

برنامه نویسی شیء گرا

مهدی سودخواه محمدی

دانشکده مهندسی کامپیوتر و فناوری اطلاعات (IT)

چکیده

در این مقاله به بررسی برنامه نویسی شیء گرا، ابعاد و ویژگیهای آن خواهیم پرداخت. بعد از ارائه تاریخچه ای از برنامه نویسی شیء گرا به فواید آن می پردازیم. پس از شرح و توضیح تک تک فواید به بررسی مفاهیم عمده برنامه نویسی شیء گرا پرداخته و سه مورد از مهمترین فواید آن را شرح خواهیم داد. پس از آن دو نوع عمده زبانهای بس را معرفی خواهیم کرد و در انتها نیز جمع بندی و نتیجه گیری کلی را خواهیم داشت.

1. تاریخچه OOP:

اولین جرقه ها در اواخر دهه 1960 در نرم افزار 67-Simula زده شد. و بعد از آن در سال 1982 در ماه آگوست این عقاید بصورت مقاله ای در مجله Byte اعلام شد. در سال 1986 سازمان ACM یک سری کنفرانس ها در مورد برنامه نویسی شیء گرا برگزار کرد.

قبل از برنامه نویسی شیء گرا عموماً نرم افزارها با روش برنامه نویسی Procedural یا ساختیافته تهیه می شدند. این روش نمی توانست در مورد نرم افزارهای با حجم بالا بخوبی جوابگو باشد. زیرا با افزایش تعداد خطوط برنامه و افزایش تعداد برنامه نویسانی که روی یک پروژه نرم افزاری کار می کردند مدیریت پروژه کار مشکلی شده و در صورت وجود خطا در نرم افزار امکان مکان یابی آن دشوار بود. در نتیجه مدت زمان و هزینه اجرای پروژه ها رو به افزایش گذاشت. از این رو شیوه برنامه نویسی شیء گرا مطرح شد.

2. فواید برنامه نویسی شیء گرا:

فواید عمده این روش برنامه نویسی عبارتند از:

برنامه نویسی از آنها استفاده نمود. یعنی برنامه نویس بدون آنکه نیاز داشته باشد برای هر بار استفاده از یک سری کد در برنامه اش آن را تایپ کند می تواند آن کد را در درون یک شیء تعبیه نماید و از آن شیء در برنامه های خود استفاده کند.

2- محلی سازی تغییرات: در برنامه نویسی شیء گرا تغییراتی که در کد برنامه اعمال میشوند عموماً در داخل یک کلاس اتفاق می افتند و بطور عمومی در کل برنامه اعمال نمی شوند. این امر موجب می شود تا کنترل تاثیرات وارد بر نرم افزار راحت تر و مطمئن تر باشد. این ویژگی را می توان متأثر از قابلیت پنهان سازی داده ها دانست که در ادامه این مقاله به آن اشاره خواهیم کرد.

3- طراحی آسان تر: سیستم ارث بری برنامه نویسی شیء گرا به برنامه نویس این امکان را می دهد تا بتواند راحت تر نرم افزار خود را طراحی کند و داشتن یک طرح کلی و اجمالی از نرم افزار به راحتی امکان پذیر است. لذا برنامه نویس حتی قبل از نوشتن کد برنامه میتواند طراحی کلی نرم افزار را انجام دهد.

4- قابلیت توسعه: در برنامه نویسی شیء گرا می تواند بعضی امکانات که خود زبان از آنها بطور مستقیم حمایت نمی کند را پیاده سازی کرد.

1- قابلیت استفاده مجدد از کد: در برنامه نویسی شیء گرا می توان از کدی که قبلاً و حتی توسط فرد دیگری ایجاد شده استفاده کرد. حتی می توان کتابخانه ای از اشیای مفید و لازم در برنامه نویسی را فراهم کرد و در

S کلاس پسر و کلاس P کلاس پدر است و کلاس فرزند از کلاس پدر ارث برده است. ارث بری موجب می شود که برنامه نویس بتواند در مورد برنامه های با حجم بالا و ساختار پیچیده، نرم افزار را به صورتی مناسب به یک ساختار درختی مناسب تبدیل کند که این امر موجب سهولت کار برنامه نویس می شود.

مسلماً با استفاده از این قابلیت حجم کار تایپی برنامه نویس کاهش می یابد و نیز برنامه نویس می تواند از کدی که قبلاً نوشته شده است دوباره استفاده کند

این دو مورد از فواید عمده وراثت در برنامه نویسی شیء گرا به شمار می روند.

5. چند شکلی

چند شکلی یا Polymorphism از ویژگیهای نسبتاً پیشرفته برنامه نویسی شیء گرا بشمار می رود. و به معنی استفاده از کلاس فرزند بجای کلاس پدر است.

در حقیقت در چندشکلی ما با دو کلاس متفاوت برخوردی یکسان داریم که در اکثر موارد برنامه نویسی شیء گرا فایده زیادی دارد.

در حقیقت ما با انواع گوناگونی از اشیا کار می کنیم بدون اینکه برایمان مهم باشد که متغیری که با آن کار می کنیم از چه نوعی است.

6. زبانهای شیء گرا:

از ابتدای بوجود آمدن برنامه نویسی شیء گرا دو نوع از زبانهای برنامه نویسی برای این سبک برنامه نویسی وجود داشته اند. نوع اول: زبانهای خالص مانند Smalltalk. در این زبانها برنامه نویس برای هر کاری بایستی از اصول برنامه نویسی شیء گرا پیروی کند حتی برای ساده ترین کارها مانند جمع دو عدد.

بعنوان مثال می توان از اعداد مختلط بعنوان یک مثال نام برد. می توان با استفاده از روشهای برنامه نویسی شیء گرا انواع عملیات مختلف را پیاده سازی کرد.

5- طراحی سریع تر نرم افزار: با استفاده از روشهای برنامه نویسی شیء گرا میتوان نرم افزارها را سریع تر از سایر روش های برنامه نویسی طراحی کرد. این امر در حقیقت نتیجه ای از استفاده مجدد از کد و طراحی سریع تر نرم افزار می باشد.

بالطبع پروژه های نرم افزاری به حجم بالا با این امر بسیار سریع ار و آسان تر انجام می شوند.

3. پنهان سازی اطلاعات

در مقایسه این گزینه با دنیای واقعی می توان از تلویزیون بعنوان یک مثال بارز نام برد. در تلویزیون اجزا و قطعات بسیار زیادی وجود دارند که همگی از دید استفاده کننده مخفی هستند زیرا لزومی ندارد که استفاده کننده با آنها کار کند و سروکار داشته باشد. در نتیجه موضوعی بنام پنهان سازی مطرح می ود که عبارت است از خارج از دسترس بودن بعضی از اطلاعات که کاربر به آنها احتیاجی ندارد. این امر در برنامه نویسی شیء گرا Datahiding نامیده می شود که منجر به Encapsulation یا جاداشدن شیء موردنظر (اطلاعات شیء موردنظر) از سایر اشیا می شود.

در یک شیء اطلاعات و توابع موجود تحت دو گروه عمومی و خصوصی تقسیم بندی می شوند. اطلاعات عمومی اطلاعاتی هستند که برای کدهای خارج از شیء نیز قابل دسترس هستند ولی اطلاعات خصوصی فقط برای خود شیء قابل دسترسی است.

4. وراثت

برنامه نویس هنگام تعریف کلاس جدید S میتواند اعلام کند که می خواهد بعضی از اطلاعات کلاسی دیگر مانند P روی کلاس S کپی شوند. به اصطلاح گفته می شود کلاس

8. منابع:

1.Object oriented programming concepts

2.Object Oriented Programming : an evolutionary approach

Author :COX,BRAD J

3.شیوه های برنامه نویسی شیء گرا

نوع دوم از زبانهای برنامه نویسی شیء گرا که بیشتر رواج دارند زبانهای مختلط هستند . در این نوع زبانهای برنامه نویسی برنامه نویس مختار است در امکانات برنامه نویسی شیء گرا استفاده کند یا نه . برای مثال می توان به زبانهای زیر اشاره کرد :

C++

Ada 95

CLOS

Scheme

Eiffel

Java

7. نتیجه گیری:

برنامه نویسی شیء گرا یک شیوه نسبتاً جدید برای انجام پروژه های نرم افزاری است که برنامه نویسان پس از احساس نیاز به شیوه ای منسجم تر برای انجام پروژه های نرم افزاری بوجود آمد که فواید عمده آن عبارتند از :

استفاده مجدد از کد

طراحی آسان تر

طراحی سریع تر

قابلیت توسعه

طراحی سریع تر نرم افزار

و از ویژگیهای این روش برنامه نویسی می توان به موارد زیر اشاره کرد:

پنهان سازی اطلاعات

چندشکلی

وراثة