

استاد : دکتر ملکی مجد	مبانی شی‌گرایی، متدها و آرایه‌ها	زمان پیشنهادی: 90 دقیقه
--------------------------	----------------------------------	-------------------------

مدیریت دانشجویان یک کلاس درس

در این سوال شما باید یک کلاس **Student** برای مدیریت اطلاعات دانشجویان و یک کلاس **Classroom** برای مدیریت کلاس درس تعریف کنید. سپس متدهایی برای عملیات روی دانشجویان و آرایه‌ای برای ذخیره اطلاعات آن‌ها طراحی کنید.

بخش 1: تعریف کلاس Student

کلاس **Student** را ایجاد کنید که شامل ویژگی‌های زیر باشد:

studentId (شماره دانشجویی - عدد صحیح)

name (نام دانشجو - رشته)

grades (آرایه‌ای از 5 عدد صحیح، برای ذخیره نمرات دانشجو در 5 درس)

یک سازنده (**Constructor**) تعریف کنید که مقدار شماره دانشجویی و نام دانشجو را دریافت کند و آرایه نمرات را مقداردهی اولیه کند (با 5 تا صفر).

- متد **SetGrade** در ورودی دو عدد صحیح می‌گیرد. یکی ایندکس درس که بین 0 تا 4 است و نمره درس که بین 0 تا 20 است. مقدار ورودی باید در متد چک شود و در صورتی که در بازه مشخص شده نبود پیغام "Error : outOfRange" در کنسول چاپ شود.
- متد **calculateAverage** را پیاده‌سازی کنید که میانگین نمرات دانشجو را محاسبه و برگرداند.
- متد **displayStudentInfo** را بنویسید که اطلاعات دانشجو (شماره دانشجویی، نام و نمرات) را نمایش دهد.

بخش 2: تعریف کلاس Classroom

کلاس Classroom را ایجاد کنید که دارای ویژگی‌های زیر باشد:

students (آرایه‌ای از اشیاء Student با ظرفیت 30 دانشجو)

studentCount (تعداد فعلی دانشجویان در کلاس)

یک کانستراکتور برای ساخت آرایه بالا و مقداردهی اولیه 0 به studentCount بنویسید

- متد **addStudent(Student newStudent)** را بنویسید که یک دانشجو را به آرایه اضافه کند. اگر کلاس پر باشد، پیام مناسبی نمایش دهد و در صورت وجود داشتن دانش آموزی در آرایه students با studentId ای برابر studentId دانش آموز جدید، پیام "Error : repetitive studentId" را در کنسول چاپ کند و دانش آموز به آرایه اضافه نشود.
- متد **RemoveStudentById** که یک عدد صحیح به عنوان آیدی دانش آموز میگیرد و دانش آموزی با آن آیدی در آرایه students را از آرایه حذف میکند. توجه کنید که بعد از حذف کردن دانش آموز نباید جایش خالی (null) بماند و دانش آموز های بعد از آن باید به چپ شیفت بخورند تا جایش را پر کنند.
- متد **findStudentById(int studentId)** را بنویسید که دانشجویی را بر اساس شماره دانشجویی جستجو کند و با استفاده از فانکشن **displayStudentInfo** اطلاعاتش را چاپ کند.
- متد **calculateClassAverage** را پیاده‌سازی کنید که میانگین نمرات کل دانشجویان کلاس را محاسبه کند و برگرداند. (از فانکشن **calculateAverage** اشیای Student استفاده کنید).
- متد **DisplayRanking** : ابتدا باید وجود دانشجو در کلاس بررسی شود (کلاس خالی نباشد). سپس دانشجویان بر اساس نمرات بصورت نزولی مرتب شوند. اگر میانگین دو دانشجو برابر بود بر اساس شماره دانشجویی به صورت صعودی مرتب شوند. در انتها دانشجویان مرتب شده به همراه رتبه نمایش داده شوند. (از بابل سورت استفاده کنید)

- متد `displayAllStudents` را بنویسید که اطلاعات تمام دانشجویان را چاپ کند. (از فانکشن `displayStudentInfo` اشیای `Student` استفاده شود.)

در `Main` چهار دانشجو بسازید و متد `setGrade` را برایشان فراخوانی کنید و نمره بدهید. سپس یک `Classroom` ایجاد کنید و همه آنها را به آن اضافه کنید. حال متد `displayAllStudents` را فراخوانی کنید و در ادامه برای یکی از دانشجویهایی که اطلاعاتش نشان داده شده `findStudentById` را فراخوانی کنید تا مطمئن شوید در کلاس حضور دارد و از کلاس با متد `removeStudentById` حذفش کنید و متد `DisplayRanking` را فراخوانی کنید.