

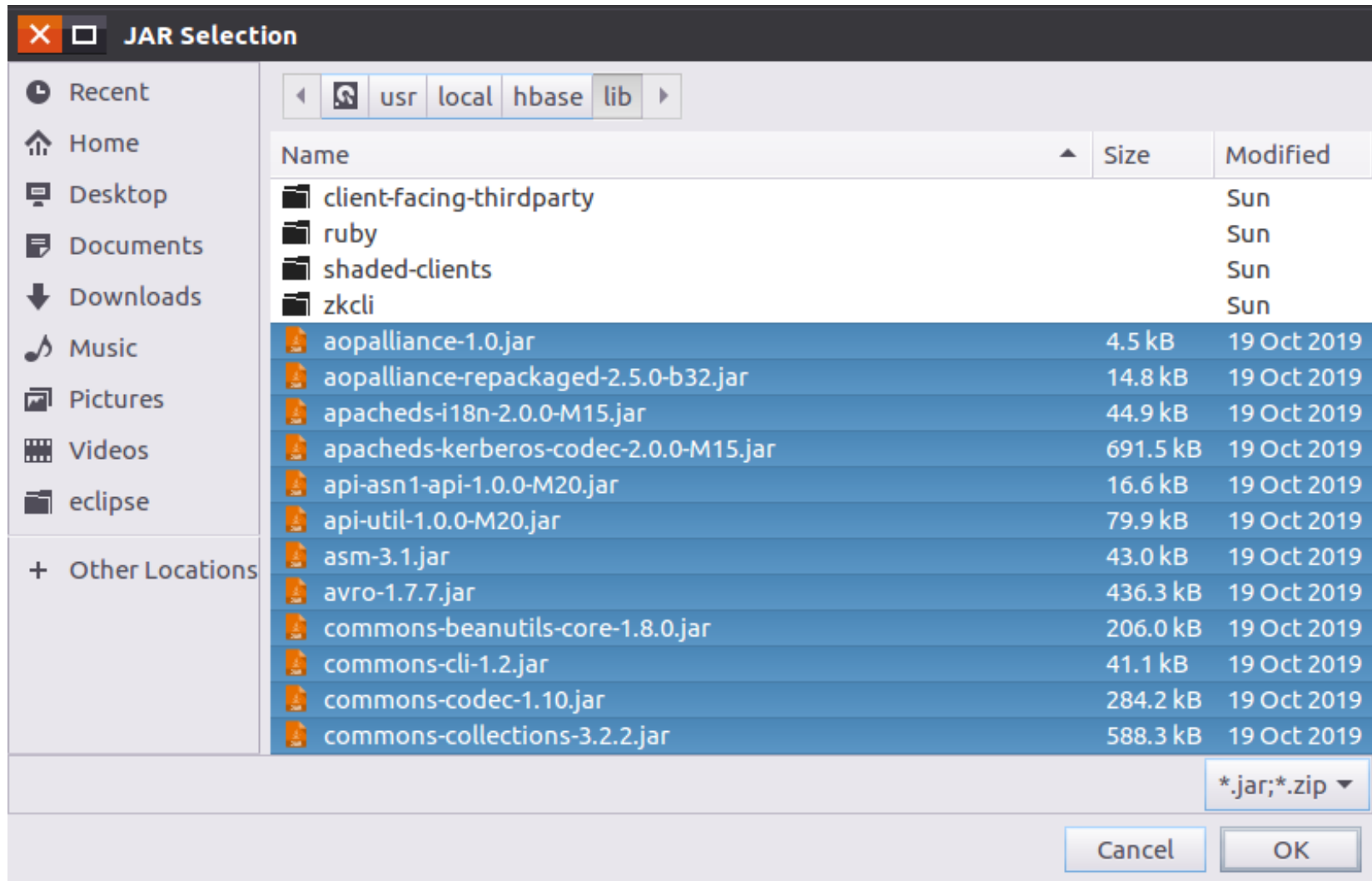


5.4.2为项目添加需要用到的JAR包



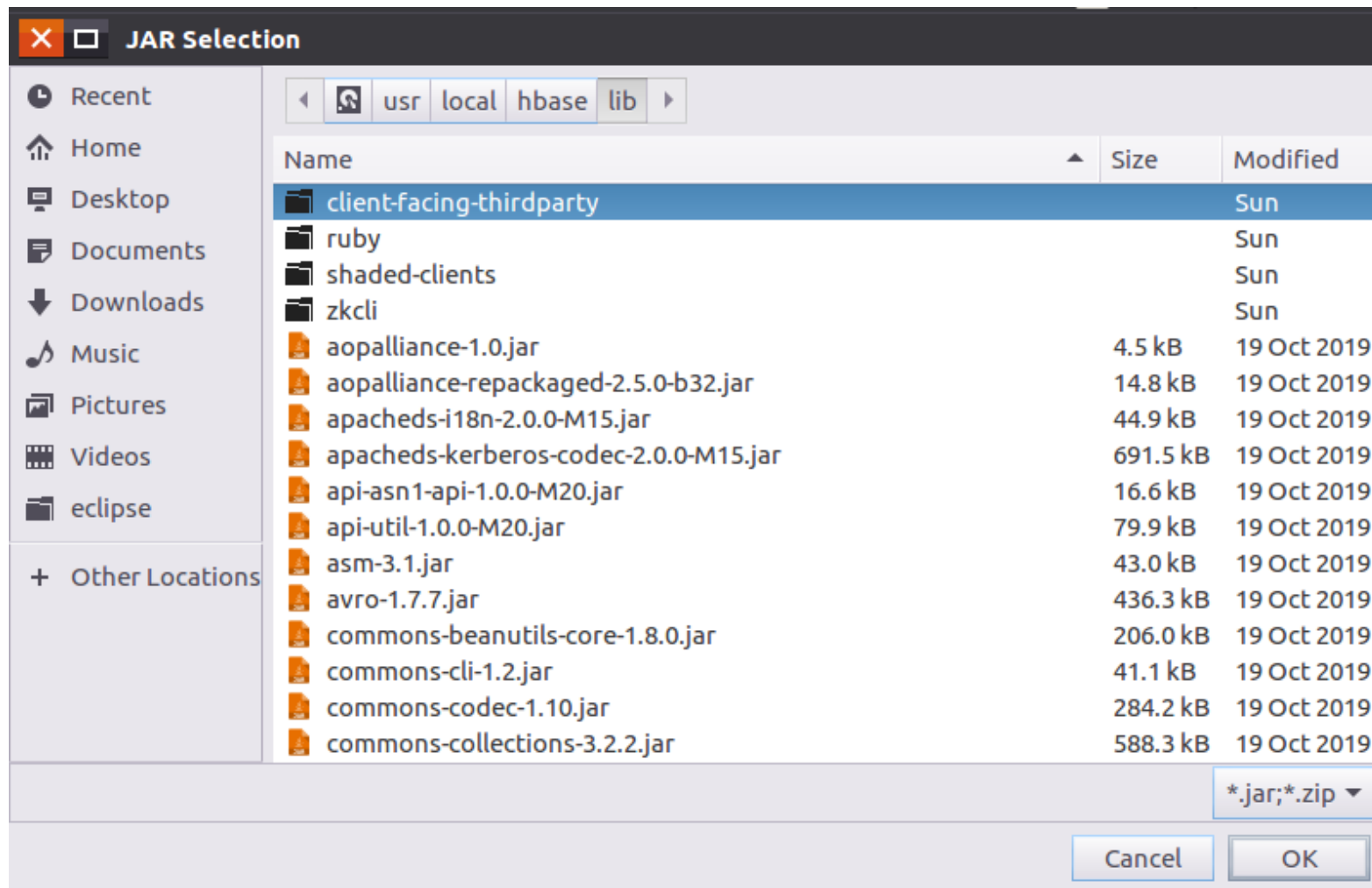


5.4.2为项目添加需要用到的JAR包



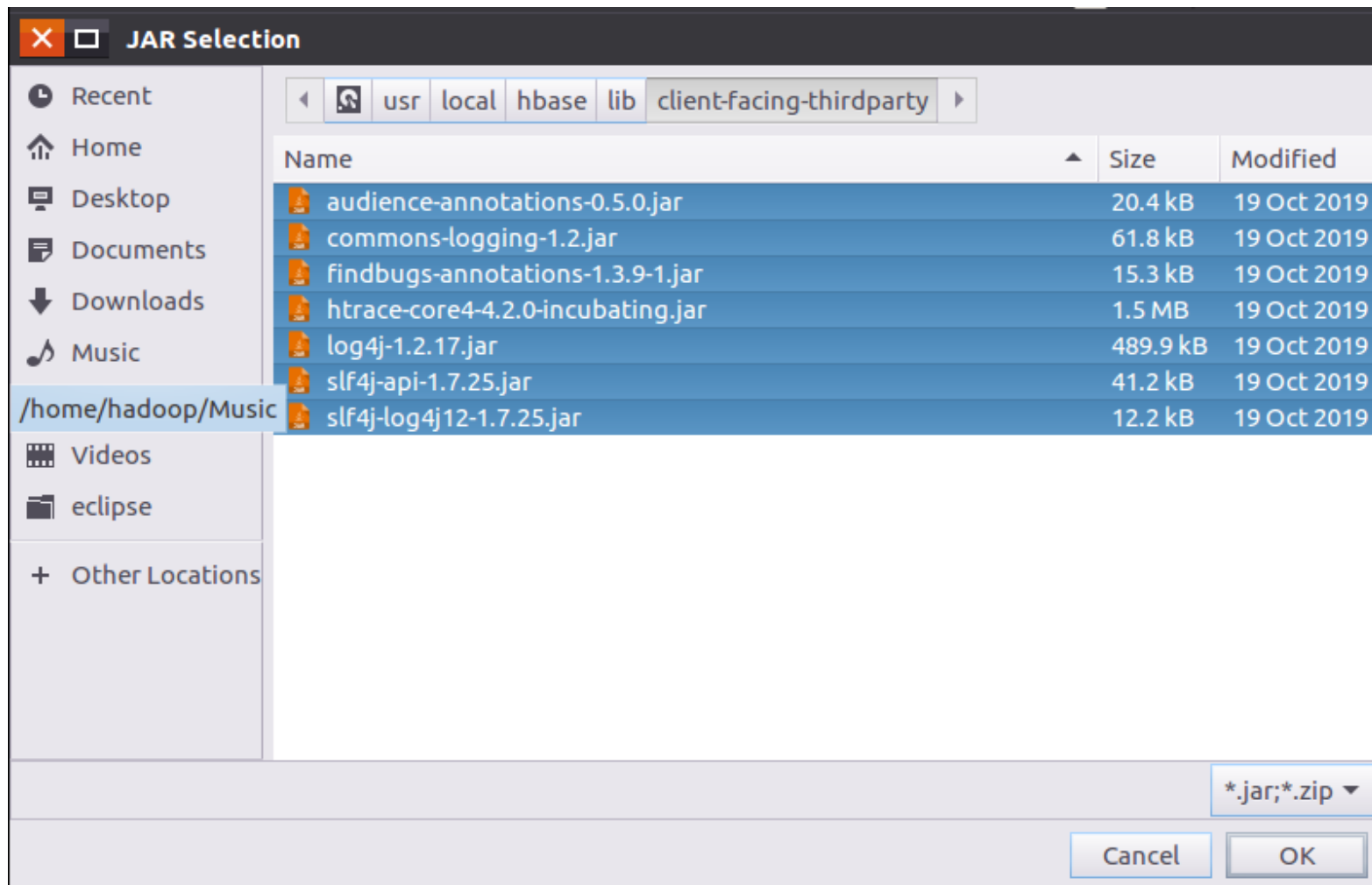


5.4.2为项目添加需要用到的JAR包



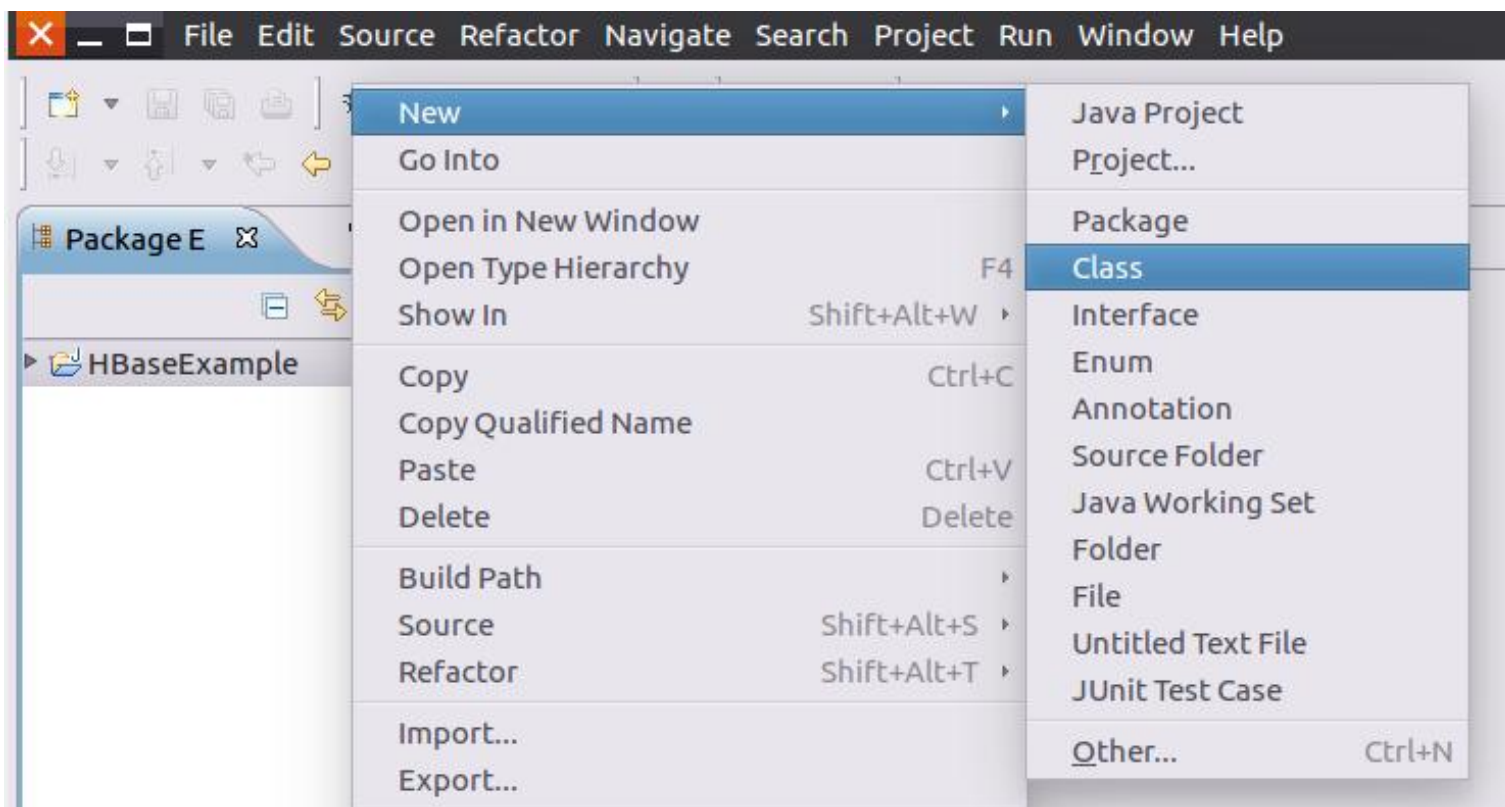


5.4.2为项目添加需要用到的JAR包





5.4.3 编写Java应用程序







5.4.3 编写Java应用程序

New Java Class

Java Class

 The use of the default package is discouraged. 

Source folder:

Package: (default)

Enclosing type:

Name:

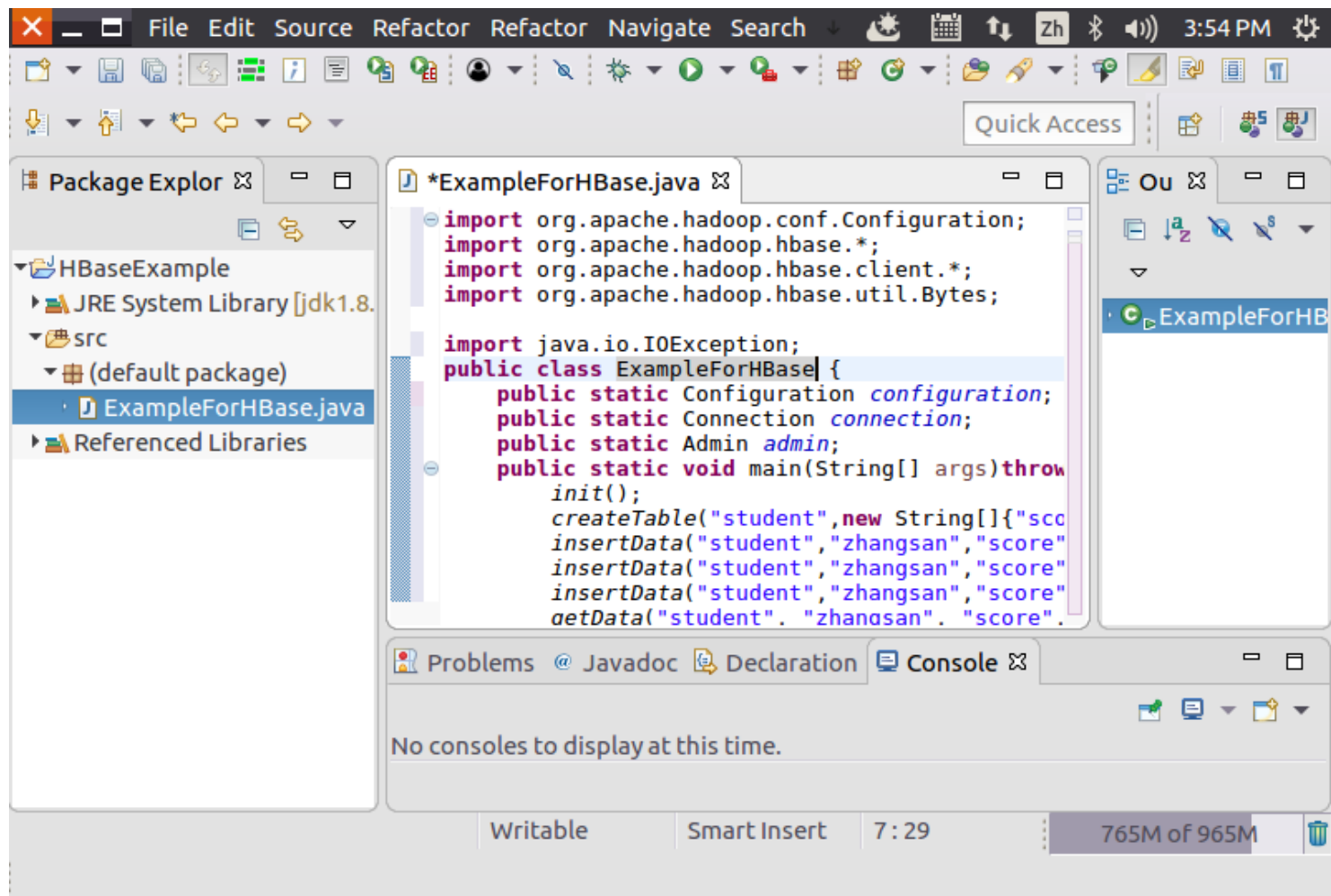
Modifiers: public package private protected
 abstract final static

Superclass:

Interfaces:



5.4.3 编写Java应用程序





5.4.3 编写Java应用程序

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.hbase.*;
import org.apache.hadoop.hbase.client.*;
import org.apache.hadoop.hbase.util.Bytes;

import java.io.IOException;
public class ExampleForHBase {
    public static Configuration configuration;
    public static Connection connection;
    public static Admin admin;
    public static void main(String[] args) throws IOException{
        init();
        createTable("student",new String[]{"score"});
        insertData("student","zhangsan","score","English","69");
        insertData("student","zhangsan","score","Math","86");
        insertData("student","zhangsan","score","Computer","77");
        getData("student", "zhangsan", "score","English");
        close();
    }
}
```



5.4.3 编写Java应用程序

```
public static void init(){
    configuration = HBaseConfiguration.create();
    configuration.set("hbase.rootdir", "hdfs://localhost:9000/hbase");
    try{
        connection = ConnectionFactory.createConnection(configuration);
        admin = connection.getAdmin();
    }catch (IOException e){
        e.printStackTrace();
    }
}
```



5.4.3 编写Java应用程序

```
public static void close(){
    try{
        if(admin != null){
            admin.close();
        }
        if(null != connection){
            connection.close();
        }
    }catch (IOException e){
        e.printStackTrace();
    }
}
```



5.4.3 编写Java应用程序

```
public static void createTable(String myTableName,String[] colFamily)
throws IOException {
    TableName tableName = TableName.valueOf(myTableName);
    if(admin.tableExists(tableName)){
        System.out.println("talbe is exists!");
    }else {
        TableDescriptorBuilder tableDescriptor =
TableDescriptorBuilder.newBuilder(tableName);
        for(String str:colFamily){
            ColumnFamilyDescriptor family =
ColumnFamilyDescriptorBuilder.newBuilder(Bytes.toBytes(str)).build();
            tableDescriptor.setColumnFamily(family);
        }
        admin.createTable(tableDescriptor.build());
    }
}
```



5.4.3 编写Java应用程序

```
public static void insertData(String tableName,String
rowKey,String colFamily,String col,String val) throws IOException {
    Table table =
connection.getTable(tableName.valueOf(tableName));
    Put put = new Put(rowKey.getBytes());
    put.addColumn(colFamily.getBytes(),col.getBytes(),
val.getBytes());
    table.put(put);
    table.close();
}
```

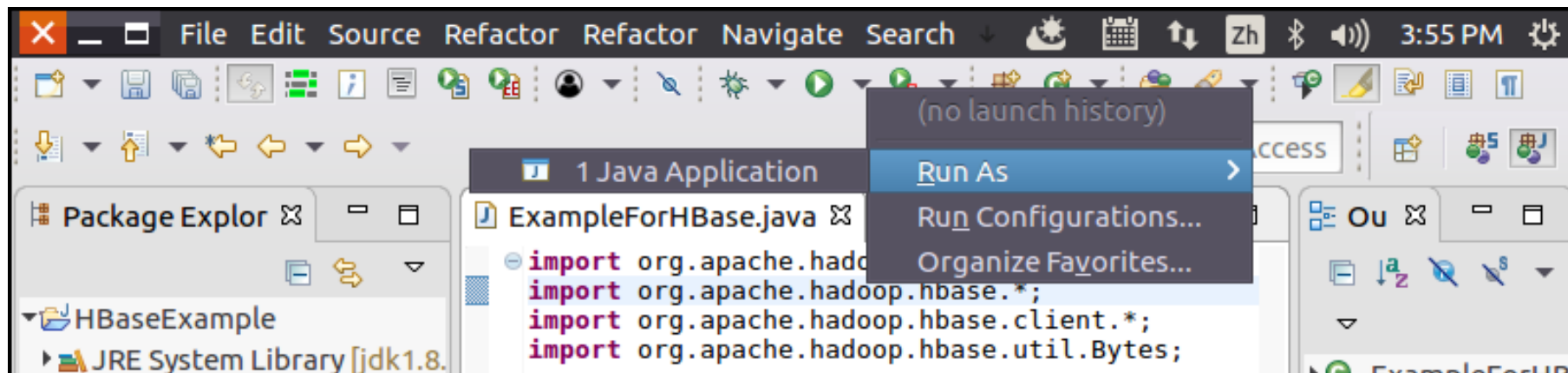


5.4.3 编写Java应用程序

```
public static void getData(String tableName,String rowKey,String
colFamily, String col)throws IOException{
    Table table = connection.getTable(TableName.valueOf(tableName));
    Get get = new Get(rowKey.getBytes());
    get.addColumn(colFamily.getBytes(),col.getBytes());
    Result result = table.get(get);
    System.out.println(new
String(result.getValue(colFamily.getBytes(),col==null?null:col.getBytes())));
    table.close();
}
}
```




5.4.4 编译运行程序





5.4.4 编译运行程序

The screenshot shows an IDE window with the following components:

- Package Explorer:** Shows the project structure with 'ExampleForHBase.java' selected.
- Code Editor:** Displays the source code for 'ExampleForHBase.java' with the following content:

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.hbase.*;
import org.apache.hadoop.hbase.client.*;
import org.apache.hadoop.hbase.util.Bytes;

import java.io.IOException;
public class ExampleForHBase {
    public static Configuration configuration;
```
- Console:** Shows the execution output for 'ExampleForHBase [Java Application]'. The output includes several log messages from 'ReadOnlyZKClient' and a final message from 'client.HBaseAdmin' indicating a successful connection to HBase. A red arrow points to the line: `[main] client.HBaseAdmin (HBaseAdmin.java:69)`.



5.4.4 编译运行程序

现在可以在Linux的终端中启动HBase Shell，来查看生成的表，启动HBase Shell的命令如下：

```
$ cd /usr/local/hbase  
$ ./bin/hbase shell
```

进入HBase Shell以后，可以使用list命令查看HBase数据库中是否存在名称为student的表。

```
hbase> list
```

```
hbase(main):010:0> list  
TABLE  
student  
1 row(s)  
Took 0.0126 seconds  
=> ["student"]
```



5.4.4 编译运行程序

再在HBase Shell交互式环境中，使用如下命令查看student表中的数据：

```
hbase(main):011:0> scan 'student'  
ROW          COLUMN+CELL  
zhangsan    column=score:Computer, timestamp=1579938991418, value=77  
zhangsan    column=score:English, timestamp=1579938991405, value=69  
zhangsan    column=score:Math, timestamp=1579938991414, value=86  
1 row(s)  
Took 0.0382 seconds
```



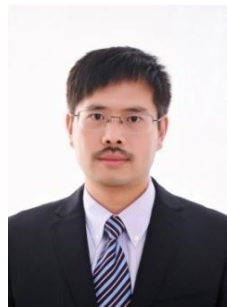
5.5 本章小结

HBase属于列族数据库，是NoSQL数据库的一种，它是Hadoop生态系统中的重要一员，借助于Hadoop的力量，**HBase**获得很好的发展空间，得到了大量的应用。

本章首先介绍了**HBase**的安装方法，包括下载安装文件、配置环境变量和添加用户权限等。然后，详细介绍了**HBase**的两种不同模式的配置方法，包括单机模式的配置和伪分布式模式的配置。在实际应用中，需要经常使用Shell命令操作**HBase**数据库，因此，本章详细介绍了一些常用的**HBase Shell**命令的使用方法，包括创建表、添加数据、查看数据、删除数据等。最后，本章详细介绍了**HBase**的Java应用程序开发方法。



附录A：主讲教师林子雨简介



主讲教师：林子雨

单位：厦门大学计算机科学系

E-mail: ziyulin@xmu.edu.cn

个人网页: <http://dblab.xmu.edu.cn/post/linziyu>

数据库实验室网站: <http://dblab.xmu.edu.cn>



扫一扫访问个人主页

林子雨，男，1978年出生，博士（毕业于北京大学），全国高校知名大数据教师，现为厦门大学计算机科学系副教授，曾任厦门大学信息科学与技术学院院长助理、晋江市发展和改革局副局长。中国计算机学会数据库专业委员会委员，中国计算机学会信息系统专业委员会委员。国内高校首个“数字教师”提出者和建设者，厦门大学数据库实验室负责人，厦门大学云计算与大数据研究中心主要建设者和骨干成员，2013年度、2017年度和2020年度厦门大学教学类奖教金获得者，荣获2019年福建省精品在线开放课程、2018年厦门大学高等教育成果特等奖、2018年福建省高等教育教学成果二等奖、2018年国家精品在线开放课程。主要研究方向为数据库、数据仓库、数据挖掘、大数据、云计算和物联网，并以第一作者身份在《软件学报》《计算机学报》和《计算机研究与发展》等国家重点期刊以及国际学术会议上发表多篇学术论文。作为项目负责人主持的科研项目包括1项国家自然科学基金青年基金项目(No.61303004)、1项福建省自然科学基金青年基金项目(No.2013J05099)和1项中央高校基本科研业务费项目(No.2011121049)，主持的教改课题包括1项2016年福建省教改课题和1项2016年教育部产学协作育人项目，同时，作为课题负责人完成了国家发改委城市信息化重大课题、国家物联网重大应用示范工程区域试点泉州市工作方案、2015泉州市互联网经济调研等课题。中国高校首个“数字教师”提出者和建设者，2009年至今，“数字教师”大平台累计向网络免费发布超过1000万字高价值的研究和教学资料，累计网络访问量超过1000万次。打造了中国高校大数据教学知名品牌，编著出版了中国高校第一本系统介绍大数据知识的专业教材《大数据技术原理与应用》，并成为京东、当当网等网店畅销书籍；建设了国内高校首个大数据课程公共服务平台，为教师教学和学生学习大数据课程提供全方位、一站式服务，年访问量超过200万次，累计访问量超过1000万次。



附录C：林子雨大数据系列教材



林子雨大数据系列教材

用于导论课、专业课、实训课、公共课

了解全部教材信息：<http://dbllab.xmu.edu.cn/post/bigdatabook/>



附录D：《大数据导论（通识课版）》教材

开设全校公共选修课的优质教材



本课程旨在实现以下几个培养目标：

- 引导学生步入大数据时代，积极投身大数据的变革浪潮之中
- 了解大数据概念，培养大数据思维，养成数据安全意识
- 认识大数据伦理，努力使自己的行为符合大数据伦理规范要求
- 熟悉大数据应用，探寻大数据与自己专业的应用结合点
- 激发学生基于大数据的创新创业热情

高等教育出版社 ISBN:978-7-04-053577-8 定价：32元

教材官网：<http://dbl原因.xmu.edu.cn/post/bigdataintroduction/>



附录E：《大数据导论》教材

- 林子雨 编著 《大数据导论》
- 人民邮电出版社，2020年9月第1版
- ISBN:978-7-115-54446-9 定价：49.80元

教材官网：<http://dbl原因.xmu.edu.cn/post/bigdata-introduction/>



开设大数据专业导论课的优质教材



扫一扫访问教材官网



附录F：《大数据技术原理与应用》教材

《大数据技术原理与应用——概念、存储、处理、分析与应用（第2版）》，由厦门大学计算机科学系林子雨博士编著，是国内高校第一本系统介绍大数据知识的专业教材。人民邮电出版社 ISBN:978-7-115-44330-4 定价：49.80元

全书共有15章，系统地论述了大数据的基本概念、大数据处理架构Hadoop、分布式文件系统HDFS、分布式数据库HBase、NoSQL数据库、云数据库、分布式并行编程模型MapReduce、Spark、流计算、图计算、数据可视化以及大数据在互联网、生物医学和物流等各个领域的应用。在Hadoop、HDFS、HBase和MapReduce等重要章节，安排了入门级的实践操作，让读者更好地学习和掌握大数据关键技术。

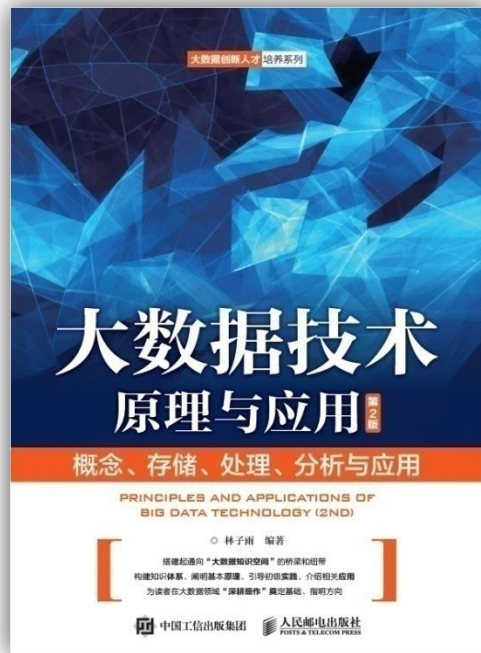
本书可以作为高等院校计算机专业、信息管理等相关专业的大数据课程教材，也可供相关技术人员参考、学习、培训之用。

欢迎访问《大数据技术原理与应用——概念、存储、处理、分析与应用》教材官方网站：

<http://dbllab.xmu.edu.cn/post/bigdata>



扫一扫访问教材官网





附录G：《大数据基础编程、实验和案例教程（第2版）》

本书是与《大数据技术原理与应用（第3版）》教材配套的唯一指定实验指导书

大数据教材



1+1黄金组合
厦门大学林子雨编著

配套实验指导书



- 步步引导，循序渐进，详尽的安装指南为顺利搭建大数据实验环境铺平道路
- 深入浅出，去粗取精，丰富的代码实例帮助快速掌握大数据基础编程方法
- 精心设计，巧妙融合，八套大数据实验题目促进理论与编程知识的消化和吸收
- 结合理论，联系实际，大数据课程综合实验案例精彩呈现大数据分析全流程

林子雨编著《大数据基础编程、实验和案例教程（第2版）》

清华大学出版社 ISBN:978-7-302-55977-1 定价：69元 2020年10月第2版

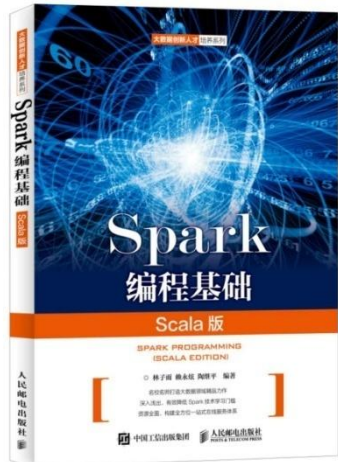


附录H: 《Spark编程基础 (Scala版)》

《Spark编程基础 (Scala版)》

厦门大学 林子雨, 赖永炫, 陶继平 编著

披荆斩棘, 在大数据丛林中开辟学习捷径
填沟削坎, 为快速学习Spark技术铺平道路
深入浅出, 有效降低Spark技术学习门槛
资源全面, 构建全方位一站式在线服务体系



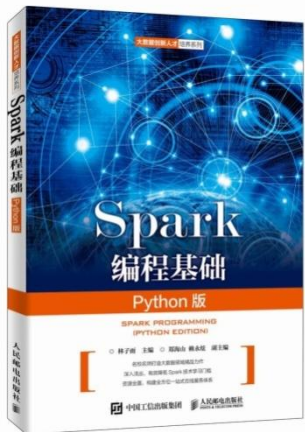
人民邮电出版社出版发行, ISBN:978-7-115-48816-9
教材官网: <http://dmlab.xmu.edu.cn/post/spark/>

本书以Scala作为开发Spark应用程序的编程语言, 系统介绍了Spark编程的基础知识。全书共8章, 内容包括大数据技术概述、Scala语言基础、Spark的设计与运行原理、Spark环境搭建和使用方法、RDD编程、Spark SQL、Spark Streaming、Spark MLlib等。本书每个章节都安排了入门级的编程实践操作, 以便读者更好地学习和掌握Spark编程方法。本书官网免费提供了全套的在线教学资源, 包括讲义PPT、习题、源代码、软件、数据集、授课视频、上机实验指南等。



附录I: 《Spark编程基础 (Python版)》

《Spark编程基础 (Python版)》



厦门大学 林子雨, 郑海山, 赖永炫 编著

披荆斩棘, 在大数据丛林中开辟学习捷径
填沟削坎, 为快速学习Spark技术铺平道路
深入浅出, 有效降低Spark技术学习门槛
资源全面, 构建全方位一站式在线服务体系

人民邮电出版社出版发行, ISBN:978-7-115-52439-3

教材官网: <http://dblab.xmu.edu.cn/post/spark-python/>



本书以Python作为开发Spark应用程序的编程语言, 系统介绍了Spark编程的基础知识。全书共8章, 内容包括大数据技术概述、Spark的设计与运行原理、Spark环境搭建和使用方法、RDD编程、Spark SQL、Spark Streaming、Structured Streaming、Spark MLlib等。本书每个章节都安排了入门级的编程实践操作, 以便读者更好地学习和掌握Spark编程方法。本书官网免费提供了全套的在线教学资源, 包括讲义PPT、习题、源代码、软件、数据集、上机实验指南等。



附录J：高校大数据课程公共服务平台



高校大数据课程

公 共 服 务 平 台

<http://dblab.xmu.edu.cn/post/bigdata-teaching-platform/>



扫一扫访问平台主页



扫一扫观看3分钟FLASH动画宣传片



附录K：高校大数据实训课程系列案例教材

为了更好地满足高校开设大数据实训课程的教材需求，厦门大学数据库实验室林子雨老师团队联合企业共同开发了《高校大数据实训课程系列案例》，目前已经完成开发的系列案例包括：

《电影推荐系统》（已经于2019年5月出版）

《电信用户行为分析》（已经于2019年5月出版）

《实时日志流处理分析》

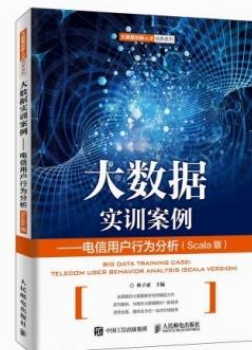
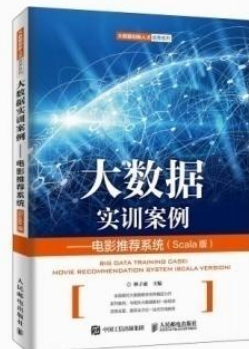
《微博用户情感分析》

《互联网广告预测分析》

《网站日志处理分析》

系列案例教材将于2019年陆续出版发行，教材相关信息，敬请关注网页后续更新！

<http://dblab.xmu.edu.cn/post/shixunkecheng/>



扫一扫访问大数据实训课程系列案例教材主页

The background of the slide features a blue gradient with several faint, light-blue silhouettes of people. At the top, there are two groups of people standing and holding hands. On the right side, a person is shown in profile, looking towards the center. On the left side, two people are shown in profile, one appearing to be speaking or gesturing towards the other. The overall composition suggests a community or a group of people.

Thank You!

Department of Computer Science, Xiamen University, 2020