متد: برای ساده تر شدن یک برنامه بزرگ می توان برنامه را به چندین برنامه کوچکتر تقسیم کرد که کار با آنها ساده تر می شود و به هریک از قسمت های کوچکتر که بخشی از عملکرد برنامه اصلی را انجام می دهد متد گفته می شود متدها درون کلاس ها تعریف می شوند و قطعه کدی هستند که وظیفه خاصی را بر عهده دارند در صورتی که قصد دارید یک عمل خاص چندین بار در برنامه اجرا شود بجای اینکه چندین بار قطعه کد را بنویسید کافی است یک بار آن را درون متد نوشته ودر صورت نیاز متد را فراخوانی کنید با این کار هم از حجم برنامه کاسته شده و هم خوانایی برناوه افزایش می یابد به اطلاعاتی که به متد ارسال می کنید پارامتر یا ورودی متد گفته می شود و مقدار بازگشتی متد خروجی متد خوانده می شود .

متدها خود به دو دسته بازگشتی و غیر بازگشتی تقسیم می شوند که هر کدام از این روش ها خود به دو دسته با آرگومان و بدون آرگومان دسته بندی می شوند.

1-متد غیر بازگشتی بدون آرگومان: متدی است که خروجی و ورودی ندارد .

 () نام متد Public void

{دستورات}

مثال:برنامه ای بنویسید که پیغام WELCOM TO C# را در خروجی نشان دهد؟

 Public void m()

{label1.text=”welcome to c#”;}

2-متد غیر بازگشتی با آرگومان: متدهایی که خروجی ندارند ولی دارای ورودی هستند.

 ( آرگومانها با انواعشان ) نام متد Public void

{دستورات}

مثال :برنامه ای بنویسید که نام و فامیل دانشجوییی را از ورودی دریافت و پیغام hello را با نام و فامیل دانشجو نشان دهد؟

 Public void merge(string name, string family)

{label1.text=”hello”+name+family;}

ودر رویداد کلیک button نام و فامیلی را از textbox دریافت کنید وسپس تابع را فراخوانی می کنیم.

 Merge ( text box1.text, textbox2.text)

3- متد بازگشتی بدون آرگومان: متدهایی که دارای خروجی هستند ولی ورودی ندارند.

 ()نام متد نوع بازگشتی Public

دستورات}

 {داده بازگشتی Return

برنامه ای بنویسید که دو عدد از ورودی دریافت و عدد اول را به توان عدد دوم برساند؟

Public int tavan()

{ int x=math.pow(textbox1.text,textbox2.text);

Return x;}

و سپس در رویداد کلیک button ،تابع را فراخوانی می کنیم .

 Label1.text=tavan().tostring();

4-متد بازگشتی با آرگومان : متدهایی هستند که آرگومان یی به متد ارسال شده و در بدنه آن اعمال روی آرگومان انجام شده و در انتها نتیجه به محل فراخوانی متد بازگشت داده می شود.

( آرگومانها و نوع آنها )نام متد نوع بازگشتی Public

دستورات}

 {داده بازگشتی Return

مثال: برنامه ای بنویسید که دو عدد صحیح دریافت و جمع دو عدد صحیح را انجام و نشان دهد؟

Public int sum(int a, int b)

Return a+b;}

ودر رویداد کلیک button تابع را فراخوانی می کنیم .

 Label1 text= sum( int . parse(text box1.text ) , int.parse(textbox2.text).tostring();

 متدهای بازگشتی : متد بازگشتی متدی است که خودش را بطور مستقیم یا غیر مستقیم فراخوانی می کند و در هر مرحله متد کپی دیگری از خودش را فراخوانی می کند تا برروی مسئله کوچکتر کار کند این فراخوانی را فراخوانی بازگشتی می نامند و این فراخوانی تا زمانی که فراخوانی اصلی به متد خاتمه نیافته است اجرا می شود ودر هر بار فراخوانی متد خودش را با نسخه ساده تری از مسئله اصلی فراخوانی می کند و دنباله ی مسئله های کوچکتر باید با حالت پایه ختم شوند تا اینکه بازگشتی به اتمام برسد.

مثال: تابع فیبو ناچی را به روش بازگشتی بنویسید؟

Long fibo(int n)

{if(n==0)

return 0;

else

return (fibo(n-2)+fib(n-1);

 مثال: تابع فاکتوریل را به روش بازگشتی بنویسید؟

Long fac(int n)

If(n<=0)

Return (1)

Else return(n\*fac(n-1))

 معایب توابع بازگشتی:فراخوانی مکرر متدها دارای سربارهای ویژه ای است این سربارها می تواند صرف دقت پردازنده یا اشغال حافظه شود هر فراخوانی بازگشتی کپی دیگری از متد را ایجاد می کند.و با این کار حافظه را اشغال می کند.

توابع کتابخانه ای:

 کتابخانه ی c# مجموعه ای است که شامل توابع از پیش تعریف شده و سایر عناصر برنامه است . این توابع و عناصر از طریق سرفایل ها قابل دستابی اند.

|  |  |
| --- | --- |
|  تابع | شرح |
| Acos(x) | کسینوس معکوس x |
| Asin(x) |  سینوس معکوس x |
| Atan(x) |  تانژانت معکوس x |
| Ceil(x) | مقدار سقف x |
| Cos(x) | کسینوس x  |
| Exp(x) |  تابع نمایی x |
| Fabs(x) |  قدر مطلق x |
| Floor(x) |  مقدار کف x |
| Log(x) |  لگاریتم طبیعی x |
| Log10(x) | لگاریتم عمومی x در پایه 10 |
| Pow(x,p) |  Xبه توانp |
| Sin(x) |  سینوس x |
| Sqrt(x) | جذر x |
| Tan(x) |  تانژانت x |

برنامه نویسی شی گرا:

برنامه نویسی شی گرا داده ها (خصوصیات) و توابع (رفتار) را در بسته هایی به نام کلاس محصور می کند .داده ها و توابع یک کلاس در نهایت در ارتباط نزدیک با هم قرار می گیرند. بدون کلاس برنامه نویس نمی تواند شی ایجاد کند و از یک کلاس برای ساختن چند شی از همان کلاس می توان چند بار استفاده کرد .کلاس ها دارای خاصیت پنهان سازی اطلاعات هستند به بیان دیگر اگر چه اشیاء کلاس ممکن است بدانند چگونه با یکدیگر از طریق واسطه های خوش تعریف ارتباط برقرار کنند اما معمولا کلاس ها نباید از چگونگی پیاده سازی کلاس های دیگر اطلاع داشته باشند جزییات پیاده سازی در داخل خود کلاس پنهان است .

تفاوت زبان برنامه نویسی c با c#

در زبان برنامه نویسی c که یک زبان برنامه نویس رویه ای است برنامه نویس تمایل زیادی به عمل گرایی دارد حال آنکه در برنامه نویسی c#تمایل به شی گرایی است و در c واحد برنامه نویس تابع است در c# واحد برنامه نویسی کلاس است که از طریق آن نهایتا اشیاء نمونه سازی می شوند برنامه نویس ها ی c توجه خود را روی نوشتن توابع متمرکز می کنند .گروه هایی از عملیات که یک کار خاص را انجام می دهند تابع را تشکیل می دهند و با گروه بندی توابع ، برنامه تشکیل می شود ودر c داده ها حائز اهمیت فراوان هستند اما نگرش اصلی این است که داده ها در درجه اول در پشتیبانی از عملیاتی وجود دارند که توابع انجام می دهند.

برنامه نویسی c# توجه خود را روی ایجاد انواع داده ای تعریف شده توسط کاربر به نام کلاس ها متمرکز می کنند کلاس ها نیز انواع داده ای تعریف شده توسط برنامه نویسی نامیده می شوند.هر کلاس حاوی داده ها و نیز مجموعه ای از توابع است که داده ها را پردازش می کنند اجزای داده ای یک کلاس عضوهای داده ای نامیده می شوند . اجزای تابعی یک کلاس توابع عضو نامیده می شوند نمونه ای از نوع داده ی تعریف شده توسط کاربر شی نامیده می شود.

طریقه ایجاد کلاس در c#:

 نام کلاس Class نوع سطح دستیابی

 {اعضای کلاس }

انواع سطح دستیابی

1. سطح دستایبی public : تعیین می کند که این کلاس خارج از فضای نامی (مجموعه ای از کلاس ها می باشد که با هم در ارتباط می باشند) که در آن تعریف می شود قابل استفاده است به عبارت دیگر سطح دستیابی public یا عمومی به معنای عدم وجود محدودیتی در دستیابی است .
2. 2- سطح دستیابی internal : مشخص می کند کلاس فقط در همان فضای نامی که تعریف شده قابل استفاده است به عبارت دیگر این نوع سطح دستیابی به معنای داخل فضای نام است .

نمونه سازی از کلاس :

پس از اینکه کلاس تعریف شد برای استفاده از آن باید نمونه ای از آن ایجاد شود ایجاد نمونه ای از کلاس را نمونه سازی و نمونه های کلاس را شی می نامند . شکل کلی نمونه سازی از کلاس بصورت زیر است:

()نام کلاس new = نام شی نام کلاس

نحوه دسترسی به اعضای شی:

پس از اینکه نمونه ای از کلاس یا شی ای از کلاس را ایجاد کردید باید بتوانید به اعضای شی دستیابی داشته باشید برای دستیابی بصورت زیر عمل می کنیم .

عضو کلاس. نام شی

سطح دستیابی اعضای کلاس می تواند یکی از موارد زیر باشد:

1. Public (عمومی): محدودیتی در دستیابی آن وجود ندارد و در خارج از کلاس قابل استفاده است معمولا متدها و خواص با این سطح دستیابی تعریف می شوند تا در خارج از کلاس قابل استفاده باشند.
2. Private(خصوصی) : اعضایی با این سطح دستیابی ، فقط در داخل همان کلاس قابل استفاده و شناخته شده اند . معمولا فیلدها و ثوابت با این سطح دستیابی معرفی می شوند.

مثال: کلاسی بنویسید که حاصل جمع دو عدد را محاسبه کند و سپس دو عدد صحیح دریافت و با استفاده از کلاس حاصلجمع آنها را محاسبه کنید؟

 Class sum

{public int add(int x, int y)

 {return x+y;}

در رویداد کلیک می نویسیم

 Sum a=new sum();

a.x=int.parse(textbox1.text)

a.y=int.parse(textbox2.text)

label1.text=a.add(a.x, a.y).tostring();

مثال: با استفاده از کلاس برنامه ای بنویسید که طول و عرض مستطیلی را دریافت و محیط و مساحت آن را محاسبه کند؟

 Class rectangle

{public float x;

Public float y;

Public int area (float x , float y)

{return x\*y;}

Public int prime(float x, float y)

{return (x+y)\*2;}

 در رویداد کلیک button می نویسیم

Rectangle b=new rectangle();

a.x=float.parse(textbox1.text)

b.y=floatt.parse(textbox2.text)