

به نام خدا

استاد درس
موضوع ارائه: مدل شی گرا_مدل رابطه ای
ارائه دهنده:

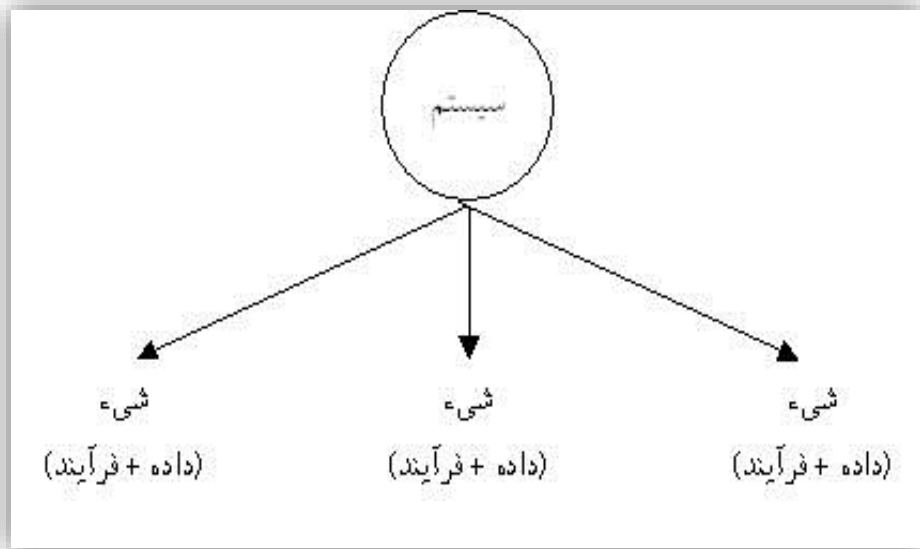
فهرست مطالب

- پایگاه داده شی گرا
- تاریخچه شی گرا
- معرفی مدل شی گرا
- ویژگی های مدل شی گرا.
- مزایا و معایب .
- جمع بندی.
- مدل رابطه ای
- تاریخچه مدل رابطه ای
- معرفی مدل رابطه ای
- مقایسه با مدل شی گرا
- مزایا .معایب
- نتیجه گیری:
- منابع

در دهه هفتاد میلادی، زبانهای چون Simula، Modula، Smalltalk طراحی شد که بر خلاف زبانهای رویه ای سنتی، زبانهای شیء گرا نامیده می شدند. اساس این زبانها که به طور خاص برای ساختن برنامه هایی در زمینه شبیه سازی و هوش مصنوعی طراحی شده بودند، بر تصویرسازی قلمرو کار بر مبنای اشیاء و کلاسها استقرار بود. در آن زمان استفاده از این زبانها چندان رواج نیافت، اما ایده های اساسی آن مانند وراثت، چند ریختی و برنامه نویسی مبتنی بر رویداد در یکی از موفقترین و متداول ترین زبانهای برنامه نویسی یعنی ++C به کار گرفته شد.

کسانی چون بوچ، فایراسمیت، کود و یوردون متدلوژیهای برای توسعه سیستم ها بر مبنای این ایده ها طراحی کردند. تحلیل و طراحی شیء گرا در دهه ۹۰ میلادی شرح و بسط زیادی پیدا کرده است و امروزه به نظر می رسد گرایش غالب در توسعه آتی متدلوژیهای مهندسی نرم افزار باشد.

در الگوی ساخت یافته، در اولین تقسیم، سیستم به دو وجه "داده" و "فرآیند" تفکیک می شود. سپس طی روندی سلسله مراتبی و مطابق با روش از بالا به پایین، هریک از این وجوه خود به اجزاء و مؤلفه های فرعی تجزیه می شوند. این روند تا به جایی ادامه پیدا می کند که عناصر قابل پیاده سازی (ساختار پرونده های فیزیکی، واحدهای برنامه ای) مشخص شوند.



اگر دو یا چند شیء در بخشی از مشخصه‌ها و رفتارهای خود مشترک باشند، می‌توان شیئی کلی‌تر از آنها را تصور کرد که در بر دارنده آنهاست. این اشیاء کلی را کلاس (رده) می‌گویند. میان اشیاء و کلاسها رابطه عضویت (تعلق) برقرار است. همچنین هر کلاسی می‌تواند خود عضو کلاس بالاتری باشد و همین‌طور الی آخر.

از دیدگاه شیء‌گرایی، قلمرو کار (یا قلمرو مسأله) از تعدادی شیء و ارتباط میان اشیاء تشکیل شده است. تعریف شیء آنقدر کلی است که تعریف دقیقی نمی‌توان از یک شیء به دست داد. هر چیزی ممکن است یک شیء باشد؛ پرونده، کاربر، دستگاه فیزیکی، واحد سازمانی، سیستم، و به‌طور کلی هر موجودیتی که دارای داده‌ها و رفتار مربوط به خود باشد داده، یعنی مشخصات اطلاعاتی که حالت شیء را در هر لحظه معین می‌کنند.

فرآیند، یعنی رفتاری که شیء در واکنش به محرکهای خارجی از خود نشان می‌دهد. بعد فرآیندی شیء را در متدلوژیهای شیء‌گرا، رفتار، سرویس و متد هم می‌نامند. داده‌های مربوط به یک شیء، به شکل مشخصه‌های آن شیء نشان داده می‌شود.

یکی از ویژگیهای متدلوژیهای شیء گرا وحدت مفهومی الگوها در کلیه مراحل زیستچرخ توسعه سیستم است. به این معنی که در طی مراحل تحلیل، طراحی و پیاده سازی سیستم، از مفاهیم، اصطلاحات و قالب فکری واحدی استفاده می شود

طراح مدل منطقی و فیزیکی سیستم را با استفاده از ساختار کلاسها و اشیاء ساخته میشود. برنامه نویس هم با استفاده از یکی از زبانهای شیء گرا، این طراحی را به کد تبدیل می کند. حتی برای طراحی بانکهای اطلاعاتی نیز روشها و ابزارهای شیء گرا وجود دارد. در این روش زبان مشترکی در کلیه مراحل توسعه سیستم به وجود می آید که امر ساخت سیستم و نگهداری و تغییر آن را آسان تر می کند.

در تحلیل و طراحی شیء گرا پنج فعالیت اساسی زیر باید انجام شود :

- یافتن کلاس - شیء ها
- شناسایی ساختارها
- تفکیک موضوعی
- تعریف مشخصه ها
- تعریف روالها

یافتن کلاس رده کسب شناخت از قلمرو کار سیستم و تشخیص رده ها و اشیایی است که نقش اساسی در کارکرد ایفا می کنند

تفکیک موضوعی

تفکیک موضوعی که می توان آن را شناسایی موضوعات نیز نام نهاد، مرحله ای است که در آن مدل تولید شده در مراحل قبل که شامل فهرست و ساختار کلاس ها و اشیاء سیستم است، به محدوده های موضوعی مختلف تفکیک می شود

فرآیند (که در روشهای مختلف شیء گرایبی به آن خدمت، روش یا رفتار نیز گفته می شود) عملیاتی است که یک شیء در پاسخ به دریافت یک پیام انجام می دهد. در این مرحله از OOA، فرآیندهای کلاس - شیءها تعیین می گردد.

لایه موضوعی در یک سیستم، می توان به تقسیم بندی تراکنش های یک مؤسسه بانکی به دریافت، پرداخت، گشایش اعتبار اشاره کرد. لایه موضوعی، خود ساختاری سلسله مراتبی دارد. یعنی هر موضوع ممکن است از چند موضوع فرعی تشکیل شده باشد.

شناسایی مشخصه ها

در این مرحله مشخصه های کلاس - شیءها تعیین می گردد. این مرحله با مرحله تعیین مشخصه ها در مدل داده ای روشهای سنتی متناظر است. نام مشخصه هایی که برای هر شیء تعریف می شوند، در قسمت میانی نماد کلاس - شیء نوشته می شود.

تعیین رابطه بین کلاس - شیءها. اگر بین دو کلاس - شیء ارتباطی وجود داشته چندبه چندی رابطه نیز در دو انتهای خط فاصل نوشته می شود.

ساختاری () Structured Query Language
فرقی ندارد. در واقع از همان ساختار SQL استفاده می
کنیم. همانطور مشخص است، اطلاعات در بانک های
اطلاعاتی معمول به صورت رکورد ذخیره می شوند.
اطلاعات در بانک های اطلاعاتی به صورت کلاس هایی از
اشیاء ذخیره می شوند.

مزیت های بانک های اطلاعاتی شیء گرا
اولین مزیتی که می توان برای بانک های اطلاعاتی شیء گرا
بیان نمود این است که طراح (Designer) می تواند ساختار
اشیاء و رفتار () behavior یا متدهای آنها را تعیین
کند. این نوع بانک های اطلاعاتی بازخورد مناسبی در زبان
های برنامه نویسی شیء گرا دارند؛ از جمله این زبانها به جاوا
(Java و C++)

از دیگر مزیت های این نوع بانک های اطلاعاتی می توان
به پنهان سازی عملیات و ساختار کلاس از دید کاربر و
همچنین متدهای تعریف شده توسط کاربر (طراح -
Designer) اشاره کرد .

زبان پرس و جو اشیاء (Object Query Language)

یکپارچگی؛ مهمترین مزیت متدولوژیهای شیء گرا، یکپارچگی مفهومی و ابزاری آنهاست. به این معنی که در طول زیست چرخ توسعه سیستم ها، از تحلیل گرفته تا پیاده سازی برای استفاده از روشهای شیء گرا، مفاهیم، واژگان و مدل‌های نسبتاً واحد و نزدیک به هم، به کار برده می شود. مفاهیمی مانند: شیء، کلاس، وراثت، چندریختی و... هم در تحلیل و طراحی شیء گرا مصداق دارند و هم در برنامه نویسی با زبانهای شیء گرا. زبان یکدست و واحدی به وجود می آید تا همه دست اندرکاران تهیه سیستم با استفاده از آن، با یکدیگر ارتباط برقرار کنند.

سازگاری با تغییرات؛ مدلهایی که با استفاده از روشهای OOA ساخته می شوند، در مقابل تغییرات نسبتاً انعطاف پذیری بالایی دارند و بسادگی می توان این مدلها را همگام با تغییرات، بهنگام کرد.

از امتیازات:متد شیء گرایی این است که می توانید یکبار اجزا را ساخته و بارها و بارها از آنها استفاده کنید شما یک برنامه را به قطعات بسیار کوچک تقسیم می کنید، که تا اندازه ای مستقل از یکدیگر می باشند.

مثال: بلوک ساختمان:یک خانه یا یک سفینه فضایی دوباره استفاده کنید

مزایا

کاربردپذیری؛ رهیافت شیء گرا به تولید واحدهایی از مدل، بانک اطلاعاتی و برنامه می انجامد که می توان آنها را مکرراً در پروژه ها یا سیستم های دیگری بکار بست.

معایب

- **دشواری**؛ مفاهیم و روشهای طراحی شیء گرا نسبتاً پیچیده است و به ویژه برای تحلیل گران و طراحانی که با روشهای ساخت یافته سنتی پرورش یافته اند، انس و آشنایی با قالب فکری شیء گرا معمولاً دشوار است.

عدم رواج؛ تعداد شرکتها و گروههایی که بتوانند پروژه های توسعه سیستم را با استفاده از روشهای OOA به صورت موفقیت آمیزی اجرا کنند، بسیار اندک است. به همین دلیل ایجاد و نگهداری سیستم ها با استفاده از روشهای شیء گرا هزینه نسبتاً بالایی را دارد.

تلفیق ناپذیری با سایر روشها؛ روشهای ساخت یافته سنتی در الگوی مفهومی و ابزارها و مراحل تا حد زیادی با هم اشتراک دارند.

اما روشهای شیء گرا به دلیل پیروی از یک قالب فکری کاملاً مجزا، بسادگی قابلیت تلفیق و ترکیب با بقیه روشها را ندارند.

جمع بندی:

متدولوژیهای شیء گرا در تحلیل و طراحی نظام آمارهای مثبتی توصیه نمی شوند. زیرا زیرساخت انسانی و فنی استفاده از این روشها به ویژه در کشور ما هنوز به حد کافی توسعه نیافته اند و از جهت دیگر هم در این مورد می توان از روشهای ساده تری که ماهیت اساساً داده گرای این سیستم ها نیز باشند، استفاده کرد.

تاریخچه مدل رابطه ای

مدل رابطه‌ای توسط E. F. Codd در سال ۱۹۷۰ ارائه شد. او مدل‌های موجود را مورد انتقاد قرار می‌داد. برای مدتی نسبتاً طولانی این مدل در مجامع علمی مورد تایید بود. اولین محصول موفق برای میکرو کامپیوترها dBASE بود که برای سیستم عامل‌های CP/M و PC-DOS/MS-DOS ساخته شد. در جریان سال ۱۹۸۰ پژوهش بر روی مدل توزیع شده (distributed database) و ماشین‌های دادگانی (database machines) متمرکز شد، اما تاثیر کمی بر بازار گذاشت.

در سال ۱۹۹۰ توجهات به طرف مدل شی گرا (object-oriented databases) جلب شد. این مدل کنترل داده‌های مرکب بود و به سادگی بر روی پایگاه داده‌های خاص، مهندسی داده (شامل مهندسی نرم افزار منابع) و داده‌های چند رسانه‌ای کار می‌کرد.

در سال ۲۰۰۰ نوآوری تازه‌ای رخ داد و دادگان اکس‌ام‌ال (XML) به وجود آمد. هدف این مدل از بین بردن تفاوت بین مستندات و داده‌ها است و کمک می‌کند که منابع اطلاعاتی چه ساخت یافته باشند یا نه در کنار هم قرار گیرند.

با این تفاوت که می‌تواند چند سطر داشته باشد. به عبارت دیگر دارای جداول چند گانه است که به طور صریح ارتباطات بین آنها بیان نمی‌شود و در عوض کلید هایی به منظور تطبیق سطر ها در جداول مختلف استفاده می‌شود. به عنوان مثال جدول کارمندان ممکن است ستونی به نام "موقعیت" داشته باشد که کلید جدول موقعیت را با هم تطبیق می‌دهد.

اوایل دهه ۸۰ به علت کارایی رابط های سطح پایین مدل سلسله مراتبی و شبکه‌ای پیشنهاد می‌شد که بسیاری از نیاز های آن زمان را برطرف می‌کرد. اما با سریعتر شدن سخت افزار به علت قابلیت انعطاف و سودمندی بیشتر سیستم های رابطه‌ای استفاده شدند.

در سال ۱۹۷۰ ارائه شد. این مدل یک مدل ریاضیاتی بود. که با مفاهیمی چون مستندات منطقی (predicate logic) و تئوری مجموعه ها (set theory) در ارتباط است. محصولات اینگرس، اراکل، DB2 و سرور (SQL Server) بر این پایه ایجاد شده. ساختار داده ها در این محصولات به صورت جدول است

بزرگترین مزیت این مدل آن بود که تنها توسط یک ساختار بنام رابطه قابل ارزیابی بود که معادل جدول دو بعدی است که برای همگان قابل فهم است.

اما نقاط ضعفی هم داشت.

۱. در مدل رابطه ای امکان دسترسی به طور همزمان ممکن نیست و در کارهای تیمی استفاده از نتایج دیگران هنگام انجام کار امکان ندارد.
۲. در این مدل تنها می توان از انواع داده هایی که تعریف شده اند استفاده کرد و امکان تعریف داده های جدید وجود ندارد.
۳. تراکنشهای بلند مدت در مدل رابطه ای وجود ندارد.

۴. ارتباط با زبانهای برنامه نویسی سطح بالایی مثل شی گرا دشوار است.

۵. پشتیبانی نکردن از پرس و جوهای بازگشتی

۶. وجود رابطه های تودرتو

و ...

این مشکلات سبب شد تا متخصصین به فکر ساخت مدل‌های جدیدی بیافتند که یکی از بهترین نوع آنها مدل شی گرائی است

نتیجه گیری

تقریباً تمام سیستمهای رابطه ای از تراکنش بلند مدت پشتیبانی نمی کند. این مسئله هنگام کار با سیستمهایی که نیاز به دسترسی اطلاعات بصورت بلند مدت را دارد با مشکل مواجه می کند.

برای غلبه بر این مشکل سیستم شی گرا تراکنش بلند مدت را پشتیبانی می کند. بانک شی گرا بصورت یک سیستم باز تعریف شده است که همواره می توان داده ها، دامنه ها و پردازش های جدیدی را به آن اضافه کرد. این عمل امکان دسترسی کاربران به مدت طولانی به پایگاه داده را فراهم می آورد.

منابع

پایگاه داده های شی گرا - Object-oriented databases، {بی تا}.

متدولوژی طراحی شی گرا، یوردون ادوارد
، مترجم: شریفی کلویی ، انتشارات آروین
۱۳۷۸

searchoracle.techtarget.com/.../object-oriented-database-management-system

[.http://pnu-club.com/pnu.thread21412.html](http://pnu-club.com/pnu.thread21412.html)

<http://reports-archive.adm.cs.cmu.edu/anon/itc/CMU-ITC-103.pdf>

<http://www.odbms.org>

http://en.wikipedia.org/wiki/Object_database