گزارش تمرین سری چهارم

هدف ازین تمرین اجرای عملیات مربوط به پرتال دانشجویی است.

برای این تمرین چندین کلاس در نظر گرفته شده است و مفاهیم ارث بری استفاده شده است.

کلاس person :

کلاسی که نشان دهنده ی هر انسان می باشد و دانشجو و استاد از این کلاس به ارث می برند. ویژگی هایی که بین دانشجو و استاد مشترک باشد در این کلاس تعریف شده است.

ویژگی های مشترک شامل نام و نام خانوادگی و آی دی هر شخص می باشد. از طرفی هر شخص چه دانشجو چه استاد دارای تعدادی واحد هستند که تعداد ان ها را در n_classes و خود واحد ها(درس ها) در لیستی از جنس *course ذخیره شده اند. پس باید بتوان برای هر شخص یک کورس را اضافه یا کم کرد.

اضافه کردن واحد بدین صورت است که ابتدا باید تعداد واحد ها افزایش دهیم سپس لیست جدید با تعداد جدید ساخته و المان های ان را ست کرده و سپس لیست قبلی را دیلیت کنیم و دوباره ان را new کرده و با لیست ساخته شده برابر قرار دهیم.

برای حذف کورس ابتدا باید پیدا کنیم کورس مورد نظر در کجای لیست قرار دارد. سپس لیست جدید را با تعداد کورس یکی کمتر می سازیم و لیست قبلی را به دو بخش قبل از اندیس کورس حذفی و بعد از اندیس کورس حذفی تقسیم میکنیم و در بخش اول ارایه های لیست جدید را برابر با لیست قبلی قرار میدهیم و در بخش دوم ارایه های لیست جدید را با یکی کمتر از i برابر ارایه های لیست قبلی قرار میدهیم.

مثلا دانشجویی ۵ درس دارد و میخواهد درس AP را حذف نماید. ابتدا اندیس AP را در لیست پیدا کرده که شماره ۲ می باشد سپس از ۰ تا ۲ (۰و۱) در ارایه ی جدید در اندیس ۰و۱ میریزیم و بعد از ان از ۳ تا ۵ (۳و۴) را در ارایه ی جدید در اندیس ۲و۳ میریزیم.

در این کلاس یک تابع virtual نوشته شده که در صورتی که این تابع در کلاس های پایینتر و تخصصی تر دوباره تعریف گردد مقدار آن با مقدار تعریف شده جایگزین شود.

کلاس prof:

در این کلاس عمده کاری که انجام میدهیم تعیرف تابع get_mean می باشد. برای این امر ابتدا یک لیستی از نمرات دانشجویان استاد درست کرده و سپس میانگین آن را محاسبه میکنیم.

این لیست نمرات همه ی دانشجویان استاد در همه ی کورس های استاد را نشان میدهد و هنگامی که تابع scoring صدا زده میشود، تغییر میکند.

کلاس student:

هنگامی که کورسی به طور کلی اضافه می شود در این کلاس آن کورس با توجه به اینکه درس تخصصی می باشد یا آزمایشگاهی به صورت مجزا به لیست در نظر گرفته شده برای دروس تخصصی و آزمایشگاهی اضافه میشود. همچنین متناسب با این اضافه شدن باید هنگام حذف کلی یک درس ان درس را نیز از لیست درس تخصصی یا آزمایشگاهی مرتبط با آن نیز حذف کنیم. برای

این منظور توابع add_theo_course,remove_theo_course,add_lab_course,remove_lab_course تعریف شده اند. در این توابع که ورودی آن ها آبجکتی از جنس کارنامه ی تخصصی یا آزمایشگاهی است، ابتدا تعداد دروس تخصصی یا آزمایشگاهی تغییر کرده و سپس کارنامه ی مرتبط با کورس به لیست اضافه یا کم میگردد.

اپراتور هایی نظیر ++ برای افزایش ترم دانشجو و =! در نظر گرفته شده است. که زمانی دو دانشجو با یکدیگر برابر نیستند که نام و نام خانوادگی آن ها با یکدیگر تفاوت داشته باشد البته چون نام پدر را نداریم آی دی آن ها را نیز باید مقایسه بکنیم.

هنگام تعریف یک دانشجو باید نام، نام خانوادگی، آی دی، معدل دانسجو و تعداد واحد های پاس شده را تعیین نماییم.

در این بین تابع get_mean معدل دانشجو و passed تعداد واحد های پاس شده دانشجو را بازمیگرداند.

کلاس er:

کلاسی است که کارنامه ای از دانشجو را ارائه میدهد که دو کلاس theorical و lab از آن ارث می برند.

به طور کلی برای هر کارنامه باید کورس، استاد، دانشجو، تعداد واحد های آن کورس، تعداد غیبت ها، نمره ی فاینال و pass/faile بودن دانشجو مشخص گردد.

دو مقدار تابع scoring و calculate_final به صورت virtual تعریف شده اند و مقدار آن ها با مقدار تعریف شده در کلاس های پایینتر جایگزین می شود.

کلاس theorical:

در این کلاس باید دو تابع scoring و calculate_final را تعریف کنیم. در تابع scoring ابتدا نمره های میانی و پایانی و تمرین ها ست می شود و سپس تابع محاسبه ی فاینال را صدا زده و مقدار فاینال را به لیست نمره های دانشجوهای استاد اضافه میکنیم. در صورتی که مقدار فاینال بین ۱۰ تا ۲۰ باشد دانشجو پاس شده است.

کلاس Lab:

در این کلاس باید دو تابع scoring و calculate_final را تعریف کنیم. در تابع scoring ابتدا نمره های امتحان پایانی و فعالیت در کلاس و گزارش ها ست می شود و سپس تابع محاسبه ی فاینال را صدا زده و مقدار فاینال را به لیست نمره های دانشجوهای استاد اضافه میکنیم. در صورتی که مقدار فاینال بین ۱۰ تا ۲۰ باشد دانشجو پاس شده است.

کلاس course:

برای تعریف یک کورس چندین راه در نظر گرفته شده است.

۱. کورس را به همراه استاد، نام کورس، تعداد واحد ها و نوع درس(تئوری یا عملی) تعریف کنیم. در این حالت هیچ دانشجو هنوز وارد این درس نشده است. این کورس به دروس استاد باید اضافه گردد.

در این کلاس اگر theorical یک باشد یعنی درس تئوری و در غیر اینصورت درس عملی است.

۲. تشکیل کورس با مقادیر قبلی ولی به همراه یک دانشجو

در این حالت جمعیت دانشجویان این کورس برابر ۱ می شود. کورس یه دروس استاد اضافه میگردد. کورس به دروس دانشجو نیز باید اضافه گردد. بدین صورت که تعداد کلاس های دانشجو یکی افزایش یافته و کورس به لیست کورس های دانشجو اضافه میگردد و با توجه به اینکه تئوری یا عملی است تعداد و لیست دروس تئوری و عملی نیز تغییر میکند.

اپراتور کوچکتر یا مساوی برای کورس ها با توجه به جمعیت هر کلاس تعریف شده است.

با اپراتور () می توانیم دانشجو به کورس مورد نظر اضافه یا کم کنیم. این اضافه کردن کورس با دو تابع جداگانه یکی برای درس تئوری و دیگری برای درس عملی انجام میشود. در این تابع اگر option ۱ باشد به معنی اضافه کردن دانشجو و اگر ۰ باشد به معنی حذف دانشجو می باشد.

برای اضافه کردن دانشجو ابتدا باید کارنامه ی درس تئوری را با مقادیر موجود بسازیم. سپس جمعیت را یکی اضافه کنیم و لیست جدید کارنامه ها را مقدار جدید جمعیت ساخته و مقادیر آن را ست کنیم. سپس لیست دانشجو ها را نیز باید آپدیت کنیم و بعد از آن درس را به درس های دانشجو اضافه کنیم. برای حذف دانشجو نیز باید به طریق مشابه عمل کنیم فقط با این تفاوت که باید اندیس دانشجویی که میخواهیم حذف کنیم را بیابیم و ان دانشجو را به همراه کارنامه ی آن حذف نماییم.