项目说明文档

数据结构课程设计

——排课软件

作 者 姓 名： 赵子昱

学 号： 1951459

指 导 教 师： 张颖

学院、 专业： 软件学院 软件工程

同济大学

Tongji University

目 录

[1 分析 1](#_Toc59276482)

[1.1 背景分析 1](#_Toc59276483)

[1.2 功能分析 1](#_Toc59276484)

[2 设计 2](#_Toc59276485)

[2.1 数据结构设计 2](#_Toc59276486)

[2.2 类结构设计 2](#_Toc59276487)

[2.3 成员与操作设计 2](#_Toc59276488)

[2.3.1 **Course**类 2](#_Toc59276489)

[2.3.1 **ScheduleSystem**类 3](#_Toc59276490)

[2.4 系统设计 4](#_Toc59276491)

[3 实现 4](#_Toc59276492)

[3.1 **Input()**输入功能的实现 4](#_Toc59276493)

[3.1.1 输入功能流程图 5](#_Toc59276494)

[3.1.2 输入功能代码 6](#_Toc59276495)

[3.1.3 输入功能截屏示例 7](#_Toc59276496)

[3.2 **SelectNCourses(…)**选择课程功能的实现 8](#_Toc59276497)

[3.2.1 选择课程功能流程图 9](#_Toc59276498)

[3.2.2 选择课程功能核心代码 10](#_Toc59276499)

[3.3 **One(...)**搜索一节小课功能的实现 11](#_Toc59276500)

[3.3.1 搜索一节小课功能流程图 11](#_Toc59276501)

[3.3.2 搜索一节小课功能核心代码 11](#_Toc59276502)

[3.4 **Two(...)**搜索两节小课功能的实现 13](#_Toc59276503)

[3.4.1 搜索两节小课功能流程图 13](#_Toc59276504)

[3.4.2 搜索两节小课功能核心代码 14](#_Toc59276505)

[3.5 **Three(...)**搜索三节小课功能的实现 15](#_Toc59276506)

[3.5.1 搜索三节小课功能流程图 15](#_Toc59276507)

[3.5.2 搜索三节小课功能核心代码 16](#_Toc59276508)

[3.6 **PaikeOneTerm (...)**排课（单学期）功能的实现 16](#_Toc59276509)

[3.6.1 排课（单学期）功能流程图 17](#_Toc59276510)

[3.6.2 排课（单学期）功能核心代码 18](#_Toc59276511)

[3.7 **Clear(...)**清空课表功能的实现 19](#_Toc59276512)

[3.7.1 清空课表功能的代码 19](#_Toc59276513)

[3.8 **WriteSchedule (...)**将课程表写入文件功能的实现 20](#_Toc59276514)

[3.8.1 将课程表写入文件功能 20](#_Toc59276515)

[3.9 **Paike ()**排课（全8学期）功能的实现 21](#_Toc59276516)

[3.9.1 将排课（全8学期）功能的代码 21](#_Toc59276517)

[3.10 主函数的实现 22](#_Toc59276518)

[3.10.1 主函数的代码 22](#_Toc59276519)

[3.10.1 主函数运行截屏 23](#_Toc59276520)

[4 测试 24](#_Toc59276521)

[4.1 功能测试 24](#_Toc59276522)

[4.1.1 排课系统功能测试 24](#_Toc59276523)

[4.2 边界测试 28](#_Toc59276524)

[4.2.1 课程数为0 28](#_Toc59276525)

[4.2.2 前几学期课程尽量多 28](#_Toc59276526)

[4.3 出错测试 31](#_Toc59276527)

[4.3.1 课程总数与每学期课程数的和不等 31](#_Toc59276528)

[4.3.2 课程数小于零 31](#_Toc59276529)

[4.3.3 当前学期课程数过多（不满足先修条件） 31](#_Toc59276530)

[4.3.4 当前学期课程数过多（课时数过多） 32](#_Toc59276531)

[4.3.5 当前学期课程数过少（少于该学期指定课程的数量） 32](#_Toc59276532)

# 1 分析

## 1.1 背景分析

排课系统是一个学校不可缺少的部分，它对于学校的管理者和学生来说都至关重要，所以一个良好的排课系统应该能够为用户提供充足的信息和功能。排课系统对于学校加强课程管理有极其重要的作用。随着学生数量和课程数量的日益庞大，如何管理如此庞大的数据显得极为复杂，传统的手工管理工作量大且容易出错。

随着计算机科学技术的不断成熟，使用计算机对排课系统进行管理，具有手工管理所无法比拟的优势。这些优点能够极大地提高学校和学生的效率，也是学校走向信息化、科学化、国际化的重要条件。因此，开发一套排课系统具有十分重要的意义。

## 1.2 功能分析

作为一个排课系统，首先应该有的功能就是输入课程信息并且可以予以保存，如果输入不合理给予提示。排课报名系统最重要的功能是排课，排课的过程应保证满足先修安排，优先满足间隔上课，课程过多给与错误提示，课表写入文件等功能。

# 2 设计

## 2.1 数据结构设计

每门课程作为一个Course对象保存，储存课程的相关信息，在排课系统类ScheduleSystem中包含一个元素为Course的动态数组来存储系统的课程信息。课表信息用一个int[5][2][5]的三维数组来存储，数组的每个元素存储课程的下标，节省了存储空间。

## 2.2 类结构设计

项目中定义了课程类（**Course**）、排课系统类（**ScheduleSystem**），**ScheduleSystem**作为**Course**的友元可以访问**Course**的私有成员。此外还用了**Stack**和**Queue**来作为辅助存储。

## 2.3 成员与操作设计

### 2.3.1 **Course**类



### 2.3.1 **ScheduleSystem**类



## 2.4 系统设计

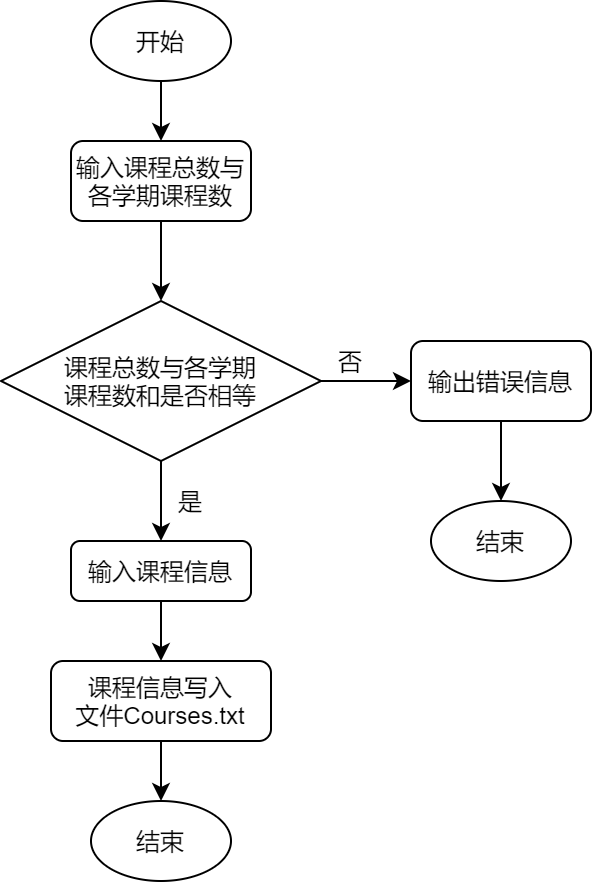
系统首先调用**Input()**函数让用户输入课程数据，若输入无误课程数据保存至Course.txt，完成课程的构建。若课程数据构建成功，则调用**Paike()** 进行排课，若排课成功将每学期的课程数据写入文件CoursesByTerm.tx**t，**将每学期的课程表写入文件Schedule.txt。

# 3 实现

## 3.1 **Input()**输入功能的实现

* **Input()**函数的功能是输入课程并将所有课程写入文件Course.txt
* 输入课程时要求每行以0结尾
* 因为txt的显示问题，文件中暂时还存在对不齐的问题，但我认为这无关紧要，无伤大雅

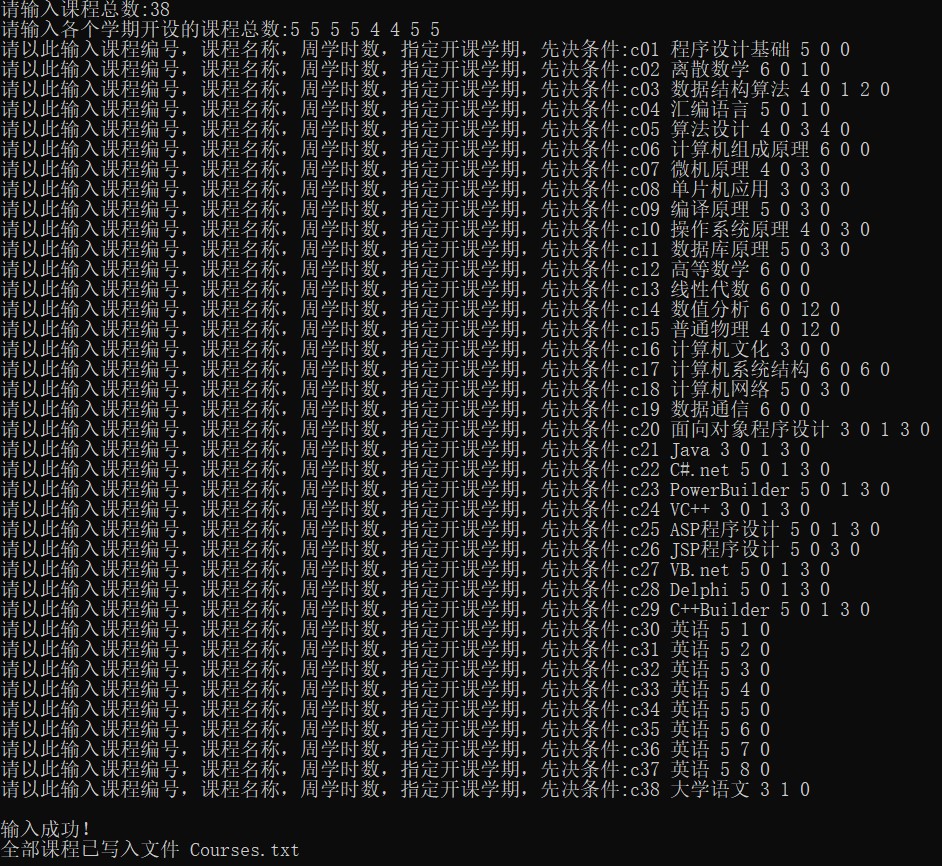
### 3.1.1 输入功能流程图



### 3.1.2 输入功能代码



### 3.1.3 输入功能截屏示例



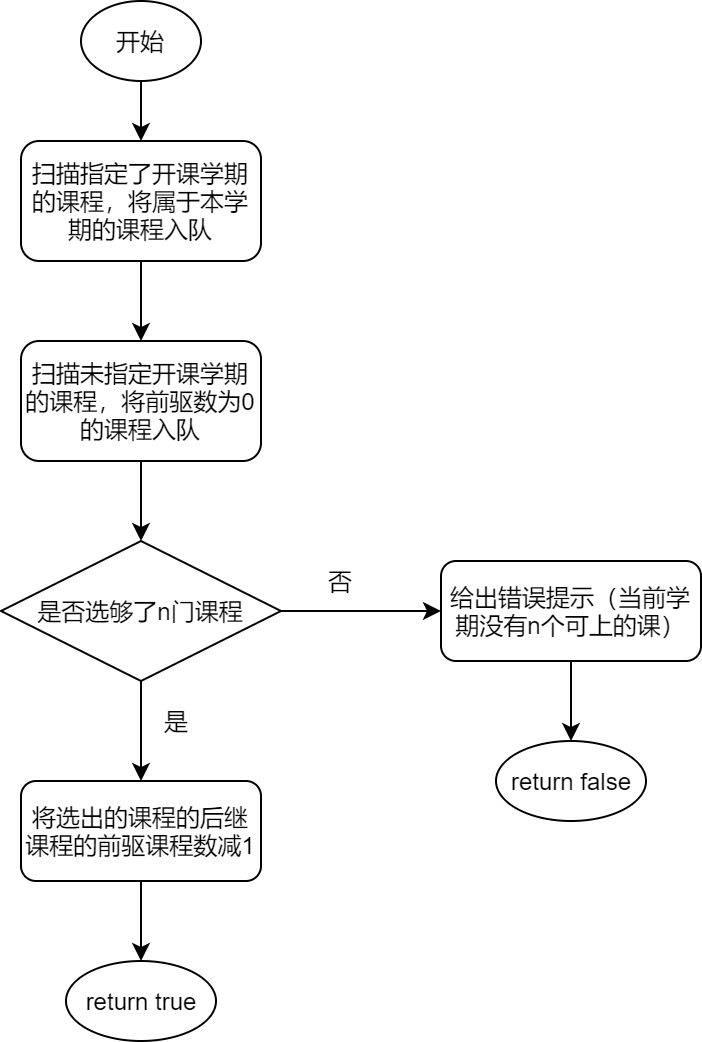
Course.txt截图：



## 3.2 **SelectNCourses(…)**选择课程功能的实现

* 选择n个本学期能上的课（先修课程在本学期前修过）
* 与拓扑排序类似，但不完全相同，选择后不立即将课程的前驱数减1，而是遍历完一轮后再减1
* 能上的课程数小于n时给出错误提示
* 参数currentTrem为当前学期数，n为要选择的课程数，队列currentCourse保存选出的课程
* 被函数PaikeOneTerm(…)调用

### 3.2.1 选择课程功能流程图



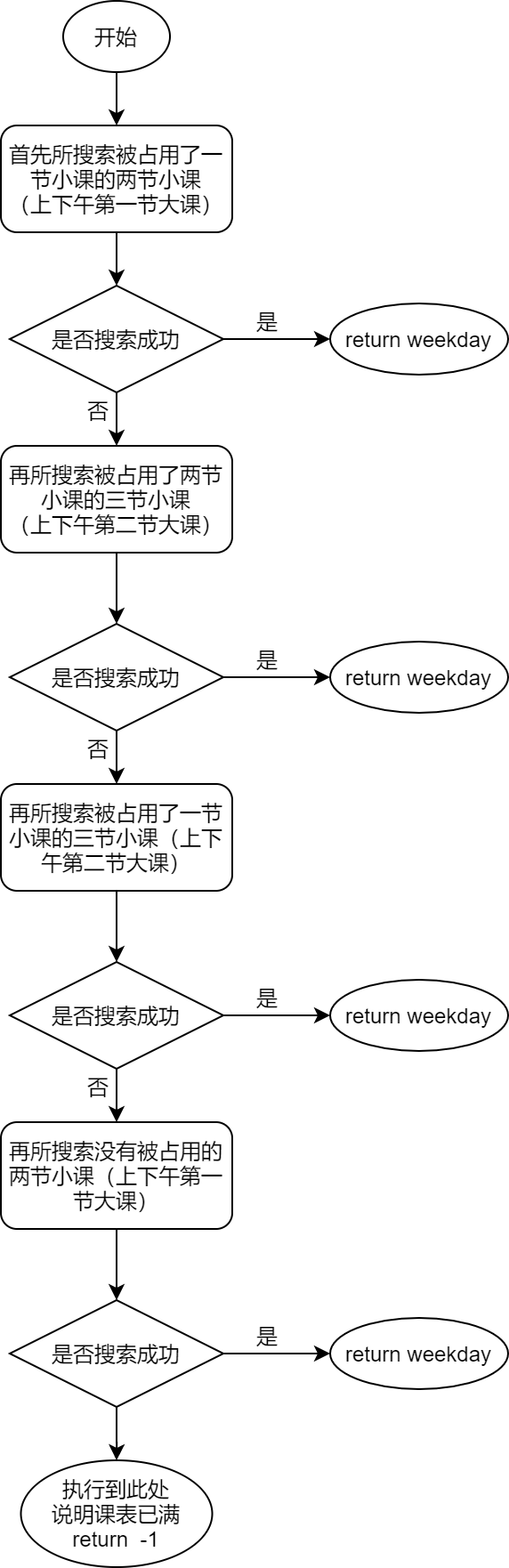
### 3.2.2 选择课程功能核心代码



## 3.3 **One(...)**搜索一节小课功能的实现

* 从星期start开始，在课程表中找到一小节课，并填入CourseNum课程，返回填入的最后一节课在星期几
* 主要思路是尽量不拆整节的大课，尽量利用零星的小课
* 参数start为起始星期数，CourseNum为要填入的课程下标，Schedule为课程表
* 被Three(…)函数调用

### 3.3.1 搜索一节小课功能流程图



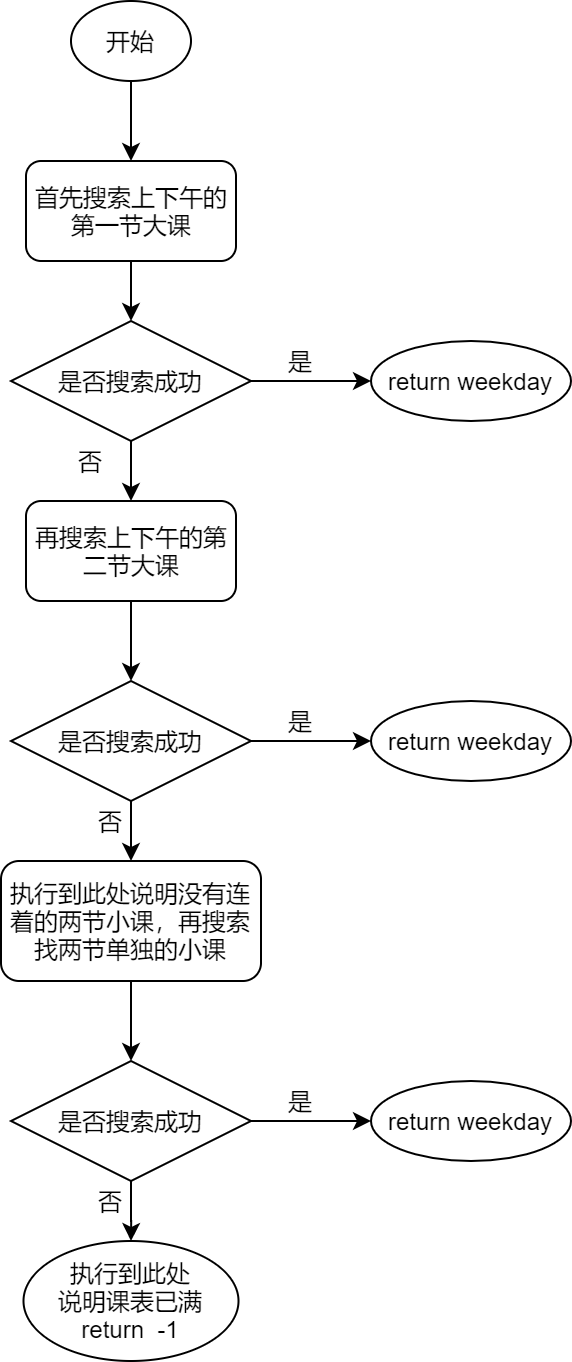
### 3.3.2 搜索一节小课功能核心代码



## 3.4 **Two(...)**搜索两节小课功能的实现

* 从星期start开始，在课程表中找到两小节课，并填入CourseNum课程，返回填入的最后一节课在星期几
* 主要思路是尽量不拆整节的大课，尽量利用零星的小课
* 参数start为起始星期数，CourseNum为要填入的课程下标，Schedule为课程表
* 被Three(…)、PaikeOneTerm (…)函数调用

### 3.4.1 搜索两节小课功能流程图



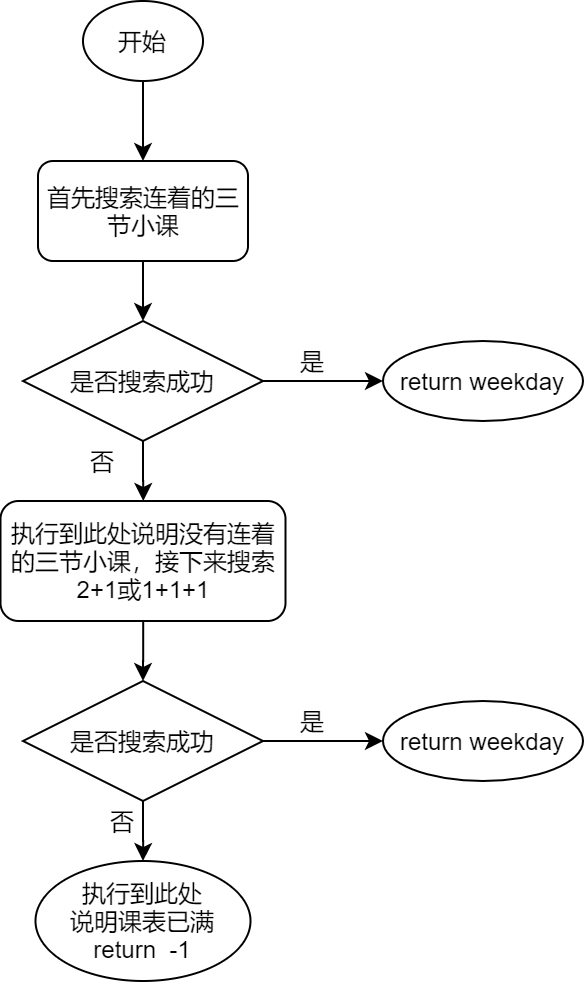
### 3.4.2 搜索两节小课功能核心代码



## 3.5 **Three(...)**搜索三节小课功能的实现

* 从星期start开始，在课程表中找到三小节课，并填入CourseNum课程，返回填入的最后一节课在星期几
* 主要思路是尽量不拆整节的大课，尽量利用零星的小课
* 参数start为起始星期数，CourseNum为要填入的课程下标，Schedule为课程表
* 被PaikeOneTerm (…)函数调用

### 3.5.1 搜索三节小课功能流程图



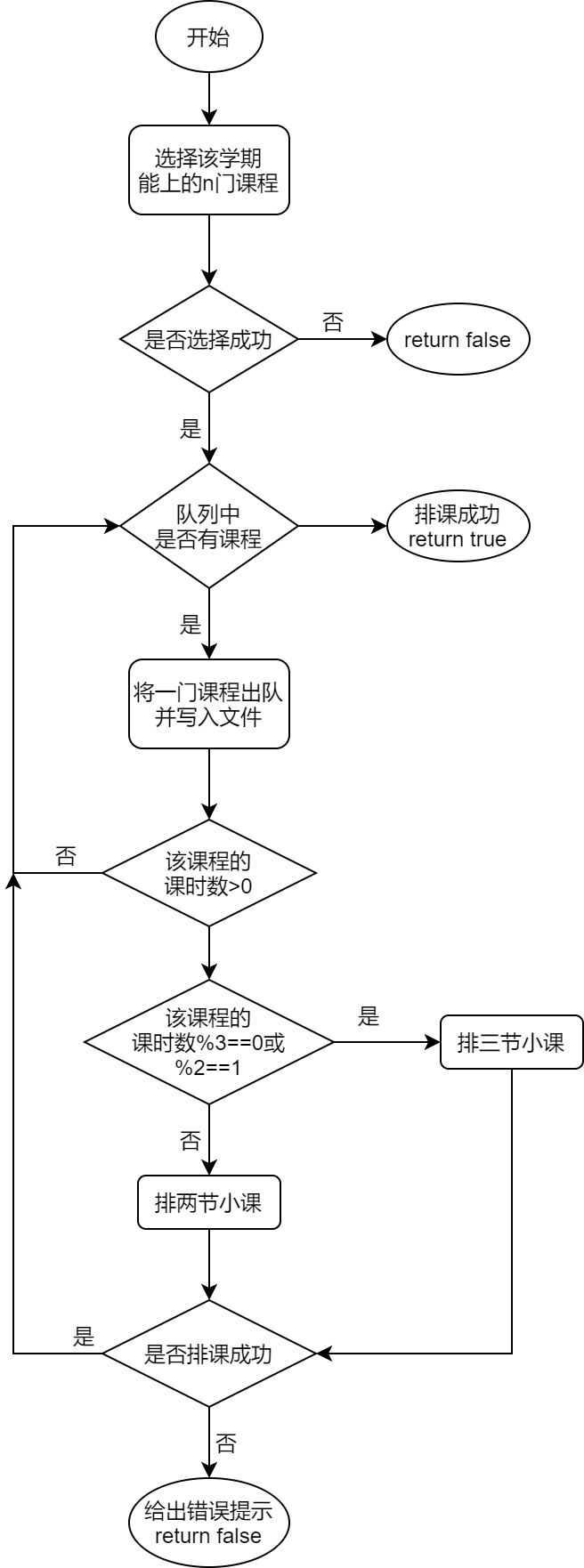
### 3.5.2 搜索三节小课功能核心代码



## 3.6 **PaikeOneTerm (...)**排课（单学期）功能的实现

* 排课（第TermNum学期），并将该学期的课程写入文件CoursesByTerm.txt
* 参数TermNum为学期数，Schedule为课程表
* 优先排三节课还是两节课的策略是本函数的关键
* 返回是否排课成功（bool）
* 被Paike ()函数调用

### 3.6.1 排课（单学期）功能流程图



### 3.6.2 排课（单学期）功能核心代码



## 3.7 **Clear(...)**清空课表功能的实现

* 被Paike ()函数调用

### 3.7.1 清空课表功能的代码



## 3.8 **WriteSchedule (...)**将课程表写入文件功能的实现

* 参数TermNum为学期数，Schedule为课表
* 被Paike ()函数调用

### 3.8.1 将课程表写入文件功能



## 3.9 **Paike ()**排课（全8学期）功能的实现

* 主函数中调用此函数

### 3.9.1 将排课（全8学期）功能的代码



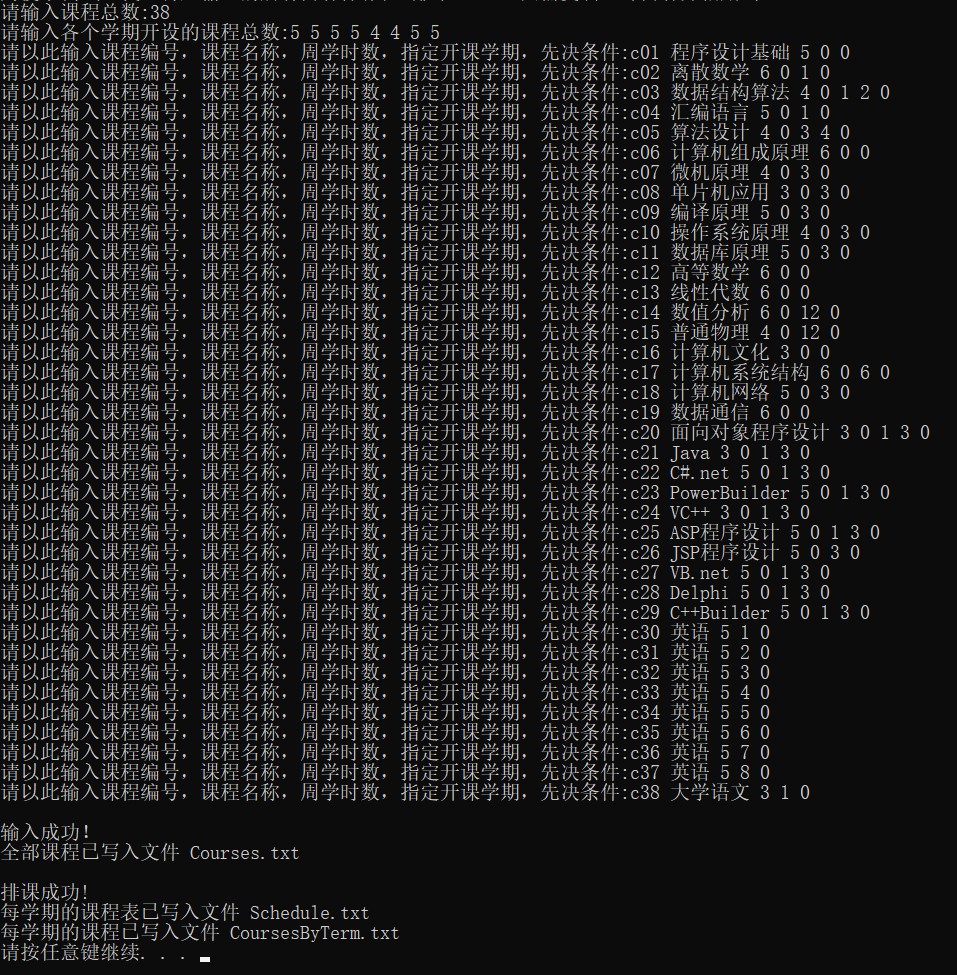
## 3.10 主函数的实现

* 主函数创建了一个ScheduleSystem类，并调用了Input()和Paike()函数
* 加入了一些错误提示

### 3.10.1 主函数的代码



### 3.10.1 主函数运行截屏



# 4 测试

## 4.1 功能测试

### 4.1.1 排课系统功能测试

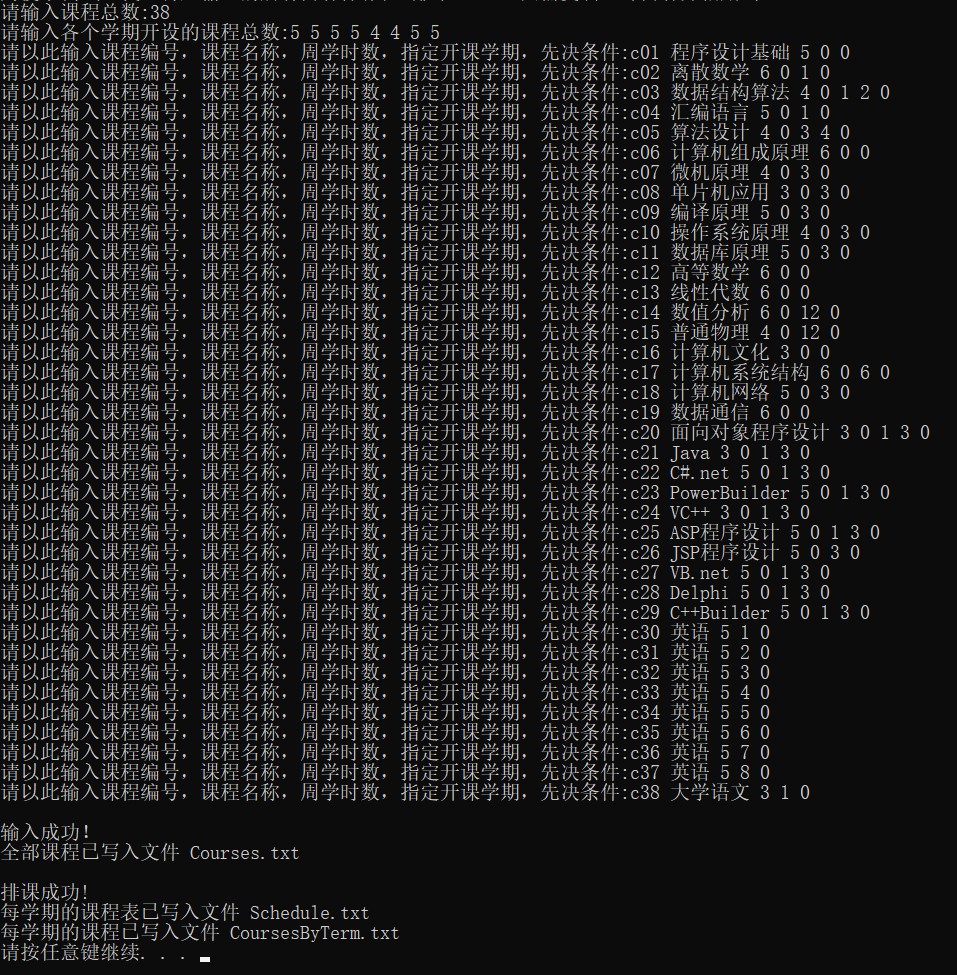
**测试用例**：

内容过多，详见'排课.txt'

**预期结果**：

正确输出全部课程文件、每学期课程文件、每学期课表文件

**实验结果**



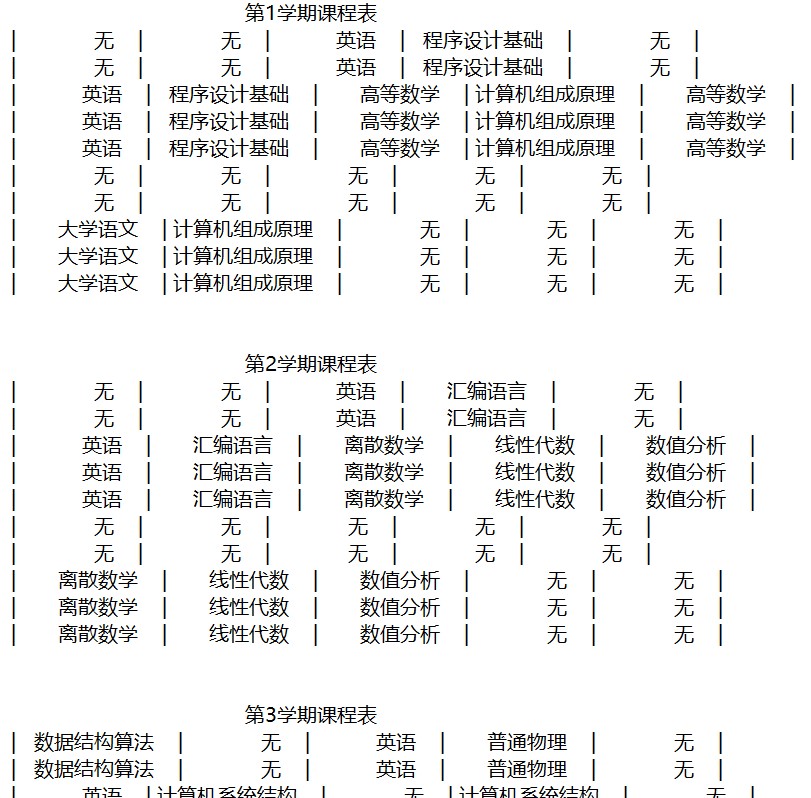
全部课程（Course.txt）:



每学期课程（CoursesByTerm.txt）:



课表（Schedule.txt）：



## 4.2 边界测试

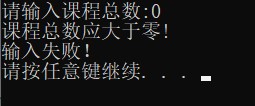
### 4.2.1 课程数为0

**测试用例：**

0

**预期结果：**给出错误提示，程序运行正常不崩溃。

**实验结果：**



### 4.2.2 前几学期课程尽量多

**测试用例：**

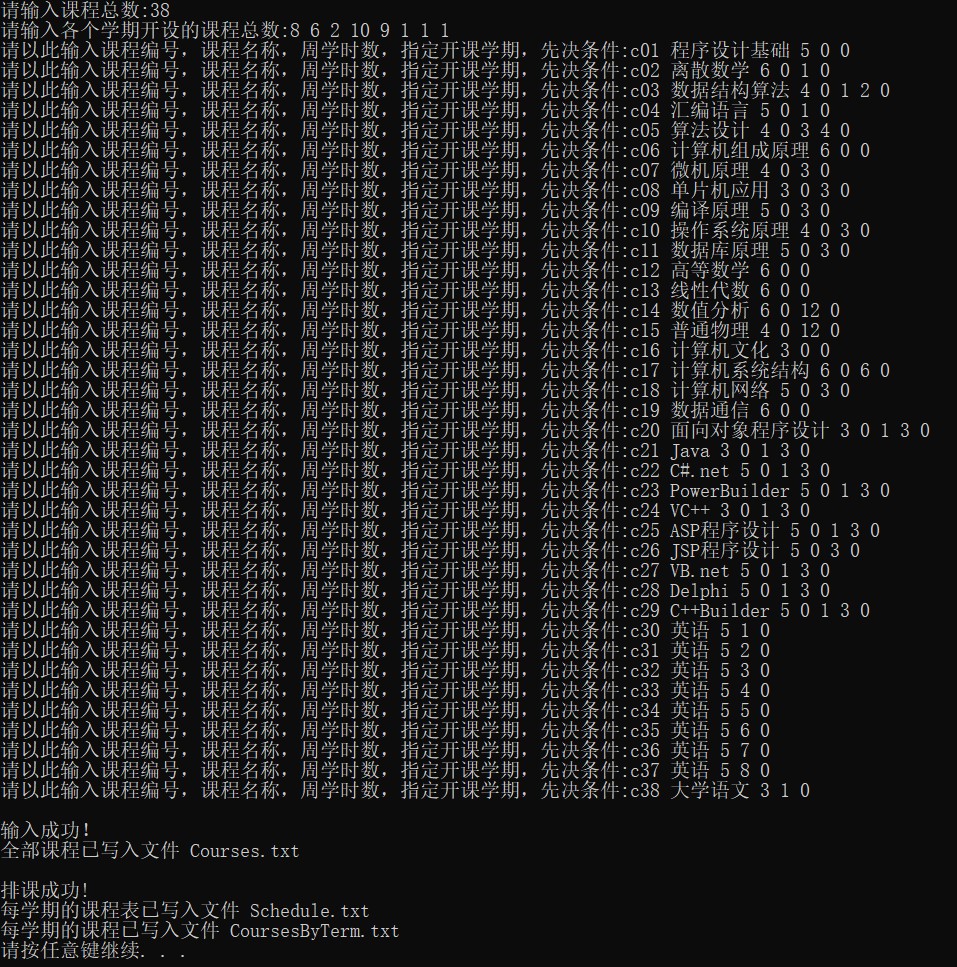
38

8 6 2 10 9 1 1 1

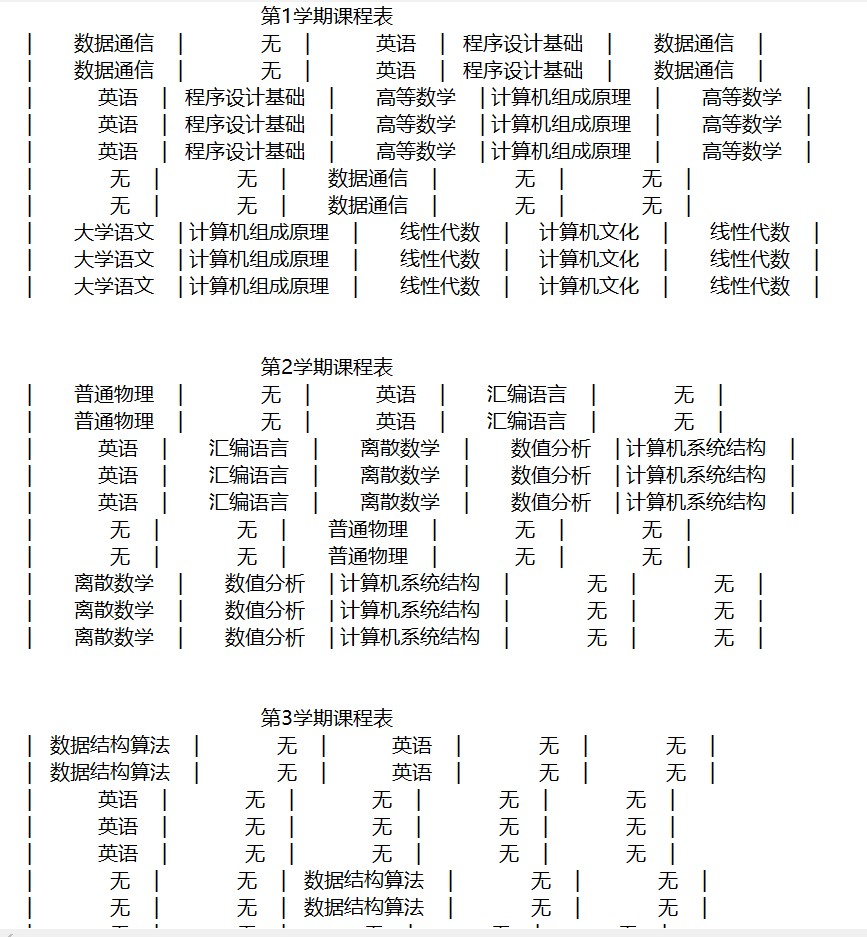
课程信息与'排课.txt'相同

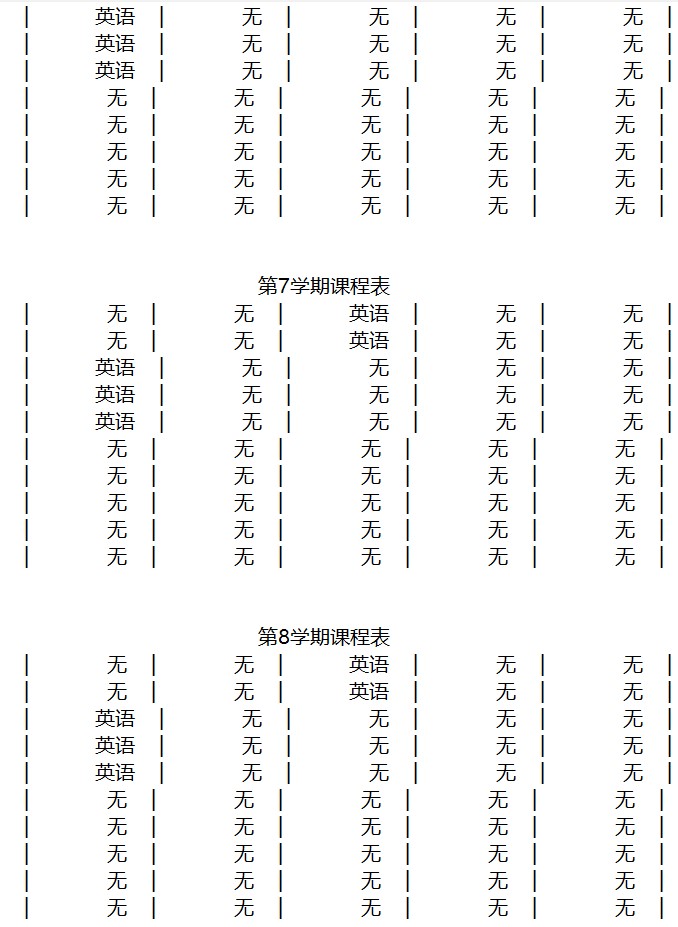
**预期结果：**程序正常运行，不崩溃。

**实验结果：**



课程表Schedule.txt：





## 4.3 出错测试

### 4.3.1 课程总数与每学期课程数的和不等

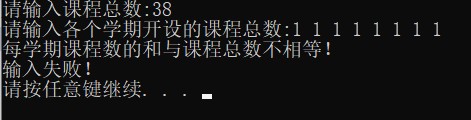
**测试用例：**

38

1 1 1 1 1 1 1 1

**预期结果：**程序给出提示信息，程序正常运行不崩溃。

**实验结果：**

****

### 4.3.2 课程数小于零

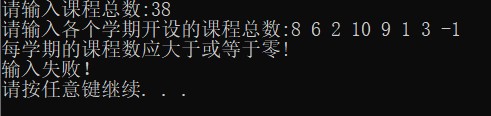
**测试用例：**

38

8 6 2 10 9 1 3 -1

**预期结果：**程序给出提示信息，程序正常运行不崩溃。

**实验结果：**

****

### 4.3.3 当前学期课程数过多（不满足先修条件）

* 也就是当前学期没有那么多能上的课（不满足先修条件）

**测试用例：**

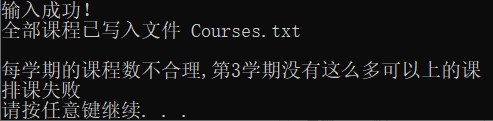
38

5 5 7 3 4 4 5 5

课程信息与'排课.txt'相同

**预期结果：**输入成功，但排课不成功，程序给出错误信息，程序正常运行不崩溃。

**实验结果：**



### 4.3.4 当前学期课程数过多（课时数过多）

**测试用例：**

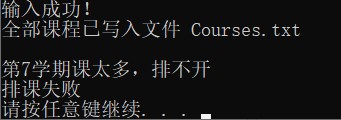
38

5 5 5 5 1 1 13 3

课程信息与'排课.txt'相同

**预期结果：**程序给出错误信息，程序正常运行不崩溃。

**实验结果：**

****

### 4.3.5 当前学期课程数过少（少于该学期指定课程的数量）

**测试用例：**

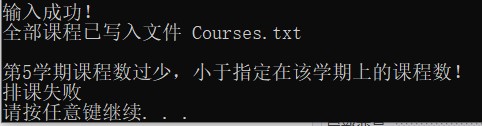
38

5 5 5 5 0 5 5 8

课程信息与'排课.txt'相同

**预期结果：**程序给出错误信息，程序正常运行不崩溃。

**实验结果：**

****