北京航空航天大学

2021－2022 学年 第二学期

《大学计算机基础（理科）》

大作业实验报告

班 级\_\_217720\_\_\_\_\_\_学 号 21377266\_

姓 名\_\_\_\_逯化\_\_\_\_\_\_\_\_成 绩 \_\_\_\_\_\_\_\_\_

2022年7月

**题目 2：简易成绩管理系统**

1. 实验目的

本题主要考察GUI设计，数据库开发，数据可视化，程序打包。GUI使用了PyQt5，数据库使用的是Sqlite3，数据可视化使用了pyqtgraph，程序打包为exe使用了Pyinstaller

二、实验任务

根据大作业发布的题目，实验任务如下：

搭建一个具有 GUI 界面的成绩管理系统，主要功能如下:

1. 教师/学生用户的注册以及登录

2. 教师用户可以在其对应的界面中：

（1）对学生成绩进行增、删、改、查操作；

（2）导出成绩为不同格式文件

（3）生成可视化的成绩图表

（4）实现数据统计分析

3. 学生用户可以在其对应的界面中查询自己的成绩和排名情况

4. 系统中的数据，如已注册的教师或学生账户以及已录入的成绩等需要使用数据库在本地文件保存

三、总体设计方案

#### 第一部分为GUI

（1）GUI界面的设计，运用了QtWidgets中的各种控件设计GUI布局。创建QMainWindow用来连接各部分的部件和包括垂直布局、水平布局等布局。

（2）为GUI界面设置信号与槽和按钮和连接

#### 第二部分为数据库

（1）建立数据库储存账号信息和考试成绩信息

（2）创建函数处理输入的信息，并储存到数据库的数据表中。读取数据表并展示在GUI中

#### 第三部分为数据分析

（1）对数据库中的成绩数据分析并生成报告。

（2）用pyqtgraph对数据库中的数据进行可视化处理

四、设计思路和关键语法

（一）首先注册和登录账号功能，需要用数据库储存。关键点有：

1. 账号需要用学工号，也就是全数字，用isdigit()方法进行判断
2. 用数据库储存密码明文的安全性太低，可以储存密码的md5值，增加安全性。节选代码如下：

import hashlib

md5pw=hashlib.md5(self.pwd\_line.text().encode('utf8')).hexdigest()

1. 账号密码错误时需要提示，这里运用QMessageBox.critical创建消息框

（二）

1. 读取和写入数据库的数据都需要连接数据库和创建指针，代码如下

        self.conn = sqlite3.connect('test.db')

        self.c = self.conn.cursor()

调用sqlite3模块，用来连接数据库。

        self.conn.commit()

        self.conn.close()

调用结束后关闭数据库连接。

2. 以下为python向数据库发送的一条语句，用来创建一个新数据表，对应新建一次考试，可以接收新建考试的名称，并初始化数据表储存数据的结构和要求。

self.c.execute('''CREATE TABLE {}

    (ID INTEGER PRIMARY KEY AUTOINCREMENT,

    CODE INT NOT NULL UNIQUE,

    CLASS INT NOT NULL,

    INRANK INT,

    OUTRANK INT,

    TOTAL INT NOT NULL,

    K INT NOT NULL,

    D INT NOT NULL,

    A INT NOT NULL);'''.format(self.name))

以下为一行语句，向数据库的特定数据表中插入一条成绩信息。

 self.c.execute('INSERT INTO {} (CODE, CLASS, K, D, A, TOTAL) VALUES ({}, {})'.format(self.name, ','.join(li), sum(list(map(int, li[2:])))))

（三）GUI

1. 设计GUI在整体上需要设计布局管理，以下为各页面采用的水平垂直混合布局的关键语法：

        self.all\_v\_layout.addLayout(self.user\_h\_layout)

        self.all\_v\_layout.addLayout(self.pwd\_h\_layout)

        self.all\_v\_layout.addLayout(self.button\_h\_layout)

        self.setLayout(self.all\_v\_layout)

将单个控件加入到一个layout中用addWidget方法，而将layout再进行整合需要用addLayout方法。

2. GUI中有很多按钮和文本框用来交互，需要配置信号和槽。

 self.student\_bt.clicked.connect(lambda:self.login\_func(0))

 self.teacher\_bt.clicked.connect(lambda:self.login\_func(1))

例如此处，将按钮的点击信号与login\_func()函数连接，由于函数需要传入参数，所以需要使用lambda方法。

3. 主页使用的时QMainWindow类，可以创建菜单栏，工具栏，状态栏。而其他布置可以用setCentralWidget方法展示在主页中。需要注意的一点，setCentralWidget方法可以接收QDialog类但不可以接收QWidget类。

      self.account\_menu = self.menuBar().addMenu('Account')

      self.account\_menu.addAction(self.login\_action

      self.login\_action.triggered.connect(self.new\_func)

以上语句为创建菜单栏中的项和子项的方法，本例有一个项和一个子项，子项的信号连接了一个函数。

1. py文件根据内容不同，所需的打开方式也会不同，启动文件需要作为主文件来启动，而其他文件应该等待被调用，于是需要一段代码控制不同启动方式的执行效果。代码如下

*if* \_\_name\_\_ == '\_\_main\_\_':

  app = QApplication(sys.argv)

    demo = Demo()

    demo.show()

    sys.exit(app.exec\_())

这一段代码只有作为主文件启动时才会运行，当被其他文件调用时会直接跳过。这就能防止文件互相调用导致的程序混乱。

1. 在老师录入成绩界面，由于每次录入的学生数量并不确定，需要有能随时增加录入数据条数的功能，见以下代码：

self.nbuton = QPushButton('new line')

self.nbuton.clicked.connect(self.new\_line)

def new\_line(self):

        self.insertRow(self.rowCount()-1)

self.setSpan(self.rowCount()-1, 0, 1, self.columnCount())

        self.setCellWidget(self.rowCount()-1, 0, self.nbuton)

以上代码的效果是在成绩录入表格界面的下方创建一个按钮，连接到一个函数，该函数可以在现有可输入的最后一行的下方再创建一行。于是可以得到足够的行数。

1. 成绩录入界面和成绩查询界面都需要方便地把多余的数据删除掉，以下代码提供了删除按钮，可以一键删除掉单选或多选的行。这段代码的关键在于dlt函数的定义，需要完整且不超出地删除选中的行并且保证不发生行数迭代。

self.delete\_button = QPushButton('Delete')

self.h\_layout.addWidget(self.delete\_button)

self.table = Demo\_table()

self.delete\_button.clicked.connect(self.table.dlt)

#上方代码在Demo(QWidget)类中，下方代码在Demo\_table(QTableWidget)

类中

ef dlt(self):

        s\_items = self.selectedIndexes()

*if* s\_items:

            selected\_rows = []

*for* i *in* s\_items:

                row = i.row()

*if* row not in selected\_rows and row != self.rowCount()-1:

                    selected\_rows.append(row)

*for* r *in* range(len(sorted(selected\_rows))):

                    self.removeRow(selected\_rows[r]-r)

（四）数据可视化

1. 首先需要读取将要进行数据化的数据，由于数据由用户选择，所有在GUI中获取数据更为妥当。见以下代码：

    def graph(self):

        self.values = []

        self.col = self.currentColumn()

*for* i *in* range(self.rowCount()):

            self.values.append(int(self.item(i, self.col).text()))

        print(self.values)

        self.plotw = PlotWidget()

        self.plotw.plot(self.values, pen='r', symbol='o')

        self.plotw.show()

这段代码定义了graph函数，这个函数绑定了一个按钮，可以由用户选择竖轴的变量一键生成可视化曲线。同时这个函数依赖了如下的模块

from pyqtgraph import PlotWidget

2. 导出为xlsx。导出功能的大部分由下面的函数实现。利用了xlsxwriter模块，这个模块可以较为方便快捷地将python中的各种数据导出为xlsx文件，并且可以高度定制格式和结构

 def xls(self):

        sheetname = self.exam\_combo.currentText()

        workbook = Workbook('{}.xlsx'.format(sheetname))

        worksheet = workbook.add\_worksheet()

        mysel = self.c.execute('select \* from {}'.format(sheetname))

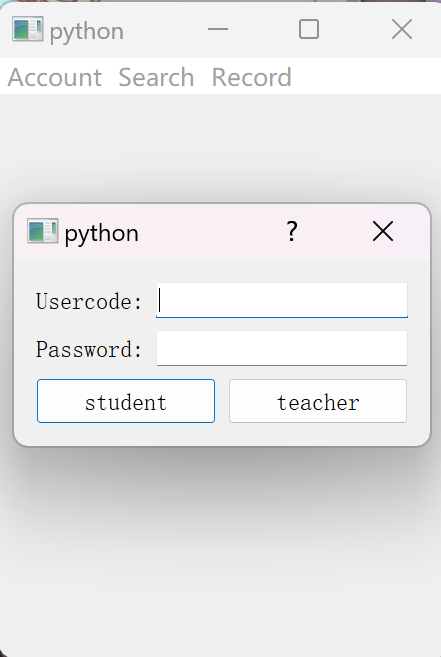
*for* i, row *in* enumerate(mysel):

*for* j, value *in* enumerate(row):

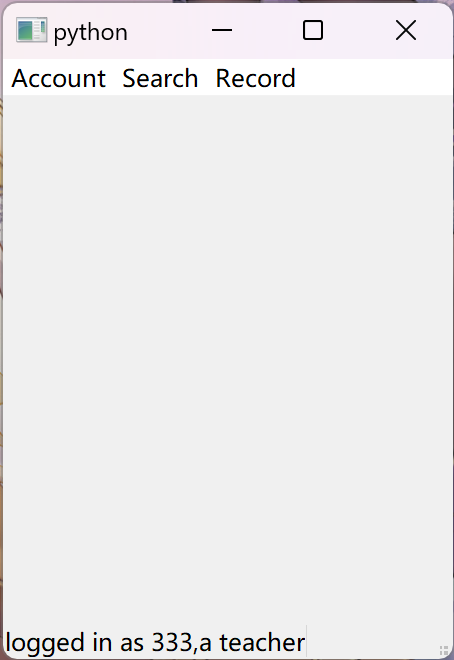
                worksheet.write(i, j, value)

        workbook.close()

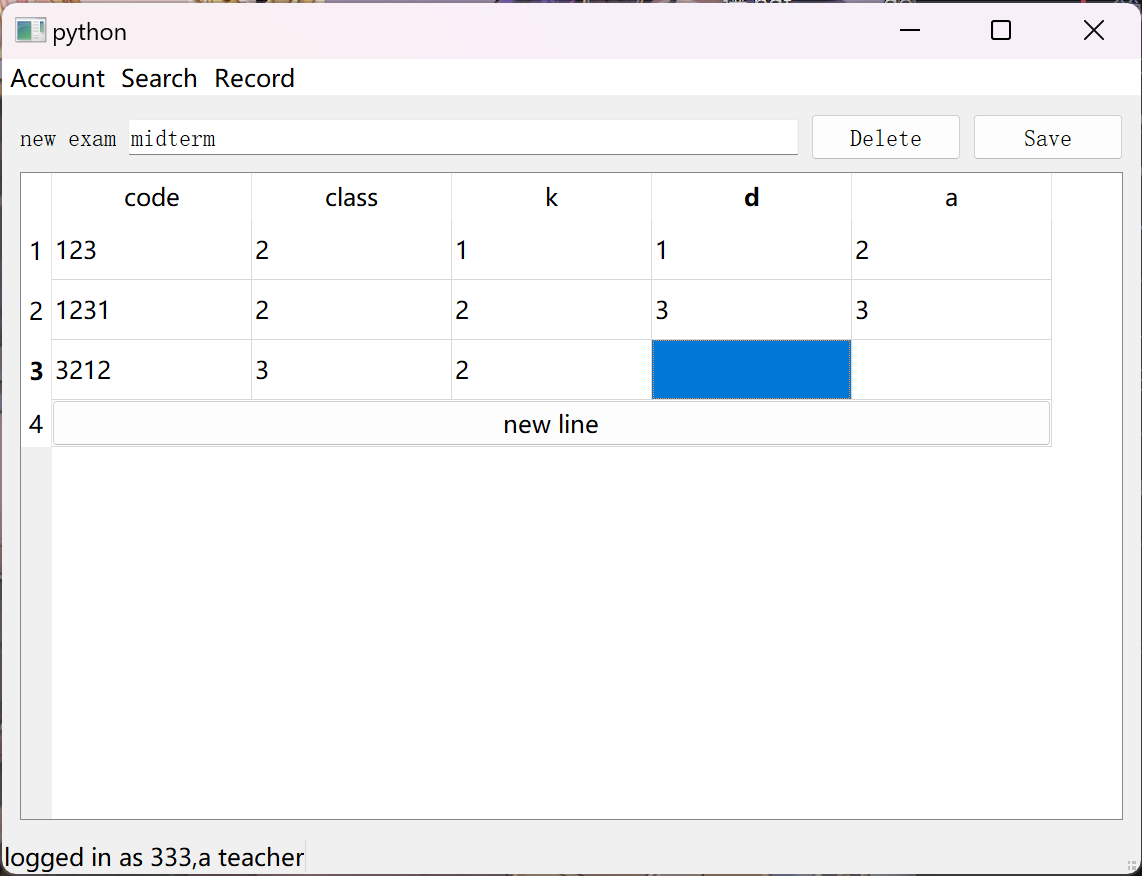
五、程序运行效果

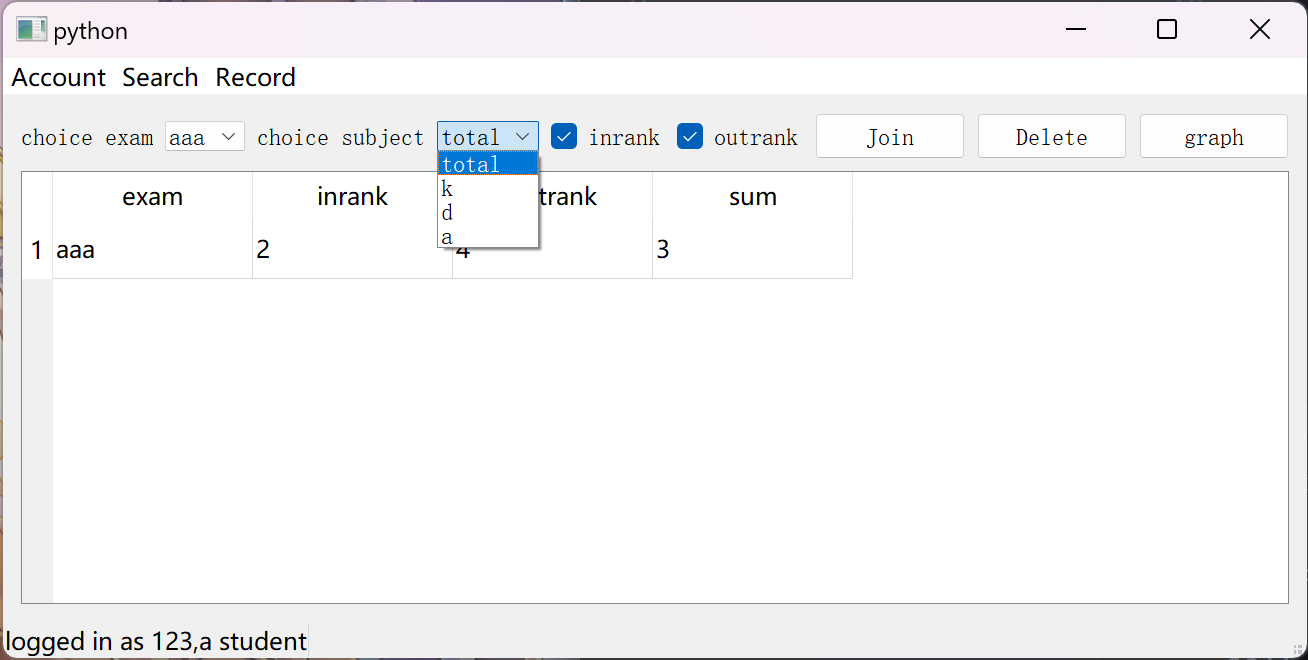


上图为登录页面 ，可以选择学生账号登录或老师账号登录。登录后在窗口下方状态栏有提示，如下图

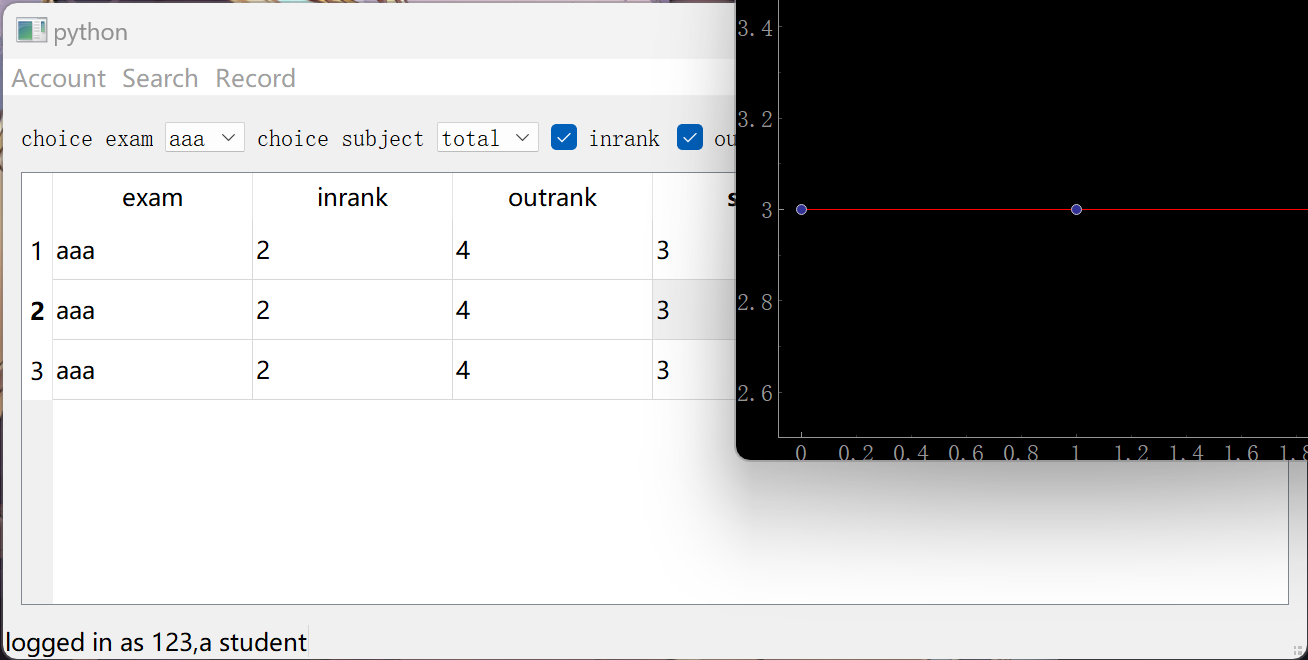


下图为教师录入成绩的页面，可以添加行数或者批量删除已输入的行。可以输入学生的学号和班级用以辨别身份

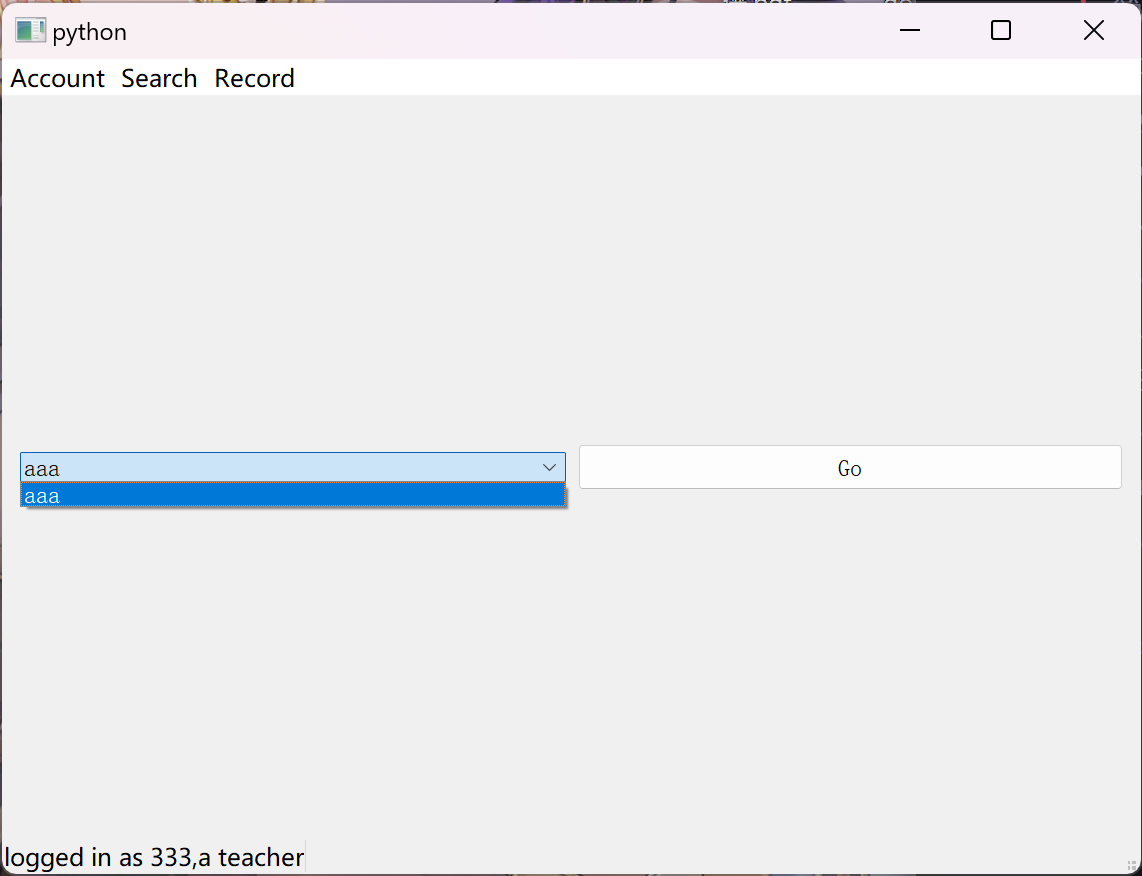




上图是学生查询成绩的页面，可以选择查询的科目，可选是否显示班内或全体的排名。选择考试框的选项通过读取数据库中的数据表名可同步数据更新。下图展示了由用户选择数据并通过“graph”键一键生成了关于“s”的图像。



下图为教师导出成绩单为xlsx的页面，与上文相同，选择考试的框可以同步更新录入考试。



六、创新之处

1. 在课上学习的是Tkinter设计GUI，而本次大作业的GUI是由PyQt5设计的。PyQt5对比Tkinter有很多优势和方便易用的布局功能。

2. 在本课的实验课时有读取写入csv和xls文件的实验。而本次实验使用的储存数据的工具是数据库。数据库相比于前两种文件有访问更快，更稳定，储存内容多样化，格式结构多样化等多种优点。

3. 在储存密码和登录检验密码时，相比于网上教程和直觉上的明文储存，本项目选择了更安全而且在互联网各大网站也有实际应用的md5值加密储存。可以防止各种原因导致的密码泄露，大大提高安全系数。

七、实验总结

1、在本实验中你遇到了哪些问题？是如何解决的？

创建了多个页面，而页面之间难以统一显示在一起，可能需要分别运行不同的py文件。

解决方法：使用了QMainWindow类，这一类能够展示子Widget与窗口中，还能添加常驻的菜单栏，解决了上述问题。代码节选如下：

class Demo(QMainWindow):

    def \_\_init\_\_(self):

        super(Demo, self).\_\_init\_\_()

        self.account\_menu = self.menuBar().addMenu('Account')

        self.search\_menu = self.menuBar().addMenu('Search')

        self.record\_menu = self.menuBar().addMenu('Record')

……

    def xls\_func(self):

        self.ox\_ = ox()

        self.setCentralWidget(self.ox\_)

……

    def record\_func(self):

*if* self.iden[1]==1:

            self.tr\_ = tr()

            self.setCentralWidget(self.tr\_)

以上为创建菜单栏项以及传入并展示子页面的语句

2、有何收获和体会？你认为大作业题目还可以从哪些角度出题？并说明理由。

收获了PyQt5，Sqlite， Python等工具的技能技巧，体会到了DDL的恐怖。大作业可以出些联网的题。因为北航的校园网很干扰普通的网络编程，希望给一些tips帮助解决难题。

八、课程学习总结

**1、课程收获和难点分析**

通过学习本课程你收获了什么？你认为现有内容中的难点在于哪些章节？

学到了计算机的基础原理，和python的使用方法。

难点毫无疑问在于动态规划相关的章节。

**2、课程评价**

从课程体系的先进性和合理性、课程内容、实验内容、考核环节、课程管理等方面进行评价。

课程体系先进领先大多数其他课程，课程安排合理，每周上机课完美影响了晚饭时间，课程内容前期很水，后期突然加速毫无预兆，按照初期的节奏完全跟不上了。实验内容虽然贴近生活但是其实还未派上用场，涉及的知识点很多，可以有效地复习巩固。考核环节有很多的报告提交比如现在这个，写报告占据了大部分时间，能否改换其他比打字更高效的方式用来沉淀吸收、课程管理严谨，从不会错过通知信息，但是期末突然通知让补mooc的分数实在是不敢恭维，希望好好管管。

**3、教师授课评价**

老师工作态度积极主动、有责任心愿意帮助学生、备课情况很明显十分充足、讲课水平高超，一点就通、教学质量很高，线上课线下课结合，充分利用了北航的平台优势。

**4、助教评价**

助教工作态度认真有责任心、在实验课上积极耐心辅导、批改作业和反馈都很及时、解答问题也很专业。

**5、课程进一步改进建议**

大作业太赶了，根本没法好好做，尤其是大计基期末的考试又尤其早，两者割裂太大，想不出来这是一门课的两大考核。

九、主要参考资料和用到的库或工具

班级群发的大作业实验指导