



دانشگاه صنعتی امیر کبیر
(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

3	مقدمه
4	شرح پروژه
5	پکیج base
5	کلاس Person
5	Field ها
5	methods
6	پکیج uni
6	کلاس Major
6	Fields
6	Methods
7	کلاس Student
7	Fields
7	Methods
8	کلاس Professor
8	Fields
8	Methods
8	کلاس Course
9	Fields
9	Methods
9	کلاس PresentedCourse
9	Fields
10	Methods
10	کلاس Transcript
10	Fields
11	Methods
12	Main

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 class‌های درون آن‌ها از جمله فیلدها، ورودی‌ها و متدها توضیح داده شده است. پروژه شما از دو پکیج مختلف به اسم base و uni تشکیل شده. دقت کنید که شما بسته به نیازتون، می‌تونید متدها و فیلدهای جدیدی برای هر کلاس در نظر بگیرید. چیزی که این‌جا اومده، صرفاً حداقل‌هایی هست که هر کلاس باید داشته باشه.

پکیج base

کلاس Person

این کلاس، نشان دهنده یک شخص در برنامه ماست. این شخص ممکنه استاد، دانشجو، کارمند یا هر چیز دیگه‌ای باشه.

Fieldها

public int id: هر آبجکت Person، باید یک ID یکتا و مختص به خودش داشته باشه. این ID نباید به هیچ object دیگه‌ای داده بشه.

public static ArrayList<Person> personList: لیستی از تمام آبجکت‌های Person که در طول برنامه ساخته می‌شن. لازمه که با ساخت هر object جدید از جنس Person، اون آبجکت داخل این آرایه قرار بگیره.

public String name: اسم هر شخص

public String nationalID: کد ملی هر شخص

methods

public Person(String name, String nationalID): کانستراکتور این کلاس، که یک آبجکت جدید از کلاس درست می‌کنه. دقت کنید که این کانستراکتور باید ID آبجکتی که می‌سازه هم مشخص کنه.

public static Person findById(int id): تابع static‌ای که با ورودی گرفتن id مختص یک person، توی آرایه personList به دنبال آبجکتی با اون id می‌گرده و وقتی پیدااش کرد، اون رو return می‌کنه. اگر آبجکتی با id مشخص شده پیدا نشد، این متد null برمی‌گردونه.

پکیج uni

کلاس Major

کلاس Major، نشان‌دهنده یک رشته دانشگاهیه. مثل «ریاضی»، «علوم کامپیوتر» و امثال اینا.

Fields

public int id: هر آبجکت Major، باید یک ID یکتا و مختص به خودش داشته باشه. این ID نباید به هیچ object دیگه‌ای داده بشه.

public static ArrayList<Major> majorList: لیستی از تمام آبجکت‌های Major که در طول برنامه ساخته می‌شن. لازمه که با ساخت هر object جدید از جنس Major، اون آبجکت داخل این آرایه قرار بگیره.

public String name: نام رشته، مثلاً «ریاضی» یا «علوم کامپیوتر»

public final int capacity: ماکسیم ظرفیت رشته. دقت کنید که این فیلد بعد از مقداردهی در constructor امکان تغییر نداره.

public final int numberOfStudents: تعداد دانشجویان حاضر در این رشته. مقدار اولیه این فیلد، صفره.

Methods

public Major(String name, int capacity): کانستراکتور کلاس که نام رشته و ظرفیت اون رو به عنوان ورودی می‌گیره. دقت کنید که این کانستراکتور باید id این رشته هم مشخص کنه.

public static Major findById(int id): این متد با گرفتن id یک Major، توی لیست majorList به دنبال اون می‌گرده و خروجی‌ش می‌ده. اگر پیدااش نکرد، null خروجی می‌ده.

public void addStudent(): این متد، تعداد دانشجویهای فعلی یک رشته رو یکی زیاد می‌کنه. قبل از این کار، چک کنید که تعداد دانشجویها به حداکثر ظرفیت رشته نرسیده باشه. اگر رسیده بود، یک خطا چاپ کنید.

کلاس Student

این کلاس، نشان‌دهنده یک دانشجو توی دانشگاه ماست که برای خودش رشته و درس و این‌ها داره.

Fields

public int id: هر آبجکت Student، باید یک ID یکتا و مختص به خودش داشته باشه. این ID نباید به هیچ object دیگه‌ای داده بشه.

public static ArrayList<Student> studentList: لیستی از تمام آبجکت‌های Student که در طول برنامه ساخته می‌شن. لازمه که با ساخت هر object جدید از جنس Student، اون آبجکت داخل این آرایه قرار بگیره.

public int personID: آی‌دی آبجکت Person‌ای که شامل اسم و مشخصات این دانشجو هست.

public final int entranceYear: سال ورود دانشجو به دانشگاه رو نشون می‌ده. این مقدار بعد از مقداردهی توی constructor، هیچ‌وقت تغییر نمی‌کنه.

public int majorID: آی‌دی آبجکت Major‌ای که نشان‌دهنده رشته این دانشجوئه.

public String studentID: شماره دانشجویی هر دانشجو. دقت کنید که این فیلد از فیلد id کاملاً متفاوت. مثلاً اگر id مقداری مثل ۴ به خودش می‌گیره، این فیلد مقداری مثل "40113009" داره.

Methods

public Student(int personID, int entranceYear, int majorID): کانستراکتور کلاس Student که آیدی دانشجو، سال ورود و آیدی رشته را مقداردهی می‌کند. سپس یک واحد به تعداد دانشجویان رشته این دانشجو اضافه می‌کند، و بعد شماره دانشجویی را با متد setStudentCode (که در ادامه توضیح داده شده) تنظیم می‌کند. حواستون باشه که توی کانستراکتور، فیلد id هم مقداردهی کنید.

public static Student findById(int id): متدی که با گرفتن آی‌دی آبجکت Student، اون رو از لیست studentList پیدا می‌کنه و خروجی می‌ده. اگر دانشجویی با id مشخص شده پیدا نشد، این متد null برمی‌گردونه.

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.

کلاس Professor

این کلاس، نشان‌دهنده یک استاد توی دانشگاه ماست که برای خودش رشته و درس و این‌ها داره.

Fields

public int id: هر آبجکت Professor، باید یک ID یکتا و مختص به خودش داشته باشه. این ID نباید به هیچ object دیگه‌ای داده بشه.

public static ArrayList<Professor> professorList: لیستی از تمام آبجکت‌های Professor که در طول برنامه ساخته می‌شن. لازمه که با ساخت هر object جدید از جنس Professor، اون آبجکت داخل این آرایه قرار بگیره.

public int personID: id متعلق به آبجکت Person ای که نشان‌دهنده مشخصات این استاده.

public int majorID: id متعلق به رشته‌ای که این استاد اون رو تدریس می‌کنه.

Methods

public Professor (int personID, int majorID): کانستراکتور کلاس Professor که آیدی فرد و آیدی رشته را مقداردهی می‌کند.

public static Professor findById (int ID): متدی که با گرفتن آیدی استاد، آبجکتی که همان آیدی را دارد را از لیست اساتید پیدا کرده و خروجی می‌دهد.

کلاس Course

این کلاس، نشان‌دهنده یک درست توی دانشگاه ماست، مثلاً درست «برنامه‌نویسی پیشرفته». این درس بعداً می‌تونه توسط استادها تدریس بشه و دانشجوها اون رو بردارن.

Fields

public int id: هر آبجکت Course، باید یک ID یکتا و مختص به خودش داشته باشد. این ID نباید به هیچ object دیگری داده بشه.

public static ArrayList<Course> courseList: لیستی از تمام آبجکت‌های Course که در طول برنامه ساخته می‌شن. لازمه که با ساخت هر object جدید از جنس Course، اون آبجکت داخل این آرایه قرار بگیره.

public string title: عنوان درس

public int units: تعداد واحدهایی که این درس به خودش اختصاص می‌ده.

Methods

public Course(String title, int 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 Course findById (int ID): متدی که با گرفتن آیدی درس، آبجکتی که همان آیدی را دارد را از لیست دروس پیدا کرده و خروجی می‌دهد.

کلاس PresentedCourse

این کلاس، نشان‌دهنده یک درس ارائه‌شده یا یک درس در حال ارائه‌ست. درس در حال ارائه، برای خودش یک استاد و یک لیست از دانشجویهای مختلف داره که اون درس رو برداشتن. دقت کنید که این مفهوم، با مفهوم Course فرق داره. اگر یک آبجکت Course، نشان‌دهنده درس «برنامه‌نویسی پیشرفته» باشه، آبجکت PresentedCourse نماد «درس برنامه‌نویسی پیشرفته دکتر قطعی که قلی و ممد و جواد برداشتن»‌ئه.

Fields

public int id: هر آبجکت PresentedCourse، باید یک ID یکتا و مختص به خودش داشته باشه. این ID نباید به هیچ object دیگری داده بشه.

public static ArrayList<PresentedCourse> presentedCourseList: لیستی از تمام آبجکت‌های PresentedCourse که در طول برنامه ساخته می‌شن. لازمه که با ساخت هر object جدید از جنس PresentedCourse، اون آبجکت داخل این آرایه قرار بگیره.

public int courseId: آی‌دی Course‌ای که داره ارائه می‌شه.

public int professorID: آی‌دی Professor‌ای که درس رو ارائه می‌ده.

public int capacity: حداکثر ظرفیت درس ارائه شده.

public ArrayList<Integer> studentIDs: آرایه‌ای از آی‌دی دانشجویهایی که این درس رو برداشتن.

Methods

public PresentedCourse (int courseId, int professorID, int 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 سایز لیست قرار می‌دهد.

public static PresentedCourse findById (int ID): 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 (int studentID): متدی که آی‌دی دانشجو را به لیست دانشجویان ثبت‌نام شده در این درس اضافه می‌کند. اگر دانشجویان این کلاس پر بودن اما این متد نباید دانشجوی جدیدی اضافه کنه و در عوض باید یک خطا بده.

کلاس Transcript

این کلاس، نشان‌دهندهٔ نمرات و کارنامهٔ یک دانشجوئه.

Fields

public int studentID: آی‌دی دانشجویی که این کارنامه متعلق به اونه.

public HashMap<Integer, Double> transcript: هش‌مپی از نمرات دانشجویان، که کلید اون نشان‌دهندهٔ آی‌دی درس ارائه شده‌ست و مقدار هر کلید، نمرهٔ اون شخص در درس مورد نظر.

Methods

public Transcript (int studentID): کانستراکتور کلاس که آی‌دی دانشجو را مقداردهی می‌کند و یک کارنامه خالی ایجاد می‌کند.

public void setGrade (int presentedCourseID, double grade): متدی که با گرفتن آی‌دی یک درس ارائه شده و نمره آن، ابتدا چک می‌کند که دانشجو در لیست دانشجویان درس ارائه شده باشد و سپس، 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. دقت کنید که این‌جا لازمه از متد `findByID` کلاس `Course` و `PresentedCourse` استفاده کنید. حتما در ابتدای این متد، اسم خود شخص و شماره دانشجویی‌ش هم چاپ کنید.

public double getGPA(): متدی که معدل وزن‌دار دانشجو را بر اساس تعداد واحدهای هر درس، محاسبه و بازمی‌گرداند.

Main

در تابع main از کلاس Main، دو رشته و پنج فرد ایجاد کنید (آیدی افراد به ترتیب ۱ تا ۵ خواهد بود). 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 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 با متد مربوطه همه کارنامه‌ها را چاپ کرده، و سپس معدل هر کدام را حساب و چاپ کنید.

نکات:

1. همانطور که گفته شد تمام کلاس‌ها به جز کلاس Transcript، دارای فیلد id و متد findById هستند. متد findById همیشه در صورتی که آبجکتی با آیدی مد نظرتون پیدا نکند null برمی‌گردونه. هر جایی که از این متد استفاده کردین، چک کنید که آبجکت خروجی‌ش null نباشه.

2. تا جای ممکن، برای دسترسی به اطلاعات موجود در کلاس‌های دیگه، از متد findById استفاده کنید. مثلاً اگر می‌خواید توی کلاس PresentedCourse به اسم درس دسترسی داشته باشین، با استفاده از متد Course.findById(courseId) به اسم کلاس دسترسی پیدا کنید. لطفاً فیلدهای تکراری مثل courseTitle توی PresentedCourse، یا personName توی student تعریف نکنید.

3. برای تعریف فیلدهایی که در کلاس‌های دیگه استفاده می‌شوند از کلیدواژه public استفاده کنید.

4. در تمام پروژه هر جا نیاز به لیست بود، از ArrayList استفاده کنید.

5. برای تبدیل عدد به استرینگ می‌توانید از متد String.valueOf(int n) استفاده کنید.

6. اصول کد تمیز رو رعایت کنید. لازمه حتما این تمرین رو با گیت انجام بدین و از طریق گیت‌هاب تحویل بدین. همچنین یک README.md خوب برای ریپوتون بنویسید که توش پروژه‌تون رو توضیح می‌دین.