实验五

一、相关知识点

1. JDBC基本概念
2. JDBC简单查询、连接查询、嵌套查询、集函数查询等

二、实验目的

理解Statement对象、ResultSet对象。

三、实验内容

1、在booklib工程的BookManager类中增加如下函数（要求采用Statement完成相关查询），并在main函数中进行测试，在实验报告中将代码补上：

```
public int getBookCount(String pubid) throws BaseException{
        //要求返回该出版社的图书数量
        int result = 0;
        Connection conn = null;
        try {
            conn = DBUtil.getConnection();
            String sql = "select * from beanpublisher where pubid=?";
            java.sql.PreparedStatement pst = conn.prepareStatement(sql);
            pst.setString(1, pubid);
            java.sql.ResultSet rs = pst.executeQuery();
            if (!rs.next()) throw new BusinessException("出版社不存在");
            sql = "select count(1) from beanbook where pubid = ?" +
            "group by pubid";
            pst = conn.prepareStatement(sql);
```

```
            pst.setString(1, pubid);
            rs = pst.executeQuery();
            if (rs.next()) result = rs.getInt(1);
        } catch (SQLException e) {
            e.printStackTrace();
            throw new DbException(e);
        } finally {
            if (conn != null)
                try {
                    conn.close();
                } catch (SQLException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
        }
        return result;
}
```

```
mysql> select * from Beanbook;
+----------+----------+-------+-------+-------+
| barcode  | bookname | pubid | price | state |
+----------+----------+-------+-------+-------+
| barcode1 | book1    | 1     |    10 | 在库  |
| barcode2 | book2    | 1     |    10 | 在库  |
| barcode3 | book3    | 2     |    10 | 在库  |
| barcode4 | book4    | 1     |    10 | 在库  |
+----------+----------+-------+-------+-------+
4 rows in set (0.00 sec)
```

![image

```
278    public static void main(String[] args) {
279        BookManager pm = new BookManager();
280        try {
281            String pubid = "1";
282            System.out.println(pm.getBookCount(pubid));
283            pubid = "10";
284            System.out.println(pm.getBookCount(pubid));
285 //           System.out.println("有图书的出版社数量为 " + pm.getPublisherCount());
286 //           System.out.println("没有图书的出版社数量为 " + pm.getNoneBookPublisherCount());
287 //           System.out.println("图书的平均价格为 " + pm.getBookAvgPrice());
288        } catch (BaseException e) {
289            // TODO Auto-generated catch block
290            e.printStackTrace();
291        }
292 //       try {
293 //           pm.deletePublisher("testpubid");
294 //       } catch (BaseException e) {
295 //           // TODO Auto-generated catch block
296 //           e.printStackTrace();
297 //       }
298    }
299
300 }
301
```

```
Problems  Javadoc  Declaration  控制台 ×
<已终止> BookManager [Java 应用程序] C:\Users\Bexh0lder\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-120
3
cn.edu.zucc.booklib.util.BusinessException: 出版社不存在
        at cn.edu.zucc.booklib.control.BookManager.getBookCount(BookManager.java:26)
        at cn.edu.zucc.booklib.control.BookManager.main(BookManager.java:284)
```

```java
public int getPublisherCount() throws BaseException{
    //要求返回图书表中出现过的出版社数量
    int result = 0;
    Connection conn = null;
    try {
        conn = DBUtil.getConnection();
        String sql = "select count(distinct pubid) from beanbook";
        java.sql.PreparedStatement pst = conn.prepareStatement(sql);
        java.sql.ResultSet rs = pst.executeQuery();
        if(rs.next()) result = rs.getInt(1);
    } catch (SQLException e) {
        e.printStackTrace();
        throw new DbException(e);
    } finally {
```

```
        if (conn != null)
            try {
                conn.close();
            } catch (SQLException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
        return result;
    }
```

```
mysql> select * from Beanbook;
+----------+----------+------+-------+-------+
| barcode  | bookname | pubid | price | state |
+----------+----------+------+-------+-------+
| barcode1 | book1    | 1     |    10 | 在库  |
| barcode2 | book2    | 1     |    10 | 在库  |
| barcode3 | book3    | 2     |    10 | 在库  |
| barcode4 | book4    | 1     |    10 | 在库  |
+----------+----------+------+-------+-------+
4 rows in set (0.00 sec)
```

```
//              System.out.println(pm.getBookCount(pubid));
            System.out.println("有图书的出版社数量为 " + pm.getPublisherCount());
        } catch (BaseException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
//      try {
//          pm.deletePublisher("testpubid");
//      } catch (BaseException e) {
//          // TODO Auto-generated catch block
//          e.printStackTrace();
```

Problems Javadoc Declaration 控制台

```
<已终止> BookManager [Java 应用程序] C:\Users\Bexh0lder\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.fu
有图书的出版社数量为 2
```

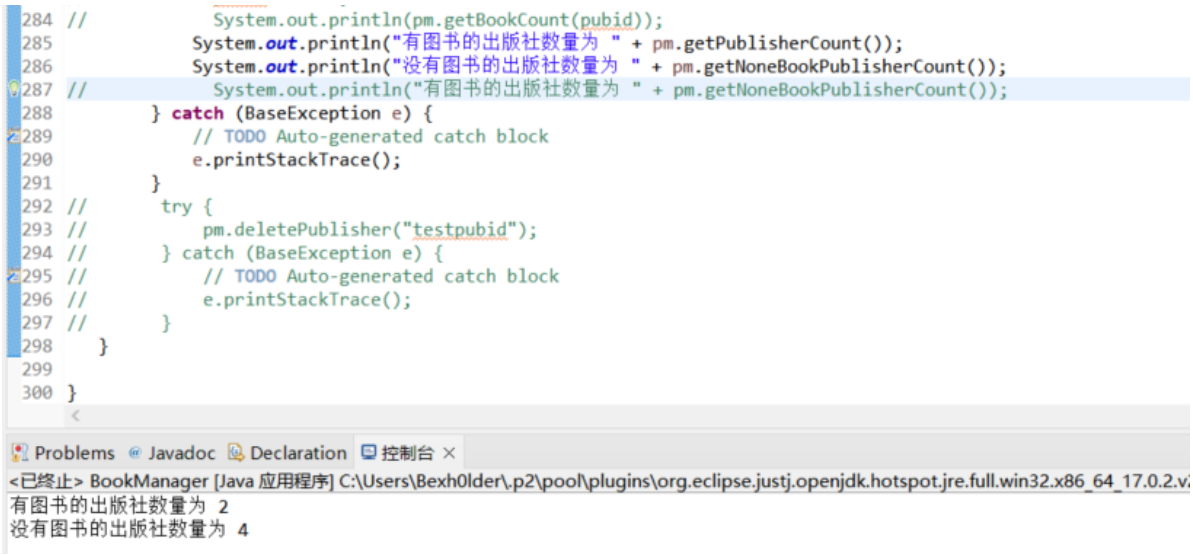
```
public int getNoneBookPublisherCount()throws BaseException{
    //要求返回没有图书的出版社数量
    int result = 0;
    Connection conn = null;
    try {
        conn = DBUtil.getConnection();
        String sql = "select count(*) from beanpublisher";
        java.sql.PreparedStatement pst = conn.prepareStatement(sql);
        java.sql.ResultSet rs = pst.executeQuery();
        if(rs.next()) result = rs.getInt(1);
        result -= (new BookManager()).getPublisherCount();
```

```
        } catch (SQLException e) {
            e.printStackTrace();
            throw new DbException(e);
        } finally {
            if (conn != null)
                try {
                    conn.close();
                } catch (SQLException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
        }
        return result;
    }
```

```
mysql> select * from Beanbook;
+----------+----------+-------+-------+-------+
| barcode  | bookname | pubid | price | state |
+----------+----------+-------+-------+-------+
| barcode1 | book1    | 1     |    10 | 在库  |
| barcode2 | book2    | 1     |    10 | 在库  |
| barcode3 | book3    | 2     |    10 | 在库  |
| barcode4 | book4    | 1     |    10 | 在库  |
+----------+----------+-------+-------+-------+
4 rows in set (0.00 sec)
```

```
mysql> select * from Beanpublisher;
+-------+---------------+---------+
| pubid | publisherName | address |
+-------+---------------+---------+
| 1     | 11            | 111     |
| 2     | 22            | 222     |
| 3     | 33            | 333     |
| 4     | 12            | 112     |
| 5     | 23            | 223     |
| 6     | 34            | 334     |
+-------+---------------+---------+
6 rows in set (0.00 sec)

mysql> select count(*) from Beanpublisher;
+----------+
| count(*) |
+----------+
|        6 |
+----------+
1 row in set (0.01 sec)
```



```
//            System.out.println(pm.getBookCount(pubid));
            System.out.println("有图书的出版社数量为 " + pm.getPublisherCount());
            System.out.println("没有图书的出版社数量为 " + pm.getNoneBookPublisherCount());
//            System.out.println("有图书的出版社数量为 " + pm.getNoneBookPublisherCount());
        } catch (BaseException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
//        try {
//            pm.deletePublisher("testpubid");
//        } catch (BaseException e) {
//            // TODO Auto-generated catch block
//            e.printStackTrace();
//        }
    }

}
```

Problems Javadoc Declaration 控制台

```
<已终止> BookManager [Java 应用程序] C:\Users\Bexh0lder\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v
有图书的出版社数量为 2
没有图书的出版社数量为 4
```

```java
public double getBookAvgPrice()throws BaseException{
        //要求返回图书的评价价格
        double result = 0;
        Connection conn = null;
        try {
            conn = DBUtil.getConnection();
            String sql = "select avg(price) from Beanbook";
            java.sql.PreparedStatement pst = conn.prepareStatement(sql);
            java.sql.ResultSet rs = pst.executeQuery();
            if(rs.next()) result = rs.getDouble(1);
        } catch (SQLException e) {
            e.printStackTrace();
            throw new DbException(e);
        } finally {
            if (conn != null)
                try {
                    conn.close();
                } catch (SQLException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
        }
        return result;
}
```

```
mysql> select * from Beanbook;
+----------+----------+------+-------+-------+
| barcode  | bookname | pubid | price | state |
+----------+----------+------+-------+-------+
| barcode1 | book1    | 1     |    10 | 在库  |
| barcode2 | book2    | 1     |    10 | 在库  |
| barcode3 | book3    | 2     |    10 | 在库  |
| barcode4 | book4    | 1     |    10 | 在库  |
+----------+----------+------+-------+-------+
4 rows in set (0.00 sec)
```

```
285 //            System.out.println("有图书的出版社数量为 " + pm.getPublisherCount());
286 //            System.out.println("没有图书的出版社数量为 " + pm.getNoneBookPublisherCount());
287             System.out.println("图书的平均价格为 " + pm.getBookAvgPrice());
288         } catch (BaseException e) {
289             // TODO Auto-generated catch block
290             e.printStackTrace();
291         }
292 //        try {
293 //            pm.deletePublisher("testpubid");
294 //        } catch (BaseException e) {
295 //            // TODO Auto-generated catch block
296 //            e.printStackTrace();
297 //        }
298     }
299
300 }
301
```

Problems Javadoc Declaration 控制台 ×

<已终止> BookManager [Java 应用程序] C:\Users\Bexh0lder\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v202

图书的平均价格为 10.0

2、在booklib工程的BookLendManager类中增加如下函数，并在main函数中进行测试，在实验报告中将代码补上：

```
1  public String loadBookLendOperator(String barcode)throws BaseException{
2      //参数为图书条码，返回这本图书最近一次被借出时的操作员姓名，要求采用连接查询实现。难点：如何识别出最近一次？假设不允许用mysql的limit关键字，也不能用嵌套查询，应该如何完成?
3      BeanBook book = (new BookManager()).loadBook(barcode);
4      if (book == null) throw new BusinessException("图书不存在");
5      String result = "";
6      Connection conn = null;
7      try {
8          conn = DBUtil.getConnection();
9          String sql = "select * from BeanBookLendRecord where bookBarcode=?";
10         java.sql.PreparedStatement pst = conn.prepareStatement(sql);
11         pst.setString(1, barcode);
12         java.sql.ResultSet rs = pst.executeQuery();
13         if (!rs.next()) {
14             throw new BusinessException("该图书没有借阅记录");
15         }
```

```
            sql = "select username,max(lendDate)" +
                  " from beanbooklendrecord" +
                  " left outer join beansystemuser on lendOperUserid = userid" +
                  " where bookBarcode=?";
            sql += " group by bookBarcode";
            pst = conn.prepareStatement(sql);
            pst.setString(1, barcode);
            rs = pst.executeQuery();
            if(rs.next()) result = rs.getString(1);
        } catch (SQLException e) {
            e.printStackTrace();
            throw new DbException(e);
        } finally {
            if (conn != null)
                try {
                    conn.close();
                } catch (SQLException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
        }
        return result;
    }
```

```
mysql> select * from beanbooklendrecord;
+----+----------+-------------+---------------------+---------------------+---------------+-----------------+----------+
| id | readerid | bookBarcode | lendDate            | returnDate          | lendOperUserid | returnOperUserid | penalSum |
+----+----------+-------------+---------------------+---------------------+---------------+-----------------+----------+
|  6 | 32001272 | barcode1    | 2022-05-09 17:36:52 | 2022-05-09 17:37:04 | admin         | admin           |        0 |
|  7 | 32001003 | barcode1    | 2022-05-09 17:37:19 | 2022-05-09 17:37:44 | admin         | admin           |        0 |
|  8 | 32001002 | barcode1    | 2022-05-09 17:38:04 | 2022-05-09 17:38:14 | admin         | admin           |        0 |
|  9 | 32001272 | barcode2    | 2022-05-09 17:41:20 | 2022-05-09 17:41:41 | admin         | admin           |        0 |
| 10 | 32001002 | barcode3    | 2022-05-09 17:41:32 | 2022-05-09 17:41:47 | admin         | admin           |        0 |
+----+----------+-------------+---------------------+---------------------+---------------+-----------------+----------+
5 rows in set (0.00 sec)

mysql>
```

```
    public static void main(String[] args) {
        BookLendManager pm = new BookLendManager();
        try {

            System.out.println("最近一次被借出时的操作员姓名为 " + pm.loadBookLendOperator("barcode1"));
//            System.out.println("最近一次被借出时的操作员姓名为 " + pm.loadBookLendOperator("barcode4"));
//            System.out.println("最近一次被借出时的操作员姓名为 " + pm.loadBookLendOperator("barcode5"));
        } catch (BaseException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
//        try {
//            pm.deletePublisher("testpubid");
//        } catch (BaseException e) {
//            // TODO Auto-generated catch block
//            e.printStackTrace();
//        }
    }

}

Problems  Javadoc  Declaration  控制台 ×
<已终止> BookLendManager [Java 应用程序] C:\Users\Bexh0lder\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1
最近一次被借出时的操作员姓名为 管理员
```

3、完成题2中的功能，要求采用嵌套查询实现。

```
1   public String loadBookLendOperator(String barcode)throws BaseException{
2       //参数为图书条码，返回这本图书最近一次被借出时的操作员姓名，要求采用连接查询实现。难点：如何识别出最近一次？假设不允许用mysql的limit关键字，也不能用嵌套查询，应该如何完成?
3       BeanBook book = (new BookManager()).loadBook(barcode);
4       if (book == null) throw new BusinessException("图书不存在");
5       String result = "";
6       Connection conn = null;
7       try {
8           conn = DBUtil.getConnection();
9           String sql = "select * from BeanBookLendRecord where bookBarcode=?";
10          java.sql.PreparedStatement pst = conn.prepareStatement(sql);
11          pst.setString(1, barcode);
12          java.sql.ResultSet rs = pst.executeQuery();
13          if (!rs.next()) {
14              throw new BusinessException("该图书没有借阅记录");
15          }
16          //连接查询
17  //          sql = "select username,max(lendDate)" +
18  //                  " from beanbooklendrecord" +
19  //                  " left outer join beansystemuser on lendOperUserid = userid" +
20  //                  " where bookBarcode=?";
21  //          sql += " group by bookBarcode";
22          //嵌套查询
23          sql = "select username from beansystemuser where userid in (" +
24                  "select lendOperUserid from beanbooklendrecord where lendDate in ("
    +
25                  "select max(lendDate) from beanbooklendrecord where bookBarcode =
    ? group by bookBarcode))";
26          pst = conn.prepareStatement(sql);
27          pst.setString(1, barcode);
28          rs = pst.executeQuery();
29          if(rs.next()) result = rs.getString(1);
30      } catch (SQLException e) {
31          e.printStackTrace();
32          throw new DbException(e);
33      } finally {
34          if (conn != null)
35              try {
36                  conn.close();
```

```
            } catch (SQLException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
        return result;
    }
```

```
    public static void main(String[] args) {
        BookLendManager pm = new BookLendManager();
        try {

            System.out.println("最近一次被借出时的操作员姓名为 " + pm.loadBookLendOperator("barcode1"));
//            System.out.println("最近一次被借出时的操作员姓名为 " + pm.loadBookLendOperator("barcode4"));
//            System.out.println("最近一次被借出时的操作员姓名为 " + pm.loadBookLendOperator("barcode5"));
        } catch (BaseException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
//        try {
//            pm.deletePublisher("testpubid");
//        } catch (BaseException e) {
//            // TODO Auto-generated catch block
//            e.printStackTrace();
//        }
    }

}

Problems  Javadoc  Declaration  控制台 ×
<已终止> BookLendManager [Java 应用程序] C:\Users\Bexh0lder\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-120
最近一次被借出时的操作员姓名为 管理员
```

4、在booklib工程的BookLendManager类中增加如下函数，并在main函数中进行测试，在实验报告中将代码补上：

```
public void showAllLendRecord(){
    //通过System.out.println方法，输出所有借阅记录的明细数据，要求结果中包括读者姓名、图书名称、所属出版社名称、借阅操作员姓名、归还操作员姓名、借阅时间、归还时间等
    //注意：需要注意未归还图书的情况
    Connection conn = null;
    try {
        List<BeanBookLendRecord> result = new ArrayList<BeanBookLendRecord>();
        conn = DBUtil.getConnection();
        String sql = "select * from BeanBookLendRecord";
        java.sql.PreparedStatement pst = conn.prepareStatement(sql);
        java.sql.ResultSet rs = pst.executeQuery();
        while(rs.next()) {
            String readerName = "";
            String bookName = "";
```

```
14              String publisherName = "";
15              String lendOperUserName = null;
16              String returnOperUserName = null;
17              BeanBookLendRecord r = new BeanBookLendRecord();
18              r.setId(rs.getInt(1));
19              r.setReaderid(rs.getString(2));
20              r.setBookBarcode(rs.getString(3));
21              r.setLendDate(rs.getTimestamp(4));
22              r.setReturnDate(rs.getTimestamp(5));
23              r.setLendOperUserid(rs.getString(6));
24              r.setReturnOperUserid(rs.getString(7));
25              r.setPenalSum(rs.getDouble(8));
26              result.add(r);
27              sql = "select readerName from beanreader where readerid = ?";
28              pst = conn.prepareStatement(sql);
29              pst.setString(1, r.getReaderid());
30              java.sql.ResultSet xs = pst.executeQuery();
31              if(xs.next()) readerName = xs.getNString(1);
32              sql = "select bookName, publisherName from beanbook,beanpublisher
    where barcode = ? and beanpublisher.pubid = beanbook.pubid";
33              pst = conn.prepareStatement(sql);
34              pst.setString(1, r.getBookBarcode());
35              xs = pst.executeQuery();
36              if(xs.next()){
37                  readerName = xs.getNString(1);
38                  publisherName = xs.getNString(2);
39              }
40              if(r.getLendOperUserid() != null)
41              {
42                  sql = "select username from beansystemuser where userid = ?";
43                  pst = conn.prepareStatement(sql);
44                  pst.setString(1, r.getLendOperUserid());
45                  xs = pst.executeQuery();
46                  if(xs.next()) lendOperUserName = xs.getNString(1);
47              }
48              if(r.getReturnOperUserid() != null)
49              {
50                  sql = "select username from beansystemuser where userid = ?";
51                  pst = conn.prepareStatement(sql);
52                  pst.setString(1, r.getReturnOperUserid());
53                  xs = pst.executeQuery();
54                  if(xs.next()) returnOperUserName = xs.getNString(1);
```

```
            }
            System.out.println("读者姓名:" + readerName + "\t图书名称:" +
                    bookName + "\t所属出版社名称:" + publisherName + "\t借阅操作员姓
名:" + lendOperUserName
                    + "\t归还操作员姓名:" + returnOperUserName + "\t借阅时间:" +
                    r.getLendDate() + "\t归还时间:" + r.getReturnDate());
        }
    } catch (SQLException e) {
        e.printStackTrace();
//          throw new DbException(e);
    } finally {
        if (conn != null)
            try {
                conn.close();
            } catch (SQLException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
    }
}
```

```
    public static void main(String[] args) {
        BookLendManager pm = new BookLendManager();
//        try {
            pm.showAllLendRecord();
//            System.out.println("最近一次被借出时的操作员姓名为 " + pm.loadBookLendOperator("barcode1"));
//            System.out.println("最近一次被借出时的操作员姓名为 " + pm.loadBookLendOperator("barcode4"));
//            System.out.println("最近一次被借出时的操作员姓名为 " + pm.loadBookLendOperator("barcode5"));
//        } catch (BaseException e) {
//            // TODO Auto-generated catch block
//            e.printStackTrace();
//        }
//        try {
//            pm.deletePublisher("testpubid");
//        } catch (BaseException e) {
//            // TODO Auto-generated catch block
//            e.printStackTrace();
//        }
    }

}
```

Problems Javadoc Declaration 控制台

```
<已终止> BookLendManager [Java 应用程序] C:\Users\Bexh0lder\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1208\jre\bin\javaw.exe  (2022年5月9日 下午9:58:38 – 下午9:58:4
读者姓名:book1    图书名称:  所属出版社名称:11    借阅操作员姓名:管理员  归还操作员姓名:管理员  借阅时间:2022-05-09 17:36:52.0    归还时间:2022-05-09 17:37:04.0
读者姓名:book1    图书名称:  所属出版社名称:11    借阅操作员姓名:管理员  归还操作员姓名:管理员  借阅时间:2022-05-09 17:37:19.0    归还时间:2022-05-09 17:37:44.0
读者姓名:book1    图书名称:  所属出版社名称:11    借阅操作员姓名:管理员  归还操作员姓名:管理员  借阅时间:2022-05-09 17:38:04.0    归还时间:2022-05-09 17:38:14.0
读者姓名:book2    图书名称:  所属出版社名称:11    借阅操作员姓名:管理员  归还操作员姓名:管理员  借阅时间:2022-05-09 17:41:20.0    归还时间:2022-05-09 17:41:41.0
读者姓名:book3    图书名称:  所属出版社名称:22    借阅操作员姓名:管理员  归还操作员姓名:管理员  借阅时间:2022-05-09 17:41:32.0    归还时间:2022-05-09 17:41:47.0
读者姓名:book1    图书名称:  所属出版社名称:11    借阅操作员姓名:管理员  归还操作员姓名:null  借阅时间:2022-05-09 20:51:38.0    归还时间:null
```

5、在**booklib**工程的**BookManager**类中增加如下函数，并在**main**函数中进行测试，在实验报告中将代码补上：

```
public void showTop5Books(){
    //通过System.out.println方法，输出借阅次数最多的5本图书及其借阅次数
    Connection conn = null;
    try {
        conn = DBUtil.getConnection();
        String sql = "select bookname, count(1) as x "
```

```
                + "from beanbooklendrecord, beanbook "
                + "where barcode = bookBarcode "
                + "group by bookname order by x DESC limit 5;";
            java.sql.PreparedStatement pst = conn.prepareStatement(sql);
            java.sql.ResultSet rs = pst.executeQuery();
            while(rs.next())
            {
                System.out.println("图书名称" + rs.getString(1) + "\t借阅次数" +
rs.getInt(2));
            }
        } catch (SQLException e) {
            e.printStackTrace();
//            throw new DbException(e);
        } finally {
            if (conn != null)
                try {
                    conn.close();
                } catch (SQLException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
        }
}
```

```
public void showTop5Publisher(){
        //通过System.out.println方法，输出被借阅图书次数最多的5个出版名称及其总借
阅次数和被借阅过的图书次数
        Connection conn = null;
        try {
            conn = DBUtil.getConnection();
            String sql = "select publishername,count(*) as x, count(distinct bookbarcode)
"
                + "from beanbooklendrecord "
                + "left outer join (beanbook left outer join beanpublisher on
beanbook.pubid = beanpublisher.pubid) on bookbarcode = barcode "
                + "group by beanbook.pubid order by x desc limit 5;";
            java.sql.PreparedStatement pst = conn.prepareStatement(sql);
            java.sql.ResultSet rs = pst.executeQuery();
            while(rs.next())
            {
                System.out.println("出版社名称" + rs.getString(1) + "\t总借阅次数" +
rs.getInt(2) + "\t被借阅过的图书数量" + rs.getInt(3));
```

```
15            }
16        } catch (SQLException e) {
17            e.printStackTrace();
18    //          throw new DbException(e);
19        } finally {
20            if (conn != null)
21                try {
22                    conn.close();
23                } catch (SQLException e) {
24                    // TODO Auto-generated catch block
25                    e.printStackTrace();
26                }
27        }
28    }
```

```
334     public static void main(String[] args) {
335         BookManager pm = new BookManager();
336 //        try {
337             pm.showTop5Publisher();
338 //          String pubid = "1";
339 //            System.out.println(pm.getBookCount(pubid));
340 //            pubid = "10";
341 //            System.out.println(pm.getBookCount(pubid));
342 //            System.out.println("有图书的出版社数量为 " + pm.getPublisherCount());
343 //            System.out.println("没有图书的出版社数量为 " + pm.getNoneBookPublisherCount());
344 //            System.out.println("图书的平均价格为 " + pm.getBookAvgPrice());
345 //        } catch (BaseException e) {
346             // TODO Auto-generated catch block
347 //            e.printStackTrace();
348 //        }
349 //       try {
350 //           pm.deletePublisher("testpubid");
351 //       } catch (BaseException e) {
352 //           // TODO Auto-generated catch block
353 //           e.printStackTrace();
354 //       }
355     }
356
357 }
358
```

Problems Javadoc Declaration 控制台

```
<已终止> BookManager [Java 应用程序] C:\Users\Bexh0lder\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-
出版社名称11        总借阅次数8        被借阅过的图书数量3
出版社名称22        总借阅次数2        被借阅过的图书数量1
出版社名称33        总借阅次数2        被借阅过的图书数量1
出版社名称23        总借阅次数2        被借阅过的图书数量1
出版社名称34        总借阅次数2        被借阅过的图书数量1
```

6、在BookLendManager中增加函数public void printDateLendRecord(String date)throws DbException，并在main函数中调用测试;要求通过该函数输出指定日期的所有借阅记录，，输出格式如下：

readerId=*,bookBarcode=**,lendDate=2020-05-01 15:17:01,returnDate=未归还

readerId=*,bookBarcode=**,lendDate=2020-05-01 15:17:01,returnDate=2020-05-12 12:00:00

说明：每个借阅记录1行输出，如果returnDate为空，则输出：“未归还”

注：时间的输出格式请使用java.text.SimpleDateFormat类实现

请提供函数代码及运行结果截图：

```
public void printDateLendRecord(String date)throws DbException{
        Connection conn = null;
        SimpleDateFormat DateFor = new SimpleDateFormat("yyyy-MM-dd hh:mm:ss");
        try {
            conn = DBUtil.getConnection();
            String sql = "select readerid, bookBarcode, lendDate, returnDate from beanbooklendrecord where lendDate = ?";
            java.sql.PreparedStatement pst = conn.prepareStatement(sql);
            pst.setString(1, date);
            java.sql.ResultSet rs = pst.executeQuery();
            if(rs.next())
                if(rs.getDate(4) != null)
                    System.out.println("readerId= " + rs.getString(1) + ",bookBarcode= " + rs.getString(2) + ",lendDate= " + DateFor.format(rs.getDate(3)) + ",returnDate=" + DateFor.format(rs.getDate(4)));
                else
                    System.out.println("readerId= " + rs.getString(1) + ",bookBarcode= " + rs.getString(2) + ",lendDate= " + DateFor.format(rs.getDate(3)) + ",未归还");
        } catch (SQLException e) {
            e.printStackTrace();
            throw new DbException(e);
        } finally {
            if (conn != null)
                try {
                    conn.close();
                } catch (SQLException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
        }
}
```

```
459    public static void main(String[] args) {
460        BookLendManager pm = new BookLendManager();
461        try {
462            pm.printDateLendRecord("2022-05-09 17:36:52");
463            pm.printDateLendRecord("2022-05-09 22:55:23");
464 //             System.out.println("最近一次被借出时的操作员姓名为 " + pm.loadBookLendOperator("barcode1"));
465 //             System.out.println("最近一次被借出时的操作员姓名为 " + pm.loadBookLendOperator("barcode4"));
466 //             System.out.println("最近一次被借出时的操作员姓名为 " + pm.loadBookLendOperator("barcode5"));
467        } catch (BaseException e) {
468 //             // TODO Auto-generated catch block
469            e.printStackTrace();
470        }
471 //       try {
472 //           pm.deletePublisher("testpubid");
473 //       } catch (BaseException e) {
474 //           // TODO Auto-generated catch block
475 //           e.printStackTrace();
476 //       }
477    }
478
479 }
480
```

Problems Javadoc Declaration 控制台 ×

<已终止> BookLendManager [Java 应用程序] C:\Users\Bexh0lder\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1208\jre\bin\javaw.

```
readerId= 32001272,bookBarcode= barcode1,lendDate= 2022-05-09 12:00:00,returnDate=2022-05-09 12:00:00
readerId= 32001272,bookBarcode= barcode1,lendDate= 2022-05-09 12:00:00,未归还
```