

دستور کار کارگاه برنامه‌نویسی پیشرفته

جلسه ششم

آشنایی با مفاهیم تحلیل و طراحی شی‌گرا

مقدمه

در این جلسه قرار است تا با نحوه اصولی تحلیل و طراحی شی‌گرا آشنا شویم. معمولاً در پروژه‌ها این امکان وجود دارد که یک نفر پروژه را در یک بازه زمانی پیاده‌سازی می‌کند و فرد دیگری کار را ادامه می‌دهد. اگر برای پیاده‌سازی سیستم به طراحی آن توجه نشده باشد، فرد جدید برای ادامه کار و همچنین نگهداری و رفع اشکال قسمت‌های قبلی با مشکل روبه‌رو می‌شود. چون فهم مسئله و راه‌حلی که توسعه‌دهنده ارائه کرده است، برای توسعه‌دهنده جدید مشکل خواهد بود. ممکن است اصلاً متوجه نشود که کد پیاده‌سازی شده چگونه کار می‌کند و به احتمال زیاد مجبور می‌شود تا قسمت‌های زیادی از کد را بازنویسی یا حذف کند. در پروژه‌های بزرگ که مسئله‌های متنوع و پیچیده‌تری وجود دارد، اگر تحلیل و طراحی درست و اصولی انجام نشود، ممکن است پیاده‌سازی پروژه به انتها نرسد و یا به قدری پیچیده شود که امکان نگهداری و توسعه‌های بعدی برای آن سیستم غیرممکن باشد. مجموعه این موضوعات در درس‌های مهندسی نرم‌افزار با بررسی انواع حالت‌های ممکن بررسی خواهد شد. در این دستور کار، به صورت مختصر با مراحل تحلیل و طراحی شی‌گرا برای یک مسئله ساده آشنا می‌شویم.

نکات آموزشی

برای انجام یک پروژه، همیشه با یک متن تعریف و صورت پروژه روبه‌رو هستیم که نیازمندی‌های موردنظر ذی‌نفعان^۱ در آن آمده است. یک مهندس نرم‌افزار با مطالعه این تعریف پروژه و در صورت نیاز برقراری ارتباط به ذی‌نفعان و کسب اطلاعات بیشتر، سعی می‌کند تا بهترین طراحی ممکن را ارائه

^۱ stakeholder

دهد. در این دستور کار ما نحوه مطالعه تعریف پروژه و به دست آوردن کلاس‌ها و ارتباط آنها را یاد می‌گیریم. همچنین نکاتی درباره طراحی خوب کلاس‌ها مطرح خواهیم کرد.

مراحل انجام کار

در این دستور کار، مراحل طراحی یک سیستم رزرو بلیت سینما را بررسی می‌کنیم. تعریف پروژه زیر داده شده است:

سیستم رزرو بلیت سینما باید رزروهای مربوط به چند سالن مختلف را ذخیره کند. هر سالن تعدادی صندلی در ردیف‌های مختلف دارد. مشتریان می‌توانند صندلی رزرو کنند که به آنها شماره صندلی و شماره ردیف در سالن داده می‌شود. مشتری می‌تواند رزرو چندین صندلی مجاور را درخواست کند.

هر رزرو مربوط به یک نمایش خاص است (منظور نمایش یک فیلم در یک زمان خاص است). نمایش‌ها در یک ساعت، تاریخ و سالن مشخص برنامه ریزی می‌شوند. سیستم شماره تلفن مشتری را ذخیره می‌کند.

- برای تشخیص کلاس‌ها از **روش فعل‌ها/اسم‌ها** استفاده می‌کنیم. در متن زیر اسم‌ها با زیرخط و **فعل‌ها** با رنگ قرمز نشان داده شده‌اند:

سیستم رزرو بلیت سینما باید رزروهای مربوط به چند سالن مختلف را **ذخیره کند**. هر سالن تعدادی صندلی در ردیف‌های مختلف دارد. مشتریان می‌توانند **صندلی رزرو کنند** (رزرو صندلی دارد) که به آنها شماره صندلی و شماره ردیف در سالن **داده می‌شود**. مشتری می‌تواند رزرو چندین صندلی مجاور را **درخواست کند**.

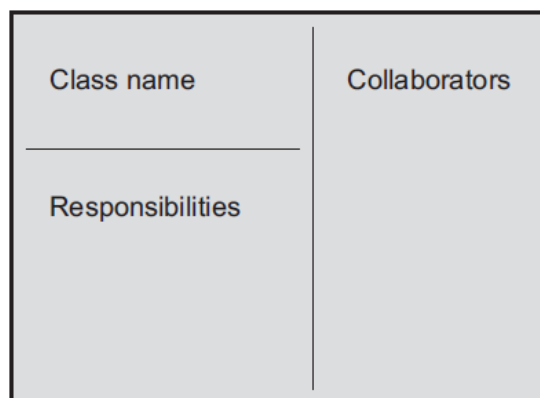
هر رزرو مربوط به یک نمایش خاص است. (منظور نمایش یک فیلم در یک زمان خاص است) نمایش‌ها در یک ساعت، تاریخ و سالن مشخص **برنامه ریزی می‌شوند**. سیستم شماره تلفن مشتری را **ذخیره می‌کند**.

بعد از مشخص کردن اسم‌ها و فعل‌ها، هر اسم به عنوان کاندیدای یک کلاس و هر فعل، عمل^۲ مربوط به کلاس‌ها در نظر گرفته می‌شود. نام کلاس‌ها را به طور مفرد به کار می‌بریم؛ مثلاً مشتری به جای مشتریان. زیرا با ساختن شی‌ها می‌توانیم چندین نوع از مشتری داشته باشیم.

^۲ Action

در این مرحله تمام اسم‌ها و فعل‌ها را مشخص می‌کنیم و چیزی را از قلم نمی‌اندازیم. برای مثال، با این فرض که «ردیف تنها می‌تواند یک مقدار int باشد، پس لازم نیست یک کلاس برای آن در نظر گرفت»، ردیف را از لیست کلاس‌ها حذف خواهیم کرد. در ادامه تحلیل پروژه، مشخص می‌شود که آیا برای موجودیت ردیف نیازمند طراحی یک کلاس هستیم یا خیر.

- در این مرحله کارت‌های CRC (Class/Responsibilities/Collaborators) را ایجاد می‌کنیم. هر کارت نماینده یک کلاس از سیستم است. شکل زیر نمای هر کارت را نشان می‌دهد:



- مرحله بعد این است که با استفاده از سناریوهای مختلف تعریف‌شده در پروژه، کارت‌های CRC را تکمیل کنیم. بهتر است که این کار به صورت گروهی انجام شود و هر فرد مسئول یک یا چند کارت باشد. سناریوها را یک به یک انجام می‌دهیم تا مشخص شود چه کلاس‌هایی در تعامل هستند و هر کلاس چه کاری انجام می‌دهد. مسئول کارت اطلاعات مربوطه را در آن وارد می‌کند.

برای مثال سناریوی زیر را در نظر بگیرید:

مشتری به سینما تلفن می‌زند و تقاضای رزرو دو بلیت برای فیلم جدایی نادر از سیمین (یا به وقت شام) برای امشب را دارد. کارمند سینما از سیستم رزرو بلیت به شرح زیر استفاده می‌کند:

۱. کارمند سینما می‌خواهد همه نمایش‌های امشب جدایی نادر از سیمین را ببیند. پس روی کارت مربوط به سیستم رزرو بلیت سینما در قسمت مسئولیت‌ها عبارت «توانایی یافتن نمایش‌ها بر اساس عنوان و روز» را اضافه می‌کنیم و به عنوان همکار کلاس Show را اضافه می‌کنیم.
۲. چطور سیستم نمایش‌ها را ذخیره می‌کند تا بعداً بتواند آنها را جست و جو کند؟ می‌تواند از Collection مثل ArrayList استفاده کند. پس روی کارت سیستم رزرو بلیت سینما در قسمت همکاران کلاس Collection را اضافه می‌کنیم.

۳. نتیجه جست‌وجو، سه سانس مختلف ۴:۰۰، ۶:۰۰ و ۸:۰۰ است. کارمند از مشتری ساعت مورد نظرش را می‌پرسد و اطلاعات مربوط به آن نمایش مثل ظرفیت، سالن و ... را در اختیار مشتری می‌گذارد. پس در مسئولیت‌های کارت سیستم رزرو بلیت سینما «نمایش جزئیات یک نمایش» را قرار می‌دهیم و روی کارت Show «ارائه جزئیات سالن و ظرفیت» در قسمت وظایف اضافه می‌کنیم.

۴. فرض کنید تعداد صندلی‌های خالی زیاد است. پس مشتری صندلی‌های ۱۳ و ۱۴ از ردیف ۵ را انتخاب می‌کند. روی وظایف کارت سیستم رزرو بلیت سینما «قبول رزرو صندلی از مشتری» را اضافه می‌کنیم.

۵. اطلاعات مربوط به نمایش در کلاس Show موجود است. پس روند رزرو به شی Show مربوطه داده می‌شود. روی کارت Show نوشته می‌شود، «توانایی رزرو صندلی».

۶. در کلاس Show برای انجام عملیات رزرو صندلی از شی Theater استفاده می‌شود که نمایی از سالن را در اختیار دارد. روی کارت Show می‌نویسیم «توانایی ذخیره سالن» و به عنوان همکار کلاس Theater را هم اضافه می‌کنیم.

۷. در کلاس Theater در قسمت وظایف «قبول درخواست رزرو» را می‌نویسیم. حال دو حالت برای ذخیره‌سازی صندلی‌ها وجود دارد: ۱- مجموعه‌ای از ردیف‌ها که هر ردیف مجموعه‌ای از صندلی‌هاست. ۲- مجموعه همه صندلی‌های سالن. مورد اول بهتر است چون برای پیدا کردن مثلاً ۴ صندلی مجاور در یک ردیف کار آسان‌تر خواهد بود. پس آن را انتخاب می‌کنیم. روی کارت در قسمت همکاران کلاس Row و Collection و در قسمت وظایف «ذخیره ردیف» را اضافه می‌کنیم.

۸. در کارت Row «ذخیره مجموعه‌ای از صندلی‌ها» و Seat را به عنوان همکار اضافه می‌کنیم. همچنین روی کارت Row «دریافت درخواست رزرو صندلی» و «یافتن صندلی با شماره» را اضافه می‌کنیم. همچنین روی کارت Seat «دریافت رزرو» و «نگهداری وضعیت صندلی(خالی/پر)» را اضافه می‌کنیم.

در ۸ مورد بالا یک سناریو را کامل بررسی کردیم و کارت‌های مربوطه را نیز تکمیل کردیم. سناریوهای مختلفی برای این سیستم موجود است:

۱. یک مشتری ۵ صندلی مجاور را می‌خواهد. چطور سیستم صندلی‌های مجاور را پیدا می‌کند و پیشنهاد می‌دهد؟

۲. مشتری می‌گوید شماره صندلی‌های که رزرو کرده است را فراموش کرده است. سیستم چطور رزرو دیروز او را نشان می‌دهد؟

۳. مشتری می‌خواهد رزرو خود را لغو کند. او نام خود و اطلاعات نمایش را دارد ولی شماره صندلی‌های خود را فراموش کرده است!
۴. یک مشتری می‌خواهد به رزرو روز گذشته خود، یک صندلی مجاور صندلی‌های قبلی اضافه کند.
۵. نمایش لغو شده است و سینما می‌خواهد به همه مشتریان خود اطلاع‌رسانی کند.
۶. سیستم باید برای تعریف سینمای جدید آماده شود که ۲ سالن با ظرفیت معلوم دارد.
۷. یک فیلم جدید قرار است برای دو هفته هر روز در سه نوبت نمایش داده شود. اطلاعات نمایش باید در سیستم ثبت شود.

انجام دهید

- با مشورت با مدرس کارگاه تعدادی از سناریوهای بالا را انتخاب کرده و کارتهای CRC مربوطه و نحوه تعامل آنها را استخراج کنید و به مدرس کارگاه ارائه دهید. همچنین در مورد تعدادی از سناریوهای ممکن دیگر در این سیستم فکر کنید و آنها را به مدرس ارائه کنید.

طراحی کلاس‌ها

در این مرحله باید اطلاعات مربوط به کارتهای CRC را به کلاس‌های به زبان جاوا تبدیل کنیم. در کارتهای CRC اطلاعات کامل در مورد هر کلاس و ارتباط بین آنها مشخص شده است. با توجه به مفاهیم cohesion و coupling که در درس نیز به آنها پرداخته شده است، سعی می‌کنیم تا بهترین طراحی را برای کلاس‌ها داشته باشیم. برای مثال، باید کمترین میزان از coupling میان کلاس‌ها وجود داشته باشد. ساده‌ترین اصل برای رعایت این موضوع، private کردن همه فیلدهای کلاس‌ها است. در کارتهای CRC با توجه به سناریوهای مختلف، کلاس‌های همکار مشخص شده‌اند. لازم است در این مرحله یک بار دیگر این اطلاعات بازبینی شوند تا از ارتباط بی مورد و بیش از حد کلاس‌ها با یکدیگر خودداری شود. همچنین باید کلاس‌ها بیشترین میزان cohesion را داشته باشند؛ یعنی این که یک متد اجراکننده یک وظیفه و یک کلاس نماینده یک موجودیت باشد. مجدداً کارتها را بررسی می‌کنیم تا ببینیم مجموعه وظایف تعریف‌شده برای یک کلاس نشان‌دهنده تنها یک موجودیت در سیستم است یا خیر. اگر این طور نباشد، باید کارتها به قسمت‌های کوچک‌تری تقسیم شوند.

همچنین در طراحی خود باید تغییرات احتمالی آینده را نیز در نظر بگیریم. برای مثال اگر قرار است در آینده سیستم دارای رابط کاربری گرافیکی باشد و فعلا ما از طریق کنسول سیستم را پیاده‌سازی می‌کنیم، باید وظیفه نمایش اطلاعات که فعلا با متد `System.out.println()` است را به یک کلاس مستقل برای نمایش و ارتباط با کاربر منتقل کنیم؛ مثلا کلاس `Display` را به نحو مناسبی طراحی و پیاده‌سازی کنیم.

انجام دهید

در ادامه، تعریف پروژه سیستم متمرکز سلامت آورده شده است. با استفاده از مطالبی که در این دستور کار با آن آشنا شده‌اید (روش فعل‌ها/اسم‌ها، کارت CRC و...) سعی کنید طراحی مناسبی از این پروژه ارائه کنید. از آنجایی که تعریف داده شده یک تعریف پروژه واقعی است، با مشورت با مدرس کارگاه حدود کار خود را مشخص کنید:

سیستم متمرکز سلامت

وزارت بهداشت خود قصد دارد سیستم متمرکز سلامت را در کشور راه‌اندازی کند. هدف از ایجاد این سیستم، نگهداری سابقه پزشکی افراد به منظور سهولت در دسترسی به آنها در نقاط مختلف جغرافیایی کشور است. علاوه بر آن وزارت بهداشت قصد دارد این اطلاعات را برای انجام تحقیقات در اختیار دانشگاه‌های تحت نظارت خود قرار دهد. علاوه بر آن بیماران نیز می‌توانند به سابقه پزشکی خود دسترسی داشته باشند و مجوز استفاده از اطلاعات پزشکی خود را که بدون ذکر نام صادر نمایند. در کنار این سیستم، با توجه به نوع بیماری و داروهای تجویز شده توسط پزشک اطلاعات در اختیار پزشک داروخانه هم قرار می‌گیرد تا بیمار بدون نیاز به استفاده از دفترچه دارو خود را دریافت کند.

اطلاعات بیمار شامل تاریخ تولد، جنسیت، نوع بیمه پایه، نوع بیمه تکمیلی، مدرک تحصیلی، شغل، محل زندگی و موقعیت جغرافیایی است.

سابقه پزشکی شامل نوع کلی بیماری، شرح حال بیمار، تشخیص پزشک، سطح امنیتی بیماری، داروی تجویز شده، تاریخ، و پزشک معالج است. نوع کلی بیمار شامل خانواده‌های اصلی بیماری‌ها می‌باشد. مانند ارتوپدی، قلب، کلیه، زنان، گوارش و غیره. سطح امنیتی بیمار نشان‌دهنده اهمیت بیماری از لحاظ افشا اطلاعات است. مثلا بیماری ایدز دارای سطح امنیتی بالا ولی سرماخوردگی در سطح امنیتی پایین است (فرض کنید سه سطح امنیتی بالا، متوسط و پایین وجود دارد).

سابقه پزشکی بیمار فقط توسط پزشک مورد اعتماد او قابل ایجاد است. اولین پزشک مورد اعتماد توسط بیمار انتخاب می‌شود. از آنجایی که بیماری‌های مختلفی وجود دارد، پزشک مورد اعتماد فقط می‌تواند برای درمان یک بیماری خاص، بیمار را به پزشک جدید ارجاع دهد. در این حالت پزشک جدید به لیست پزشکان مورد اعتماد فرد اضافه می‌شود و به سابقه پزشکی فرد دسترسی پیدا خواهد کرد. این دسترسی می‌تواند به دو صورت تعریف شود: دسترسی به کل سوابق پزشکی و یا تنها دسترسی به سوابق مربوط به بیماری‌هایی است که نوع کلی آن با بیماری فعلی یکسان است.