共築夢想校園

動機:

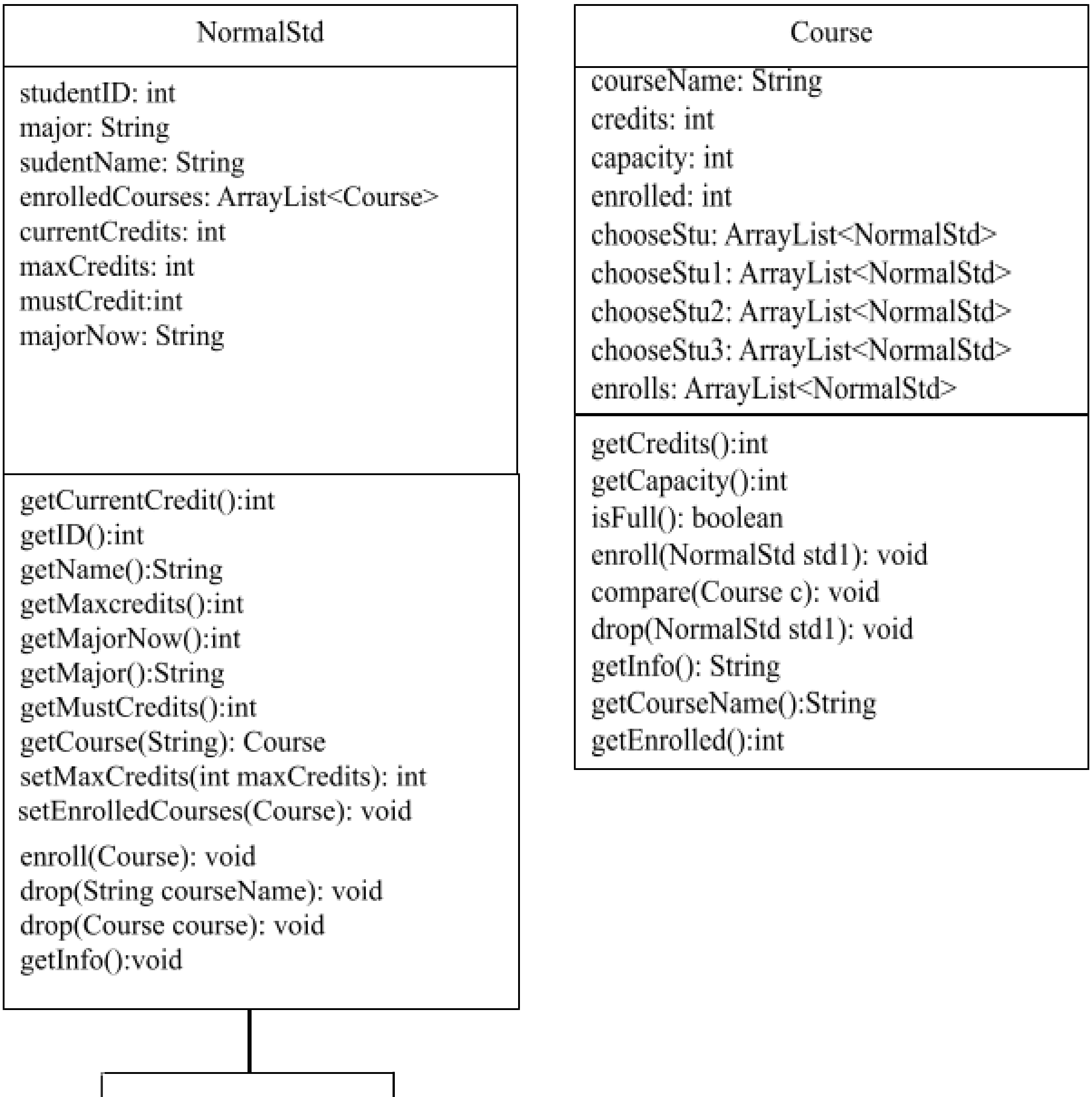
1. 選課身分別限制讓本系必修學生選不到課, 反而其他學院的可以;即便課堂人數未滿, 也因為選課身分只能跑加簽。

   解方:第一階段選課照舊, 第二階段不限制身分但本系優先
2. 學生因學分計算或選課疏忽而導致無法畢業的問題。

   解方:使用者可以藉由選課以及退課的方式增加或減少已選課程庫的資料。如果有選上課, 系統會輸出有選上的科目:如果沒有, 會輸出未選上課的結果。此外, 會印出已修的學分並告知是否可以畢業。

概念:將學生分為一般生、雙主修生、輔系生, 並設置僅能雙輔資管。選課時, 以雙輔生為優先, 若有餘額, 則依序遞補商院生(以企管系為代表)、非商院生(以社會系為代表)。同時, 若學生排序一樣, 則系統隨機選擇誰能選上這堂課。

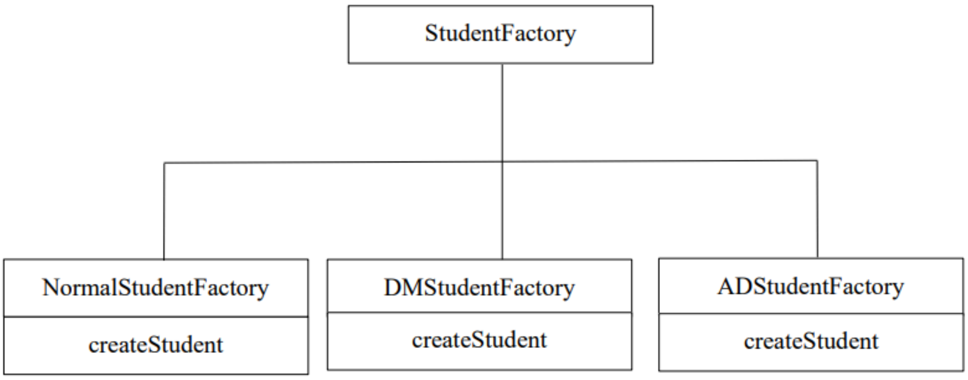
系統架構:



Design Pattern:

我們使用的是Factory Pattern，因為我們的程式中需要輸入多種類型的學生(如DMStd、ADStd、NormalStd)而這些學生都需要一些相似的初始化工作，這樣可以將初始化邏輯抽象到一個Factory中，並根據不同的情況return相應的學生。

我們創建了三個學生類型的Factory：BAStudentFactory、DMStudentFactory 和 ADStudentFactory，它們分別實現了 StudentFactory 接口。這樣的方式讓我們可以輕鬆擴展系統，添加新的學生類型時只需實現新的Factory即可，而不需要修改現有的程式碼。



分工：
109204014呂喬榆：程式發想及撰寫
110305075繆孟珊：程式測試及修改、Design Pattern、Java Doc
112305009陳奕伶：程式測試及修改、PPT、Java Doc
112305101王鈞毅：程式發想及撰寫

	NormalStd
Modifier and type	**Method (or Variable) and description**
Instance variable	
int	studentID ID of a student
String	studentName The student’s name
String	major Major of the student
ArrayList<Course>	enrolledCourses An ArrayList that holds all courses which have been token by the student.
int	currentCredits currentCredits
int	maxCredits credits limit
int	mustCredit The required courses students have passed in such major
int	majorNow Users input numbers to identify their identity, input “1” means double major or auxiliary department MIS, “2” means Commerce, and “3” means Sociology.
Constructor	

NormalStd(String studentName, int studentID, String major, int maxCredits, int majorNow, int mustCredit) Enable to construct a normalStd object with given s*tudent id, studentName, major, maxCredits, majorNow, mustCredit*, and set currentCredits as 0.	
Instance methods	
-	Getter for getID(), getName(), getCurrentID(), getMaxCredits(), getMajorNow(), getMajor(), getMustCredits().
int	setMaxCredits(int) update the maxCredits.
void	setEnrolledCourses(Course) update courses students have enrolled.
Course	getCourse(String courseName) Require a Course object to represent a course, and search the list of enrolledCourses by getCourseName() method. It also records the occurrence count and index of the specific course.If found, return the Course object. Otherwise, return null.
void	drop(String courseName) Create a new NormalStd student object, copies the enrolled courses, and updates the student's enrolled courses. If the course is found, remove it from *enrolledCourses*(Don't forget to revise the current number of students of this course and adjust the current credits of this student); If it's not found, show the information at the console. (See output below)
void	drop(Course course) Removethe course by method getCourseName().

void	getInfo() Determine whether the student is able to graduate, and print a String description. (The limitation is 51 for BA, 42 for SOC)

Course	
Modifier and type	**Method (or Variable) and description**
Instance variable	
String	courseName The course name of this course.
int	credits The credits of the course.
int	capacity The maximum number of students in this course
int	enrolled The current number of students in this course.
ArrayList<NormalStd>	chooseStu Courses the student have chosen.
ArrayList<NormalStd>	chooseStu1 Students who choose the course and majorNow equal to 1.
ArrayList<NormalStd>	chooseStu2 Students who choose the course and majorNow equal to 1.
ArrayList<NormalStd>	chooseStu3 Students who choose the course and majorNow equal to 1.
ArrayList<NormalStd>	enrolls Numbers of students enrolled into the course.
Constructor	
Course(String courseName, int credits, String department) Enable to construct a Course object with given *courseName, credits, and set capacity as 3* and *enrolled* as 0.	
Instance methods	
-	Getter for getCredits(), getCapacity(), getCourseName(),

	getEnrolled().
boolean	**isFull()** Check whether this course is full. If it is full then return true; otherwise return false.
void	**enroll()** Check if maxCredits after adding such course of the student will be more than maxCredits. If so, show a refusal message; otherwise, add the student into the course.
void	**compare()** ● First, use for-each loop to categorize students into chooseStu1, chooseStu2, chooseStu3 by majorNow. ● Second, decide who is able to be added into the course by comparing the size of each chooseStu with capacity. ● If the size of chooseStu1is larger than capacity, we only enroll students in chooseStu1. If the size of chooseStu1 is smaller than capacity, we first add all students in chooseStu1 into the course and then compare chooseStu2, and then compare chooseStu3. For those who are in the same chooseStu arraylist, use method random() to decide who is able to enroll the course. ● Finally, update the information for those enrolled students.
void	**drop()** When a student drops a course, invoke this method to reduce current number of students and reduce the student from the course.
String	**getInfo()** Return a String description for the *courseName, credits, enrolled and capacity* of the course.

DMStd	
Modifier and type	**Method (or Variable) and description**
Instance variable	
Constructor	
DMStd(String studentName, int studentID, String major, int maxCredits. int majorNow, int mustCredit) Enable to construct a DMStd object with given *studentName, studentID, major, maxCredits, majorNow, mustCreditsand* by method super().	
Instance methods	
-	Getter for getMaxCredits().
void	getInfo() Determine whether the student is able to graduate, and print a String description. (Hint: the limitation will be different from NoramlStd.) (The limitation is 87 for BA, 99 for SOC)

ADStd	
Modifier and type	**Method (or Variable) and description**
Instance variable	
Constructor	
DMStd(String studentName, int studentID, String major, int maxCredits. int majorNow, int mustCredit) Enable to construct a ADStd object with given *studentName, studentID, major, maxCredits, majorNow, mustCreditsand* by method super().	
Instance methods	
-	Getter for getMaxCredits().

void	getInfo() Determine whether the student is able to graduate, and print a String description. (Hint: the limitation will be different from NoramlStd.) (The limitation is 69 for BA, 72 for SOC)

Tester

Tester

```
public class Tester {
   public static void main(String[]args){
      //測試人員

      StudentFactory normalStudentFactory = new NormalStudentFactory();
      StudentFactory dmStudentFactory = new DMStudentFactory();
      StudentFactory adStudentFactory = new ADStudentFactory();

      //BA NormalStd /Can’t Graduate
      NormalStd s1 = normalStudentFactory.createStudent("Jason", 109305001, "BA", 31, 1,
86);

      Course mis1 = new Course("程式設計一",2,"MIS");
      mis1.enroll(s1);

      //BA雙MIS選課
      NormalStd s2 = dmStudentFactory.createStudent("Max", 110305003, "BA", 31, 1, 50);

      //BA輔MIS選課
      mis1.enroll(s2);

      //BA Auxiliary Department MIS /Can’t Graduate
      NormalStd s3 = adStudentFactory.createStudent("Mary", 112305002, "BA", 25, 3, 40);
      mis1.enroll(s3);

      Course mis2 = new Course("計算機概論",2,"MIS");
      Course mis3 = new Course("資料結構",3,"MIS");


      //BA普通生選課
      mis1.enroll(s3);
      mis2.enroll(s3);
      mis3.enroll(s3);



      //SOC Double Major MIS /Can’t Graduate
      NormalStd s4 = dmStudentFactory.createStudent("Molly", 110204001, "SOC", 31,
1,40);
      //SOC雙MIS選課
      mis1.enroll(s4);
      mis2.enroll(s4);
      mis3.enroll(s4);
```

```
        //SOC  Auxiliary Department MIS/Graduate
        NormalStd s5 = adStudentFactory.createStudent("David", 109204002,"SOC", 31,
1,70);
        //SOC輔MIS選課
        mis1.enroll(s5);
        mis2.enroll(s5);

        //SOC NormalStd /Graduate
        NormalStd s6 = normalStudentFactory.createStudent("Peter", 109204003, "SOC", 25,
3,41);


        //SOC普通生選課
        mis3.enroll(s6);

        //課程排序身分別選課

        mis1.compare(mis1);
        mis2.compare(mis2);
        mis3.compare(mis3);


        s1.getInfo();
        s2.getInfo();
        s3.getInfo();
        s4.getInfo();
        s5.getInfo();
        s6.getInfo();

        //選上的課
        s1.getCourseInfo();
        s2.getCourseInfo();
        s3.getCourseInfo();
        s4.getCourseInfo();
        s5.getCourseInfo();
        s6.getCourseInfo();
    }



}
```

output

```
Congratulations, Jason, your must credit is 88, you are able to graduate with BA degree!
Sorry, Max, you still leave 37 credits before graduation, since you are a double major
student.
Sorry, Mary, you still leave 24 credits before graduation, since you are an auxiliary
department student.
```

```
Sorry, Molly, you still leave 52 credits before graduation, since you are a double major
student.
Congratulations, David, your must credit is 74, you are able to graduate with SOC degree,
and MIS as auxiliary degree!
Congratulations, Peter, your must credit is 44, you are able to graduate with SOC degree!
Jason 已經選上的課:
程式設計一

Max 已經選上的課:
很抱歉, 你沒有選上課

Mary 已經選上的課:
計算機概論
資料結構

Molly 已經選上的課:
程式設計一
計算機概論
資料結構

David 已經選上的課:
程式設計一
計算機概論

Peter 已經選上的課:
資料結構
```