

# Python Project - Training Institution Management System

💡 This document is translated from Chinese. Some of the images and presentations are still in Chinese. If you have any questions, please feel free to ask 张子博 🙏

## 👁️ Goal:

Practice coding with object-oriented programming concepts.

## 📝 Process

### Training Institution Management System

Use **object-oriented** thinking to design a training institution management system and design a console interaction system.

### Code details requirements

- Given the scale of the system, only parts of the logic are shown below. Students only need to achieve the following **Requirements**.
- Focus on the big picture of the system instead of minor details, such as functions like phone number format validation.
- All the following requirements are merely suggestions, so you are free to add or delete content according to your understanding.

### Requirements

As the system administrator, the following operations can be performed:

1. Headquarters and Branches Class

- a. Set up a headquarters and three branches.
  - b. View the number of employees, number of teachers, and number of students in any branch.
  - c. View the revenue of any one of the branches.
2. Student Class
    - a. Fill in student information (name, phone number, branch, student ID), join the class, and pay tuition.
    - b. View the class and branch the student belongs to, and the course the student is participating in.
    - c. Student drop-out.
    - d. View the personal information of the student (name, phone number).
3. Teacher Class
    - a. Add basic information for new teachers (name, phone number, branch, employee number).
    - b. Set the courses taught by the teacher (multiple courses are possible, please display the list of teaching teachers for user selection).
    - c. View the basic information of the teacher.
    - d. View the branch where the teacher is located, and the courses taught.
4. Course Class
    - a. Add a new course, input basic information (name, price, branch).
    - b. Assign a teacher to the course (**one**, please display the list of teachers for user selection).
    - c. View the student list of the course.
5. Employee Class
    - a. Employees are of three types: logistics, finance, and administration.
    - b. Add new employees, and you need to specify the employee type (using **inheritance**) when adding, including the personal information of the employee (name, employee number, branch).
    - c. Display employee information.

## Document Writing

- Write a **Lark(飞书/Feishu) cloud document** to explain the functions and implementation details. The content of the document includes:

- Implemented functions (use mind maps or other forms).
- Object-oriented design details (such as showing inheritance relationships with class diagrams).
- Anything else you want to show.
- The document doesn't need to be very long, but it has to include everything above.

## Tips:

In the cloud document, you can insert codes, mind maps, UML diagrams, and other content. Please make good use of it~



- 交互!
- 面向!
- 测试!
- 其他!
- 文档!

提示:  
在云文档中

提交

1. 实验代码
2. 实验截图
3. 实验报告

提交

作业提交系:

## Things to submit

1. Experiment code, naming is not limited.

2. **Experiment report**, to be submitted in the form of a **Lark(飞书/Feishu) cloud document**, naming is not limited.



## The way to submit

We use Lark(飞书/Feishu) to submit assignments. Please click the link and fill in the form.

<https://nankai.feishu.cn/share/base/form/shrcnn40xc30G2pYdHc3geLd85e>



### Submission Notes:

- Submit individual assignment files one by one, or select multiple files locally and then submit, there is no need to submit a compressed file.
- When submitting assignments multiple times, please indicate in the remarks that this is a multiple submission, otherwise, it may cause an abnormality in the score.