بنام خدا

عنوان مقاله : Object Oriented

ارائه دهنده :فهیمه علی اکبر وفا

نام استاد:آقای مهندس سلیمی

تاریخ:89/5/8

با روش شيءگرايي (object-oriented) يك برنامه را به قطعات خيلي كوچك بنام آبجكت‌هايي تقسيم مي‌كنيم كه تا اندازه‌اي مستقل از يكديگر مي‌باشند و به آن مانند ساختماني از بلوك‌ها نگاه مي‌كنيم. پس ابتدا بايد انواع مختلف بلوك‌ها را بسازيم .اولين باري كه ما اين بلوك‌ها را داريم، مي‌توانيم آنها را كنار هم گذاشته ساختمان را بسازيم و در اين صورت مي‌شود از اين آبجكت‌ها استفاده مجدد كرد.

پس از اين از كاربران بايد بپرسيم كه چه نيازهايي دارند و كاري نداريم كه با اطلاعاتي كه نياز دارند چه كاري مي‌خواهد انجام گيرد يا رفتار سيستم چگونه است. اين كارdata-centric (مبتني بر داده) ناميده مي‌شود و مدل مذبور مخصوص طراحي پايگاه داده و گرفتن اطلاعات خيلي با اهميت است. اما انتخاب اين روش در زمان طراحي مشكلاتي دارد چون درخواست‌هايsystem چند بار عوض مي‌شود. ولي بايد بدانيم كه تغيير رفتار سيستم زياد هم ساده نيست و روش شيءگرايي براي همين ايجاد شده است چون مي‌توانيم هم بر اطلاعات و هم بر رفتار سيستم تمركز كنيم. پس بايد encapsulation و polymorphism و inheritance كه بنياد آبجكت‌گرايي است را بشناسيم.

مزاياي رويكرد شيءگرا

مطرح بودن و كاربرد روز افزون رويكرد شيءگرا در ايجاد و توسعه سيستم‌هاي اطلاعاتي، اصولاً مبتني بر دلايلي منطقي است . اين رويكرد ، حل سه مسأله عمده موجود در رويكردهاي متداول توسعه سيستم يعني كيفيت، بهره‌وري و انعطاف را نشانه گرفته است . سيستم‌هاي توسعه‌يافته در رويكردهاي قبلي شديداً مستعد اشكال، پرهزينه ، و نامنعطف بوده‌اند . رويكردهاي شيءگرا پتانسيل كاستن از اشكالات و هزينه‌ها و افزايش ميزان انعطاف را بنا به طبيعت و امكانات ذاتي خود دارا مي‌باشد.

اصلاح و گسترش امكانات سيستم با سهولت بيشتري امكان‌پذير و عملي است ، و چون هر كدام از اشياء واحدهايي خودكافي هستند كه بدون درگيري با بقيه سيستم قابل تغيير و جايگزيني مي‌باشند. اين منافع و مزايا، نيروي محركه بالقوه حركت شيءگرايي را تشكيل مي‌دهند .

و اما اگر بخواهيم به طور خلاصه مزاياي شي گرايي را نام ببريم بايد بگوييم:

1 - مخفي بودن جزئيات پياده سازي اشياء

2 - ارتباط ساده بين اشياء (توسعه پذيري ساده)

3 - تسهيل استفاده از متخصصين

4 - قابل پشتيباني

شيىء (Object) :

به هر شيئ يا مفهوم يا هر چيز قابل دركي كه بشود با خصوصيات و رفتار مستقل آنرا از يک محيط بازشناسي كرد Object گفته مي شود . هر شيئ يكسري خصوصيات دارد كه به آنهاAttribute گفته مي شود كه در واقع يک مقدار يا ارزش مشخصي براي آن به ازاي هر شيئ مي تواند وجود داشته باشد . از طرف ديگر هر شيئ يكسري رفتار دارد كه به آنها Method گفته مي شود كه در واقع پاسخ هايي است كه آن شيئ كه به تحريكات يا پيغامها يا رخدادهايي كه از محيط خود صادر ميشود مي دهد . وبالاخره هر شيء را چيزي مي‌دانيم كه افراد راجع به آنها مي‌انديشند، آن را شناسايي مي‌كنند، روي آن كار مي‌كنند ، يا مفاهيمي را در رابطه با آن به كار مي‌برند. افراد مختلف مي‌توانند از يك شيء واحد درك‌هاي متفاوتي داشته باشند ، و به همين دليل واقعيت شيء هم به ديدگاه مشاهده كننده بستگي دارد، و هر كس شيء را با توجه به مشخصه‌ها و رفتارهاي مهم و مورد نظر خود توصيف مي‌نمايد .

انواع اشياء در سيستم‌هاي رايانه‌اي

اگر بتوانيم هر چيزي را يك شيء به حسات آوريم، در اين صورت پذيرش هر چيز به كار رفته در سيستم رايانه‌اي به عنوان شيء هم ميسر خواهد شد .

شیء در سیستم =» نقشها ، چیز های ملموس ، رویدادها با تعاملات ، دستگاه های جانبی ، واحد های سازمانی ، محل سایتها.

اشياءسيتم‌هاي رايانه‌اي بر اساس نوع كاربرد يا به عنوان جزئي از سيستم كه از ديدگاه ايجاد كننده اهميت دارد طبقه‌بندي مي‌شود :

» رابطه‌هاي كاربران

رابطه‌هاي كاربران اشيائي هستند كه روي صفحه نمايش رايانه ظاهر مي‌شوند و كاربران نهايي به طور مستقيم با آنها تعامل دارند. اين اشياء مؤلفه‌هايي دارند، رفتارهايي دارند،بايكديگر در تعامل هستند ، و مهمتر از همه اينكه ما هم با آنها تعامل داريم. كاركردن مستقيم يك كاربر نهايي با اشياء رابط يك وضعيت كاملاً شيءگرا است . اشياء رابط بسياري وجود دارند كهwidgetياكنترل ناميده مي‌شوند و استفاده از آنها متداول است ،مثل button، Scroll bar،

text box، down list drop، radio button و غيره . وقتي كه بعضي از ايجادكنندگان سيستم در فرآيند توسعه سيستمها سخن از كاربرد فناوري شيءگرا به ميان ميآورند ، در واقع مقصدشان استفاده از اشياء موسوم به رابطهاي كاربران به منظور ايجاد رابط گرافيكي كاربرد در سيستم مورد نظر است ،و بقيه سيستم ممكن است كه در اصل شيءگرا هم نباشد. به همين دليل است كه براي بعضي از افراد،شيءگرا با رابط گرافيكي كاربرد مترادف است .

» اشياء محيط عملياتي

شيء محيط عملياتي، نوع ديگري از اشياء سيستم رايانهاي است كه در جايي از يك شبكه رايانهاي قرار دارد يا به وسيله سيستمعامل كنترل ميشود . به عنوان مثالهر يك از دو مفهومserver وclient در معماري client/server شيءبه حساب ميآيند . اين معماري به واقع مانند سيستمي از اشياء متعامل است و به همين دليل بعضي از ايجادكنندگان سيستم ، شيءگرايي را مترادف با اين معماري ميپندارند .

» اشياء مربوط به كار

اين اشياء كه براي انجام كارها مورد استفاده قرار ميگيرند، چيزهايي هستند كه كاربردهاي رايانهاي با آنها سروكار داشته يا آنها را ايجاد ميكنند . اشيائي از نوع مستندات ، اشيائي از نوع كاربردهاي چند رسانهاي و چيزهايي كه آنها را اشياء زمينه كار يا اشياء حوزه كار خواهيم ناميد، در اين گروه قرار ميگيرند .

» اشياء از نوع مستندات

مستندات هم اشيائي هستند كه چيزهايي را مي شناسندو نحوه انجام كارهايي را مي دانند.در هنگام استفاده از يك واژه پرداز, ابتدا از يك مستند مي خواهيم كه باز شودو سپس مي توانيم ازآن بخواهيم كه مقداري متن جديد (مثلا چند پاراگراف جديد) را بپذيرد, شكل يك پاراگراف راتغيير دهد, فونتهاراعوض كند, يا قطعه متني راازجايي حذف و به جاي ديگري اضافه نمايد.ميتوانيم بخواهيم كه از لحاظ املايي كنترل كند يا خود راذخيره و سپس ببندد.
ميتوانيم بخواهيم كه از لحاظ املايي بسياري مانند نام, عنوان, حاشيه هاي عمودي, اندازه و شكل حروف, تعداد صفحات, مشخصات ايجاد كننده, وتاريخ ايجاد خود را هم ميداند.Worksheet هم شيء ديگري از انواع مستندات است و اكثر كاربران با آن تعامل و آشنايي دارند . اين شيء هم چيزهايي را ميشناسد و نحوه انجام كارهاي زيادي را ميداند كه در اجراي محاسبات مكرر بسيار مفيدند . نمايش اسلايد هم كه با استفاده از يك كاربرد گرافيكي ايجاد

مي‌شود ،يك شيء‌از نوع مستند است . كاربرد مي‌تواند اشيائي از نوع مستندات را كه در يك كاربرد ايجاد شده‌اند ، در درون مستندي كه با استفاده از كاربرد ديگري ايجاد گرديده است ، جاسازي نمايد و اين نياز در كاربردهاي اداري بسيار پيش مي‌آيد . با توجه به اين موارد ، جاي تعجب نيست كه بسياري از ايجادكنندگان سيستم ، شيء‌گرايي را مترادف با پردازش مستندات تلقي مي‌كنند .

» اشياء چند رسانه اى

سيستم‌هاي چند رسانه‌اي، نوع مهم ديگري از كاربردها هستند كه حاوي صدا، تصوير، انيميشن ،و ويدئو مي‌باشند. سيستم‌هاي چند رسانه‌اي همچنين شامل اشيائي هستند كه گاهي اشياء بزرگ دودوئي ناميده مي‌شوند. كاركردن با سيستم‌هاي چند رسانه‌اي بايد امري كاملاً طبيعي باشد ، زيرا اشياء آنها رفتاري همانند اشياء دنياي واقعي داشته و تعامل ما با خود را آسان مي‌كنند. درخواست براي ايجاد سيستم‌هاي كاربردي چند رسانه‌اي ، يا گنجاندن امكانات چند رسانه‌اي در بيشتر كاربردها روند فزاينده‌اي دارد. بعضي از ايجادكنندگان سيستم اساساً شيء‌گرائي راچند رسانه‌اي مترادف مي‌دانند.

» اشياء زمينه يا حوزه كار

تصور اشيائي از نوع مستندات يا چند رسانه‌اي به عنوان اشياء يك سيستم رايانه‌اي نسبتاً آسان است، چون اولاً : اكثر آنها را ديده و با آنها تعامل داشته‌ايم ، و ثانياً : اغلب شبيه اشياء دنياي واقعي بوده و مانند آنها رفتار مي‌كنند . آخرين گروه از اشياءْ مربوط به كار در سيستم‌هاي رايانه‌اي متفاوت از موارد مذكور است . اكثر افراد تماس و تعاملي با آنها نداشته‌اند. اين اشياء در سيستم‌هاي رايانه‌اي از لحاظ شكل و رفتار معمولاً با همانند خود در دنياي واقعي مشابهت ندارند . اين اشياء متناظر با چيزهايي هستند كه در هنگام مدلسازي داده‌ها مشخص مي‌شوند . براي بسياري از دست‌اندركاران ايجاد سيستم و به ويژه تحليلگران سيستم‌هاي تجاري، شيء‌گرايي با اشياء زمينه يا حوزه‌كاري مترادف است .

كـلاس Class :

براي دسته‌بندي اشياء مفهومي به نام كلاس تعريف مي‌گردد . كلاس نشان دهنده خصوصيات و رفتار گروه خاصي ازاشياء مي‌باشد كه در آن خصوصيات و رفتارها مشترک هستند. در واقع كلاس يک مفهوم انتزاعي است و در محيط عملياتي كلاس واقعا وجود ندارد وobjectاي ازآن كلاس وجود دارد . به عنوان يک مثال ديگر كلاس دانشجو -object دانشجو به شماره ۱۲۳۴۵۶۷۸ . پس كلاس يک مفهوم‌است وobject يک واقعيت .

هر كلاس هم مانند شيئ يک نام دارد - يکسريAttribute و يکسري Method . که در واقع Attribute ها رفتار object هاي آن کلاس هستند و Method ها پاسخ هاي object هاي آن کلاس به حساب مي آيند. اما روشن است که اشياء داخل يك كلاس بايد به طريقي قابل شناسايي و بازشناسي از يکديگر باشند.براي شناسايي هر شيءمعمولاً يك مؤلفه يا بخشي از اطلاعات مشخصهها آن به کار مي‌رود . در مدل‌سازي داده‌ها هم يك شناسه لازم است . بسياري از اشياء داراي شناسه طبيعي هستند ،مواردي هم پيش مي‌آيد که شيء فاقد شناسه منحصربه‌فرد مي‌باشد ، در اين ضروري است که شناسه‌اي توسط سيستم تعيين گردد .

تمايز بين کلاس‌ها و اشياء بسيار مهم است . شايد رويکرد شيءگرا بتواند تفاوت آنها را نشان دهد ، تفاوتي که اهميت آن در زمان تعريف رفتارهاي کلاس و شيء به خوبي نمايان مي‌شود .

Instantiation: به معناي ايجاد يک شيئ از يک کلاس است. يعني نمونه‌اي از آن کلاس که همان شيئ مي‌باشد توليد مي‌گردد .

مؤلفه‌هاي يك كلاس

هر کلاس از اشياء غالباً به وسيله مؤلفه‌هايي تعريف مي‌شود که در بين همه اشياء آن کلاس مشترك هستند .هر مؤلفه عنوان يك اطلاعي است که ما مي‌خواهيم درباره اشياء کلاس داشته باشيم، و بديهي است که يك مؤلفه از هر شيء مي‌تواند مقداري متفاوت از ساير اشياءهمان کلاس داشته باشد . مؤلفه‌هاي هر کلاس از اشياء ، معادل با مؤلفه‌هاي يك نوع موجوديت داده‌اي است .

روابط اشياء

طبيعي است که هر شيء مي‌تواند با اشياء ديگري مرتبط باشد. روابط اشياء به روابط موجود در مدل‌هاي داده‌اي شباهت زيادي دارد . رابطه عبارت از يك هم‌بستگي است که بر پايه زمينه نگرش ما به اشياء شکل مي‌گيرد . هر شيء از اشياء مرتبط با خود هم چيزهايي مي‌داند. اشيائي هم که تحت عنوان رابط کاربر معرفي گرديدند، مي‌توانند با اشياء ديگر مرتبط باشند. روابط بين اشياء هم معمولاً مانند روابطي که در مدل‌هاي داده‌اي ديده‌ايم در دو جهت نامگذاري يا تعريف مي‌شوند.

ماهيت روابط بين اشياء را هم عيناً مانند مدل‌سازي داده‌ها مي‌توان با دقت بسيار تعريف نمود، هر رابطه مي‌تواند اختياري يا اجباري باشد . منظور از رابطه اختياري اين است که يك شيء ممکن است با شيء ديگري مرتبط باشد . رابطه اجباري حاکي از حتمي بودن و ضرورت وجود ارتباط بين اشياء مورد نظر است . در مدل‌سازي داده‌ها معمولاً قواعدي براي نشان دادن يا اجباري بودن رابطه‌ها وجود دارد، و اين قواعد به همان صورت براي اشياء هم قابل

استفاده است .

جنبه ديگري از روابط اشياء كه با مدل‌سازي داده‌ها شباهت كامل دارد، كاردينالیته هر رابطه است كه وضعيت رابطه بين اشياء را نشان مي‌دهد و يك رابطه يك به يك است . رابطه بين اشياء مي‌تواند چند به چند هم باشد ، البته هر رابطه چند به چند نشان‌دهنده دو رابطه مستقل يك به چند است .

Encapsulation:

در سيستم‌هاي object oriented اين رفتارها را در يك object بسته‌بندي مي‌كنيم و اسم اين عمل پنهان‌سازي يا encapsulation است. مثلا در مورد بانك، همه تغييرات سيستم بانكي مربوط به حساب‌ها مي‌شود به آساني در آبجكت حساب انجام گيرد. از ديگر مزاياي پنهان‌سازي اين است كه تاثيرات اعمال شده به سيستم را محدود مي‌كند.

اگر به سيستم (system) به عنوان بستري از آب و به تغيير درخواست‌ها به عنوان صخره نگاه كنيم وقتي صخره را در آب مي‌اندازيم امواج در همه جهت‌ها ايجاد مي‌شود، آنها در سر تا سر درياچه حركت مي‌كنند، به كرانه ضربه مي‌زنند، طنين‌افكن مي‌شوند و با ديگر امواج برخورد مي‌كنند و حتي ممكن است مقداري آب بر روي ساحل و خارج از درياچه بريزد. يعني برخورد صخره با آب باعث ايجاد ميزان زيادي موج‌هاي كوچك مي‌شود،حالا درياچه خود را پنهان مي‌كنيم پس آن را به تكه‌هاي كوچكي از آب با موانعي ميان آنها تقسيم مي‌كنيم ، بعد ضربات سيستم تغيير مي‌كند قبل از اين امواج در همه جهت‌ها ايجاد مي‌شدند اما حالا امواج فقط مي‌توانند از يكي از اين منابع عبور و بعد متوقف مي‌شوند ، پس با نهان‌سازي درياچه ما تاثير موج كوچك حاصل از انداختن صخره در آب را محدود كرديم.

در مورد سيستم بانكي هم وضع به همين روال است. مثلا: اخيرا مديريت بانك تصميم گرفته است كه اگر مشتري در بانك يك حساب اعتباري دارد بتواند از حساب يك مبلغ برداشت كرده و براي حساب جاري استفاده كند. در يك سيستم غيرپنهان سازي كار را با يك روش اجباري شروع مي‌كنيم تا تجزيه و تحليل كاراتر شود. كلا بايد همه جا را بگرديم و همه جاهايي كه از عمليات برداشت از حساب در سيستم استفاده شده است را پيدا كنيم. آن وقت درخواست جديد را يكپارچه مي‌نماييم، اگر واقعا كار به درستي انجام گرفته باشد احتمالا80 درصد موارد برداشت از حساب را پيدا كرديم، با يك سيستم نهان‌سازي ما نيازي به استفاده از روش اجباري براي تجزيه نداريم. ما به مدل سيستم خود نگاه مي‌كنيم و به آساني جايي كه رويداد برداشت از حساب پنهان شده بود را پيدا مي‌كنيم. بعد از اينكه عمليات را در حساب قرار داديم يك بار درخواستمان را فقط در آن آبجكت تغيير مي‌دهيم و ديگر هيچ كاري نداريم.

مفهوم ديگر نهان‌سازي information hiding يا پنهان سازي اطلاعات است. پنهان‌سازي اطلاعات توانايي است كه جزئيات مبهم يك شي را از بقيه پنهان مي‌كند. در يك آبجكت دنياي خارج يعني هر چيزي خارج از همان شي است. پنهان سازي اطلاعات همان مزيت نهان سازي را فراهم مي‌سازد.

وراثت:

وراثت يكي ديگر از مفاهيم object گرايي است در سيستم شي گرائي وراثت به ما اجازه مي‌دهد تا شي جديد را بر مبناي شي قبلي بسازيم يعني بعضي خصوصيات شي قبلي را به ارث مي‌برد. اگر مثالمان را در مورد جانوراني به كار بريم كه پستاندار باشند، سگ‌ها، گربه‌ها، انسان‌ها، نهنگ‌ها و ... هستند كه هر كدام ويژگي خاصي دارند و داشتن مو، خونگرم بودن و غذا دادن به بچه‌ها در همه‌شان مشابه است در شي پستانداران اين شيparent، اشياي (object) فرزند هم دارند مثلا انسان و نهنگ. يكي از مزيت‌هاي وراثت راحتي در نگهداري است يعني وقتي چيزي تغيير كند فقط كافي است parent object را تغيير داد و تغييرات به خودي خود روي فرزندان سرايت مي‌كند و آنها اين تغييرات را به ارث مي‌برند.اگر ما 125 عدد تابلو داشته باشيم و يك مشتري درخواست يك پيغام انصراف را بر روي همه تابلوها مي‌دهد، اگر وراثت نداشته باشيم بايد به هر يك از125 تابلو برويم و تغيير را در آن به وجود آوريم. اگر object oriented در سيستم باشد، چون تمام تابلوها را از يك شيparent ارث مي‌بريم كافي است تا به شي پدر رفته و يك بار آن را تغيير دهيم تا همه تابلوها خود به خود تغييرات را به ارث مي‌برند.

چند ريختي:

سومين مفهوم بنيادي object oriented چند ريختي است. چند ريختي يعني ما اشكال بسياري از يك تابع ويژه داشته باشيم مثل وراثت، چند ريختي هم در دنياي واقعي ديده مي‌شود. مثلا در فرمان صحبت كردن يك انسان مي‌گويد، شما چطوريد، هر يك از جانداران به زبان خود. ولي همه اينها در اصل يك عكس‌العمل است.

يكي از مزايا چند ريختي مثل ديگر اصول object oriented سهولت نگهداري است.

آناليز شي گراء (OOA)

آناليز شي گراء و يا OOA ، يک متدولوژي قدرتمند براي تجزيه و تحليل فرآيند پياده سازي نرم افزار است . در زمان استفاده از OOA ، هر چيز در فرآيند پياده سازي نرم افزار بمنزله كلاس در نظر گرفته خواهد شد (اين طرز تفكر مي بايست محور آناليز سيستم قرار گيرد) . مثلا" در يک بيمارستان هر يک از عناصر موجود نظير : دكتر ، پرستار ، بيمار و ملاقات كننده ، بمنزله يک كلاس در نظر گرفته مي شوند . هر نسخه جديدي كه از يک كلاس ايجاد مي گردد ،

بمنزله يک نمونه (Instance) از کلاس در نظر گرفته خواهد شد . محوريت فرآيند آناليز شي گراء ، تاکيد بر ايجاد کلاس هاي مورد نياز سيستم است .

مهمترين و اصلي ترين رويکردOOA ،يافتن پاسخ مناسب براي سوالاتي است که باWhat شروع و در فرآيند پياده سازي نرم افزار حضوري موثر دارند . نمونه سوالاتOOA در اين زمينه عبارتند از : - چه کلاس هائي در برنامه وجود دارد؟- چه چيزي را برنامه انجام خواهد داد ؟- هر يک از کلاس ها در برنامه چه عملياتي را بمنظور حل مسئله انجام خواهند داد ؟- مسئوليت هر کلاس در برنامه چيست ؟ درOOA ، تاکيد بر آناليز اشياء ، فعاليت ها و مسئوليت هاي سيستم نرم افزاري است .

مروري بر فرآيند تجزيه و تحليل شـيءگرا

تجزيه و تحليل شيءگرا مشتمل بر چهار فعاليت اصلي تعريف سيستم ، تعريف وظيفهمندي سيستم يا سناريوهاي موارد کاربرد آن،ساخت مدل شيء ، و تدوين نهايي مستندات تجزيه و تحليل است . در اينجا فقط به فرآيند تجزيه و تحليل ميپردازيم ، که نوعاً براي تعريف مجموعه اطلاعات موجود در مدل نيازمنديها به کار ميرود.

» تعريـف نيازمندي

براي تعريف نيازمنديها در رويکرد شيءگرا نيز مانند رويکردهاي ساختيافته، راههاي مختلفي وجود دارد . تدوين سناريوها يا موارد کاربرد ، راهي طبيعي را براي تقسيم سيستم به بخشهاي کوچکتر قابل شناخت و ارزيابي در اختيار ما ميگذارد . ميخواهيم کارهايي را تعيين و تعريف کنيم که با اجراي آنها نياز کاربران مرتفع ميگردد . در ابتدا لازم است که رويدادهاي محتمل در محيط سيستم مشخص شوند و سپس به تدريج رفتار سيستم و مدل شيء تعيين و با استفاده از شرح سناريوها مستند گردد.

» ساختن مدل

اکثر متدهاي شيءگراي تجزيه و تحليل براي ساختن مدل شيء مجموعه فعاليتهاي تقريباً مشابهي را معرفي يا پيشنهاد ميکنند.

براي تهيه مدل شيء مفمولاً گامهاي زير اجرا ميشود :

1- شناسايي و تعيين اشياء و انواع کلاسها

2- شناسايي و تعيين سلسله مراتب طبقهبندي و سلسله مراتب کل-جزء

3- شناسايي و تعيين مؤلفههاي مهم

4- تعيين روابط اشياء

5- تعيين خدمات و پيام‌ها

6- مشخص كردن رفتارهاي وابسته به زمان

7- دسته‌بندي كلاس‌ها

بايد توجه داشت كه اين گام‌ها همواره به همين ترتيب و به‌طور كاملاً متوالي اجرا نمي‌شوند،ومعمولاًدر ين فعاليت‌هاي مختلف شاهدانجام كارهاي تكراري هم‌هستيم.

» يافتن اشياء و كلاس‌ها

كار مدل‌سازي شيءگرا توسط مشاورين، از شناسايي انواع اشياء و كلاس‌هايي آغاز مي‌شود كه در فرآيند كرايه تجهيزات نقشي دارند. ابداع كنندگان تجزيه و تحليل شيءگرا براي تحقق اين شناسايي رويكردهاي نسبتاً متفاوتي را ارائه كرده‌اند، اما در عمل ملاحظه مي‌كنيم كه شناخت زمينه مورد مطالعه، قواعد عرفي و تجربه شخصي تحليلگران، راهنمايي اصلي آنها در انتخاب راه و شناخت مطلوب اشياء است .

رويدادهاي كليدي و موارد كاربرد اشياء

به منظور حصول اطمينان از تناسب و كفايت كلاس‌هاي تعيين شده براي تهيه مدل نيازمندي‌ها، لازم است كه با شناسايي رويدادهاي كاري و موارد كاربرد اشياء يا تدوين سناريوهاي ممكن ، تصويري هر چند ابتدايي از موارد و راه‌هاي استفاده از انواع كلاس‌هاي اشياء در سيستم به دست آيد . بدين ترتيب ، معلوم مي‌شود كه مدل شيء در دست تهيه تا چه حد مي‌تواند پاسخ‌گوي رويدادها و موارد كاربرد اشياء در حوزه فعاليت مورد نظر ما باشد .

» شناسايي سلسله مراتب‌هاي طبقه‌بندي

به منظور پالايش هر كلاس در درون يك سلسله مراتب طبقه‌بندي مي‌توان از يك رويكرد از بالا به پايين يا از پايين به بالا استفاده نمود . در رويكرد بالا به پايين ،يك كلاس مورد نظر با هدف ايجاد يك سلسله مراتب طبقه‌بندي ، به كلاس‌هاي فرعي و تخصصي‌تري بسط داده مي‌شود . در رويكرد از پايين به بالا برعكس اين حالت ، كلاس‌هاي خاص را گرفته و در جهت رسيدن به يك كلاس كلي يا عمومي‌تر تلاش مي‌نمايند .

» شناسايي و تعيين مؤلفه‌ها

در اين مرحله به تعيين مؤلفه‌هاي كلاس‌هاي تعبيه شده در مدل شيء مي‌پردازند. بعضي از مؤلفه‌ها در طي تهيه سلسله مراتب‌هاي طبقه‌بندي و سلسله مراتب كل-جزء شناخته شده‌اند . به طور كلي ، در اكثر سيستم‌هاي

اطلاعاتي وجود شناخت نسبي از مؤلفه‌هاي مهم در طي فرآيند تحليل شيءگرا ضروري است ، اگر چه در بعضي از متدها شيءگرا تاطراحي فيزيكي و يا حتي پياده‌سازي سيستم نيازي به مؤلفه‌ها وجود ندارد. نكته قابل توجه اين است كه ، تكميل قطعي سيستم در اين مرحله اجباري نيست و در گام‌هاي بعدي هم ممكن است مؤلفه‌هاي بيشتري به آن اضافه شود .

» تعيين دسته‌ها يا موضوعات

در صورت پيچيده‌تر بودن مدل شيء يك نمايش ساده شده گرافيكي از آن مي‌تواند وسيله مناسبي براي مطالعه سيستم باشد . دسته‌ها يا گروه‌هايي از اشياء را مي‌توان به صورت واحدهايي انتخاب و با اضافه كردن ارتباطات بين آنها به عنوان مدل به مديريت يا كاربران ارائه نمود . اين واحدهاي يا سطح بالاتر را موضوع نيز مي‌نامند . براي هر سلسله مراتب طبقه‌بندي يا سلسله مراتب كل-جزء مي‌توانيم يك موضوع تعريف كنيم. در صورت بسيار بزرگ بودن سيستم در دست ايجاد بهتر است كه موضوعات مطرح سيستم در آغاز فرآيند تجزيه و تحليل تعريف شود، و در اين صورت پروژه هم بر اساس موضوعات قابل تقسيم خواهد بود .

طراحي شي گراء (OOD)

منظور از طراحي سيستم، تهيه و تدوين مشخصات تفصيلي اقدامات لازم براي پياده‌سازي سيستم رايانه‌اي است. بعضي از تصميمات طراحي متضمن فناوري خاصي نمي‌باشد، اما در هر صورت اين مشخصات معمولاً به عنوان يك مدل فيزيكي از سيستم شناخته مي‌شود . در حالي كه مي‌دانيم مشخصات تهيه شده در فرآيند تجزيه و تحليل سيستم يك مدل منطقي است ، كه بدون پرداختن به چگونگي پياده‌سازي طرح آنچه را كه لازم و ضروري به نظر آيد شرح مي‌دهد.

به اين ترتيب ، در طراحي شيءگرا لازم است كه با توجه به زبان برنامه‌نويسي و محيط توسعه‌اي كه براي پياده‌سازي به كار خواهد رفت به امر طراحي سيستم پرداخته شود . نكته اساسي در طراحي شي گراء ، تاكيد و سرو كار داشتن با سوالاتي است كه با How شروع و در فرآيند پياده سازي نرم افزار حضوري فعال و موثر خواهند داشت . " چگونه اين كلاس داده را جمع آوري مي كند ؟" . " چگونه اين كلاس گزارش را چاپ مي نمايد ؟" ، نمونه سوالاتي در اين زمينه مي باشند .در نمونه مثال بيمارستان، وضعيت فوق به خصلت ها ، صفات و متدهاي يک كلاس مرتبط مي گردد .

بنابراين OOA ، كلاس هاي مورد نظر و ضروري بمنظور نيل به اهداف نرم افزار را مشخص مي نمايد و محور عمليات بر جستجوو تبين جايگاه يک كلاس در برنامه متمركزاست . درOOD ، تاكيد بر پياده سازي كلاس ها ، صفات و

خصايصي است كه بمنزله هسته يک كلاس مطرح مي گردند . تركيب هر يک از فعاليت هاي فوق (آناليز شي گراء و طراحي شي گراء) بهمراه پياده سازي لينک هائي كه با كلاس ها سروكار دارند جملگي بعنوان بخشي از فرآيند OOP (برنامه نويسي شي گراء) محسوب مي گردند.