



دانشگاه صنعتی امیرکبیر  
( پلی تکنیک تهران )

دانشکده ریاضی و علوم کامپیوتر

برنامه سازی پیشرفته و کارگاه

تمرین شی گرای

استاد درس

دکتر مهدی قطعی

استاد دوم

بهنام یوسفی مهر

نگارش

سانیا عزتی

بهار ۱۴۰۳

## فهرست

مقدمه.....	3
شرح پروژه.....	3
Package edu.uni.party.....	3
• فرد: Class Person.....	3
Package edu.uni.entities.....	4
Class Major: رشته دانشگاهی.....	4
Class Student : دانشجو.....	5
Class Professor : استاد.....	6
Class Course : درس.....	6
Package edu.uni.components.....	7
درس ارائه شده : Class PresentedCourse.....	7
کارنامه دانشجو : Class Transcript.....	8
Class Main.....	9

## مقدمه

تا به اینجا با مفاهیم شی‌گرایی و کلاس در جاوا آشنا شدید. در این تمرین هدف شما طراحی یک سیستم مدیریت ساده برای دانشگاه است. در این برنامه شما باید با ایجاد class های مختلف، مفاهیم و عوامل مختلف دانشگاه مثل دروس ارائه شده، دانشجو، استاد و ... را تعریف کنید و از روی آن‌ها، شی‌های منحصر به فرد بسازید.

## شرح پروژه

در ادامه، درباره ساختار پکیج‌ها و مشخصات class های درون آن‌ها از جمله فیلدها، ورودی‌ها و متدها توضیح داده شده است.

برای فیلدها، اطلاعات نوشته شده درون پرانتز نوع آن فیلد را مشخص می‌کنند.

برای متدها، اطلاعات نوشته شده درون پرانتز ورودی‌های متد هستند. نوع خروجی‌های متد با علامت ">" نشان داده شده است.

## Package edu.uni.party

- **فرد: Class Person**

- +Fields:**

- personID (int) : آیدی فرد
    - personList (static - List<Person>) : لیستی از تمام آبجکت‌های ساخته شده از روی این کلاس - مقدار اولیه: لیست خالی
    - name (String) : نام فرد
    - nationalID (final - String) : کد ملی فرد

- +methods:**

- public Person (name, nationalID) : کانستراکتور کلاس که نام و کد ملی را به لیست افراد اضافه می‌کند. به ازای هر آبجکت جدید، ابتدا آن را به لیست افراد اضافه می‌کند، سپس آیدی فرد را برابر با سائز لیست قرار می‌دهد.

- متدی که با ورودی گرفتن آیدی فرد، : Person -> (ID) findById static public  
آبجکتی که همان آیدی را دارد را از لیست افراد پیدا کرده و خروجی می‌دهد.

## Package edu.uni.entities

### رشته دانشگاهی: Class Major

#### + Fields:

- آیدی رشته: majorID (int)
- لیست رشته‌ها، لیستی از تمام آبجکت‌های :<Major> List (static - majorList)  
ساخته شده از روی این کلاس - مقدار اولیه: لیست خالی
- نام رشته: name (String)
- حداکثر تعداد دانشجویان قابل پذیرش در این رشته: maxCapacity (final - int)
- تعداد فعلی دانشجویان در این رشته - مقدار اولیه: ۰ : numberOfStudents (int)

#### + Methods:

- کانستراکتور کلاس که نام رشته :<name> Major (name, maxStudentNumber) public  
و حداکثر تعداد دانشجویان را مقداردهی می‌کند. به ازای هر آبجکت جدید، آن را به  
لیست رشته‌ها اضافه می‌کند، سپس آیدی رشته را برابر با سایز لیست قرار  
می‌دهد.
- متدی که با گرفتن آیدی رشته، آبجکتی :<Major> (ID) findById static public  
که همان آیدی را دارد را از لیست رشته‌ها پیدا کرده و خروجی می‌دهد.
- متدی که تعداد دانشجویان رشته را یک واحد :<>() addStudent void public  
افزایش می‌دهد. اگر تعداد دانشجویان به حداکثر ظرفیت رسیده باشد، پیامی  
چاپ می‌کند.

## دانشجو : Class Student

### + Fields:

- studentID (int): آیدی دانشجو
- studentList (static - List<Student>): لیست دانشجویان، لیستی از تمام آبجکتهای ساخته شده از روی این کلاس - مقدار اولیه: لیست خالی
- personID (int): آیدی فردی که این دانشجو از روی آن ساخته شده
- entranceYear (final - int): سال ورود دانشجو
- majorID (int): آیدی رشته‌ای که دانشجو در آن ثبت‌نام کرده است
- studentCode (String): شماره دانشجویی

### + Methods:

- public Student (personID, entranceYear, majorID): کانستراکتور کلاس که آیدی دانشجو، سال ورود و آیدی رشته را مقداردهی می‌کند. سپس ابتدا با متد مربوطه یک واحد به تعداد دانشجویان این دانشجو اضافه می‌کند، و بعد شماره دانشجویی را با متد مربوطه (که در ادامه توضیح داده شده) تنظیم می‌کند. در نهایت به ازای هر آبجکت، ابتدا آن را به لیست افراد اضافه می‌کند، سپس آیدی دانشجو را برابر با سائز لیست قرار می‌دهد.
- public static findById (ID) -> Student: متدی که با گرفتن آیدی دانشجو، آبجکتی که همان آیدی را دارد را از لیست دانشجویان پیدا کرده و خروجی می‌دهد.
- public void setStudentCode(): متدی که شماره دانشجویی را بر اساس سال ورود، آیدی رشته و تعداد دانشجویان حال حاضر رشته تنظیم می‌کند. برای مثال ۴۰۱۰۲۰۳ شماره دانشجویی سومین دانشجوی ورودی ۴۰۱ دومین رشته دانشگاه است.

## استاد : Class Professor

### + Fields:

- professorID (int): آیدی استاد
- professorList (static - List<Professor>): لیستی از تمام آجکت‌های ساخته شده از روی این کلاس - مقدار اولیه: لیست خالی
- personID (int): آیدی فردی که این استاد از روی آن ساخته شده
- majorID (int): آیدی رشته‌ای که استاد در آن تدریس می‌کند

### + Methods:

- public Professor (personID, majorID): کانستراکتور کلاس که آیدی فرد و آیدی رشته را مقداردهی می‌کند. به ازای هر آجکت جدید، ابتدا آن را به لیست اساتید اضافه می‌کند، سپس آیدی استاد را برابر با سائز لیست قرار می‌دهد.
- public static findById (ID) -> Professor: متدی که با گرفتن آیدی استاد، آجکتی که همان آیدی را دارد را از لیست اساتید پیدا کرده و خروجی می‌دهد.

## درس : Class Course

### +Fields:

- courseID (int): آیدی درس
- courseList (static - List<Course>): لیست دروس، لیستی از تمام آجکت‌های ساخته شده از روی این کلاس - مقدار اولیه: لیست خالی
- title (String): عنوان درس
- units (int): تعداد واحدهای درس

### + Methods:

- public Course (title, units): کانستراکتور کلاس که عنوان درس و تعداد واحدها را مقداردهی می‌کند. به ازای هر آبجکت جدید، ابتدا آن را به لیست دروس اضافه می‌کند، سپس آیدی درس را برابر با سائز لیست قرار می‌دهد.
- public static findById (ID) -> Course: آبجکتی که با گرفتن آیدی درس، آبجکتی که همان آیدی را دارد را از لیست دروس پیدا کرده و خروجی می‌دهد.

## Package edu.uni.components

### Class PresentedCourse : درس ارائه شده

### + Fields:

- presentedCourseID (int): آیدی درس ارائه شده
- presentedCourseList (static - List<PresentedCourse>): لیست دروس ارائه شده، لیستی از تمام آبجکت‌های ساخته شده از روی این کلاس - مقدار اولیه: لیست خالی.
- courseID (int): آیدی درسی که این درس ارائه شده از روی آن ساخته شده
- professorID (int): آیدی استاد ارائه‌دهنده
- maxCapacity (int): حداکثر تعداد دانشجویان قابل ثبت‌نام در این درس
- studentIDList (List<Integer>): لیست آیدی‌های دانشجویان ثبت‌نام شده در این درس

### + Methods:

- public PresentedCourse (courseID, professorID, maxCapacity): کانستراکتور کلاس که آیدی درس، آیدی استاد و حداکثر تعداد دانشجویان را مقداردهی می‌کند. سپس یک لیست خالی با گنجایشی برابر با حداکثر تعداد دانشجویان این درس، برای آیدی‌های دانشجویان ثبت‌نام شده می‌سازد. در نهایت

به ازای هر آبجکت، ابتدا آن را به لیست دروس ارائه شده اضافه می‌کند، سپس آیدی درس ارائه شده را برابر با سائز لیست قرار می‌دهد.

- `public static findById (ID) -> PresentedCourse`: متدی که با گرفتن آیدی درس ارائه شده، آبجکتی که همان آیدی را دارد را از لیست دروس ارائه شده پیدا کرده و خروجی می‌دهد.
- `public void addStudent (studentID)`: متدی که آیدی دانشجو را به لیست دانشجویان ثبت‌نام شده در این درس اضافه می‌کند.

## کارنامه دانشجو : Class Transcript

### + Fields:

- `studentID (int)`: آیدی دانشجو
- `transcript (Map<Integer, Double>)`: کارنامه دانشجو که دیکشنری‌ای شامل آیدی دروس ارائه شده دانشجو و نمره آن‌هاست.

### + Methods:

- `public Transcript (studentID)`: کانستراکتور کلاس که آیدی دانشجو را مقداردهی می‌کند و یک کارنامه خالی ایجاد می‌کند.
- `public void setGrade (presentedCourseID, grade)`: متدی که با گرفتن آیدی یک درس ارائه شده و نمره آن، ابتدا چک می‌کند آیدی دانشجوی این کارنامه در لیست آیدی‌های درس ارائه شده باشد، سپس در آن صورت نمره دانشجو را برای آن درس ارائه شده تنظیم می‌کند.
- `public void printTranscript()`: متدی که کارنامه دانشجو را چاپ می‌کند. خروجی چاپ شده باید به صورت «عنوان درس: نمره» باشد.
- `public double getGPA() -> double`: متدی که معدل وزن‌دار دانشجو را بر اساس تعداد واحدهای هر درس، محاسبه و بازمی‌گرداند.



نکات:

1. همانطور که گفته شد تمام کلاس‌ها به جز کلاس Transcript، دارای دو فیلد static List<X> XList و int XID، و متد findById (X XID) هستند که X نام کلاس است. در کد خود برای دسترسی به آبجکت‌های هر کلاس از متد findById استفاده کنید.
2. برای تعریف فیلدهایی که در کلاس‌های دیگر استفاده می‌شوند از کلیدواژه public استفاده کنید.
3. در تمام پروژه هر جا نیاز به لیست بود، از ArrayList استفاده کنید.
4. برای تبدیل عدد به استرینگ می‌توانید از متد String.valueOf (int n) استفاده کنید.

## Class Main

در تابع main از کلاس Main، دو رشته و پنج فرد ایجاد کنید. سپس سه شی دانشجو تعریف کنید که دو نفر اول در رشته اول و نفر آخر در رشته دوم تحصیل کنند. سپس اسم و شماره دانشجویی همه دانشجوها را چاپ کنید.

در ادامه با دو فرد باقی مانده، دو شی استاد تعریف کنید. هر کدام از استادها در یکی از رشته‌های تعریف شده تدریس می‌کنند. سپس اسم و آیدی اساتید را چاپ کنید.

سه درس تعریف کنید و از روی آن‌ها، سه درس ارائه شده تعریف کنید. دروس اول و دوم را استاد اول و درس سوم را استاد دوم ارائه می‌کند. در درس اول دانشجوی اول و دوم، در درس دوم دانشجوی اول و دوم و سوم و سوم، و در درس سوم دانشجوی سوم شرکت کرده باشد.

در ادامه یک شی کارنامه برای هر دانشجو تعریف کنید و به دروسی که اخذ کرده‌اند نمره بدهید. در انتها همه کارنامه‌ها را چاپ کرده و معدل هر کدام را حساب و چاپ کنید.