《Java 程序设计》实验报告 12

学生姓名：李季鸿　　班级：2021 级计本（3）班　　学号：20213002624

实验地点：9 教 404　　指导教师：张春元

实验日期：2023-05-22　　共 2 学时

实验环境：Win10+JDK1.8+ IntelliJ IDEA 2022.1.1

1. 实验目的

学习数据库编程和多线程编程。

2. 实验内容

（1）用集成化开发工具完成实验教材 P106 实验 1 内容（数据库采用 MySQL 数据库）。

（2）用集成化开发工具完成实验教材 P108 实验 2 内容（数据库采用 MySQL 数据库）。

（3）用集成化开发工具完成实验教材 P113 实验 1 内容。

（4）用集成化开发工具完成实验教材 P115 实验 2 内容。

3. 实验过程

报告撰写具体要求：截屏显示或直接写出实验 1 至实验 4 的源码和运行结果。

实验内容（1）：

ComputerAverPrice.java:

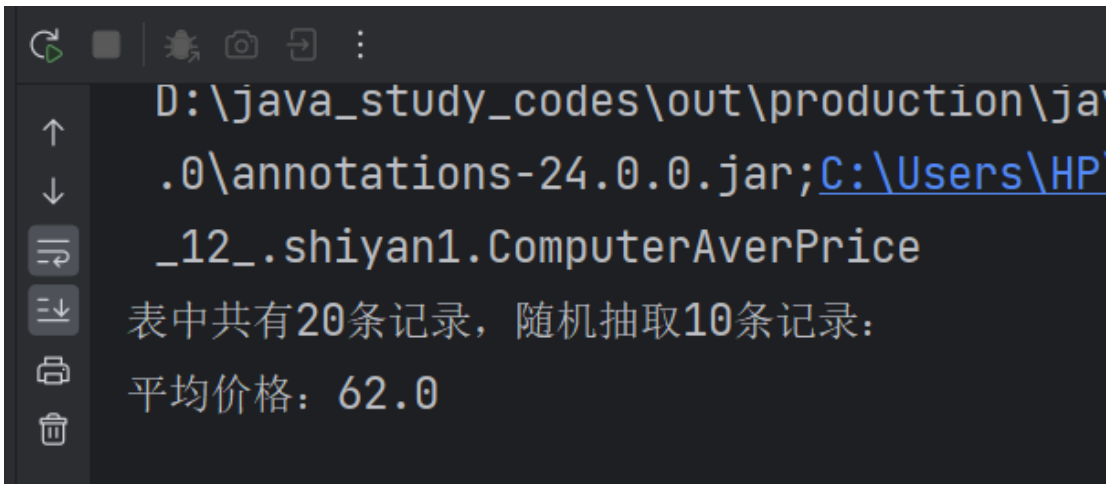
```
package _12_.shiyan1;

import java.sql.*;

/**
 * \* Created with IntelliJ IDEA.
 * \* @ProjectName: java_study_codes
 * \* @FileName: ComputerAverPrice
 * \* @author: li-jihong
 * \* Date: 2023-05-22 16:47
 */
public class ComputerAverPrice {
    public static void main(String[] args) {
        Connection con = null;
        Statement sql;
        ResultSet rs;
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
        } catch (Exception e) {}
        try{
            con = DriverManager.getConnection("jdbc:mysql://localhost:3306/shiyan12?useSSL=TRUE", "root","");
        } catch (SQLException e) {
```

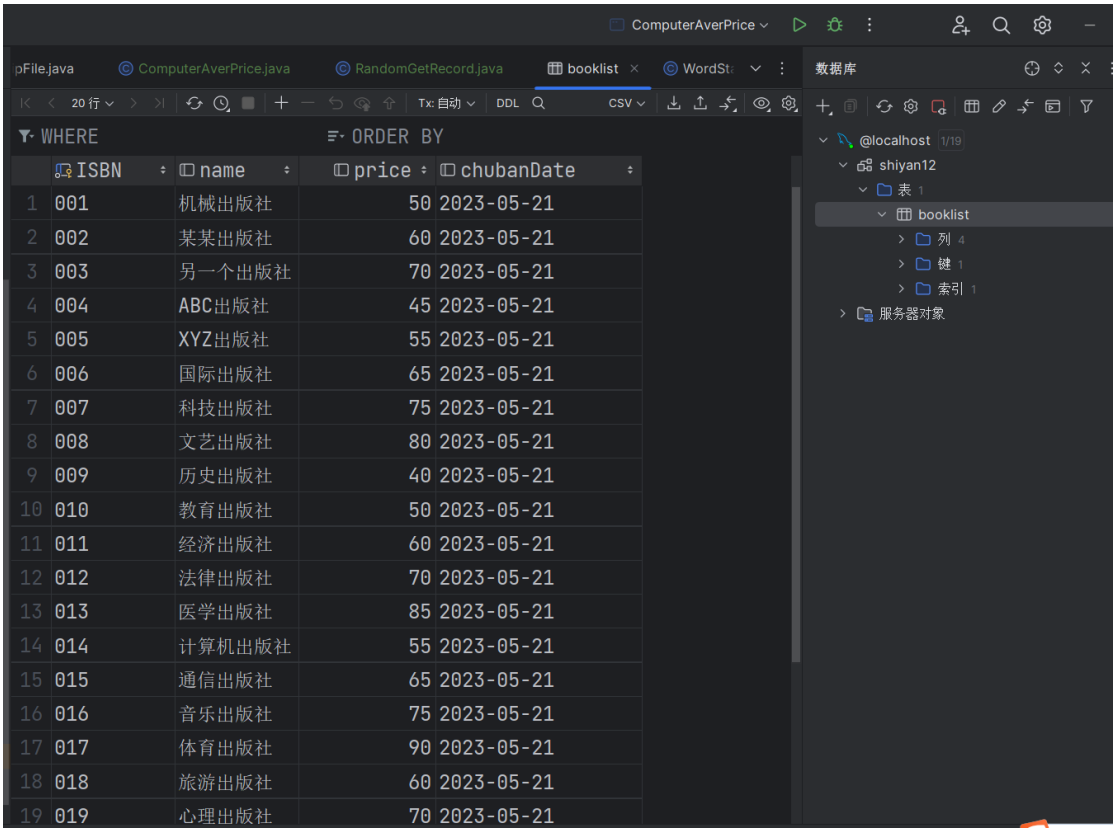
```
                System.out.println(e);
            }
            try{
                sql                                                                    =
con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_READ_ONL
Y);
                rs = sql.executeQuery("select * from booklist");
                rs.last();
                int max    = rs.getRow();
                System.out.println("表中共有" + max + "条记录，随机抽取 10 条记录：  ");
                int [] a = RandomGetRecord.getRandomNUmber(max, 10);
                float sum = 0;
                for(int i : a){
                    rs.absolute(i);
                    float price = rs.getFloat(3);
                    sum = sum + price;
                }
                con.close();
                System.out.println("平均价格：" + sum/a.length);
            }
            catch (SQLException e){}
        }
}
```

RandomGetRecord.java:

```
package _12_.shiyan1;

import java.util.Random;
import java.util.Vector;

/**
 * \* Created with IntelliJ IDEA.
 * \* @ProjectName: java_study_codes
 * \* @FileName: RandomGetRecord
 * \* @author: li-jihong
 * \* Date: 2023-05-22 16:30
 */
public class RandomGetRecord {
    public static int[] getRandomNUmber(int max, int amount) {
        Vector<Integer> vector = new Vector<Integer>();
        for (int i = 1; i <= max; i++) {
            vector.add(i);
        }
        int result[] = new int[amount];
```

```
            while (amount > 0) {
                int index = new Random().nextInt(vector.size());
                int m = vector.elementAt(index);
                vector.removeElementAt(index);
                result[amount - 1] = m;
                amount--;
            }
            return result;
        }
}
```





	ISBN	name	price	chubanDate
1	001	机械出版社	50	2023-05-21
2	002	某某出版社	60	2023-05-21
3	003	另一个出版社	70	2023-05-21
4	004	ABC出版社	45	2023-05-21
5	005	XYZ出版社	55	2023-05-21
6	006	国际出版社	65	2023-05-21
7	007	科技出版社	75	2023-05-21
8	008	文艺出版社	80	2023-05-21
9	009	历史出版社	40	2023-05-21
10	010	教育出版社	50	2023-05-21
11	011	经济出版社	60	2023-05-21
12	012	法律出版社	70	2023-05-21
13	013	医学出版社	85	2023-05-21
14	014	计算机出版社	55	2023-05-21
15	015	通信出版社	65	2023-05-21
16	016	音乐出版社	75	2023-05-21
17	017	体育出版社	90	2023-05-21
18	018	旅游出版社	60	2023-05-21
19	019	心理出版社	70	2023-05-21

课后练习：

Students.java:

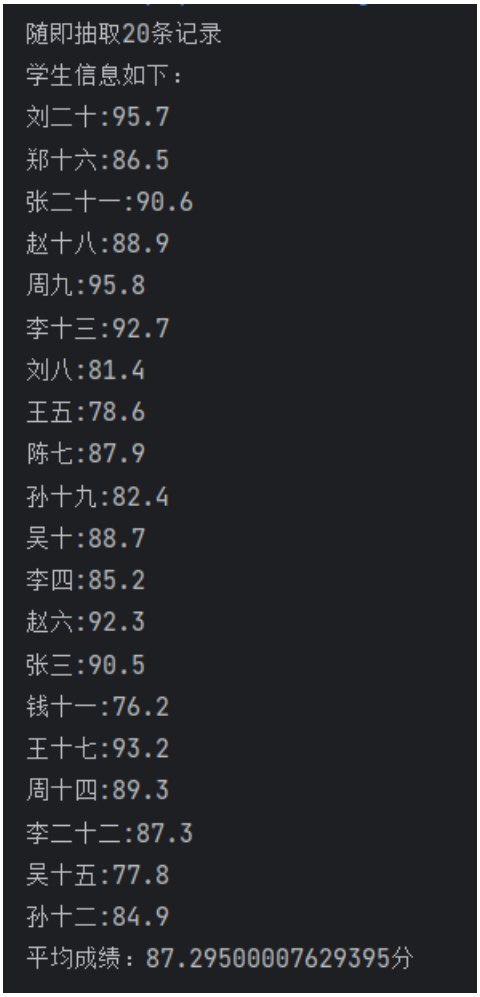
```
package _12_.shiyan1;

import java.sql.*;
import java.util.Random;
import java.util.Vector;

/**
 * \* Created with IntelliJ IDEA.
 * \* @ProjectName: java_study_codes
 * \* @FileName: students
 * \* @author: li-jihong
 * \* Date: 2023-05-22 17:29
 */
public class Students {
    public static void main(String args[]){
        int wantRecordAmount = 20;
        Random random = new Random();
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
        }
        catch (ClassNotFoundException e){
            System.out.print(e);
        }
        Connection con;
        Statement sql;
        ResultSet rs;
        try {
            String uri =
"jdbc:mysql://127.0.0.1:3306/shiyan12?user=root&password=123456&useSSL=true";
            con = DriverManager.getConnection(uri);
            sql =
con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_READ_ONL
Y);
            rs = sql.executeQuery("SELECT * FROM score");
            rs.last();
            int count = rs.getRow();
            Vector<Integer> vector = new Vector<Integer>();
            for(int i=1;i<=count;i++){
                vector.add(i);
            }
            int itemAmount = Math.min(wantRecordAmount,count);
            System.out.println("随即抽取"+itemAmount+"条记录");
```

```
                System.out.println("学生信息如下：");
                double sum = 0,n = itemAmount;
                while(itemAmount>0){
                    int randomIndex = random.nextInt(vector.size());
                    int index = (vector.elementAt(randomIndex)).intValue();
                    rs.absolute(index);
                    String name = rs.getString("name");
                    float grade = rs.getFloat("score");
                    System.out.println(name+":"+grade);
                    sum = sum +grade;
                    itemAmount--;
                    vector.removeElementAt(randomIndex);
                }
                con.close();
                double aver = sum/n;
                System.out.println("平均成绩："+aver+"分");
            }
            catch (SQLException e){
                System.out.println(""+e);
            }
        }
}
```

```sql

-- 插入20条数据
INSERT INTO score (name, score)
VALUES
  ('张三', 90.5),
  ('李四', 85.2),
  ('王五', 78.6),
  ('赵六', 92.3),
  ('陈七', 87.9),
  ('刘八', 81.4),
  ('周九', 95.8),
  ('吴十', 88.7),
  ('钱十一', 76.2),
  ('孙十二', 84.9),
  ('李十三', 92.7),
  ('周十四', 89.3),
  ('吴十五', 77.8),
  ('郑十六', 86.5),
  ('王十七', 93.2),
  ('赵十八', 88.9),
  ('孙十九', 82.4),
  ('刘二十', 95.7),
  ('张二十一', 90.6),
  ('李二十二', 87.3);
SELECT * FROM `score`
```

信息 | 结果 1 | 剖析 | 状态

id	name	score
1	张三	90.5
2	李四	85.2
3	王五	78.6
4	赵六	92.3
5	陈七	87.9
6	刘八	81.4
7	周九	95.8
8	吴十	88.7



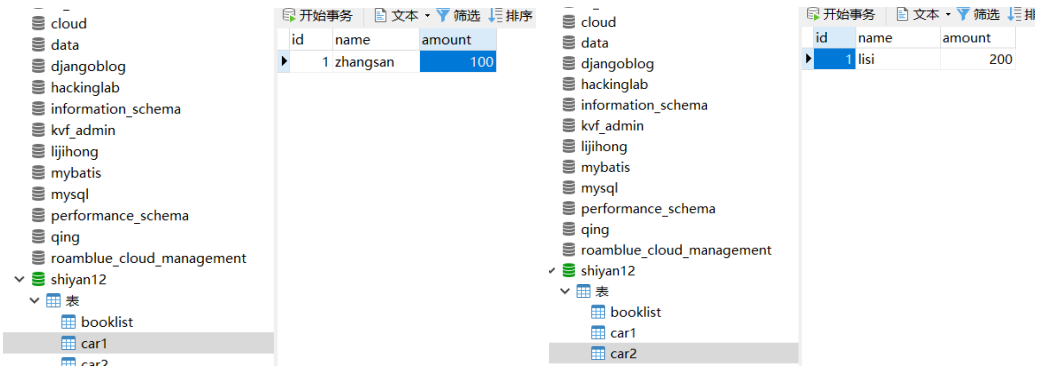
实验内容（2）：

TurnMoney.java:

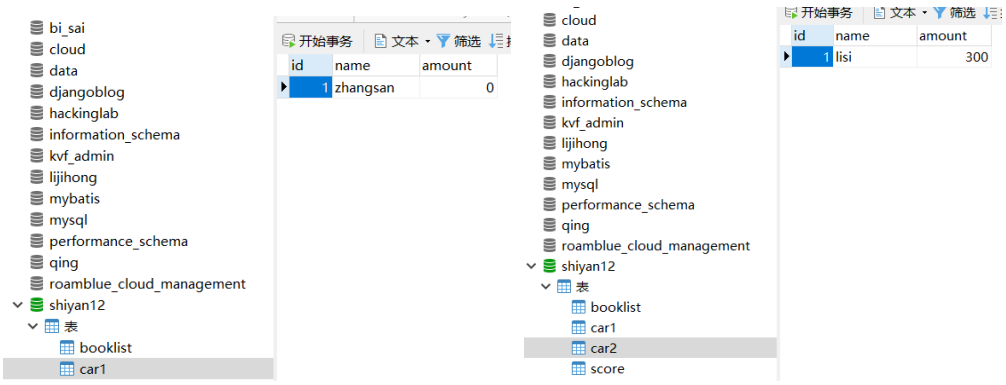
```
package _12_.shiyan2;

/**
 * \* Created with IntelliJ IDEA.
 * \* @ProjectName: java_study_codes
 * \* @FileName: TurnMoney
 * \* @author: li-jihong
 * \* Date: 2023-05-22 17:36
 */
import java.sql.*;
public class TurnMoney {
    public static void main(String args[]) {
        Connection con = null;
        Statement sql;
        ResultSet rs;

        try {
```

```
                double n = 100;
                String uri = "jdbc:mysql://127.0.0.1:3306/shiyan12?user=root&password=123456&useSSL=true";
                con = DriverManager.getConnection(uri);
                con.setAutoCommit(false);
                sql = con.createStatement();
                rs = sql.executeQuery("SELECT * FROM car1 WHERE name = 'zhangsan'");
                rs.next();
                double amountOne = rs.getDouble("amount");
                System.out.println("转账操作之前 zhangsan 的钱款数额：" + amountOne);
                rs = sql.executeQuery("SELECT * FROM car2 WHERE name = 'lisi'");
                rs.next();
                double amountTwo = rs.getDouble("amount");
                System.out.println("转账操作之前 lisi 的钱款数额：" + amountTwo);
                amountOne = amountOne - n;
                amountTwo = amountTwo + n;
                sql.execute("UPDATE car1 SET amount = " + amountOne + " WHERE name = 'zhangsan'");
                sql.execute("UPDATE car2 SET amount = " + amountTwo + " WHERE name = 'lisi'");
                con.commit();
                con.setAutoCommit(true);
                rs = sql.executeQuery("SELECT * FROM car1 WHERE name = 'zhangsan'");
                rs.next();
                amountOne = rs.getDouble("amount");
                System.out.println("转账操作之后 zhangsan 的钱款数额：" + amountOne);
                rs = sql.executeQuery("SELECT * FROM car2 WHERE name = 'lisi'");
                rs.next();
                amountTwo = rs.getDouble("amount");
                System.out.println("转账操作之后 lisi 的钱款数额：" + amountTwo);
                con.close();
            } catch (SQLException e) {
                try {
                    con.rollback();
                } catch (SQLException exp) {
                }
                System.out.println(e.toString());
            }
        }
}
```

转账前：



转账后：



在上面基础上：

```
package _12_.shiyan2;

import java.sql.*;

public class NewTransaction {
    public static void main(String[] args) {
        Connection con = null;
        Statement sql;
        ResultSet rs;
```

```
        try {
            double amount = 500;
            String uri = "jdbc:mysql://127.0.0.1:3306/shiyan12?user=root&password=123456&useSSL=true";
            con = DriverManager.getConnection(uri);
            con.setAutoCommit(false);
            sql = con.createStatement();
            rs = sql.executeQuery("SELECT * FROM car1 WHERE name = 'zhangsan'");
            rs.next();
            double zhangsanAmount = rs.getDouble("amount");
            System.out.println("转账操作之前 zhangsan 的钱款数额：" + zhangsanAmount);
            rs = sql.executeQuery("SELECT * FROM car2 WHERE name = 'lisi'");
            rs.next();
            double lisiAmount = rs.getDouble("amount");
            System.out.println("转账操作之前 lisi 的钱款数额：" + lisiAmount);

            if (zhangsanAmount >= amount) {
                zhangsanAmount -= amount;
                lisiAmount += amount;

                sql.execute("UPDATE car1 SET amount = " + zhangsanAmount + " WHERE name = 'zhangsan'");
                sql.execute("UPDATE car2 SET amount = " + lisiAmount + " WHERE name = 'lisi'");

                System.out.println("转账成功！");
                System.out.println("转账操作之后 zhangsan 的钱款数额：" + zhangsanAmount);
                System.out.println("转账操作之后 lisi 的钱款数额：" + lisiAmount);

                con.commit();
            } else {
                System.out.println("转账失败：zhangsan 的钱款数额不足！");
                con.rollback();
            }

            con.setAutoCommit(true);
            con.close();
        } catch (SQLException e) {
            try {
                con.rollback();
            } catch (SQLException exp) {
            }
```

```
                System.out.println(e.toString());
            }
        }
}
```

```
转账操作之前zhangsan的钱款数额：0.0
转账操作之前lisi的钱款数额：300.0
转账失败：zhangsan的钱款数额不足！
```

实验内容（3）：

Letter.java:

```
package _12_.shiyan3;

/**
 * \* Created with IntelliJ IDEA.
 * \* @ProjectName: java_study_codes
 * \* @FileName: Letter
 * \* @author: li-jihong
 * \* Date: 2023-05-22 18:08
 */
public class Letter {
    char c = '\0';
    public   void   setChar(char c){
        this.c = c;
    }
    public char getChar(){
        return c;
    }
}
```

TypeKey.java:

```
package _12_.shiyan3;

/**
 * \* Created with IntelliJ IDEA.
 * \* @ProjectName: java_study_codes
 * \* @FileName: TypeKey
 * \* @author: li-jihong
 * \* Date: 2023-05-22 18:07
 */
public class TypeKey {
    public static void main(String args[]){
```

```
        System.out.println("键盘练习（输入#结束程序）：");
        System.out.println("输入显示的字母（回车）\n");
        Letter letter;
        letter = new Letter();
        GiveLetterThread giveChar;
        InputLetterThread typeChar;
        giveChar = new GiveLetterThread();
        typeChar = new InputLetterThread();
        giveChar.setLetter(letter);
        giveChar.setSleepLength(3200);
        typeChar.setLetter(letter);
        giveChar.start();
        typeChar.start();
    }
}
```

GiveLetterThread.java:

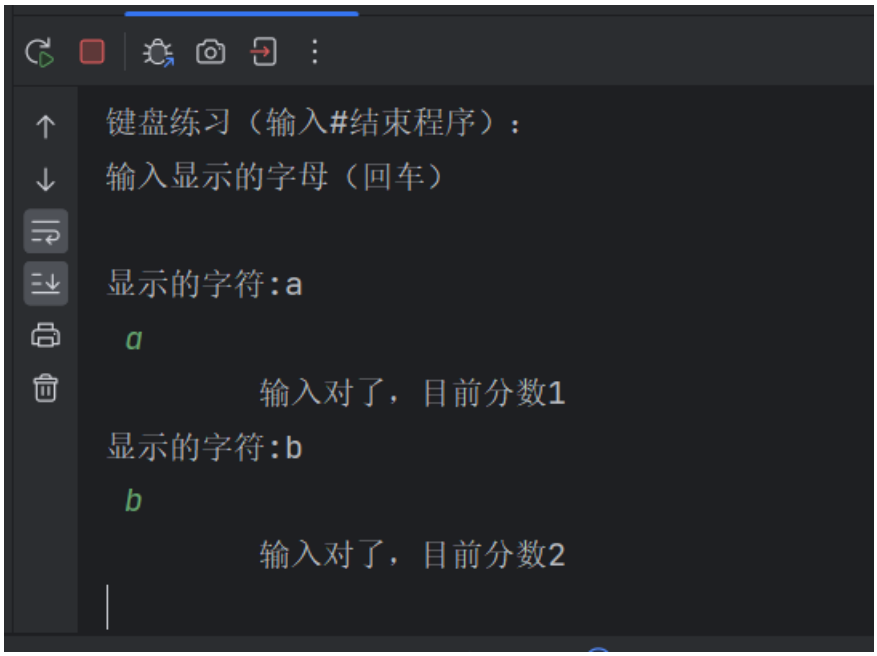
```
package _12_.shiyan3;

/**
 * \* Created with IntelliJ IDEA.
 * \* @ProjectName: java_study_codes
 * \* @FileName: GiveLetterThread
 * \* @author: li-jihong
 * \* Date: 2023-05-22 18:10
 */
public class GiveLetterThread extends Thread {
    Letter letter;
    char startChar = 'a', endChar = 'z';
    int sleepLength = 5000;

    public void setLetter(Letter letter) {
        this.letter = letter;
    }

    public void setSleepLength(int n) {
        sleepLength = n;
    }

    public void run() {
        char c = startChar;
        while (true) {
            letter.setChar(c);
```

```
                System.out.printf("显示的字符:%c \n ", letter.getChar());
                try {
                    Thread.sleep(sleepLength);
                } catch (InterruptedException e) {
                }
                c = (char) (c + 1);
                if (c > endChar)
                    c = startChar;
            }
        }
}
```

InputLetterThread.java:

```
package _12_.shiyan3;

import java.util.Scanner;

/**
 * \* Created with IntelliJ IDEA.
 * \* @ProjectName: java_study_codes
 * \* @FileName: InputLetterThread
 * \* @author: li-jihong
 * \* Date: 2023-05-22 18:11
 */
public class InputLetterThread extends Thread {
    Scanner reader;
    Letter letter;
    int score = 0;

    InputLetterThread() {
        reader = new Scanner(System.in);
    }

    public void setLetter(Letter letter) {
        this.letter = letter;
    }

    public void run() {
        while (true) {
            String str = reader.nextLine();
            char c = str.charAt(0);
            if (c == letter.getChar()) {
                score++;
```

```
                System.out.printf("\t\t 输入对了，目前分数%d\n" ,score);
            } else {
                System.out.printf("\t\t 输入错了，目前分数%d\n ", score);
            }
            if (c == '#')
                System.exit(0);
        }
    }
}
```



实验内容（4）：

TwoThreadGuessNumber.java:

```
package _12_.shiyan4;

/**
 * \* Created with IntelliJ IDEA.
 * \* @ProjectName: java_study_codes
 * \* @FileName: TwoThreadGuessNumber
 * \* @author: li-jihong
 * \* Date: 2023-05-26 20:05
 */
public class TwoThreadGuessNumber {
    public static void main(String args[]) {
        Number number = new Number();
        number.giveNumberThread.start();
        number.guessNumberThread.start();
//          number.guessNumberThread1.start();
```

```
//              number.guessNumberThread2.start();
        }
}
```

Number.java:

```
package _12_.shiyan4;

/**
 * \* Created with IntelliJ IDEA.
 * \* @ProjectName: java_study_codes
 * \* @FileName: Number
 * \* @author: li-jihong
 * \* Date: 2023-05-26 20:06
 */
public class Number implements Runnable {
        //      private static int theNumber;
        final int SMALLER = -1, LARGER = 1, SUCCESS = 8;
        int realNumber, guessNumber, min = 0, max = 100, message = SMALLER;
        //      int guessNumber1, guessNumber2, min = 0, max = 100, message = SMALLER,
message2 = SMALLER;
        boolean pleaseGuess = false, isGiveNumber = false;
        //          boolean pleaseGuess = false, isGiveNumber = false, pleaseGuess2 = false,
isGiveNumber2 = false;
        Thread giveNumberThread, guessNumberThread;
//      Thread giveNumberThread, guessNumberThread1, guessNumberThread2;

        Number() {
                giveNumberThread = new Thread(this);
                guessNumberThread = new Thread(this);
//              guessNumberThread1 = new Thread(this);
//              guessNumberThread2 = new Thread(this);
        }

        public void run() {
                for (int count = 1; true; count++) {
                        setMessage(count);
                        if (message == SUCCESS)
                                return;
                }
        }

//      public void run() {
//              for (int count = 1; true; count++) {
//                      setMessage(count);
```

```
//                  if (message == SUCCESS || message2 == SUCCESS)
//                      return;
//              }
//      }

    public synchronized void setMessage(int count) {

        if (Thread.currentThread() == giveNumberThread && isGiveNumber == false) {
            realNumber = (int) (Math.random() * 100) + 1;
            System.out.println("随机给你一个 1 至 100 之间的数，猜猜是多少？");
            isGiveNumber = true;
            pleaseGuess = true;
        }
        if (Thread.currentThread() == giveNumberThread) {
            while (pleaseGuess == true)
                try {
                    wait();
                } catch (InterruptedException e) {
                }
            if (realNumber == guessNumber) {
                message = SMALLER;
                System.out.println("你猜小了");
            } else if (realNumber < guessNumber) {
                message = LARGER;
                System.out.println("你猜大了");
            } else {
                message = SUCCESS;
                System.out.println("恭喜，你猜对了");
            }
            pleaseGuess = true;
        }
        if (Thread.currentThread() == guessNumberThread && isGiveNumber == true) {
            while (pleaseGuess == false)
                try {
                    wait();
                } catch (InterruptedException e) {
                }
            if (message == SMALLER) {
                min = guessNumber;
                guessNumber = (min + max) / 2;
                System.out.println("A 第" + count + "次猜这个数是：" + guessNumber);
            } else if (message == LARGER) {
                max = guessNumber;
```

```
                guessNumber = (min + max) / 2;
                System.out.println("A 第" + count + "次猜这个数是：" + guessNumber);
            }
            pleaseGuess = false;
        }
        notifyAll();
    }
}
```

```
随机给你一个1至100之间的数，猜猜是多少？
A第1次猜这个数是：50
你猜大了
A第2次猜这个数是：25
恭喜，你猜对了

进程已结束,退出代码0
```

实验后的练习：

TwoThreadGuessNumber.java:

```
package _12_.shiyan4;

/**
 * \* Created with IntelliJ IDEA.
 * \* @ProjectName: java_study_codes
 * \* @FileName: TwoThreadGuessNumber
 * \* @author: li-jihong
 * \* Date: 2023-05-26 20:05
 */
public class TwoThreadGuessNumber {
    public static void main(String args[]) {
        Number number = new Number();
        number.giveNumberThread.start();
        number.guessNumberThread1.start();
        number.guessNumberThread2.start();
    }
}
```

Number.java:

```
package _12_.shiyan4;

/**
 * \* Created with IntelliJ IDEA.
```

```
 * \* @ProjectName: java_study_codes
 * \* @FileName: Number
 * \* @author: li-jihong
 * \* Date: 2023-05-26 20:06
 */
public class Number implements Runnable {
        private static int theNumber;
    final int SMALLER = -1, LARGER = 1, SUCCESS = 8;
//    int realNumber, guessNumber, min = 0, max = 100, message = SMALLER;
        int guessNumber1, guessNumber2, min = 0, max = 100, message = SMALLER, message2 = SMALLER;
//    boolean pleaseGuess = false, isGiveNumber = false;
        boolean pleaseGuess = false, isGiveNumber = false, pleaseGuess2 = false, isGiveNumber2 = false;
//    Thread giveNumberThread, guessNumberThread;
    Thread giveNumberThread, guessNumberThread1, guessNumberThread2;

    Number() {
        giveNumberThread = new Thread(this);
        guessNumberThread1 = new Thread(this);
        guessNumberThread2 = new Thread(this);
    }

//    public void run() {
//        for (int count = 1; true; count++) {
//            setMessage(count);
//            if (message == SUCCESS)
//                return;
//        }
//    }

    public void run() {
        for (int count = 1; true; count++) {
            setMessage(count);
            if (message == SUCCESS || message2 == SUCCESS)
                return;
        }
    }

//    public synchronized void setMessage(int count) {
//
//        if (Thread.currentThread() == giveNumberThread && isGiveNumber == false) {
//            realNumber = (int) (Math.random() * 100) + 1;
```

```
//                 System.out.println("随机给你一个 1 至 100 之间的数，猜猜是多少？");
//                 isGiveNumber = true;
//                 pleaseGuess = true;
//             }
//             if (Thread.currentThread() == giveNumberThread) {
//                 while (pleaseGuess == true)
//                     try {
//                         wait();
//                     } catch (InterruptedException e) {
//                     }
//                 if (realNumber == guessNumber) {
//                     message = SMALLER;
//                     System.out.println("你猜小了");
//                 } else if (realNumber < guessNumber) {
//                     message = LARGER;
//                     System.out.println("你猜大了");
//                 } else {
//                     message = SUCCESS;
//                     System.out.println("恭喜，你猜对了");
//                 }
//                 pleaseGuess = true;
//             }
//             if (Thread.currentThread() == guessNumberThread && isGiveNumber == true) {
//                 while (pleaseGuess == false)
//                     try {
//                         wait();
//                     } catch (InterruptedException e) {
//                     }
//                 if (message == SMALLER) {
//                     min = guessNumber;
//                     guessNumber = (min + max) / 2;
//                     System.out.println("A 第" + count + "次猜这个数是：" + guessNumber);
//                 } else if (message == LARGER) {
//                     max = guessNumber;
//                     guessNumber = (min + max) / 2;
//                     System.out.println("A 第" + count + "次猜这个数是：" + guessNumber);
//                 }
//                 pleaseGuess = false;
//             }
//             notifyAll();
//         }
```

```
    public synchronized void setMessage(int count) {

        if (Thread.currentThread() == giveNumberThread && isGiveNumber == false) {
            theNumber = (int) (Math.random() * 100) + 1;
            System.out.println("随机给你一个 1 至 100 之间的数，猜猜是多少？" +
theNumber);
            isGiveNumber = true;
            pleaseGuess = true;
            pleaseGuess2 = true;
        }
        if (Thread.currentThread() == giveNumberThread) {
            while (pleaseGuess == true || pleaseGuess2 == true)
                try {
                    wait();
                } catch (InterruptedException e) {
                }
            if (theNumber == guessNumber1 || theNumber == guessNumber2) {
                message = SUCCESS;
                System.out.println("恭喜，你猜对了");
                return;
            }
            pleaseGuess = true;
            pleaseGuess2 = true;
        }
        if (Thread.currentThread() == guessNumberThread1 && isGiveNumber == true) {
            while (pleaseGuess == false)
                try {
                    wait();
                } catch (InterruptedException e) {
                }
            if (message == SMALLER) {
                min = guessNumber1;
                guessNumber1 = (min + max) / 2;
                System.out.println("A 第" + count + "次猜这个数是：" + guessNumber1);
            } else if (message == LARGER) {
                max = guessNumber1;
                guessNumber1 = (min + max) / 2;
                System.out.println("A 第" + count + "次猜这个数是：" + guessNumber1);
            }
            if (theNumber > guessNumber1) {
                message = SMALLER;
                System.out.println("你猜小了");
            } else if (theNumber < guessNumber1) {
```

```
                message = LARGER;
                System.out.println("你猜大了");
            }

            pleaseGuess = false;
        }
        if (Thread.currentThread() == guessNumberThread2 && isGiveNumber == true) {
            while (pleaseGuess2 == false)
                try {
                    wait();
                } catch (InterruptedException e) {
                }
            if (message2 == SMALLER) {
                min = guessNumber2;
                guessNumber2 = (min + max) / 2;
                System.out.println("B 第" + count + "次猜这个数是：" + guessNumber2);
            } else if (message2 == LARGER) {
                max = guessNumber2;
                guessNumber2 = (min + max) / 2;
                System.out.println("B 第" + count + "次猜这个数是：" + guessNumber2);
            }
            if (theNumber > guessNumber2) {
                message2 = SMALLER;
                System.out.println("你猜小了");
            } else if (theNumber < guessNumber2) {
                message2 = LARGER;
                System.out.println("你猜大了");
            }
            pleaseGuess2 = false;
        }
        notifyAll();
    }
}
```

```
随机给你一个1至100之间的数，猜猜是多少？92
B第1次猜这个数是：50
你猜小了
A第1次猜这个数是：50
你猜小了
A第2次猜这个数是：75
你猜小了
B第2次猜这个数是：75
你猜小了
B第3次猜这个数是：87
你猜小了
A第3次猜这个数是：87
你猜小了
A第4次猜这个数是：93
你猜大了
B第4次猜这个数是：93
你猜大了
B第5次猜这个数是：90
你猜小了
A第5次猜这个数是：90
你猜小了
A第6次猜这个数是：91
你猜小了
B第6次猜这个数是：91
你猜小了
B第7次猜这个数是：92
A第7次猜这个数是：92
恭喜，你猜对了
```

4. 实验总结

写出实验中的心得体会（对第 11 章理论课重点简述）。

一、控制游标

.Statement sql=con.createStatement();创建的 Statement 对象的执行查询所获得的结果集的游标只能用过 next()顺序向下移动。

为了得到一个游标可滚动的结果集，需使用下述方法获得一个 Statement 对象:
Statement stmt = con.createStatement(int type,int concurrency);
注: type 参数用来指明游标滚动方式;concurrency 参数用来指明是否用查询结果集更新数据库。
二、使用预处理语句
问题
Statement sql=con.createStatement();ResultSet rs = sql.executeQuery(sql 语句);int ok = sql.executeUpdate(sql 语句);
前面讲的查询或更新 SQL 语句在执行时需要事先由数据库中的 SQL 解释器解释成底层的内部命令才能运行，增大了数据库的负担，降低了执行的效率。
解决办法:采用“预处理语句+通配符”将要执行的 SQL 语句的一般形式提前解释好，真正运行时只需将通配符代表的具体值送进去即可。
三、java 中使用 mysql，使用预处理语句作用或者说好处是什么
在 Java 中使用 MySQL 数据库时，使用预处理语句（Prepared Statements）有以下好处：
1. 提高性能：预处理语句可以预编译 SQL 查询，并将其存储在数据库中，以便重复执行。这样可以减少每次执行查询时的解析和编译开销，从而提高查询的性能。
2. 防止 SQL 注入攻击：预处理语句使用参数化查询，将用户输入的数据作为参数传递给 SQL 语句，而不是将用户输入直接嵌入到 SQL 语句中。这样可以有效防止 SQL 注入攻击，因为参数化查询会对输入数据进行转义和验证，确保查询的安全性。
3. 简化 SQL 语句拼接：通过使用预处理语句，可以将 SQL 语句和参数分开处理，避免手动拼接 SQL 字符串。这样可以提高代码的可读性和可维护性，并降低出错的风险。
4. 提供数据类型安全性：预处理语句可以根据参数的数据类型进行类型检查和转换。这样可以确保传递给数据库的数据类型正确匹配，并减少数据转换错误的可能性。
5. 重复使用查询计划：预处理语句可以缓存查询计划，使其可重复使用。这在需要多次执行相同查询的情况下非常有用，避免了每次查询都重新生成查询计划的开销。
总而言之，使用预处理语句可以提高数据库查询的性能、安全性和可维护性。它是一种推荐的做法，特别是当需要执行频繁的 SQL 查询时。
四、事务及事务处理
✓ 事务由一组 SQL 语句组成。
✓ 事务处理是指：应用程序保证事务中的 SQL 语句要么全部 都执行，要么一个都不执行。
JDBC 事务处理步骤(下述方法均来自数据库连接对象)
✓ 1. 用 con.setAutoCommit(booean b)关闭自动提交模式 注：要先关闭自动提交模式，再获取 Statement 对象 sql
✓ 2. 用 con.commit()让事务中的全部语句生效 注：要先执行完事务中的 sql 语句，再执行 commit()方法
✓ 3. 用 con.rollback()处理事务失败 注：一般放在 catch(){}语句块中