

---

# 《Java 程序设计》课程设计

## 一、需求分析

### 1. 背景介绍

随着当今时代学生数量的逐渐增加，各个学校对于学生信息的管理压力也越来越大，曾几何时，对于学生信息的管理还处于人工校对、人工管理的阶段。然而由于当今时代信息技术的飞速发展，特别是计算机科学与技术的广泛应用，计算机的产生以及应用极大地提高了信息处理的能力以及效率，因此将计算机引入学生信息的管理是十分有必要的，一方面能够减少出错的几率，从而提高学生信息管理的精准程度，另一方面能够大大减轻人工劳动力的成本，从而将复杂繁琐的工作交由计算机进行处理。计算机的出现，大大提高了生产劳作的效率，减轻了人工成本，提高了数据操作的规模，对于许多复杂功能的实现也变得简单起来，将计算机引入学生信息管理系统，不仅能够提高对学生信息管理的效率，还能完成诸如选课，考试安排等复杂功能的实现。学生信息管理系统是针对学校人事处的大量业务处理工作而开发的管理软件，主要用于学校学生信息管理，总体任务是实现学生信息关系的系统化、科学化、规范化和自动化，其主要任务是用计算机对学生各种信息进行日常管理，如查询、修改、增加、删除，另外还考虑到学生选课，针对这些要求设计了学生信息管理系统。学生信息档案的管理对于学校的管理者来说至关重要，学生信息是高等学校非常重要的一项数据资源，是一个教育单位不可缺少一部分。特别是近几年来，国家政策的调整，我国高等院校大规模的扩招，给高等院校的教学管理、学生管理、后勤管理等方面都带来不少的冲击。其包含的数据量大，涉及的人员面广，而且需要及时更新，故较为复杂，难以单纯地依靠人工管理，而且传统的人工管理方式既不易于规范化，管理效率也不高，我国各类高等院校中还有相当一部分学生档案管理还停留在纸介质的基础上，尤其是中、小学对学生档案的管理更是落后，这样的管理机制已经不能适应时代发展的要求，其管理方法将浪费许多人力和物力。随着科学技术的不断提高，计算机科学与技术日渐成熟，计算机应用的普及已进入人类社会生活的各个领域，并发挥着越来越重要的作用。这种传统的手工管理模式必然被以计算机为物质基础的信息管理方法所取代。

作为计算机应用的一部分，使用计算机对学生档案进行管理，有着手工管理所无法比拟的优点，如：检索迅速、查找方便、可靠性高、存储量大、保密性好、寿命长、成本低等。这些优点能够极大地提高学生档案管理的效率，也是学校向科学化、正规化管理发展的必要条件，更是各个高等院校与世界接轨的重要条件。

---

## 2. 研究现状

现如今，几乎所有的高等院校都已经将计算机引入学生信息的管理之中，并且出现了诸如选课系统等新的使用功能，其典型案例有：“学生信息管理系统”，“数字化校园学工信息系统”，“教学管理平台”，“高校学工信息管理系统”，“曲阜师范大学教务系统”等。通过了解这些学生信息管理系统，从中抽象出学生信息管理的简单功能，对其进行模仿，使用 Java 编程语言，结合 MySQL 数据库编写一个简易的学生信息管理系统。由于市面上成熟的类似产品种类繁多，因此针对几款热门的学生管理系统，从中抽象出最普遍的本质特性，制作建议的带有 MySQL 数据库连接的学生管理系统，以此作为 MySQL 数据库练习的一个具体实例，加深对 MySQL 数据库相关操作的理解，提升对 Java JDBC 和 MySQL 数据库的操作熟练度，为今后实际开发运用奠定基础。系统针对学校学生信息的特点以及管理中实际需要而设计，能够有效地实现学生信息管理的信息化，减轻管理人员的工作负担，高效率、规范化地管理大量的学生信息，并避免人为操作的错误和不规范行为。

## 二、概要设计

### 1. 系统框架图

此为系统框架图：

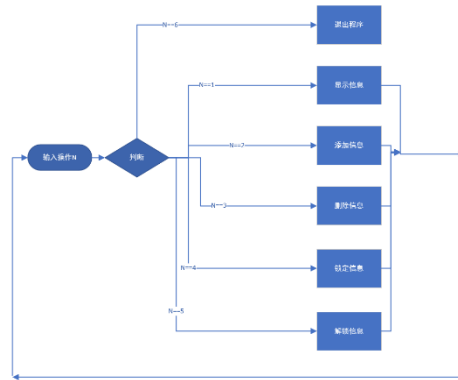
#### 学生信息管理系统框架图

- 学生信息管理系统
  - 数据操作
    - show all 显示所有的数据元素
    - add user 添加学生
    - delete user 删除学生
  - 锁操作
    - lock user 锁定学生信息
    - unlock user 解锁学生信息

### 2. 系统流程图

---

此为系统流程图：



### 三、详细设计

#### 1、显示操作

当数据库中还有数据时，进行打印输出数据的操作

#### 2、添加操作

输入要添加者的姓名，学号，锁定状态，进行 MySQL 数据库添加操作，添加学生信息，然后更新数据库信息

#### 3、删除操作

输入要删除者的姓名，进行 MySQL 数据库删除操作，删除学生信息，然后更新数据库信息

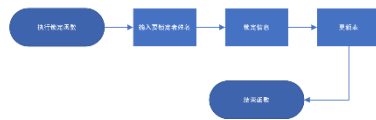
#### 4、锁定操作

输入要锁定者的姓名，进行 MySQL 数据库锁定操作，锁定学生信息，然后更新数据库信息

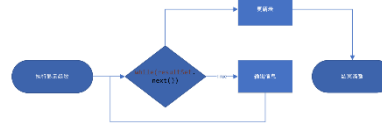
#### 5、解锁操作

输入要解锁者的姓名，进行 MySQL 数据库解锁操作，解锁学生信息，然后更新数据库信息

锁定操作



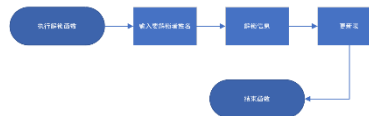
显示操作



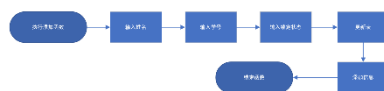
删除操作



解锁操作



添加操作



## 四、附录

核心代码：

```
package dbStudents;
```

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Scanner;
```

```
public class main {
    public static void main(String[] args) {
        Connection connection = null;
        PreparedStatement preparedStatement = null;
        ResultSet resultSet = null;
        String url = "jdbc:mysql://localhost:3306/student_manager";
        String username = "root";
        String password = "070011";
        String driver = "com.mysql.jdbc.Driver";
```

---

```
        try {
            Class.forName("com.mysql.jdbc.Driver");
            connection = DriverManager.getConnection(url, username,
password);

            System.out.println("Connect database successful !");
            String s1 = "select * from s1";
            preparedStatement = connection.prepareStatement(s1);
            resultSet = preparedStatement.executeQuery(s1);

            int choice = 0;
            Scanner scanner = new Scanner(System.in);
            do {
                System.out.println("Please Choose Your Action: ");
                System.out.println("1. show all");
                System.out.println("2. add user");
                System.out.println("3. delete user");
                System.out.println("4. lock user");
                System.out.println("5. unlock user");
                System.out.println("6. exit");
                choice = scanner.nextInt();
                switch (choice) {
                    case 1:
                        show_all(preparedStatement, resultSet);
                        break;
                    case 2:
                        add_user(connection,                preparedStatement,
resultSet);
                        break;
                    case 3:
                        delete_user(connection,                preparedStatement,
resultSet);
                        break;
                    case 4:
                        lock_user(connection,                preparedStatement,
resultSet);
                        break;
                    case 5:
                        unlock_user(connection,                preparedStatement,
```

---

```

resultSet);

        break;
    case 6:
        System.out.println("Exit the system...");
    }
} while (choice != 6);
scanner.close();

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

public static void show_all(PreparedStatement preparedStatement,
ResultSet resultSet) {
    String s1 = "select * from s1";
    try {
        resultSet = preparedStatement.executeQuery(s1);
        System.out.println("*****");
        System.out.println("number" + "\t" + "name" + "\t" + "sex"
+ "\t" + "islocked");
        System.out.println("*****");
        while (resultSet.next()) {
            System.out.println(resultSet.getInt(1) + "\t" +
resultSet.getString(2) + "\t" + resultSet.getString(3)
+ "\t" + resultSet.getInt(4));

System.out.println("*****");
        }
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
}

```

---

```
public static void add_user(Connection connection,
PreparedStatement preparedStatement, ResultSet resultSet) {

    String s2 = "insert into s1 values(?, ?, ?, ?)";

    Scanner scanner = new Scanner(System.in);
    System.out.println("Please enter the number of the student you
want to add :");
    int num = scanner.nextInt();
    System.out.println("Please enter the name of the student you
want to add :");
    String name = scanner.next();
    System.out.println("Please enter the sex of the student you
want to add :");
    String sex = scanner.next();
    System.out.println("Please enter the islocked of the student
you want to add :");
    int islocked = scanner.nextInt();

    try {
        preparedStatement = connection.prepareStatement(s2);
        preparedStatement.setInt(1, num);
        preparedStatement.setString(2, name);
        preparedStatement.setString(3, sex);
        preparedStatement.setInt(4, islocked);
        preparedStatement.execute();
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```

```
public static void delete_user(Connection connection,
PreparedStatement preparedStatement, ResultSet resultSet) {
    String s3 = "delete from s1 where name=?";
    try {
        Scanner scanner = new Scanner(System.in);
```

---

```
        System.out.println("Please enter the name of the student
you want to add :");
        String name = scanner.next();
        preparedStatement = connection.prepareStatement(s3);
        preparedStatement.setString(1, name);
        preparedStatement.executeUpdate();
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```

```
    public static void lock_user(Connection connection,
PreparedStatement preparedStatement, ResultSet resultSet) {
        System.out.println("Please enter the name of the student you
want to lock :");
        String s4 = "update s1 set islocked=? where name=?";
        try {
            Scanner scanner = new Scanner(System.in);
            String name = scanner.next();
            preparedStatement = connection.prepareStatement(s4);
            preparedStatement.setInt(1, 1);
            preparedStatement.setString(2, name);
            preparedStatement.executeUpdate();
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```

```
    public static void unlock_user(Connection connection,
PreparedStatement preparedStatement, ResultSet resultSet) {
        System.out.println("Please enter the name of the student you
want to unlock :");
        String s4 = "update s1 set islocked=? where name=?";
        try {
            Scanner scanner = new Scanner(System.in);
            String name = scanner.next();
```



---

```
        preparedStatement = connection.prepareStatement(s4);
        preparedStatement.setInt(1, 0);
        preparedStatement.setString(2, name);
        preparedStatement.executeUpdate();
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

}

}
```