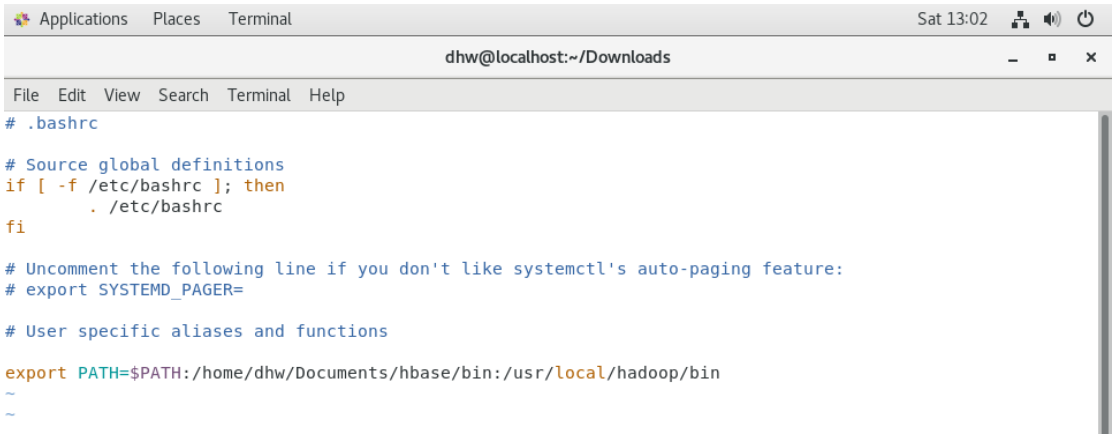
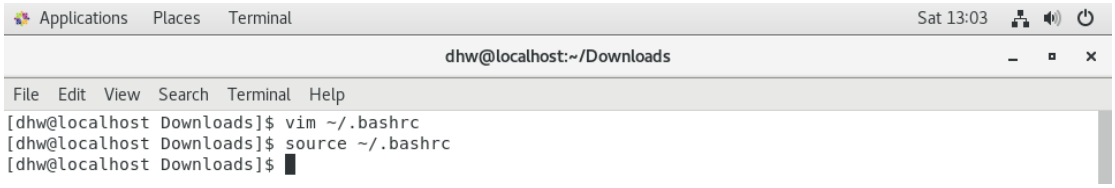
一、 准备工作

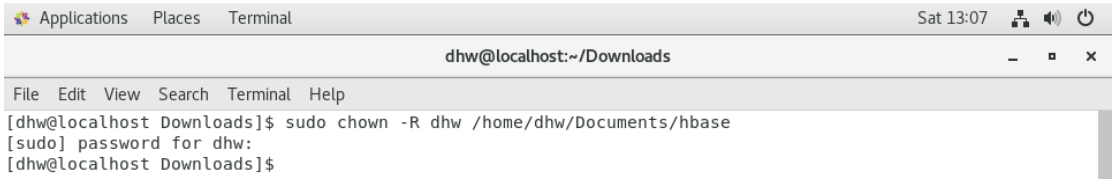
下载 HBase：http://archive.apache.org/dist/hbase/
解压；

配置环境变量（顺便配置 Hadopp 环境变量）：

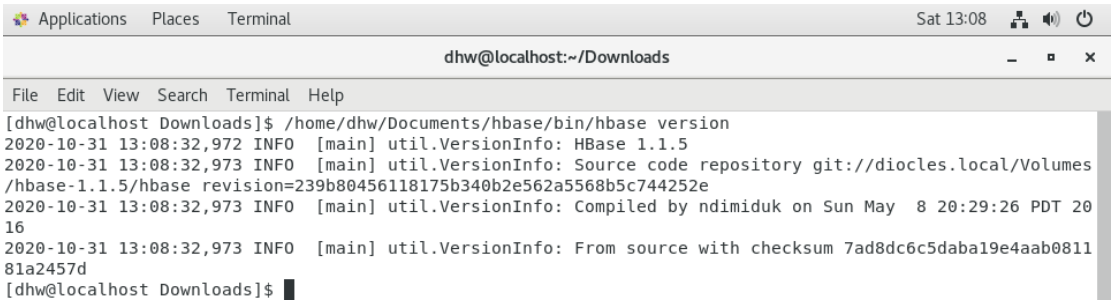




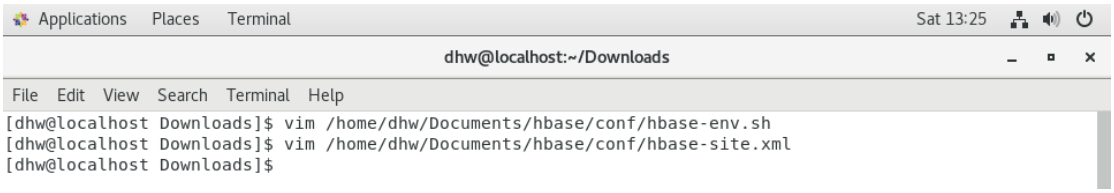
为当前登录 Linux 系统的 dhw 用户添加访问 HBase 目录的权限：



查看 Hbase 版本信息：



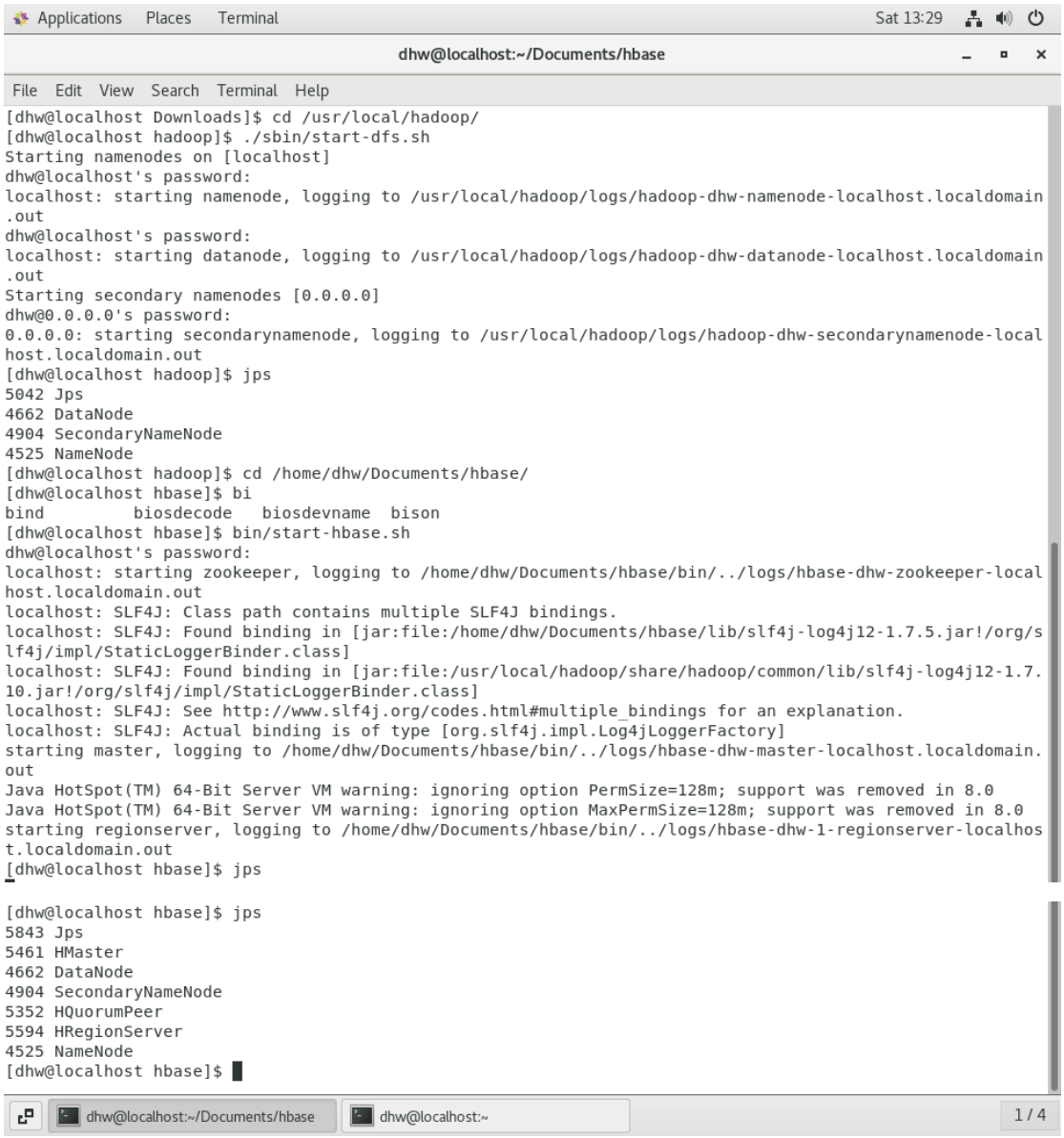
HBase 伪分布式配置：




```
# Seconds to sleep between slave commands.  Unset by default.  This
# can be useful in large clusters, where, e.g., slave rsyncs can
# otherwise arrive faster than the master can service them.
# export HBASE_SLAVE_SLEEP=0.1

export JAVA_HOME=/usr/local/jdk1.8
export HBASE_CLASSPATH=/home/dhw/Documents/hbase/conf
# Tell HBase whether it should manage it's own instance of Zookeeper or not.
export HBASE_MANAGES_ZK=true

# The default log rolling policy is RFA, where the log file is rolled as per the
# RFA appender. Please refer to the log4j.properties file to see more details on
# In case one needs to do log rolling on a date change, one should set the envir
```

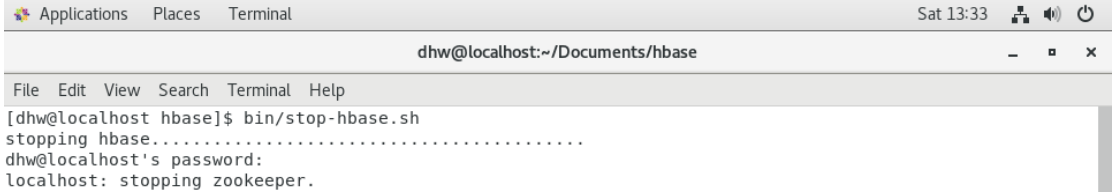
```
 * limitations under the License.
 */
-->
<configuration>
  <property>
    <name>hbase.rootdir</name>
    <value>hdfs://localhost:9000/hbase</value>
  </property>
  <property>
    <name>hbase.cluster.distributed</name>
    <value>true</value>
  </property>
</configuration>

~
~
~
```

启动：

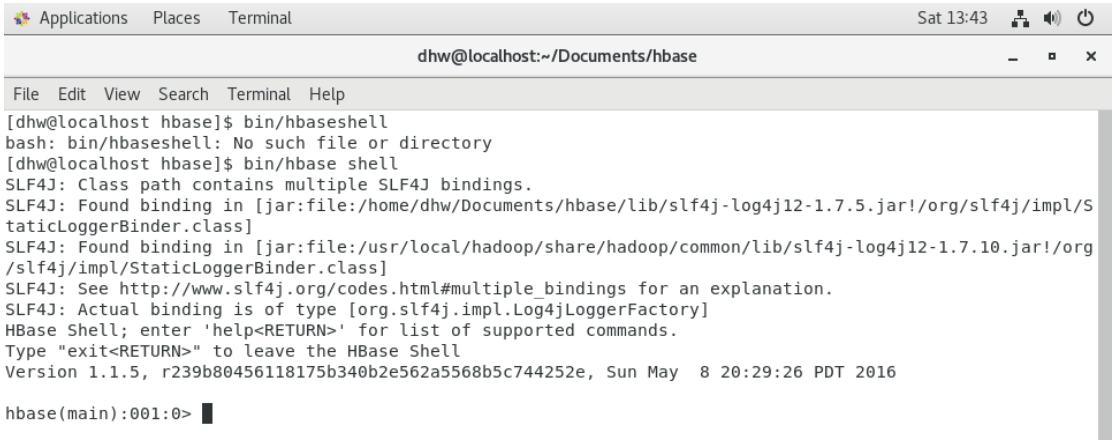


关闭 Hbase：

最后需要注意的是，启动关闭 Hadoop 和 HBase 的顺序一定是：启动 Hadoop→启动 HBase→关闭 HBase→关闭 Hadoop。



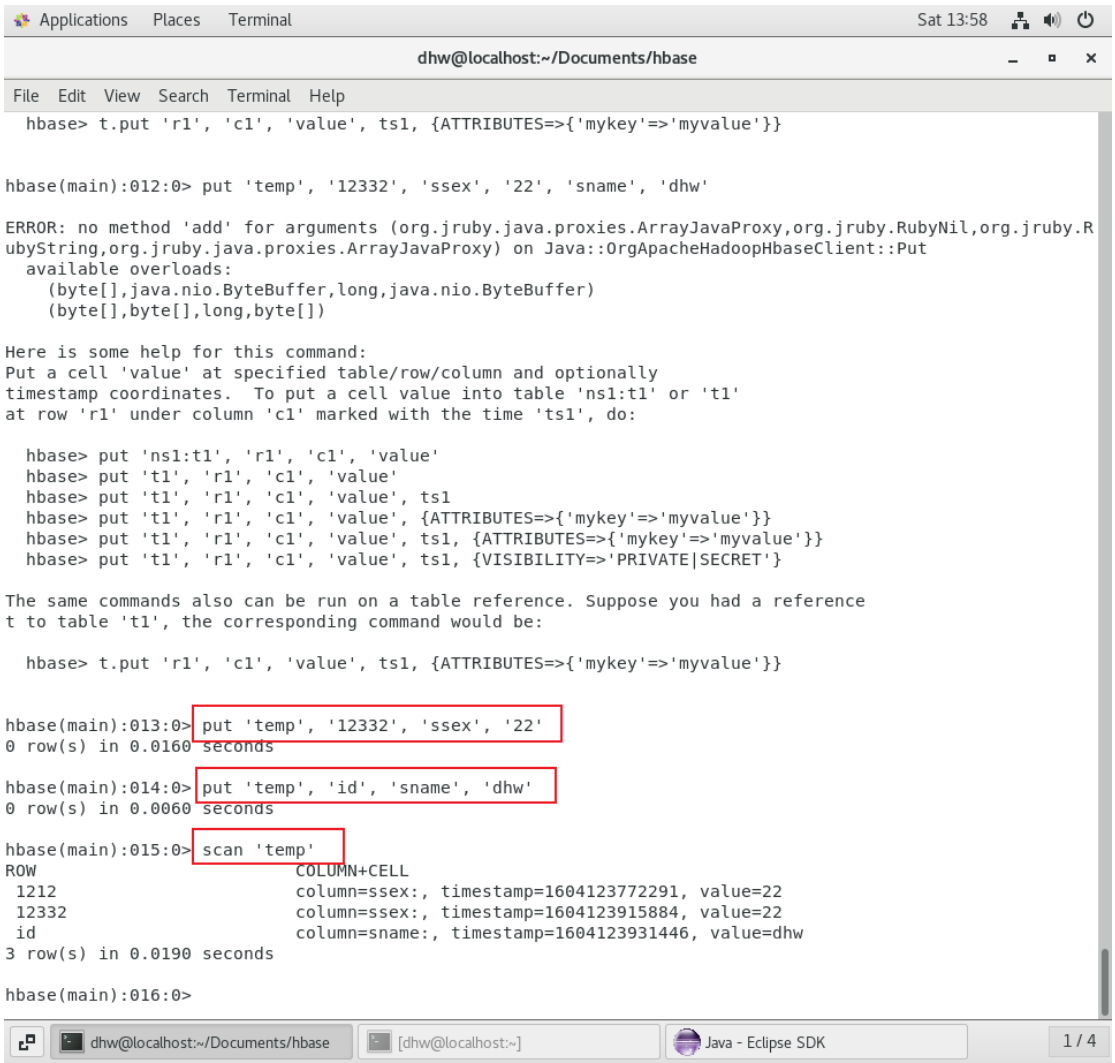
再次启动后进入 Shell 命令行：



创建表并查看:

```
hbase(main):001:0> create 'temp', 'sname', 'ssex', 'sage', 'course'
0 row(s) in 10.9620 seconds

=> Hbase::Table - temp
hbase(main):002:0> list
TABLE
temp
1 row(s) in 0.1420 seconds

=> ["temp"]
```

插入数据:



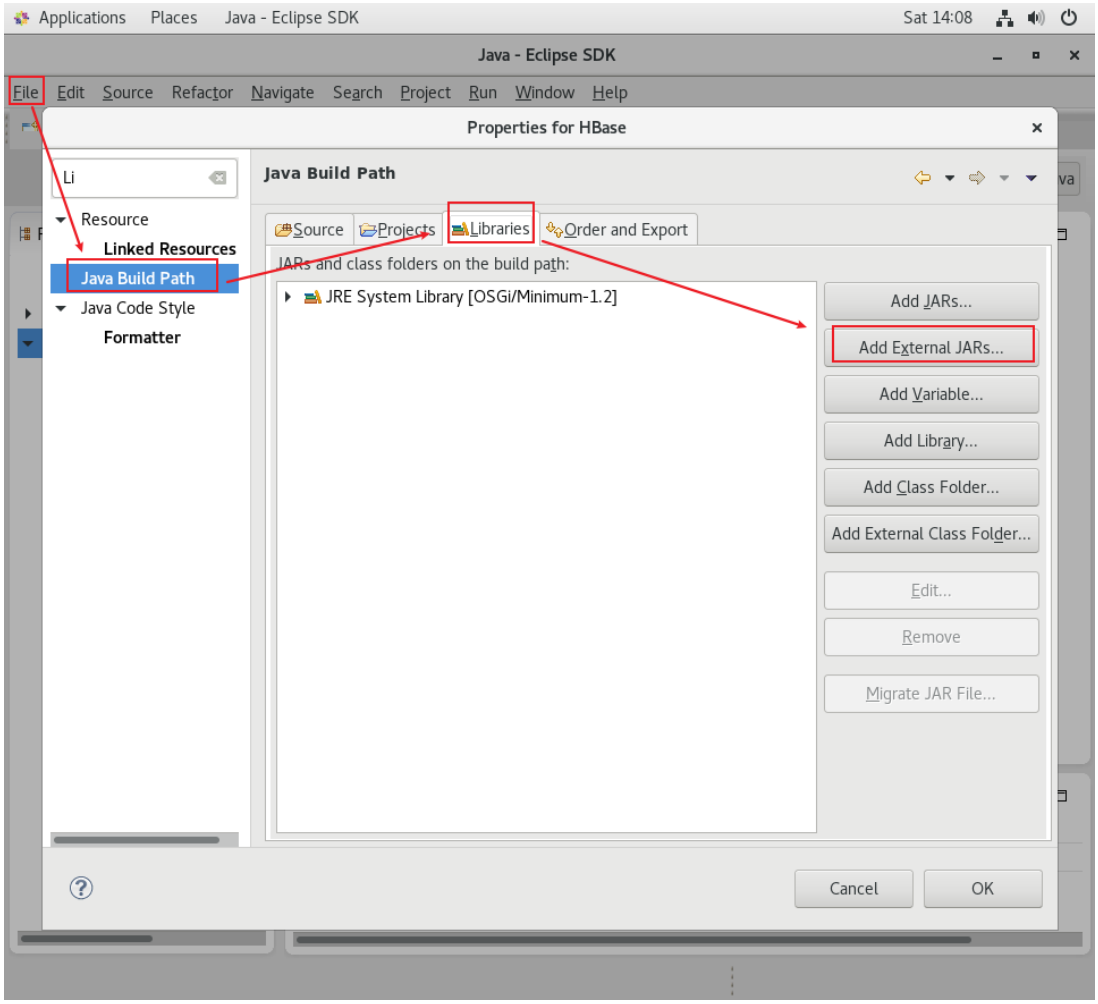
删除数据：

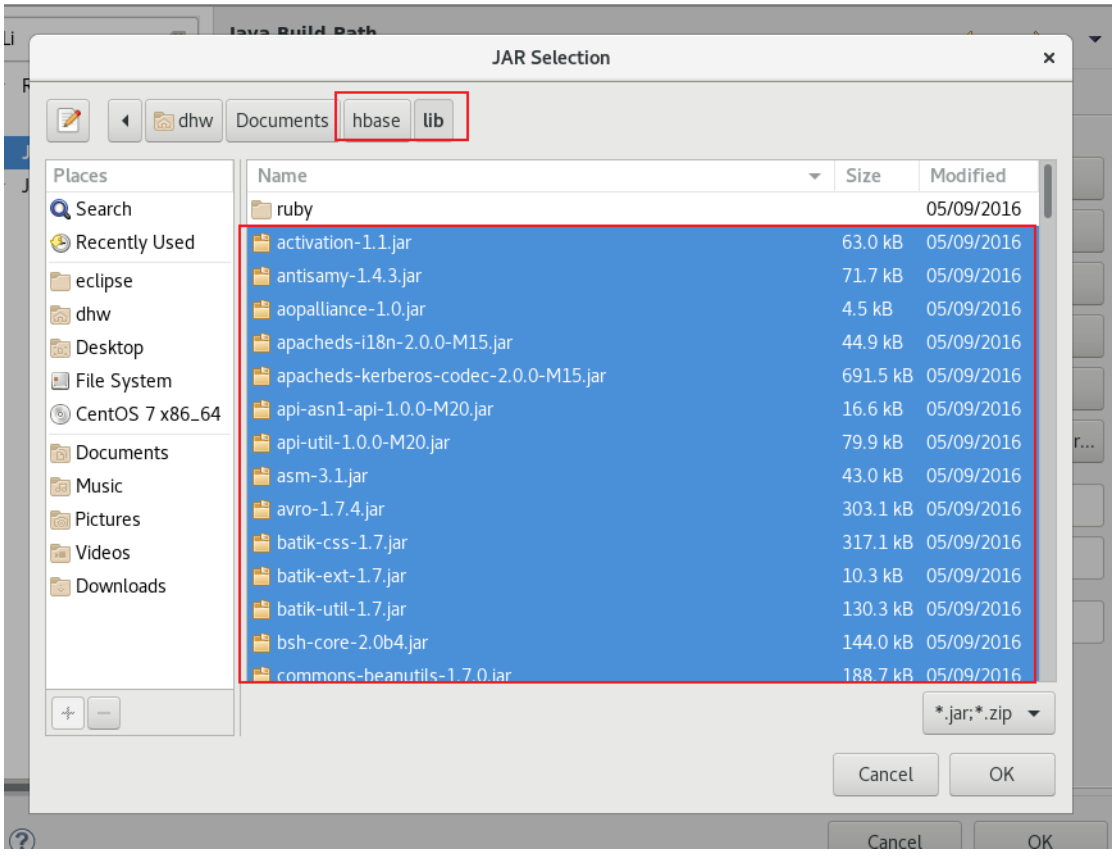
```
hbase(main):016:0> delete 'temp', 'id', 'sname'
0 row(s) in 0.0820 seconds

hbase(main):017:0> scan 'temp'
ROW                           COLUMN+CELL
 1212                         column=ssex:, timestamp=1604123772291, value=22
 12332                        column=ssex:, timestamp=1604123915884, value=22
2 row(s) in 0.0330 seconds

hbase(main):018:0> delete 'temp', '1212'

ERROR: wrong number of arguments (2 for 3)

Here is some help for this command:
Put a delete cell value at specified table/row/column and optionally
timestamp coordinates.  Deletes must match the deleted cell's
coordinates exactly.  When scanning, a delete cell suppresses older
versions. To delete a cell from  't1' at row 'r1' under column 'c1'
marked with the time 'ts1', do:

  hbase> delete 'ns1:t1', 'r1', 'c1', ts1
  hbase> delete 't1', 'r1', 'c1', ts1
  hbase> delete 't1', 'r1', 'c1', ts1, {VISIBILITY=>'PRIVATE|SECRET'}

The same command can also be run on a table reference. Suppose you had a reference
t to table 't1', the corresponding command would be:

  hbase> t.delete 'r1', 'c1',  ts1
  hbase> t.delete 'r1', 'c1',  ts1, {VISIBILITY=>'PRIVATE|SECRET'}


hbase(main):019:0> deleteall 'temp', '1212'
0 row(s) in 0.0190 seconds

hbase(main):020:0> scan 'temp'
ROW                           COLUMN+CELL
 12332                        column=ssex:, timestamp=1604123915884, value=22
1 row(s) in 0.0150 seconds

hbase(main):021:0>
```

删除表：首先设置不可用，最后删除

```
hbase(main):021:0> disable 'temp'
0 row(s) in 4.4060 seconds

hbase(main):022:0> drop 'temp'
0 row(s) in 2.2760 seconds

hbase(main):023:0> scan 'temp'
ROW                           COLUMN+CELL

ERROR: Unknown table temp!
```

设置 eclipse 编程环境：

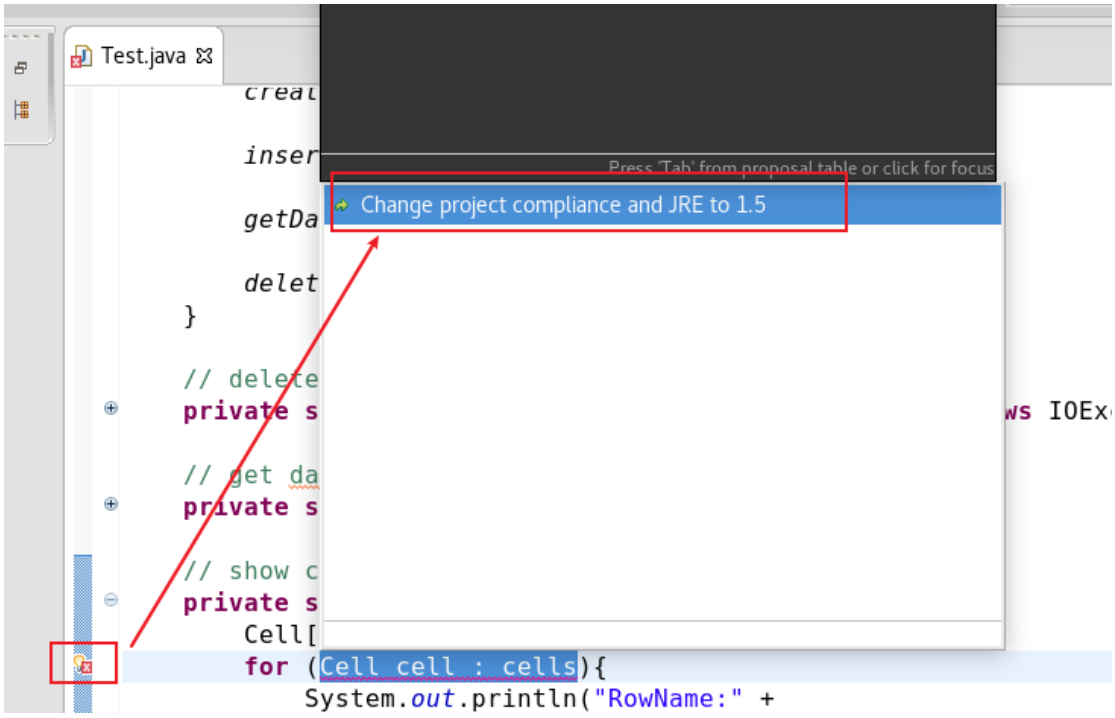

Applications
Places
Java - Eclipse SDK
Sat 14:08
Java - Eclipse SDK
File
Edit
Source
Refactor
Navigate
Search
Project
Run
Window
Help
Properties for HBase
Li
Resource
Linked Resources
Java Build Path
Java Code Style
Formatter
Java Build Path
Source
Projects
Libraries
Order and Export
JARs and class folders on the build path:
JRE System Library [OSGi/Minimum-1.2]
Add JARs...
Add External JARs...
Add Variable...
Add Library...
Add Class Folder...
Add External Class Folder...
Edit...
Remove
Migrate JAR File...
Cancel
OK


JAR Selection
dhw
Documents
hbase
lib
Places
Search
Recently Used
eclipse
dhw
Desktop
File System
CentOS 7 x86_64
Documents
Music
Pictures
Videos
Downloads
Name
Size
Modified
ruby
05/09/2016
activation-1.1.jar
63.0 kB
05/09/2016
antisamy-1.4.3.jar
71.7 kB
05/09/2016
aopalliance-1.0.jar
4.5 kB
05/09/2016
apacheds-i18n-2.0.0-M15.jar
44.9 kB
05/09/2016
apacheds-kerberos-codec-2.0.0-M15.jar
691.5 kB
05/09/2016
api-asn1-api-1.0.0-M20.jar
16.6 kB
05/09/2016
api-util-1.0.0-M20.jar
79.9 kB
05/09/2016
asm-3.1.jar
43.0 kB
05/09/2016
avro-1.7.4.jar
303.1 kB
05/09/2016
batik-css-1.7.jar
317.1 kB
05/09/2016
batik-ext-1.7.jar
10.3 kB
05/09/2016
batik-util-1.7.jar
130.3 kB
05/09/2016
bsh-core-2.0b4.jar
144.0 kB
05/09/2016
commons-beanutils-1.7.0.jar
188.7 kB
05/09/2016
.jar;.zip
Cancel
OK

编写代码进行测试：

其中出现错误：Syntax error, 'for each' statements are only available if source level is 1.5 or greater

解决方法：



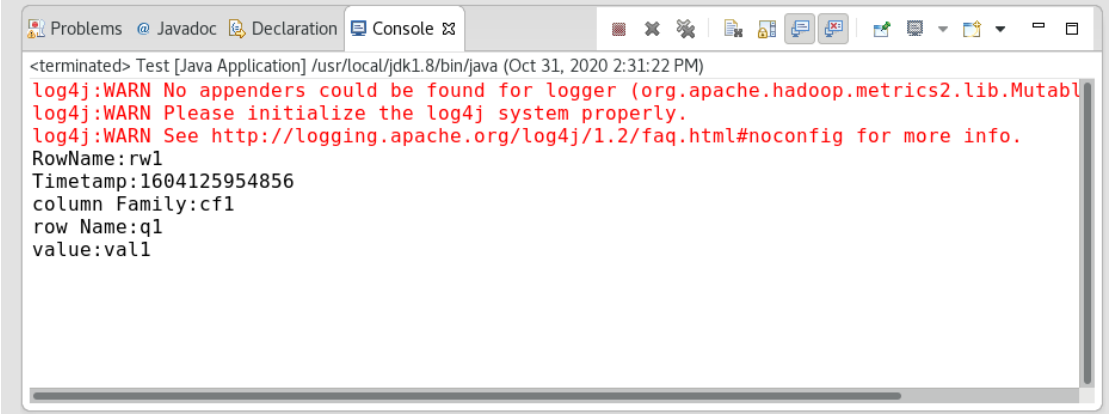
完整代码如下：

```
import java.io.IOException;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.hbase.Cell;
import org.apache.hadoop.hbase.CellUtil;
import org.apache.hadoop.hbase.HBaseConfiguration;
import org.apache.hadoop.hbase.HColumnDescriptor;
import org.apache.hadoop.hbase.HTableDescriptor;
import org.apache.hadoop.hbase.TableName;
import org.apache.hadoop.hbase.client.Admin;
import org.apache.hadoop.hbase.client.Connection;
import org.apache.hadoop.hbase.client.ConnectionFactory;
import org.apache.hadoop.hbase.client.Get;
import org.apache.hadoop.hbase.client.Put;
import org.apache.hadoop.hbase.client.Result;
import org.apache.hadoop.hbase.client.Table;



public class Test {

    public static Configuration configuration;
```

```
    public static Connection connection;
    public static Admin admin;

    public static void main(String[] args) throws IOException {


            createTable("t2",new String[]{"cf1", "cf2"});

            insertRow("t2", "rw1", "cf1", "q1", "val1");

            getData("t2", "rw1", "cf1", "q1");

            deleteTable("t2");
    }

    // delete table
    private static void deleteTable(String tableName) throws IOException {
        init();

        TableName tn = TableName.valueOf(tableName);
        if (admin.tableExists(tn)) {
            admin.disableTable(tn);
            admin.deleteTable(tn);
        }

        close();
    }

    // get datas
    private static void getData(String tableName,
                    String rowKey, String colFamily, String col) throws IOException{
        init();

        Table table = connection.getTable(TableName.valueOf(tableName));

        Get get = new Get(rowKey.getBytes());
        get.addColumn(colFamily.getBytes(),col.getBytes());

        Result result = table.get(get);
        showCell(result);

        table.close();
        close();
    }
```

```
    // show cells
    private static void showCell(Result result){
        Cell[] cells = result.rawCells();
        for (Cell cell : cells){
            System.out.println("RowName:" +
                                    new String(CellUtil.cloneRow(cell)) + " ");
            System.out.println("Timetamp:" + cell.getTimestamp() + " ");
            System.out.println("column Family:" +
                                    new String(CellUtil.cloneFamily(cell)) + " ");
            System.out.println("row Name:" +
                                    new String(CellUtil.cloneQualifier(cell)) + " ");
            System.out.println("value:" +
                                    new String(CellUtil.cloneValue(cell)) + " ");
        }
    }

    // put rows
    private static void insertRow(String tableName,
                    String rowKey, String colFamily,
                    String col,String val) throws IOException {
        init();

        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();
        close();
    }

    // create table
    private static void createTable(String myTableName,String[] colFamily)
                    throws IOException {
            init();

            TableName tableName = TableName.valueOf(myTableName);
        if(admin.tableExists(tableName)){
            System.out.println("talbe is exists!");
        }else {
            HTableDescriptor hTableDescriptor = new HTableDescriptor(tableName);
```

```
                for (String colstr : colFamily) {
                    HColumnDescriptor hColumnDescriptor =
                                        new HColumnDescriptor(colstr);
                    hTableDescriptor.addFamily(hColumnDescriptor);
                }
                admin.createTable(hTableDescriptor);
            }

            close();
        }

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

        // create connection
        private 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();
            }
        }

}
```



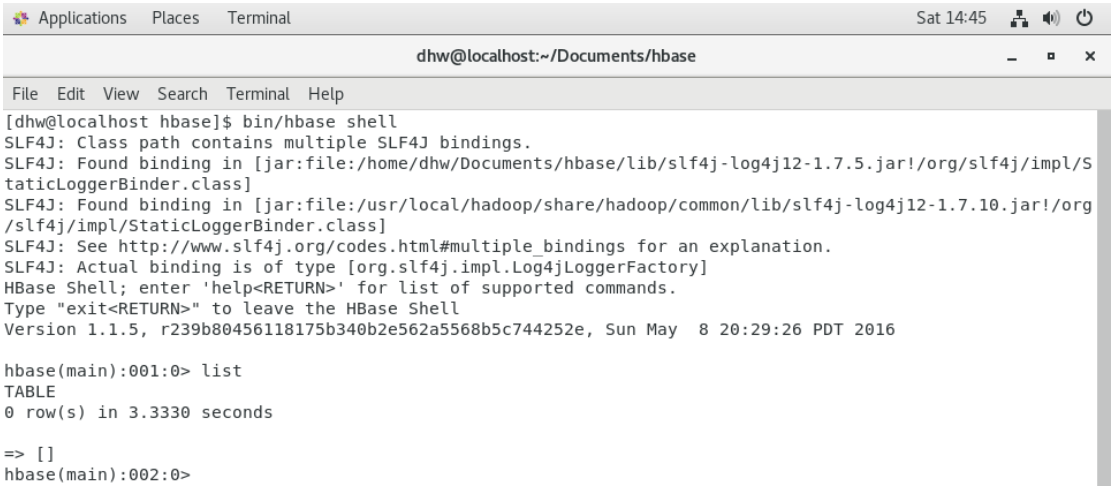
之后可以把该程序部署到 Hadoop 平台上运行，方法见作业 3。

【备忘录】Hadoop 安装目录：/usr/local/hadoop
HBase 安装目录：/home/dhw/Documents/hbase
Eclipse 安装目录：/home/dhw/Documents/eclipse

二、 实验

(一)

（1）列出表信息（如表名）



Eclipse 实现：（之前创建表删除则没有输出），其中 init()和 close()代码之后将不再列出，统一为以下形式。

```java
// create connection
private 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();
    }
}

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

Test.java t1.java

```java
import java.io.IOException;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.hbase.HBaseConfiguration;
import org.apache.hadoop.hbase.HTableDescriptor;
import org.apache.hadoop.hbase.client.Admin;
import org.apache.hadoop.hbase.client.Connection;
import org.apache.hadoop.hbase.client.ConnectionFactory;


public class t1 {

    public static Configuration configuration;
    public static Connection connection;
    public static Admin admin;

    public static void main(String[] args) throws IOException {
        init();

        HTableDescriptor[] hTableDescriptor = admin.listTables();
        for (HTableDescriptor h : hTableDescriptor) {
            System.out.println(h.getNameAsString());
        }

        close();
    }
```

（2）终端打印指定表的所有记录（按准备工作中的代码创建表 t2 再操作）

```
hbase(main):001:0> list
TABLE
0 row(s) in 3.3330 seconds

=> []
hbase(main):002:0> list
TABLE
t2
1 row(s) in 0.2980 seconds

=> ["t2"]
hbase(main):003:0> scan 't2'
ROW                           COLUMN+CELL
 rw1                          column=cf1:q1, timestamp=1604127293478, value=val1
1 row(s) in 1.1490 seconds

hbase(main):004:0>
```

Eclipse 实现

下述方法 showCell()之后将不再展示，统一为下述内容：

```
  // show cells
  private static void showCell(Result result){
      Cell[] cells = result.rawCells();
      for (Cell cell : cells){
          System.out.println("RowName:" +
                             new String(CellUtil.cloneRow(cell)) + " ");
          System.out.println("Timetamp:" + cell.getTimestamp() + " ");
          System.out.println("column Family:" +
                             new String(CellUtil.cloneFamily(cell)) + " ");
          System.out.println("row Name:" +
                             new String(CellUtil.cloneQualifier(cell)) + " "
          System.out.println("value:" +
                             new String(CellUtil.cloneValue(cell)) + " ");
      }
  }
```

Test.java t2.java

```java
import java.io.IOException;


public class t2 {

    public static Configuration configuration;
    public static Connection connection;
    public static Admin admin;

    public static void main(String[] args) throws IOException {
        String tableName = "t2";
        getTableByName(tableName);
    }

    private static void getTableByName(String tableName) throws IOException {
        init();

        Table table = connection.getTable(TableName.valueOf(tableName));
        Scan scan = new Scan();
        ResultScanner scanner = table.getScanner(scan);
        for (Result result : scanner) {
            showCell(result);
        }

        close();
    }
```

Console

```
<terminated> t2 [Java Application] /usr/local/jdk1.8/bin/java (Oct 31, 2020 3:01:53 PM)
log4j:WARN No appenders could be found for logger (org.a
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.h
RowName:rw1
Timetamp:1604127293478
column Family:cf1
row Name:q1
value:val1
```

（3）向表添加和删除指定的列族或列

给 t2 表 rw1 行的 cf2 列添加数据

```
hbase(main):008:0> scan 't2'
ROW                           COLUMN+CELL
 rw1                          column=cf1:q1, timestamp=1604127969928, value=add
 rw1                          column=cf1:q2, timestamp=1604128078435, value=add
1 row(s) in 0.0570 seconds

hbase(main):009:0> put 't2', 'rw1', 'cf2', 'add'
0 row(s) in 0.2650 seconds

hbase(main):010:0> scan 't2'
ROW                           COLUMN+CELL
 rw1                          column=cf1:q1, timestamp=1604127969928, value=add
 rw1                          column=cf1:q2, timestamp=1604128078435, value=add
 rw1                          column=cf2:, timestamp=1604128223335, value=add
1 row(s) in 0.0570 seconds

hbase(main):011:0>
```

Eclipse 实现

其中 insertRow()方法如下，之后将不再重复展示。

Test.java t3.java

```java
import java.io.IOException;


public class t3 {

    public static Configuration configuration;
    public static Connection connection;
    public static Admin admin;

    public static void main(String[] args) throws IOException {
        insertRow("t2", "rw1", "cf2", "i1", "eclipse");
        System.out.println("insert success");
    }

    // put rows
    private static void insertRow(String tableName,
            String rowKey, String colFamily,
            String col,String val) throws IOException {
        init();

        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();
        close();
    }
```

```
hbase(main):010:0> scan 't2'
ROW                         COLUMN+CELL
 rw1                        column=cf1:q1, timestamp=1604127969928, value=add
 rw1                        column=cf1:q2, timestamp=1604128078435, value=add
 rw1                        column=cf2:, timestamp=1604128223335, value=add
1 row(s) in 0.0570 seconds

hbase(main):011:0> scan 't2'
ROW                         COLUMN+CELL
 rw1                        column=cf1:q1, timestamp=1604127969928, value=add
 rw1                        column=cf1:q2, timestamp=1604128078435, value=add
 rw1                        column=cf2:, timestamp=1604128223335, value=add
 rw1                        column=cf2:i1, timestamp=1604128645083, value=eclipse
1 row(s) in 1.6340 seconds

hbase(main):012:0>
```

删除刚刚分别创建的列;

```
hbase(main):012:0> delete 't2', 'rw1', 'cf2'
0 row(s) in 1.0440 seconds

hbase(main):013:0> scan 't2'
ROW                         COLUMN+CELL
 rw1                        column=cf1:q1, timestamp=1604127969928, value=add
 rw1                        column=cf1:q2, timestamp=1604128078435, value=add
 rw1                        column=cf2:i1, timestamp=1604128645083, value=eclipse
1 row(s) in 0.0810 seconds

hbase(main):014:0>
```

Eclipse 实现

Test.java t3.java

```
    public static void main(String[] args) throws IOException {
//      insertRow("t2", "rw1", "cf2", "i1", "eclipse");
//      System.out.println("insert success");

        deleteRow("t2", "rw1", "cf2", "i1");
        System.out.println("delete success");
    }

    private static void deleteRow(String tableName, String rowKey,
            String colFamily, String col) throws IOException {
        init();

        Table table = connection.getTable(TableName.valueOf(tableName));
        Delete delete = new Delete(rowKey.getBytes());

        System.out.println("1.delete colFamily info or 2.delete col info");
        switch (new Scanner(System.in).next()) {
            case "1":
                // delete colFamily
                delete.addFamily(colFamily.getBytes());
                table.delete(delete);
                break;
            case "2":
                // delete col
                delete.addColumn(colFamily.getBytes(), col.getBytes());
                table.delete(delete);
                break;
            default:
                System.out.println("error");
        }
```

Console

```
<terminated> t3 [Java Application] /usr/local/jdk1.8/bin/java (Oct 31, 2020 3:29:48 PM)
log4j:WARN No appenders could be found for logger (org.a
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.h
1.delete colFamily info or 2.delete col info
2
ok
delete success
```

```
hbase(main):013:0> scan 't2'
ROW                     COLUMN+CELL
 rw1                    column=cf1:q1, timestamp=1604127969928, value=add
 rw1                    column=cf1:q2, timestamp=1604128078435, value=add
 rw1                    column=cf2:i1, timestamp=1604128645083, value=eclipse
1 row(s) in 0.0810 seconds

hbase(main):014:0> scan 't2'
ROW                     COLUMN+CELL
 rw1                    column=cf1:q1, timestamp=1604127969928, value=add
 rw1                    column=cf1:q2, timestamp=1604128078435, value=add
1 row(s) in 0.1980 seconds

hbase(main):015:0>
```

（4）清空表的所有记录数据

```
hbase(main):014:0> scan 't2'
ROW                          COLUMN+CELL
 rw1                         column=cf1:q1, timestamp=1604127969928, value=add
 rw1                         column=cf1:q2, timestamp=1604128078435, value=add
1 row(s) in 0.1980 seconds

hbase(main):015:0> truncate 't2'
Truncating 't2' table (it may take a while):
 - Disabling table...
 - Truncating table...
0 row(s) in 28.6020 seconds

hbase(main):016:0> scan 't2'
ROW                          COLUMN+CELL
0 row(s) in 1.3250 seconds

hbase(main):017:0>
```

Eclipse 实现：（按准备工作中重新创建表）

Test.java | t4.java

```java
public class t4 {

    public static Configuration configuration;
    public static Connection connection;
    public static Admin admin;

    public static void main(String[] args) throws Exception {
        clearTable("t2");
    }

    private static void clearTable(String tableName) throws Exception {
        init();

        // table copy
        HBaseAdmin admin2 = new HBaseAdmin(configuration);
        HTableDescriptor tableDescriptor =
                admin2.getTableDescriptor(Bytes.toBytes(tableName));

        TableName tableName2 = TableName.valueOf(tableName);

        // delete ori table
        admin.disableTable(tableName2);
        admin.deleteTable(tableName2);

        // create ori table
        admin.createTable(tableDescriptor);

        close();
    }
```

```
hbase(main):017:0> scan 't2'
ROW                          COLUMN+CELL
 rw1                         column=cf1:q1, timestamp=1604130054305, value=val1
1 row(s) in 0.2150 seconds

hbase(main):018:0> scan 't2'
ROW                          COLUMN+CELL
0 row(s) in 0.4960 seconds

hbase(main):019:0>
```

（5）统计表的行数（同理再次创建表）

```
hbase(main):019:0> scan 't2'
ROW                           COLUMN+CELL
 rw1                          column=cf1:q1, timestamp=1604130226712, value=val1
1 row(s) in 0.0530 seconds

hbase(main):020:0> count 't2'
1 row(s) in 1.5630 seconds

=> 1
hbase(main):021:0>
```

Eclipse 实现

Test.java | t5.java

```
import java.io.IOException;


public class t5 {

    public static Configuration configuration;
    public static Connection connection;
    public static Admin admin;

    public static void main(String[] args) throws Exception {
        countRows("t2");
    }

    private static void countRows(String tableName) throws IOException {
        init();

        Table table = connection.getTable(TableName.valueOf(tableName));
        Scan scan = new Scan();
        ResultScanner scanner = table.getScanner(scan);

        int num = 0;
        Result temp = scanner.next();
        while (temp != null) {
            temp = scanner.next();
            num ++;
        }
        System.out.println("rowNum: " + num);

        scanner.close();
        close();
    }
```

Console

```
<terminated> t5 [Java Application] /usr/local/jdk1.8/bin/java (Oct 31, 2020 3:49:51 PM)
log4j:WARN No appenders could be found for logger (org.a
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.h
rowNum: 1
```

(二)

（1）将标数据存入 HBase 中

创建学生表如下：

```
hbase(main):021:0> create 'Student', 'S_No', 'S_Name', 'S_Sex', 'S_Age'
0 row(s) in 5.3750 seconds

=> Hbase::Table - Student
hbase(main):022:0> put 'Student', 's01', 'S_No', '2018001'
0 row(s) in 0.2550 seconds

hbase(main):023:0> put 'Student', 's02', 'S_No', '2018002'
0 row(s) in 0.0270 seconds

hbase(main):024:0> put 'Student', 's03', 'S_No', '2018003'
0 row(s) in 0.0220 seconds

hbase(main):025:0> put 'Student', 's01', 'S_Name', 'Zhangsan'
0 row(s) in 0.0670 seconds

hbase(main):026:0> put 'Student', 's02', 'S_Name', 'Mary'
0 row(s) in 0.0300 seconds

hbase(main):027:0> put 'Student', 's03', 'S_Name', 'LiSi'
0 row(s) in 0.0440 seconds

hbase(main):028:0> put 'Student', 's01', 'S_Sex', 'male'
0 row(s) in 0.0050 seconds

hbase(main):029:0> put 'Student', 's02', 'S_Sex', 'female'
0 row(s) in 0.0230 seconds

hbase(main):030:0> put 'Student', 's03', 'S_Sex', 'male'
0 row(s) in 0.0160 seconds

hbase(main):031:0> put 'Student', 's01', 'S_Age', '23'
0 row(s) in 0.0550 seconds

hbase(main):032:0> put 'Student', 's02', 'S_Age', '22'
0 row(s) in 0.0070 seconds

hbase(main):033:0> put 'Student', 's03', 'S_Age', '24'
0 row(s) in 0.0260 seconds

hbase(main):034:0>
```

```
hbase(main):034:0> scan 'Student'
ROW                           COLUMN+CELL
 s01                          column=S_Age:, timestamp=1604131102681, value=23
 s01                          column=S_Name:, timestamp=1604131014474, value=Zhangsan
 s01                          column=S_No:, timestamp=1604130939121, value=2018001
 s01                          column=S_Sex:, timestamp=1604131065144, value=male
 s02                          column=S_Age:, timestamp=1604131110232, value=22
 s02                          column=S_Name:, timestamp=1604131028978, value=Mary
 s02                          column=S_No:, timestamp=1604130974326, value=2018002
 s02                          column=S_Sex:, timestamp=1604131077786, value=female
 s03                          column=S_Age:, timestamp=1604131119648, value=24
 s03                          column=S_Name:, timestamp=1604131041111, value=LiSi
 s03                          column=S_No:, timestamp=1604130988796, value=2018003
 s03                          column=S_Sex:, timestamp=1604131087045, value=male
3 row(s) in 0.1780 seconds

hbase(main):035:0>
```

创建课程表如下

```
=> Hbase::Table - Course
hbase(main):036:0> put 'Course', 'c01', 'C_No', '123001'
0 row(s) in 0.1160 seconds

hbase(main):037:0> put 'Course', 'c02', 'C_No', '123002'
0 row(s) in 0.0060 seconds

hbase(main):038:0> put 'Course', 'c03', 'C_No', '123003'
0 row(s) in 0.0310 seconds

hbase(main):039:0> put 'Course', 'c01', 'C_Name', 'Math'
0 row(s) in 0.0410 seconds

hbase(main):040:0> put 'Course', 'c02', 'C_Name', 'Computer Science'
0 row(s) in 0.0500 seconds

hbase(main):041:0> put 'Course', 'c03', 'C_Name', 'English'
0 row(s) in 0.0110 seconds

hbase(main):042:0> put 'Course', 'c01', 'C_Credit', '2.0'
0 row(s) in 0.0050 seconds

hbase(main):043:0> put 'Course', 'c02', 'C_Credit', '5.0'
0 row(s) in 0.0160 seconds

hbase(main):044:0> put 'Course', 'c03', 'C_Credit', '3.0'
0 row(s) in 0.1880 seconds

hbase(main):045:0> scan 'Course'
ROW                         COLUMN+CELL
 c01                        column=C_Credit:, timestamp=1604131454824, value=2.0
 c01                        column=C_Name:, timestamp=1604131398734, value=Math
 c01                        column=C_No:, timestamp=1604131350236, value=123001
 c02                        column=C_Credit:, timestamp=1604131463271, value=5.0
 c02                        column=C_Name:, timestamp=1604131415451, value=Computer Science
 c02                        column=C_No:, timestamp=1604131359001, value=123002
 c03                        column=C_Credit:, timestamp=1604131471224, value=3.0
 c03                        column=C_Name:, timestamp=1604131429311, value=English
 c03                        column=C_No:, timestamp=1604131366051, value=123003
3 row(s) in 0.1860 seconds

hbase(main):046:0>
```

创建选课表如下

```
hbase(main):046:0> create 'SC', 'SC_Sno',
hbase(main):047:0*
hbase(main):048:0* 'SC_Cno', 'SC_Score'
0 row(s) in 9.2920 seconds

=> Hbase::Table - SC
hbase(main):049:0> put 'SC', 'sc01', 'SC_Sno', '2018001'
0 row(s) in 0.0880 seconds

hbase(main):050:0> put 'SC', 'sc01', 'SC_Cno', '123001'
0 row(s) in 0.0810 seconds

hbase(main):051:0> put 'SC', 'sc01', 'SC_Score', '86'
0 row(s) in 0.0230 seconds

hbase(main):052:0> put 'SC', 'sc02', 'SC_Sno', '2018001'
0 row(s) in 0.0560 seconds

hbase(main):053:0> put 'SC', 'sc02', 'SC_Cno', '123003'
0 row(s) in 0.0790 seconds

hbase(main):054:0> put 'SC', 'sc02', 'SC_Score', '69'
0 row(s) in 0.0160 seconds

hbase(main):055:0> put 'SC', 'sc03', 'SC_Sno', '2018002'
0 row(s) in 0.0140 seconds

hbase(main):056:0> put 'SC', 'sc03', 'SC_Cno', '123002'
0 row(s) in 0.0200 seconds

hbase(main):057:0> put 'SC', 'sc03', 'SC_Score', '77'
0 row(s) in 0.0050 seconds

hbase(main):058:0> put 'SC', 'sc04', 'SC_Sno', '2018002'
0 row(s) in 0.0230 seconds

hbase(main):059:0> put 'SC', 'sc04', 'SC_Cno', '123003'
0 row(s) in 0.0040 seconds

hbase(main):060:0> put 'SC', 'sc04', 'SC_Score', '99'
0 row(s) in 0.0050 seconds

hbase(main):061:0> put 'SC', 'sc05', 'SC_Sno', '2018003'
0 row(s) in 0.0030 seconds
```

```
hbase(main):061:0> put 'SC', 'sc05', 'SC_Sno', '2018003'
0 row(s) in 0.0030 seconds

hbase(main):062:0> put 'SC', 'sc05', 'SC_Cno', '123001'
0 row(s) in 0.0040 seconds

hbase(main):063:0> put 'SC', 'sc05', 'SC_Score', '98'
0 row(s) in 0.0040 seconds

hbase(main):064:0> put 'SC', 'sc06', 'SC_Sno', '2018003'
0 row(s) in 0.0070 seconds

hbase(main):065:0> put 'SC', 'sc06', 'SC_Cno', '123002'
0 row(s) in 0.0470 seconds

hbase(main):066:0> put 'SC', 'sc06', 'SC_Score', '95'
0 row(s) in 0.0050 seconds

hbase(main):067:0> scan 'SC'
ROW                          COLUMN+CELL
 sc01                        column=SC_Cno:, timestamp=1604131653297, value=123001
 sc01                        column=SC_Score:, timestamp=1604131668838, value=86
 sc01                        column=SC_Sno:, timestamp=1604131609372, value=2018001
 sc02                        column=SC_Cno:, timestamp=1604131724364, value=123003
 sc02                        column=SC_Score:, timestamp=1604131741356, value=69
 sc02                        column=SC_Sno:, timestamp=1604131697705, value=2018001
 sc03                        column=SC_Cno:, timestamp=1604131811333, value=123002
 sc03                        column=SC_Score:, timestamp=1604131824196, value=77
 sc03                        column=SC_Sno:, timestamp=1604131768205, value=2018002
 sc04                        column=SC_Cno:, timestamp=1604131855863, value=123003
 sc04                        column=SC_Score:, timestamp=1604131869414, value=99
 sc04                        column=SC_Sno:, timestamp=1604131842156, value=2018002
 sc05                        column=SC_Cno:, timestamp=1604131926185, value=123001
 sc05                        column=SC_Score:, timestamp=1604131937973, value=98
 sc05                        column=SC_Sno:, timestamp=1604131900136, value=2018003
 sc06                        column=SC_Cno:, timestamp=1604131966503, value=123002
 sc06                        column=SC_Score:, timestamp=1604131976997, value=95
 sc06                        column=SC_Sno:, timestamp=1604131955242, value=2018003
6 row(s) in 0.1560 seconds

hbase(main):068:0>
```

（2）编程实现以下功能

1、创建表，参数 tableName 为表的名称，字符串数组 fields 为存储记录各个字段名称的数组。要求当 HBase 已经存在名为 tableName 的表的时候，先删除原有的表，然后再创建新的表。

```
private static void createTable(String talbeName,
                                String[] fields) throws Exception {
    init();

    TableName tableName = TableName.valueOf(talbeName);
    if (admin.tableExists(tableName)) {
        // first delete table
        // then create new table
        System.out.println("the table already exists");
        admin.disableTable(tableName);
        admin.deleteTable(tableName);
    }

    HTableDescriptor hTableDescriptor = new HTableDescriptor(tableName);
    for (String str : fields) {
        HColumnDescriptor hColumnDescriptor = new HColumnDescriptor(str);
        hTableDescriptor.addFamily(hColumnDescriptor);
    }

    admin.createTable(hTableDescriptor);
    System.out.println("create new table success");

    close();
}
```

t5.java | hbase.java

```
import org.apache.hadoop.hbase.client.Admin;
import org.apache.hadoop.hbase.client.Connection;
import org.apache.hadoop.hbase.client.ConnectionFactory;


public class hbase {

    public static Configuration configuration;
    public static Connection connection;
    public static Admin admin;

    public static void main(String[] args) throws Exception {
        String tableName = "testTable";
        String[] fileds = {"Id", "Score"};
        createTable(tableName, fileds);
    }
```

Console

```
<terminated> hbase [Java Application] /usr/local/jdk1.8/bin/java (Oct 31, 2020 4:24:54 PM)
log4j:WARN No appenders could be found for logger (org.a
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.h
create new table success
```

```
Specify a FORMATTER for all columns of a column family.

Scan can also be used directly from a table, by first getting a reference to a
table, like such:

  hbase> t = get_table 't'
  hbase> t.scan

Note in the above situation, you can still provide all the filtering, columns,
options, etc as described above.


hbase(main):069:0> scan 'testTable'
ROW                           COLUMN+CELL
0 row(s) in 0.4540 seconds

hbase(main):070:0>
```

再次运行时结果（原表已被删除后重新创建）

Console

```
<terminated> hbase [Java Application] /usr/local/jdk1.8/bin/java (Oct 31, 2020 4:26:36 PM)
log4j:WARN No appenders could be found for logger (org.a
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.h
the table already exists
create new table success
```

2、addRecord(String tableName, String row, String[] fields, String[] values)

向表 tableName、行 row（用 S_Name 表示）和字符串数组 fields 指定的单元格中添加对应的数据 values。其中，fields 中每个元素如果对应的列族下还有相应的列限定符的话，用

“columnFamily:column”表示。例如，同时向“Math”、“Computer Science”、“English”三列添加成绩时，字符串数组 fields 为{“Score:Math”, "Score:Computer Science", "Score:English"}，数组 values 存储这三门课的成绩。

hbase2.java

```
    public static void main(String[] args) throws Exception {
        String tableName = "testTable";
        String[] fields = {"Score:Math", "Score:Computer", "Score:English"};
        String[] values = {"99", "95", "90"};
        addRecord(tableName, "S_Name", fields, values);
    }

    private static void addRecord(String tableName, String row,
                    String[] fields, String[] values) throws IOException {
        init();

        Table table = connection.getTable(TableName.valueOf(tableName));
        for (int i = 0; i < fields.length; i ++) {
            Put put = new Put(row.getBytes());
            String[] cols = fields[i].split(":");
            if (cols.length == 1) {
                // not extra col limit
                put.addColumn(cols[0].getBytes(),
                        "".getBytes(),
                        values[i].getBytes());
            } else {
                put.addColumn(cols[0].getBytes(),
                        cols[1].getBytes(),
                        values[i].getBytes());
            }
            table.put(put);
        }

        System.out.println("add success");

        table.close();
```

```
hbase(main):069:0> scan 'testTable'
ROW                         COLUMN+CELL
0 row(s) in 0.4540 seconds

hbase(main):070:0> scan 'testTable'
ROW                         COLUMN+CELL
 S_Name                     column=Score:Computer, timestamp=1604133521400, value=95
 S_Name                     column=Score:English, timestamp=1604133522012, value=90
 S_Name                     column=Score:Math, timestamp=1604133520504, value=99
1 row(s) in 1.5380 seconds

hbase(main):071:0>
```

3、scanColumn(String tableName, String column)

浏览表 tableName 某一列的数据，如果某一行记录中该列数据不存在，则返回 null。要求当参数 column 为某一列族名称时，如果底下有若干个列限定符，则要列出每个列限定符代表的列的数据；当参数 column 为某一列具体名称（例如“Score:Math”）时，只需要列出该列的数据。

hbase3.java

```
    public static void main(String[] args) throws Exception {
        String tableName = "testTable";
        scanColum(tableName, "Score:Math");
    }

    private static void scanColum(String tableName, String colum)
            throws Exception {
        init();

        Table table = connection.getTable(TableName.valueOf(tableName));
        Scan scan = new Scan();
        String[] cols = colum.split(":");
        if (cols.length == 1) {
            // not col limit
            scan.addFamily(Bytes.toBytes(colum));
        }
        else {
            scan.addColumn(Bytes.toBytes(cols[0]), Bytes.toBytes(cols[1]));
        }

        ResultScanner scanner = table.getScanner(scan);
        Result result = scanner.next();
        while (result != null) {
            showCell(result);
            result = scanner.next();
        }

        table.close();
        close();
    }
```

Console

```
<terminated> hbase3 [Java Application] /usr/local/jdk1.8/bin/java (Oct 31, 2020 4:49:17 PM
log4j:WARN No appenders could be found for logger (org.a
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.h
RowName:S_Name
Timetamp:1604133520504
column Family:Score
row Name:Math
value:99
```

```
hbase(main):070:0> scan 'testTable'
ROW                     COLUMN+CELL
 S_Name                 column=Score:Computer, timestamp=1604133521400, value=95
 S_Name                 column=Score:English, timestamp=1604133522012, value=90
 S_Name                 column=Score:Math, timestamp=1604133520504, value=99
1 row(s) in 1.5380 seconds

hbase(main):071:0>
```

4、modifyData(String tableName, String row, String column)

修改表 tableName，行 row（可以用学生姓名 S_Name 表示），列 column 指定的单元格的数据。

hbase4.java

```java
public class hbase4 {

    public static Configuration configuration;
    public static Connection connection;
    public static Admin admin;

    public static void main(String[] args) throws Exception {
        String tableName = "testTable";
        modifyData(tableName, "S_Name", "Score:Math", "100");
    }

    private static void modifyData(String tableName, String row,
            String colum, String value) throws Exception {
        init();

        Table table = connection.getTable(TableName.valueOf(tableName));
        Put put = new Put(row.getBytes());
        String[] cols = colum.split(":");
        if (cols.length == 1) {
            // not col limit
            put.addColumn(colum.getBytes(), "".getBytes(), value.getBytes());
        }
        else {
            put.addColumn(cols[0].getBytes(), cols[1].getBytes(), value.getByte
        }
        table.put(put);
        System.out.println("modify success");

        table.close();
        close();
    }
```

Console

```
<terminated> hbase4 [Java Application] /usr/local/jdk1.8/bin/java (Oct 31, 2020 4:56:55 PM
log4j:WARN No appenders could be found for logger (org.a
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.h
modify success
```

```
hbase(main):070:0> scan 'testTable'
ROW                     COLUMN+CELL
 S_Name                 column=Score:Computer, timestamp=1604133521400, value=95
 S_Name                 column=Score:English, timestamp=1604133522012, value=90
 S_Name                 column=Score:Math, timestamp=1604133520504, value=99
1 row(s) in 1.5380 seconds

hbase(main):071:0> scan 'testTable'
ROW                     COLUMN+CELL
 S_Name                 column=Score:Computer, timestamp=1604133521400, value=95
 S_Name                 column=Score:English, timestamp=1604133522012, value=90
 S_Name                 column=Score:Math, timestamp=1604134641211, value=100
1 row(s) in 0.5860 seconds

hbase(main):072:0>
```

5、deleteRow(String tableName, String row)

删除表 tableName 中 row 指定的行的记录。

hbase5.java

```java
import java.io.IOException;


public class hbase5 {

    public static Configuration configuration;
    public static Connection connection;
    public static Admin admin;

    public static void main(String[] args) throws Exception {
        String tableName = "testTable";
        deleteRow(tableName, "S_Name");
    }

    private static void deleteRow(String tableName, String row)
            throws Exception {
        init();

        Table table = connection.getTable(TableName.valueOf(tableName));
        Delete delete = new Delete(row.getBytes());
        table.delete(delete);

        System.out.println("delete success");

        table.close();
        close();
    }
```

Console

```
<terminated> hbase5 [Java Application] /usr/local/jdk1.8/bin/java (Oct 31, 2020 5:00:49 PM
log4j:WARN No appenders could be found for logger (org.a
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.h
delete success
```

```
hbase(main):071:0> scan 'testTable'
ROW                         COLUMN+CELL
 S_Name                     column=Score:Computer, timestamp=1604133521400, value=95
 S_Name                     column=Score:English, timestamp=1604133522012, value=90
 S_Name                     column=Score:Math, timestamp=1604134641211, value=100
1 row(s) in 0.5860 seconds

hbase(main):072:0> scan 'testTable'
ROW                         COLUMN+CELL
0 row(s) in 1.2940 seconds

hbase(main):073:0>
```

关闭:

```
hbase(main):001:0> exit()
[dhw@localhost hbase]$ bin/stop-hbase.sh
stopping hbase.............................
dhw@localhost's password:
localhost: stopping zookeeper.
[dhw@localhost hbase]$ /usr/local/hadoop/sbin/stop-dfs.sh
Stopping namenodes on [localhost]
dhw@localhost's password:
localhost: stopping namenode
localhost: namenode did not stop gracefully after 5 seconds: killing with kill -
9
dhw@localhost's password:
localhost: stopping datanode
Stopping secondary namenodes [0.0.0.0]
dhw@0.0.0.0's password:
0.0.0.0: stopping secondarynamenode
[dhw@localhost hbase]$
```

注意：参考书《大数据基础编程、实验和案例教程.pdf》中关于打开 HBase 的 shell 命令有误，中间少加空格。