

پیاده‌سازی کد دانشکده:

کلاس‌های Faculty (دانشکده)، Course، Lab، Student، Professor را پیدا سازی و تکمیل کنید.
- در صورت نیاز، علاوه بر متدهای فعلی، متدهای جدید نیز میتوانید تعریف کنید.

توجه ۱: شرح برخی فیلدها و متدها داده شده است، سایر متدهایی که گفته نشده‌اند ولی در کد موجودند نیز باید پیاده‌سازی شوند.

توجه ۲: موارد سبزرنگ اختیاری اند.

:Faculty

دانشکده دارای اسم، تعدادی استاد، تعدادی دانشجو و تعدادی درس آموزشی است.

- متد isStudentValid وجود یک دانشجو را در دانشکده بررسی میکند.

- متد isCourseValid وجود یک درس آموزشی را در دانشکده بررسی میکند.

- متد isProfessorValid وجود یک استاد را در دانشکده بررسی میکند.

دانشکده باید قابلیت اضافه کردن دانشجو، استاد و درس را داشته باشد.

```
public class Faculty {  
  
    private String name;  
    private Professor[] professors;  
    private Student[] students;  
    private Course[] courses;  
  
    public Faculty(String name) {}  
  
    public String getName() {}  
    public void setStudents(Student[] students) {}  
    public Student[] getStudents() {}  
    public void setCourses(Course[] courses) {}  
    public void setProfessors(Professor professors) {}  
    public Professor[] getProfessors() {}  
    public Course[] getCourses() {}  
    public void enrollStudent(Student student) {}  
    public void addCourse(Course course) {}  
    public void addProfessor(Professor professor) {}  
    public boolean isStudentValid(Student student) {}  
    public boolean isCourseValid() {}  
    public boolean isProfessorValid() {}  
}
```

Course:

هر درس دارای اسم، یک استاد، تعدادی دانشجو، ظرفیت و تعدادی آزمایشگاه است.

- در صورتی که دانشجو به درس اضافه شود، اتومات باید به اولین Lab داری ظرفیت اضافه شود، در صورت پر بودن Lab اول، به Lab دوم اضافه شود و ... (نیازی به بررسی حالتی که تمامی آزمایشگاه‌ها پر باشند نیست)

- در صورت پر بودن کلاس، پیام Class is full چاپ شود و درس به دانشجو اضافه نشود.

درس باید قابلیت اضافه کردن دانشجو و آزمایشگاه را داشته باشد.

```
public class Course {  
  
    private String name;  
    private Professor professor;  
    private Student[] students;  
    private Lab[] labs;  
    private int maxCapacity;  
    private int currentCapacity;  
  
    public Course(String name, Professor professor, int credit){}  
  
    public void setName(String name){}  
    public String getName(){}  
    public void setProfessor(Professor professor){}  
    public Professor getProfessor(){}  
    public void setStudents(Student[] students){}  
    public Student[] getStudents(){}  
    public int getCurrentCapacity(){}  
    public Lab[] getLabs(){}  
    public void addLab(Lab lab){}  
    //Add student in course and the first lab that has capacity  
    public void enrollStudent(Student student){}  
  
}
```

:Lab

هر آزمایشگاه متعلق به یک درس است (فیلد course در کد) و دارای یک TA، روز، تعدادی دانشجو و ظرفیت است.

آزمایشگاه باید قابلیت اضافه کردن دانشجو را داشته باشد.

- در صورت پر بودن آزمایشگاه، پیام Lab is full چاپ شود و آزمایشگاه به دانشجو اضافه نشود.

```
public class Lab {  
  
    private Course course;  
    private String TA;  
    private String day;  
    private Student[] students;  
    private int currentSize;  
    private int capacity;  
  
    public Lab(String TA, String day, Course course, int capacity){}  
    public void setTA(String TA){}  
    public String getTA(){}  
    public void setDay(String day){}  
    public String getDay(){}  
    public void setStudents(Student[] students){}  
    public Student[] getStudents(){}  
    public boolean hasCapacity(){}  
    public void enrollStudent(Student student){}  
  
}
```

:Professor

هر استاد دارای اسم و تعدادی درس است.

قابلیت اضافه شدن درس به استاد باید پیاده‌سازی شود.

```
public class Professor {  
    private String firstName;  
    private String lastName;  
    private Course[] courses;  
  
    public Professor(String firstName, String lastName){}  
  
    public String getFirstName(){}  
    public Course[] getCourses(){}  
    public void setCourses(Course[] courses){}  
    public void addCourse(Course course){}  
  
}
```

:Student

هر دانشجو دارای نام، شماره دانشجویی، لیست نمرات، معدل، تعداد واحدها و تعدادی درس است.
- متد calculateAverage برای محاسبه ی معدل دانشجو است.(فرض شود تمامی نمرات وزن یکسانی دارند)

```
public class Student {  
  
    private String firstName;  
    private String lastName;  
    private String id;  
    private int[] grades;  
    private int average;  
    private Course[] courses;  
  
    public Student(String fName, String lname, String sID) {}  
    public String getFirstName() {}  
    public String getLastName() {}  
    public String getId() {}  
    public int[] getGrades() {}  
    //All grades have equal weight  
    public void calculateAverage() {}  
    public int getAverage() {}  
  
}
```

:Main

این کلاس را برای تست کد خود بسازید و متد main را در این کلاس قرار دهید.

موفق باشید