یک شیوه برنامه نویسی است که ساختار یا بلوک اصلی اجزای آن، شی‌ها می‌باشند. در واقع در این روش برنامه نویسی، برنامه به شیء گرایش پیدا می‌کند. برنامه نویسی شیئ گرا یک مفهوم طراحی است. با استفاده از برنامه نویسی شیئ گرا می توانید برنامه های تحت وب را ایجاد کنید.

توضیح مفاهیم اولیه

1. شیئ -(Object)یک شیئ نمونه ای از یک کلاس است و تعریفی شبیه به اشیائ دنیای واقعی دارد. اشیاء می توانند هرچیزی مانند خودکار، مداد، کتاب، میز و غیره باشند. یک شیئ موجود خصوصیاتی نیز به همراه دارد و این خصوصیات در اشیاء مختلف متفاوت هستند.

برای نمونه یک ماشین را مثال می زنیم. یک ماشین خصوصیت هایی مانند رنگ، مدل، برند و غیره را دارد، همچنین رفتارهایی مانند حرکت به جلو یا عقب را نیز دارد. اشیاء مختلف خصوصیت ها و رفتارهای متفاوتی نیز دارند.

مثال دیگر می توان تلوزیون را نام برد. تلوزیون خصوصیت هایی مانند رنگ، برند، اندازه و مدل را دارد و رفتارهایی مانند روشن یا خاموش شدن نیز به آن تعلق دارد. در این جا می بینیم که هر دو شیئ رفتارهای متفاوتی دارند.

2. کلاس -کلاس ها بسته هایی هستند که روی یک شیئ تعریف می شوند. به عبارت دیگر می توان گفت که کلاس یک طرح اولیه است.کلاس تمام خواص و رفتار یک شیئ را توصیف می کند. به تعریف دیگر کلاس ها صفات و متدهای اشیاء از نوع خود را تعریف می کنند.

مثال: همان ماشینی که در بالا ذکر کردیم نشان می دهد که یک ماشین باید خصوصیاتی مانند رنگ، تعداد در و غیره را داشته باشد و این یک شیئ است که 4 در دارد و رنگ آن قرمز است.یا از لحاظ برنامه نویسی می توانیم بگوییم که این ماشین نمونه ای از کلاس ماشین است. بنابراین بازنمایی ساختاری از شیئ را کلاس می گویند. شما باید قبل از ایجاد هر نوع شیئ ای یک کلاس بسازید بسازید.

مزایای استفاده از برنامه نویسی شیئ گرا

مزایای بسیاری برای OOP در برنامه نویسی شیئ گرا وجود دارد.

قابلیت استفاده مجدد: فرض کنید یک کلاس ایجاد شده با نام Carدارید و و در برخی از نقاط اپلیکیشن خود از آن استفاده کرده اید.پس می توانید همان کلاس را در اپلیکیشن های مشابه نیز استفاده کنید.

نگهداری آسان: نگهداری و تغییر کدهای موجود بسیار آسان است چرا که می توانیم یک شیئ جدید فقط با ایجاد تفاوت های کوچک در نمونه موجود، ایجاد کنیم.

ویژگی های برنامه نویسی شیئ گرا

1. در اینجا شما می توانید داده و توابعی را در صورت نیاز به راحتی اضافه کنید.
2. در اینجا داده ها مخفی هستند و نمی توانند توسط توابع دیگر در دسترس قرار بگیرند.
3. برنامه ها به موجودیت های شناخته شده با نام شیئ تقسیم شده است .

موجودیت ها :(Entity)یک موجودیت اغلب گروهی از مردم را نشان می دهد(مثل فرزندان، اپلیکیشن ها) اما همچنین می تواند گروهی از اشیاء (مانند کتاب های درسی)، فعالیت ها (مانند تکالیف) و یا مفاهیم را نیز نشان دهد.

یک برنامه نویس شیئ گرا باید شبیه اشیاء فکر کند. برنامه نویسان سنتی مانند کامپیوتر فکر می کنند.

برای مدیریت کلاس های مختلف یک سیتم نرم افزاری و برای کاهش پیچیدگی آن، تکنیک های متعددی وجود دارد. که به آن مفاهیم OOP می گویند. 4 مفهوم اصلی در OOPS استفاده می شود. که به صورت زیر هستند:

ارث بری:(Inheritance) قابلیتی برای ایجاد کلاس جدید از یک کلاس موجود را ارث بری می گویند. با استفاده از مثال زیر معنای ارث بری را بهتر درک می کنیم.

اجازه دهید یک مثال از زندگی روزمره بیان کنیم. فرض کنیم که شخصی مرده است پس تمام اموال او برای فرزندانش به ارث می رسد(اموال می تواند، حقوق، خصوصیت ها و غیره باشد). در روشی مشابه این مسئله در مورد OOPاتفاق میافتد که کلاس پایه (یا کلاس والد) متدها و خصوصیت هایی دارد که توسط زیر کلاس ها از آن به ارث برده می شوند.(برخی اوقات به آنها کلاس فرزند می گویند). در اینجا مشاهده می کنیم که اشیاء توسط کلاس ها تعریف می شوند و کلاس ها می توانند خصوصیت ها و رفتارهایی ازکلاس های که قبل ایجاد شده را به ارث ببرند(کلاس پایه). کلاس نتیجه نیز به عنوان زیر کلاس یا کلاس فرزند یا کلاس مشتق شده شناخته شده است. کلاس مشتق شده می تواند تعدادی رفتار ها و خصوصیت های اضافه شده نیز داشت باشند.

کپسوله سازی:(Encapsulation) این بدان معنی است که دادها یا پیاده سازی جزئیات یک ماژول از ماژول های دیگر مخفی شود. کپسوله سازی می تواند به صورت محدود کردن دسترسی به خواص مشخصی نیز تعریف شود Encapsulation .تکنیکی است که برای مخفی کردن داده ها استفاده می شود. این کار برای کاهش پیچیدگی سیستم مفید است.

Abstraction: این یک فرایند برای نشان دادن ویژگی های ضروری یک موجودیت توسط مخفی کردن جزئیات مهم مشخص شده است. این عمل یکی دیگر از ویژگی های خوب OOPSمی باشد و برای نشان دادن جزئیات ضروری به استفاده کننده از شیئ می باشد.اجازه دهید که یک مثال بیان کنیم: زمانی که شما مانیتور خود را روشن یا خاموش می کنید آیا می دانید که چگونه این مانیتور روشن یا خاموش می شود؟ ما نیازی نداریم که

بدانیم چه چیزی داخل کامپیوتر اتفاق میافتد. یا زمانی که شما با ماشین رانندگی می کنید و دنده را عوض می کنید آیا در مورد مکانیسم داخلی چرخ دنده نگرانی دارید؟ مطمئنا نه! در کل این روش یک فرایند برای افشای جزئیات ضروری شیئ و مخفی کردن جزئیات مهم که مورد نیاز کاربر نیست، می باشد. این کار برای قابلیت استفاده مجدد از کد مفید است.

Polymorphism (چند ریختی) : پلی مورفیسم از ترکیب دو کلمه POLY و Morphمشتق شده است POLY .به معنای تعدادی و Morphبه معنای تغییر جزئی می باشد. بنابراین پلی مورفیسم داشتن یک نام با شکل های مختلف است. در روش می توانیم متدهایی با نام مشابه داشته باشیم اما تغییرات جزئی در عملکرد داشته باشند. دو نوع پلی مورفیسم وجود دارد.

Overriding: این یک چند ریختی در زمان اجرا نامیده می شود. برای Overridingمتدهایی که متد آنها استفاده خواهند شد فقط در زمان اجرا تعیین می شوند.

Overloading: این یک پلی مورفیسم در زمان کامپایل نام دارد که متد آن اجرا شده و توسط کامپایلر تعیین می شود. این تصمیم زمانی که کدکامپایل شده دریافت می شود گرفته می شود.