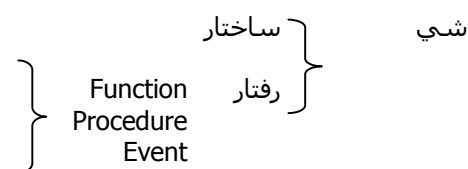


روند تکمیلی زبانهای برنامه نویسی :

شیء گرا	ساخت یافته	برنامه های اولیه
شیء ۱	Function	خط ۱
شیء ۲	Function	خط ۲
شیء ۳	Function	خط ۳
	Function	خط n
از تعامل ، همکاری و پیغام دهی بین اشیا	از ترکیب فراخوانی چندین function	توالی خطوط برنامه

در مدل‌های ساخت یافته اجزای تشکیل دهنده برنامه Function و Procedure ها هستند که از فراخوانی Call تعدادی Function و Procedure برنامه ایجاد می شود. در مدل شیء گرا برنامه از تعدادی شیء تشکیل می شود که با استفاده از تعاملات بین اشیا برنامه اجرا می شود.

مفهوم شیء: هر موجودیتی در دنیای مجازی که از روی یک واقعیت در دنیای واقعی ای الگو برداری شده است . یک شیء است از ساختار و رفتار تشکیل شده است .



مثال (ماشین، دوچرخه، کارمندو....) همه اینها بسته به موضوع برنامه نویسی ما میتواند شیء باشد. هر شیء مجموعه ای از ساختارها و رفتارهاست که هدف خاصی را پیاده سازی میکند این هدف پیاده سازی یک مفهوم از دنیای واقعی در دنیای مجازی است. ساختارها شامل مشخصات و ویژگی هاست. رفتارهای اشیا شامل Function و Procedure و Enter هایی هستند که با اشیا دیگر تبادل می کنند. پس معادله سازی یک شیء در دنیای کامپیوتر را Abstraction (انتزاع) یک شیء یک طرح برنامه پذیر است.

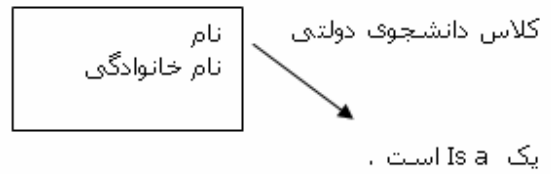
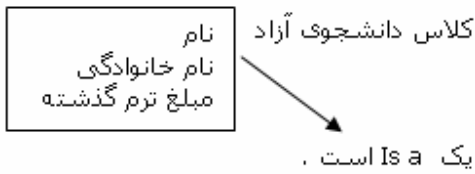
(ساختار و رفتار) که چیزهایی را در دنیای واقعی مانند ماشین، دوچرخه و کامپیوتر نشان می دهد. در برنامه شما یک شیء ممکن است یک فرم یک Button (دکمه) یا یک Connection، Database، (پایگاه داده) هر شیء یک واحد عملیاتی کامل است (به طور

مستقل کار می کند) و تمام داده ها و اعمال مورد نظرش را درون خود دارد.

دنیای واقعی	دنیای مجازی
پیست اتومبیل رانی	صفحه کد پیست را نمایش میدهد
(ماشین) شیء	شیء (ماشین)

Method	Field	Data	property	Attribute
procedure	شیء	operation		Function

رابطه کلاس با شیء
 دانشجوی دانشگاه آزاد و دانشجوی دانشگاه دولتی هر دو کلاس‌هایی هستند که دانشجویان در یکی از این طبقه ها قرار می گیرند کلیه ی دانشجویان اشیایی از این کلاس هستند.



کلاس ها قالب هایی برای اشیاء هستند. کلاس ها را می توان طرح های اولیه برای اشیاء دانست که رابطه ی isa یا است یک برقرار است. به عنوان مثال همه ی شما یک دانشجوی دانشگاه آزاد هستید یا اینکه رنو است یک ماشین. هر محیط برنامه نویس که بر اساس اشیا با تعاریف فوق شکل گیرد شئی گرا نامیده می شود. مثال شی و کلاس در VB

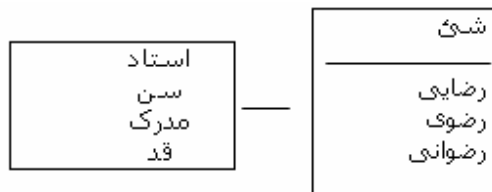
Dim myform as formi
Myform = new form()
کلاس شی

مثال: در زبان Vb با استفاده از کلمه ی New یک شی را تعریف می کنیم و با استفاده از کلمه ی کلیدی New از نوع کلاس مشخص ان را مقدار دهی می کند در هنگام مقدار دهی یک نمونه از کلاس درون حافظه ساخته می شود که به شی مربوطه اختصاص داده میشود. در صورت تعریف شی دیگری از نوع این کلاس فضای دیگری به ان شی اختصاص داده میشود وقتی که یک نمونه کلاس ساخته میشود یک کپی از ساختار کلاس در حافظه ایجاد میشود که این کپی مستقل از دیگر کپی هاست به این ترتیب اطلاعات هر شی مستقل از اشیا دیگر نگهداری میشود در هر لحظه محدودیتی روی تعداد کپی های یک کلاس وجود ندارد به عنوان مثال از دنیای واقعی اگر ماشین یک شی باشد این کلاس که برای ماشین مورد استفاده قرار گیرد و تغییرات بر روی این ماشین تغییری روی دیگر ماشین ها نمی گذارد.

اشیا ترکیبی از اعضا هستند خصوصیات فیلدها، متدها و رویدادها و هر چیزی که اطلاعات و عملیاتی را در برداشته باشد. متدها اعمالی هستند که شی میتواند انجام دهد و رویدادها اطلاعاتی هستند که شی به اشیا دیگر می فرستد یا از انها دریافت میکند. Enter همچنین زمانی که اتفاقی در برنامه می افتد یک رویداد ایجاد میشود. مثال: در شی ماشین فیلدها و خصوصیات مانند رنگ مدل، نوع سوخت، تیپ تعریف میشود. این اطلاعات وضعیت شی ماشین را توصیف میکند همچنین شی ماشین شاید عملیاتی مانند (برو به جلو، برو به عقب، ترمز یا تعویض دنده را داشته باشد. همچنین میتواند رویدادهای مثل موتور داغ شده است. Heating و Enginover یا انجکتور خراب است را ارسال کند و شی راننده از جریانات داخل موتور خویش آگاه کند همچنین اگر ماشین باشی درخت تعامل کند رویداد تصادف اتفاق می افتد.)
کپسوله سازی Encapsulation

شی
Go forward
رفتارهای عمومی
اجزای موتور خصوصی

کپسول سازی روشی است که یک شی را مستقل از اینتر فیس پیاده سازی کنیم منظور از اینتر فیس ساختارها و رفتارهای عمومی شی است و هر برنامه ای تنها اینتر فیس شی را در اختیار دارد تا زمانی که اینتر فیس شی ثابت باشد برنامه میتواند با ان تعامل کند حتی اگر خصوصیات عمومیشان با یکدیگر تعامل میکنند داده های داخل شی نباید در اینتر فیس قرار گیرد بنابراین فیلدها را به ندرت عمومی Pablick تعریف می کنیم. در مثال ماشین شی ماشین با راننده تعامل میکند اینتر فیس ماشین (رفتارهای عمومی ماشین مثل: برو به جلو، برو به عقب، ترمز است) این همه اطلاعاتی است که راننده برای تعامل ماشین احتیاج دارد ماشین شاید شامل موتور نیز باشد اما راننده نیازی به شناخت موتور ندارد بنابراین اگر موتور ماشین تغییر کند تا زمانی که اینتر فیس ان به درستی کار کند ماشین به کار خود ادامه میدهد و تفاوتی برای راننده ندارد.
تولید Encapsulation: ۱. مدیریت برنامه را ساده می کند ۲. تغییرات را محلی نگه میدارد ۳. اشیا میتواند با ساختار و رفتار تضمین شده با یکدیگر تعامل کنند خواص دیگری نیز وجود دارد که با تکمیل مفاهیم شی گرایبی به انها پی خواهیم برد.
تمرین سیستم ثبت نام ۱. ۴ درس ارائه شده ۲. ۲ استاد ۳. ۱۰ دانشجو داریم.

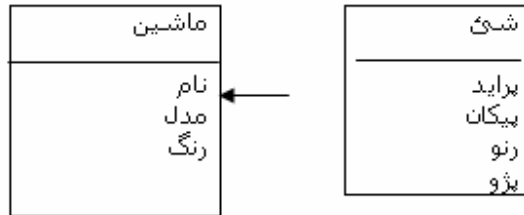


دانشجو
معدل دیپلم - سن
شهریه پرداختی

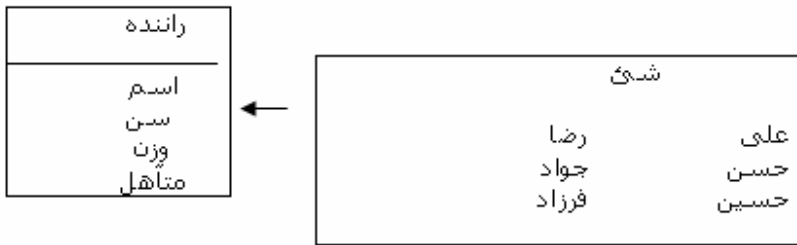
شیء	
علی، ۱	رضا، ۶
حسن، ۲	جواد، ۷
حسین، ۳	فرزاد، ۸
محمد، ۴	وحید، ۹
میثم، ۵	سعید، ۱۰

تاکسی سرویس که تعداد ۴ عدد ماشین و تعداد راننده (۶ نفر) در آن فعالیت میکنند.

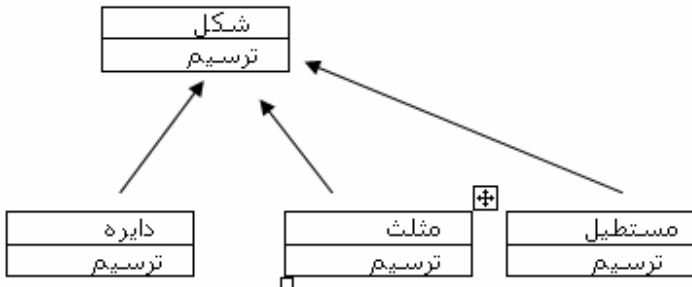
کلاس - ماشین



کلاس - راننده



چند ریختی Polymorphism:



رفتار های مشترک ترسیم دارند

مثال: (مستطیل ۵ و ۲ ، دایره به شعاع ۲ ، مستطیل ۲ و ۴ ، دایره به شعاع ۲)

تمامی اینها نیز رفتار مشترک دارند . به طور مثال همگی ترمز روغنی دارند امی کامیونت و مینی بوس ترمز بادی دارند .

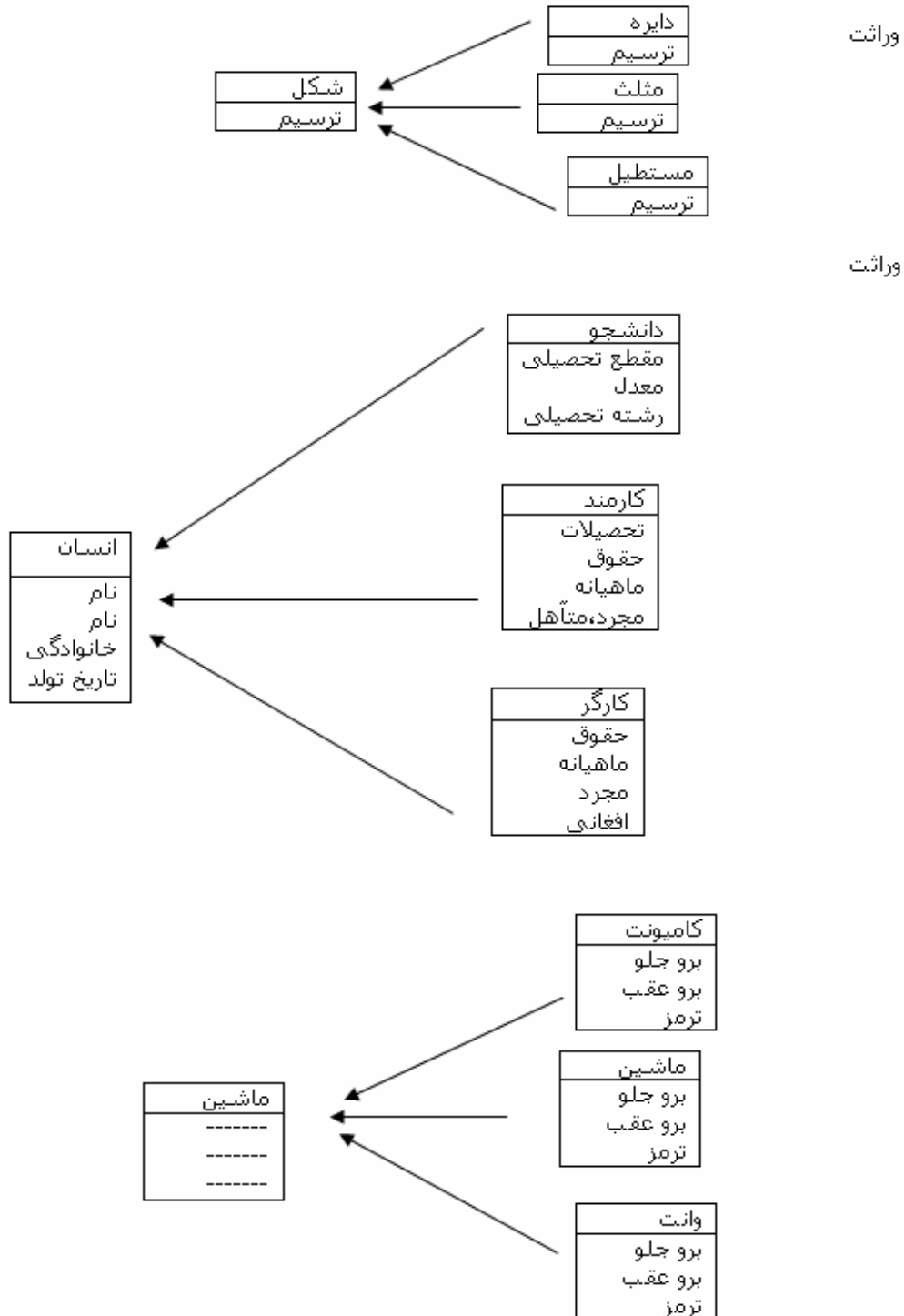
کامیونت	مینی بوس	انواع سواری	وانت	ماشین سبک
---------	----------	-------------	------	-----------

چند شکلی Polymorphism

توانایی کلاس ها متفاوت در پیاده سازی های مختلف از اینترفیس های عمومی مشابه است. به عبارت دیگر چند شکلی به متد ها و خصوصیات یک شیء اجازه میدهد بدون توجه به چگونگی پیاده سازی اعضای آن مورد استفاده قرار میگیرد. برای مثال در قوانین راهنمایی و رانندگی ایران قانون رانندگی برای گواهینامه پایه دوم دارای شرایط حداکثر وزن ماشین ۲۵۰۰kg است بدین ترتیب راننده میتواند با ماشین های وانت انواع سواری کامیونت و مینی بوس رانندگی کند. این بدان معناست که در دنیای کامپیوتر اگر شیء راننده باشی ماشین در تعامل است. با توجه به اینکه اشیایی از این نوع اینترفیس مشابهی دارند. شیء راننده میتواند بدون توجه به جزئیات خاص پیاده سازی هر ماشین با آن تعامل کند

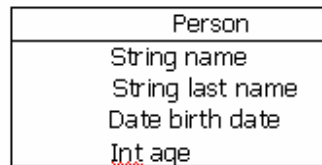
انواع چند ریختگی

- ◆ چندریختگی با اینترفیس interface polymorphism
- ◆ چند ریختگی با وراثت inheritance polymorphism



وراثت رابطه ای بین اشیا است که طبق آن اشیا مشابه به ساختار و رفتار مشابه را به اشتراک می گذارد. طبق این رابطه ساختار و رفتار مشترک در کلاسی به نام کلاس پدر قرار میگیرند بدین ترتیب مجموعه ساختارها و رفتارهایی که در کلاس پدر تعریف شده است توسط کلاس های فرزند مثال: کارگر کارمند و دانشجو به ارث برده میشود وراثت خواص زیر را به برنامه اضافه میکند :

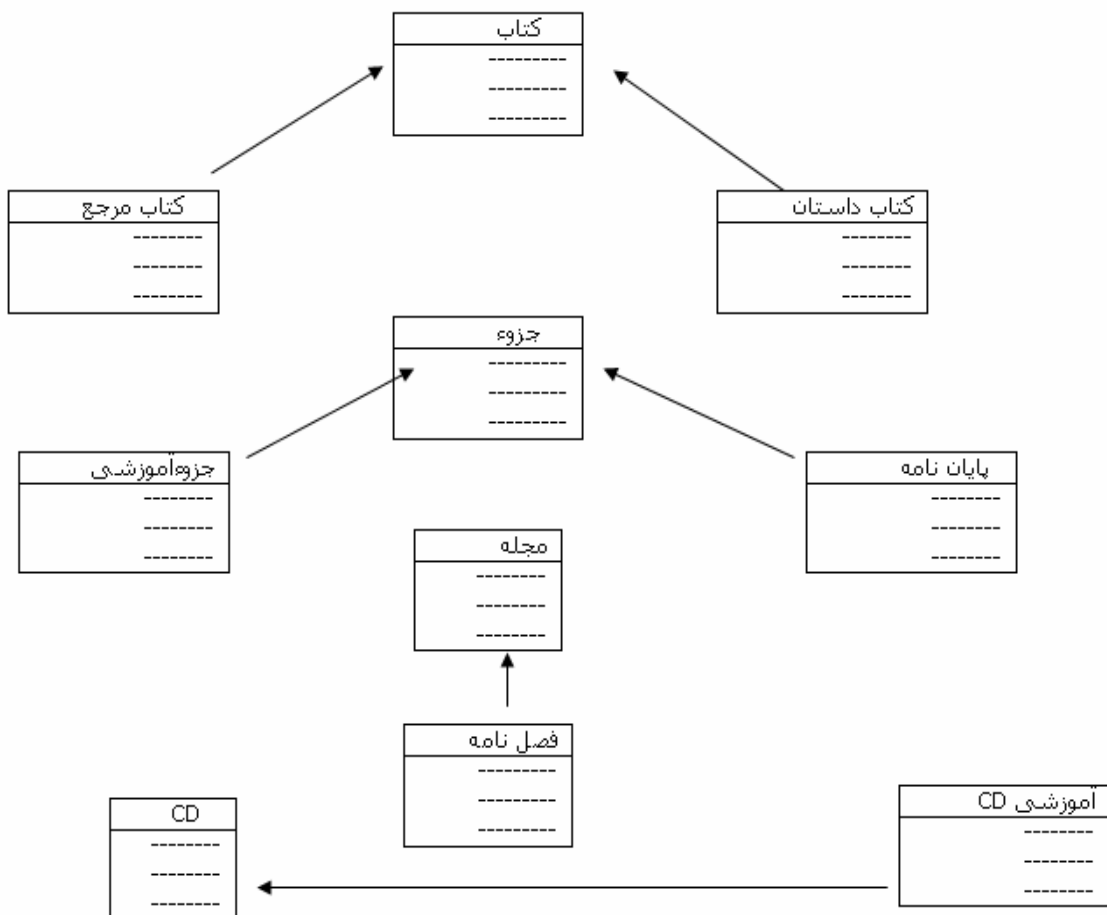
۱. استفاده مجدد بدین معناست که ساختار و رفتار مشترک که در کلاس پدر تعریف شده اند نیازی به تعریف مجدد ندارد و میتوان در کلاس های فرزند مورد استفاده قرار گیرد.
۲. چند ریختی: توضیحات بخش چند ریختگی. کلاس به فرم زیر نمایش داده میشود.



نام کلاس، نامی منحصر به فرد است که به هر کلاس نسبت داده میشود مثلاً ماشین (Car) یا انسان Person میتواند نام کلاس ها باشند هر کلاس متناسب با ساختار خود تعدادی مشخصه یا ویژگی دارد. مشخصه ها دارای نام و نوع هستند بعنوان مثال: نام مشخص (انسان) یک مشخصه از کلاس انسان است که نوع آن String (رشته، حرفی) است و یا سن شخص از نوع عددی میباشد هر کلاس متناسب با رفتار مورد انتظار از آن دارای تعدادی متد میباشد. هر کلاس متناسب با رفتار مورد انتظار از آن دارای تعدادی متد میباشد. متدها دارای نام پارامترهای ورودی و نوع خروجی هستند مثال: عنوان: انسانهای دانشگاهی

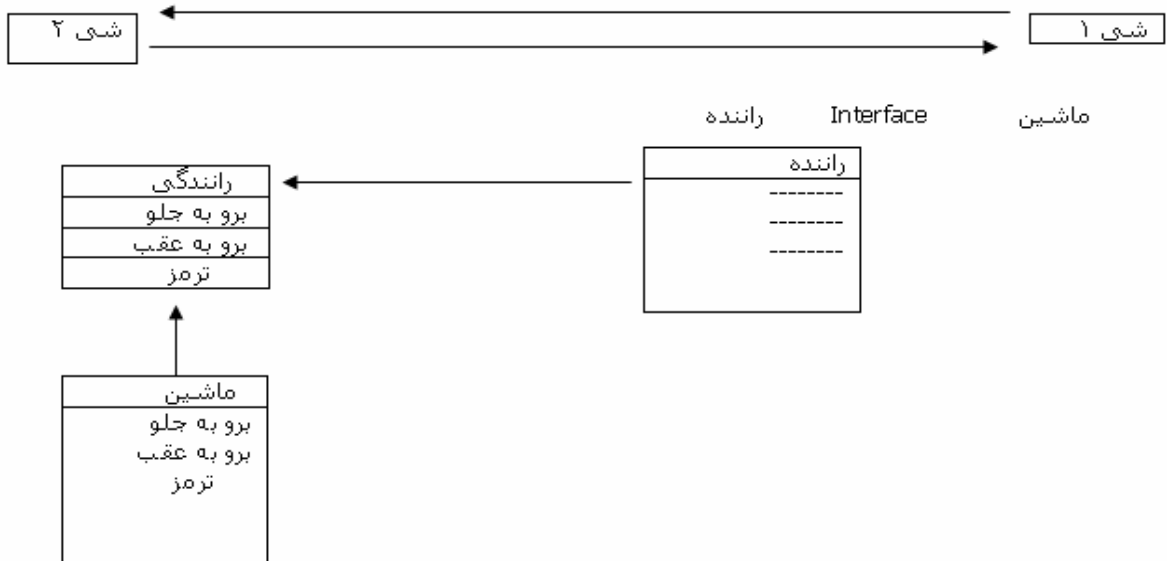
پارامترهای ورودی و نوع خروجی هستند مثال: انسانهای دانشگاه

عنوان: محتویات یک کتابخانه

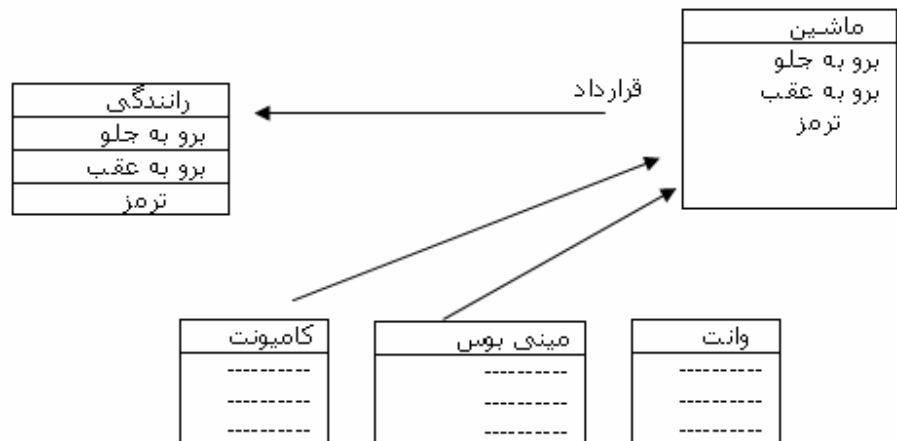


نوع دیگری از وراثت که میتواند چند پدر داشته باشد.

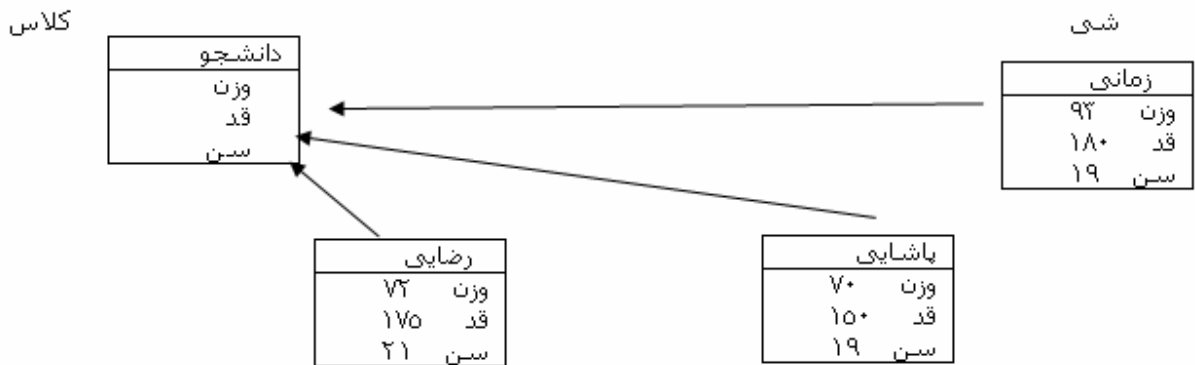
قرارداد



اینترفیس یک قرارداد برای ساختار و رفتار بین اشیا است در واقع اینترفیس اعضای یک کلاس را تعیین میکند اما توضیحاتی در باره ی پیاده سازی همه جزئیات نمیدهد. اشیا یی که طبق قرارداد اینترفیس با هم تعامل میکنند، ساختار و رفتار (ثابتی) را از آن انتظار دارند. بدین ترتیب شیء طرف مقابل قرارداد میتواند با رعایت قرارداد تعویض شود یک شیء میتواند اینترفیس های زیاد و متفاوتی داشته باشد و کلاس های متفاوت زیادی میتواند یک اینترفیس مشابه را پیاده سازی کند همه اشیا که اینترفیس مشابهی را پیاده سازی میکنند میتوانند با دیگر اشیا یی که از اینترفیس استفاده میکنند تعامل داشته باشند بعنوان مثال شیء ماشین و شیء راننده را در نظر میگیریم انتظار راننده از شیء ماشین این است که بتواند با آن رانندگی کند بدین ترتیب قرارداد (اینترفیس رانندگی) بین دو شکل برقرار است با این قرارداد طبق مثال ماشین میتواند وانت مینی بوس کامیونت و.... باشد و برای راننده تفاوتی نمیکند که جزئیات آن ماشین را و نحوه عملکرد آن را بداند.



در چند شکلی وراثتی با استفاده از وراثت این امکان ایجاد میشود که ساختار و رفتار مشترک بعنوان یک قرارداد در کلاس پدر تعریف شود تا کلید کلاس های فرزند از آن طبیعت کند در مثال ماشین تمام ماشین ها ساختار و رفتار مشترک خود را در کلاسی به نام ماشین قراردادده اند بدین ترتیب راننده بدون توجه به نوع ماشین و تنها با این حقیقت که تمام ماشین ها از کلاس ماشین به ارث گرفته شده اند و میتوانند رانندگی کنند کلاسی که از کلاس دیگر ارث می برد مشتق از کلاس یا وارث آن یا Sub class آن یا کلاس فرزند نامیده میشود. کلاسی که از آن ارث بری میشود کلاس پدر یا Super class یا کلاس پایه نامیده میشود. هر کلاسی که به ارث گرفته میشود اعضای مشابهی با کلاس پایه دارد و اعضای اضافی مورد نیاز میتواند اضافه شوند کلاس های وارث همه ویژگی های کلاس پایه را دارد و میتواند با اشیا یی که با کلاس پایه تعامل میکنند در ارتباط باشند. در دنیای واقعی: از اشیا یی مختلف یک شکل را انتظار داریم به آن چند شکلی میگویند. مثال: از تمام اشکال انتظار داشته باشیم.



رابطه Is a (است یک)

قابلیت ترسیم داشته باشند یا لز تمام ماشین ها انتظار حرکت داشته باشیم. تعریف شی گرای: به هر محیطی که اجزای تشکیل دهنده ان اشیا باشند شی گرا گویند. برنامه نویسی شی گرا: به برنامه ای که اجزای تشکیل دهنده ان اشیا هستند. در برنامه نویسی شی گرا مفاهیم صورت مسئله را به صورت اشیا و کلاس های ان ها طراحی و در برنامه پیاده سازی میکنیم. زبانهای مثل C و پاسکال و غیر شی گرا است اما سی پلاس پلاس و جاوا و محیط هایی مانند Vb.Net شی گرا هستند. Vb.6 شی گرا نیست چون نمیتواند در ان اشیا جدید تعریف کرد.

مفهوم Overloading (چند باره نویسی)



Overloading به شما امکان میدهد که چندین عضو با نام مشابه داشته باشید هر عضوی که هم نام با عضو دیگر است باید تعریف متفاوتی داشته باشد مثال: در دنیای اشکال شی دایره دارای یک متد ترسیم است که دایره را در مبدأ مختصات ترسیم میکند ممکن است نیاز باشد که متدی دیگر تعریف کنید تا دایره را در نقطه X و Y ترسیم نماید بدین ترتیب متد دوم که دارای تعریف متفاوتی و پارامترهای متفاوتی نسبت به متد اول است تعریف میشود و متداول را Overload میکند.

مشخصه: String Name
نام مشخصه نوع مشخصه

نام پارامتر نوع پارامتر
متد: String Concat(string a1, string a2)
پارامترهای ورودی نام متد نوع خروجی

شبه کد :

Clas student
{String name مشخصه
{String last name مشخصه
Ing age
{ String return full name()