

مستندات پروژه

موضوع پروژه : برنامه گل فروشی

نام استاد محترم:

تهیه کننده:

الحمد لله  
الرحمن الرحيم

۴	چکیده .....
۵	مقدمه .....
۶	فصل اول: چشم انداز پروژه (VISION) .....
۷	۱-۱- VISION چیست .....
۸	۲-۱- موضوع پروژه .....
۸	۳-۱- اهداف پروژه .....
۸	۴-۱- انجام پروژه چه سودی دارد .....
۹	۵-۱- ذینفعان پروژه .....
۹	۶-۱- سابقه انجام پروژه های مشابه .....
۱۰	۷-۱- معایب پروژه های گذشته .....
۱۱	۸-۱- زبان برنامه نویسی .....
۱۱	۹-۱- پایگاه داده .....
۱۱	۱۰-۱- سیستم عامل مورد نیاز برای پیاده سازی .....
۱۲	فصل دوم: طراحی پایگاه داده سیستم گل فروشی .....
۱۳	نمودار ER کلی پایگاه داده سیستم گل فروشی .....
۱۴	فصل سوم: بررسی C# SOURCE CODE .....
۱۵	در مورد زبان برنامه نویسی .....
۱۵	مروری بر تاریخچه برنامه نویسی .....
۱۷	برنامه نویسی ساخت یافته .....
۱۹	تاریخچه سی شارپ .....
۲۰	اهداف طراحی زبان .....
۲۰	تاریخچه .....
۲۱	ویژگی ها .....
۲۲	سیستم یکپارچه شده .....
۲۲	انواع داده .....
۲۳	BOXING و ENBOXING .....
۲۳	ویژگی های جدید در C# 2.0 .....
۲۳	کلاسهای PARTIAL .....
۲۴	GENERIC ها .....
۲۴	کلاس های STATIC .....
۲۴	یک شکل جدید از تکرار کننده با استفاده از سازنده توابع .....
۲۵	DELEGATE های ناشناس .....
۲۵	CONTRAVARIANCE DELEGATE COVARIANCE AND .....

۲۵	..... NULLABLE نوع داده
۲۷	..... کدهای مربوط به فرم ورود به سیستم
۲۹	..... کدهای مربوط به فرم اصلی سیستم گل فروشی
۳۳	..... کدهای مربوط به فرم ثبت دسته بندی گل
۳۷	..... کدهای مربوط به فرم ثبت واحد شمارش گل
۴۱	..... کدهای مربوط به فرم تعریف عنوان گل
۴۶	..... کدهای مربوط به فرم ثبت موجودی گل
۵۱	..... کدهای مربوط به فرم ثبت مشخصات مشتری
۵۷	..... کدهای مربوط به فرم ثبت فاکتور فروش
۶۶	..... کدهای مربوط به فرم ثبت کاربران سیستم
۷۳	..... چاپ فاکتور
۷۵	..... <b>فصل چهارم: جمع بندی ، نتیجه گیری و پیشنهادات</b>
۷۶	..... جمع بندی
۷۷	..... نتیجه گیری
۷۷	..... پیشنهادات
۷۹	..... منابع
۷۹	..... کتابها
۷۹	..... وب سایتها

## چکیده

در این نوشتار ، به بررسی و تحلیل سیستماتیک یکی از نرم افزارهای تحت ویندوز با عنوان گل فروشی پرداخته شده است.

هدف از تحلیل این سیستم نرم افزاری ، یافتن راه حلهای مناسب برای پیاده سازی می باشد. یافتن راه حل مناسب نیاز به بررسی دارد. برای بررسی بایستی از یکی از روشهای تحلیل مناسب استفاده نمود. روشی که برای تحلیل این سیستم استفاده شده است روش RUP می باشد که برای نمایش برخی از مراحل آن از نمودارهای UML استفاده شده است. برای این منظور:

- در فصل اول Vision طرح مورد بررسی قرار گرفته است
- در فصل دوم نمودار Use Case منطقی سیستم به همراه سناریوهای هر یک از Use Case های ارائه شده است
- در فصل سوم نمودار کلاس مورد بررسی قرار گرفته است
- در فصل چهارم ، بر اساس نمودار کلاس ، پایگاه داده طراحی و پیاده سازی گردیده است
- در فصل پنجم نمودار توالی ارائه شده است
- در فصل ششم نمودار همکاری بر اساس نمودارهای توالی ارائه شده است
- در فصل هفتم Source Code های برنامه به همراه واسط کاربر ارائه گردیده اند
- نهایتاً در فصل هشتم به جمع بندی و نتیجه گیری پرداخته شده است.

## مقدمه

نرم افزار مدیریت فروش گل فروشی ابزاری نیرومند است که به شایستگی قادر است پاسخ گوی تمام نیازهای مدیریتی باشد. در واقع این نرم افزار به همراه مشاوره و همکاری کارشناسان و متخصصان مجرب پیاده سازی سیستم گل فروشی ها تهیه گردیده که موجبات صرفه جویی در زمان و بر طرف نمودن کلیه مشکلات شیوه های دستی و قدیمی در گل فروشی ها را فراهم آورده است.

نرم افزار فوق قابل استفاده در کلیه گل فروشی ها و کلیه مراکز و محلهای مشابه می باشد. این نرم افزار دارای برنامه حسابداری فاکتور می باشد این امکان را به مدیریت می دهد که با اطمینان کامل و بهتری بتواند مجموعه خود را کنترل و مدیریت کند. در ادامه به بررسی ، تحلیل و پیاده سازی پروژه گل فروشی خواهیم پرداخت.

## **فصل اول: چشم انداز پروژه (Vision)**

در این فصل ابتدا تعریفی از Vision ارائه می دهیم و سپس به بررسی مواردی می پردازیم که باید در چشم انداز پروژه یا همان Vision باید مورد بررسی قرار گیرند.

## ۱-۱- Vision چیست

Vision یا چشم انداز پروژه در حقیقت یک سری مستندات اولیه پروژه است که شامل بررسی موارد ذیل می باشد:

- ۱- موضوع پروژه
- ۲- اهداف پروژه
- ۳- انجام پروژه چه سودی دارد
- ۴- ویژگیهای پروژه
- ۵- ذینفعان پروژه چه کسانی هستند
- ۶- سابقه انجام پروژه های مشابه
- ۷- معایب پروژه های گذشته
- ۸- زبان برنامه نویسی
- ۹- پایگاه داده
- ۱۰- سیستم عامل مورد استفاده
- ۱۱- روش تحلیل
- ۱۲- نرم افزارهای مورد نیاز برای تحلیل و پیاده سازی



## ۱-۲- موضوع پروژه

موضوع پروژه جاری پیاده سازی یک نرم افزار با عنوان سیستم گل فروشی می باشد.

## ۱-۳- اهداف پروژه

- هدف از پیاده سازی ، سیستمی است که با آن بتوان عملیات فروش گل و ایجاد فاکتور مشتری را با سطوح دسترسی مختلف انجام داد.
- در پیاده سازی این پروژه اهداف تجاری مد نظر نمی باشد. بلکه فقط اهداف آموزشی زیر مد نظر می باشد: (البته ممکن است از این سیستم در مقاصد تجاری نیز استفاده شود)

۱- آشنایی با برنامه های تحت ویندوز

۲- آشنایی با تحلیل این گونه برنامه ها

۳- آشنایی با روش پیاده سازی این گونه نرم افزارها

۴- آموزش برنامه نویسی سیستمهای تجاری

۵- آموزش نحوه استفاده از ابزارهای برنامه نویسی برای پیاده سازی این گونه سیستمها

۶- آموزش نحوه ارتباط بین پایگاه داده و فرمها در این گونه سیستمها

۷- آشنایی با تکنولوژی برنامه نویسی LINQ

## ۱-۴- انجام پروژه چه سودی دارد

- هزینه دوره های آموزشی خارج از دانشگاه برای دانشجو حذف می شود
- انجام این پروژه از نظر کارآموزی و آموزش نحوه انجام یک پروژه تجاری برای کسانی که به نحوی با پروژه سرکار دارند بسیار مفید است

- Component هایی که در این پروژه تولید می شوند اکثراً در پروژه های آینده و یا پروژه های مشابه نیز قابل استفاده اند و این مساله خود باعث صرفه جویی در زمان اجرای پروژه های دیگر خواهد شد

## ۱-۵- ذینفعان پروژه

### • تحلیل گر

- کسی است که عملیات تحلیل سیستم را انجام می دهد و با سیستم به خوبی آشنایی دارد.

### • برنامه نویس

- کسی است که عملیات پیاده سازی و برنامه نویسی را با استفاده از مستندات تولید شده توسط تحلیلگر، انجام می دهد.

### • افراد تحت تعلیم دوره

- افراد تحت تعلیم دوره معمولاً دانشجویانی هستند که در این دوره آموزشی شرکت نموده اند.

### • مشتریان

- مشتریان کسانی هستند که در آینده این نرم افزار را خریداری می نمایند و به نحوی از این برنامه استفاده می نمایند.

## ۱-۶- سابقه انجام پروژه های مشابه

نسخه های بسیار متوعی از برنامه گل فروشی در سرتاسر دنیا به زبانهای مختلف (چه از لحاظ برنامه نویسی و چه از لحاظ زبان ملی) پیاده سازی شده اند که زمینه فعالیت آنها به شرح ذیل می باشد:

### ۱- فعالیتهای علمی

a. دانشگاهی

b. پژوهشی

c. آموزشی

## ۲- فعالیتهای تجاری

a. خرید

b. فروش

c. بازاریابی

## ۳- فعالیتهای عمومی

a. درج خلاصه فروش

b. معرفی گل

## ۷-۱- معایب پروژه های گذشته

در بررسی به عمل آمده از ایرادهای زیر مشاهده شد که بیشتر این ایرادات در نرم افزارها معمولا مشترک می باشند:

- عدم وجود امکان جستجوی عادی
- وجود اشکال در جستجو با حروف فارسی در هنگام تایپ (مثلا با حرف ی و گ بیشتر مشکل دارند و نتایج مختلفی ارائه می شود و گاهی اوقات نتیجه مطلوبی از جستجو حاصل نمی شود)
- فرم ثبت نام مشتری بزرگ
- عدم وجود جستجوی پیشرفته برای اعضا ( در برخی موارد به یک جستجوی ساده اکتفا شده بود که آن هم نتایج مطلوبی ارائه نمی کرد)
- استفاده از کنترلهای غیر استاندارد
- گرافیک بسیار خسته کننده
- عدم پاسخدهی به موقع در بخش پشتیبانی
- عدم پشتیبانی از سیستم عاملهای مختلف مختلف

- عدم وجود دسته بندی گل

- ....

البته ایرادات فنی بیشماری در نرم افزارها موجود می باشد که به علت طولانی شدن بحث از ارائه آنها خودداری می شود.

## ۱-۸- زبان برنامه نویسی

زبان برنامه نویسی Base پروژه C# 2010 می باشد که C# در مجموعه Visual Studio 2010 موجود و قابل نصب می باشند. تکنولوژی مورد استفاده نیز LINQ می باشد.

## ۱-۹- پایگاه داده

پایگاه داده مورد استفاده در سیستم SQL Server می باشد که مناسب برای نصب بر روی ویندوز است.

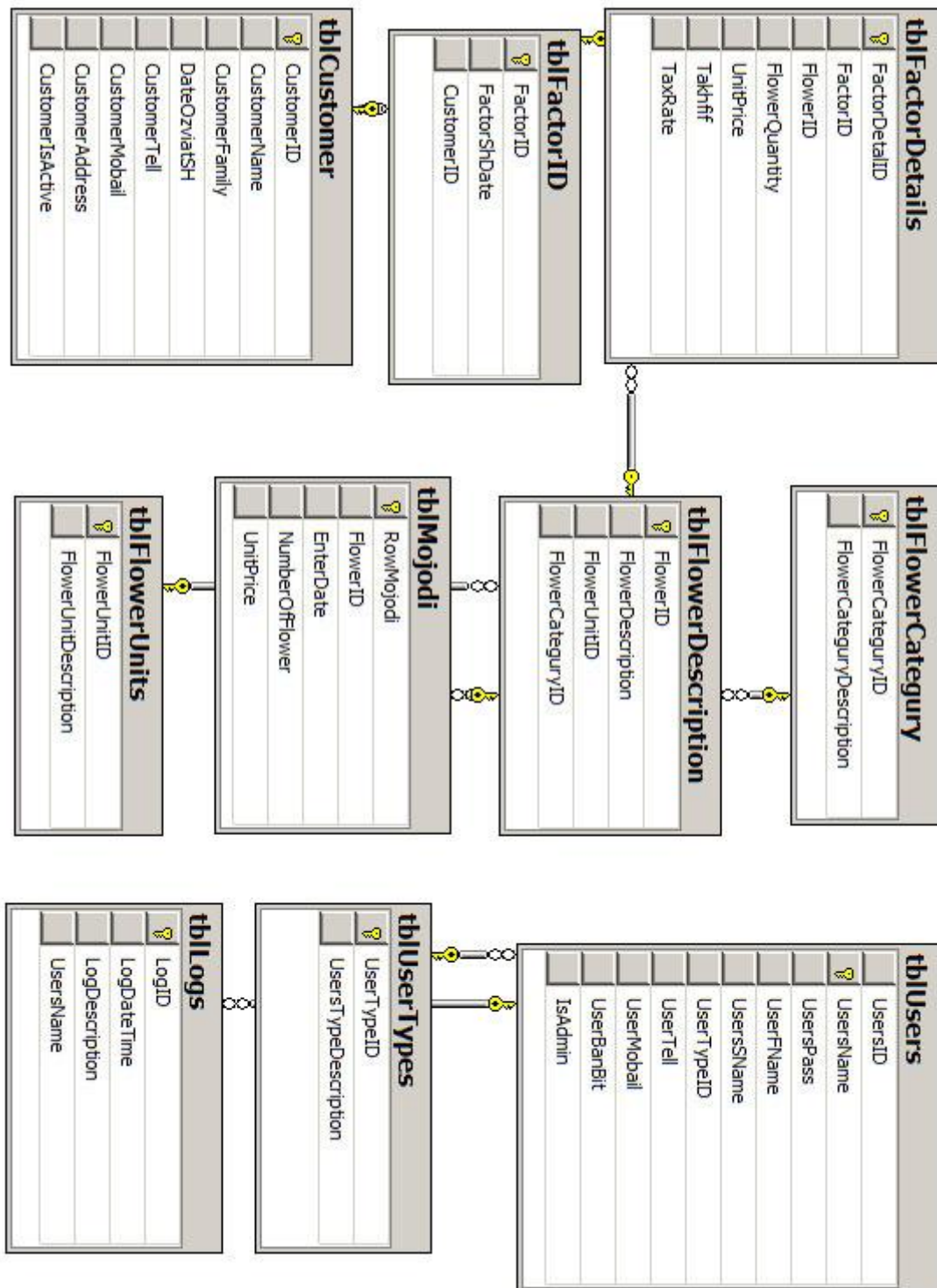
## ۱-۱۰- سیستم عامل مورد نیاز برای پیاده سازی

- سیستم عامل استاندارد برای کارهای حرفه ای Windows Server می باشد. ولی در این مورد ما از

سیستم عامل ویندوز ۱۰ استفاده خواهیم نمود.

## **فصل دوم: طراحی پایگاه داده سیستم گل فروشی**

## نمودار ER کلی پایگاه داده سیستم گل فروشی



نمایه 1- نمودار ER

## فصل سوم: بررسی Source Code C#

## در مورد زبان برنامه نویسی

در این فصل ابتدا توضیحاتی در مورد زبان برنامه نویسی ارائه خواهد شد و سپس به بررسی کدهای نوشته شده در کنار شکل فرمها پرداخته خواهد شد.

## مروری بر تاریخچه برنامه نویسی

برنامه نویسی رایانه به گونه ای که ما امروز می شناسیم ، در اواخر دهه ی ۵۰ میلادی شروع شد و در دهه ی ۶۰ میلادی سرعت رشد نمود . در آن دوران برنامه نویسی یک کار تخصصی و مهندسی تلقی نمیشد زیرا رایانه ها بسیار محدود بودند و کارهای ساده ای انجام می دادند . اغلب اینگونه بود که افراد بسیار سریعتر از رایانه ها پاسخ محاسبات ریاضی را می یافتند . لذا برنامه نویسی برای چنین دستگاه هایی چندان ارزش و اهمیت نداشت و اولین برنامه ها فقط چند دستور ساده ی متوالی بودند . پس از اجرای این دستورها رایانه باید از نو راه اندازی میشد تا برای اجرای دوباره ی برنامه آماده شود .

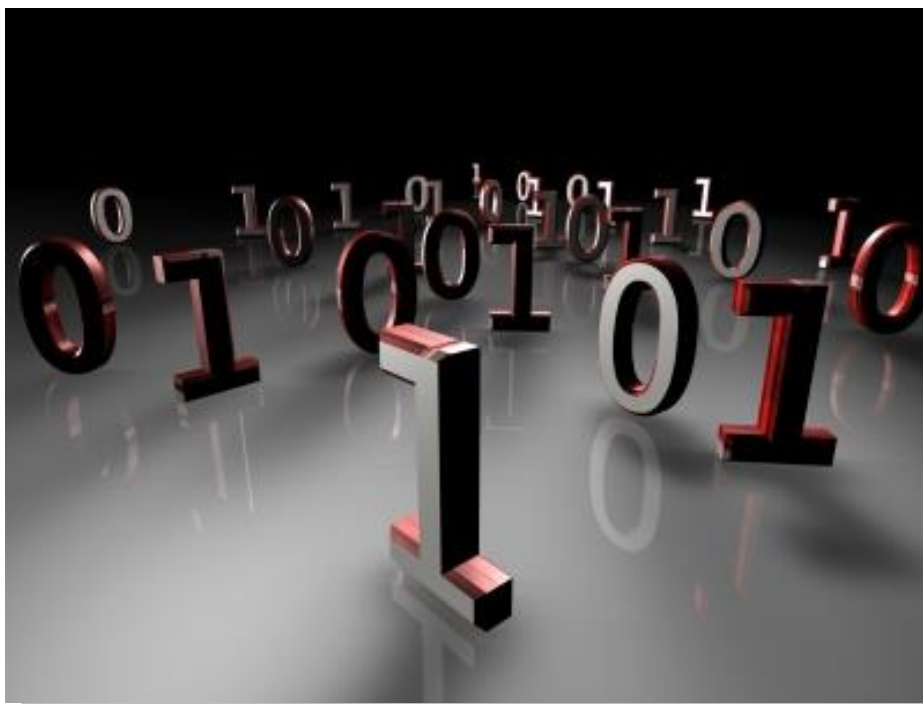
کم کم سخت افزار متحول شد و رایانه ها قدرتمندتر و کارآمدتر شدند اما به همان نسبت تولید برنامه برای آنها دشوار گشت طوری که برنامه های ساده ی اولیه ، مشکل جدی بر سر راه کاربری رایانه محسوب شدند . پس تلاش هایی صورت گرفت تا برنامه های کارآمدتری نوشته شود . اما طبق یک قانون کلی ، پیچیدگی نرم افزار بیشتر و بیشتر می گردد تا جایی که دیگر انسان نمی تواند آن را مدیریت کند و از همان جا هرج و مرج شروع می شود .

بزودی نهضت همگانی صورت گرفت تا راهکاری برای مدیریت پیچیدگی و کارایی برنامه ها یافت شود . سرانجام در اواخر دهه ی ۶۰ " برنامه نویسی ساخت یافته " ( Procedural programming ) بعنوان راه حل پذیرفته شد. برنامه نویسی ساخت یافته امکان میداد که برنامه به چند تابع تقسیم شود و چند نفر بطور همزمان کار تولید نرم افزار را پیش ببرند و همین امر سبب افزایش سرعت تولید نرم افزار و کاهش خطاهای برنامه نویسی گردید و برنامه های بزرگ و مفید ساخته شدند . برنامه هایی که در ابتدا با این روش ایجاد شدند چیزی بین ۱۰ تا ۲۰ هزار خط کد داشتند ، از ساختمان داده ی کوچکی بهره می جستند و الگوریتم های ساده



ای در آنها به کار می رفت . با استفاده از برنامه نویسی ساخت یافته براحتی میشد چنین برنامه ای را مدیریت کرد .

برنامه ها آرام آرام به قصد بهبود کارایی و حجیم تر شدند . الگوریتم های پیشرفته ای بکار گرفته شد و ساختمان داده ی پیچیده تری در ترکیب برنامه ها استفاده شد . برنامه نویسی ساخت یافته دیگر نمی توانست از پس کدهای طولانی چنین برنامه هایی برآید ( سیستم عامل ویندوز NT بیش از ۱۰ میلیون خط کد دارد ) .

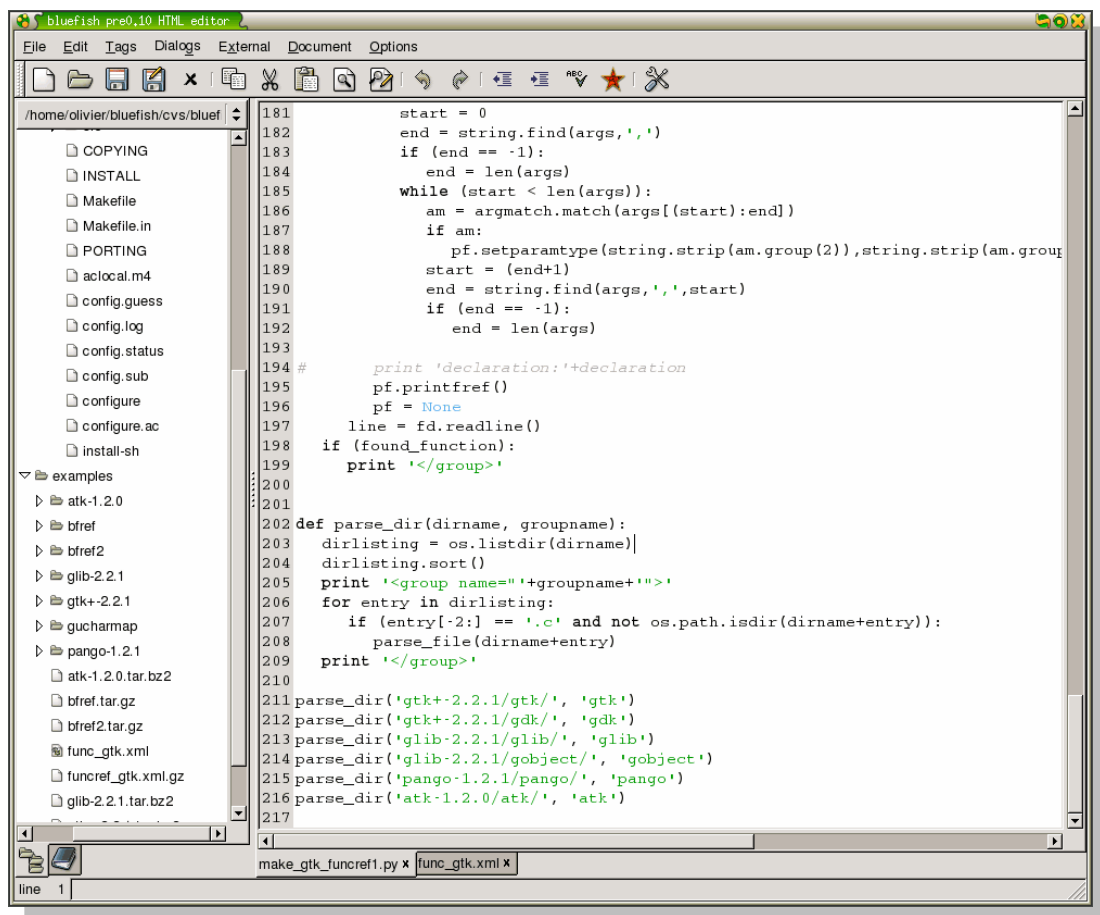


نمایه 2 - اعداد باینری

بنظر میرسید که روش دیگری لازم است تا تولید کنندگان نرم افزار را از دست متغیرهای فراوان و توابع طولانی و سرگیجه آور خلاص کند . بر اساس این تفکر بود که برنامه نویسی شیء گرا ( Object oriented programming ) پیش کشیده شد . تفکر شیء گرا تقریباً همزمان با تفکر ساخت یافته مطرح شد اما فقط برای طراحی برنامه استفاده میشد . پس از عاجز ماندن برنامه نویسی ساخت یافته ، شیء گرایی در برنامه نویسی هم استفاده شد . حالا برنامه نویسان می توانند در برنامه به پانزده شیء فکر کنند که این اشیاء با چهل پیام مشخص با هم ارتباط دارند ، حال آنکه همین برنامه اگر به سبک ساخت یافته ایجاد شود باید به ۱۰۰ متغیر ، ۲۰ ثابت ، ششصد تابع و ... فکر کرد .

## برنامه نویسی ساخت یافته

برنامه نویسی ساخت یافته روش منظمی برای تولید برنامه هاست . برنامه هایی که به این شیوه نوشته میشوند خوانایی زیادی دارند ( یعنی راحت درک میشوند ) , آزمایش و اشکال زدایی و اصلاح آنها نیز آسان است . در این روش برنامه بصورت مجموعه ای از فعالیت ها تصور میشود که باید بر روی داده ها انجام شوند . هر کار پیچیده ای به چند کار کوچکتر تجزیه میشود تا این کارها براحتی قابل پیاده سازی باشند . سپس برای انجام هریک از این کارها , زیر برنامه ی خاصی که تابع ( Function ) نام گرفته نوشته میشود . به همین دلیل به این شیوه " برنامه نویسی بالا به پایین " ( Up-Down programming ) نیز می گویند . برنامه نویسی ساخت یافته روش موفقی در حل مسائل پیچیده است . توابع قابلیت استفاده ی مجدد دارند . یعنی میتوان توابع یک برنامه را در برنامه های دیگر نیز بکار برد . به این ترتیب کد نویسی کمتر میشود و تولید نرم افزار با سرعت بیشتری صورت می گیرد .



نمایه 3 - محیط ویرایشگر کد

توابع امکان می دهند تا تولید نرم افزار با استفاده از گروه ممکن شود . به این شکل که پس از مشخص شدن توابع مورد نیاز ، هریک از اعضای گروه وظیفه ی نوشتن و آزمایش توابع مشخصی را برعهده می گیرد و به این شکل اعضای گروه بصورت همزمان روی بخش های مختلف برنامه کار می کنند بدون اینکه یکی منتظر دیگری بماند . در نهایت این توابع به هم پیوند می خورند تا جریان منطقی برنامه برقرار شود .

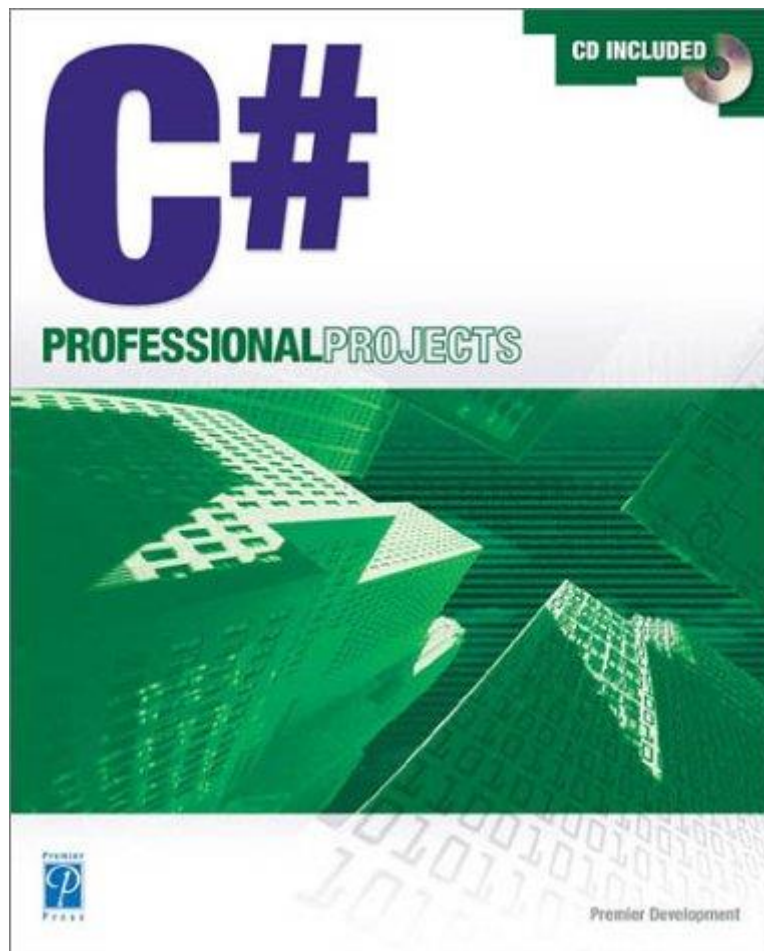
بدیهی است که عیب یابی و اصلاح برنامه نیز آسان میشود ، زیرا با بررسی جریان برنامه ، تابع معیوب بسرعت شناسایی و اصلاح میشود بدون اینکه بقیه ی بخش های برنامه تغییر یابند . زبان های C و PASCAL دو نمونه از زبان های برنامه نویسی ساخت یافته هستند .

برای زبان C# همانند بیشتر زبانهای برنامه سازی دو نوع کامپایلر وجود دارد. یک نوع کامپایلر که به کامپایلر Command Line معروف است و نوع دیگر کامپایلر Visual است. کامپایلرهای Command Line محیطی شبیه به محیط DOS دارند و با دادن یک سری دستورات به اجرا در می آیند. کامپایلرهای Visual محیطی همانند ویندوز دارند که با دارا بودن محیط گرافیکی و ابزارهای خاص، برنامه نویس را در امر برنامه سازی کمک می کنند. از نمونه های هر یک از کامپایلرها، می توان به Microsoft C# Command Line Compiler که یک کامپایلر Command Line و Microsoft Visual C# که یک کامپایلر Visual است، اشاره کرد. البته در حال حاضر بیشتر از کامپایلرهای ویژوال استفاده می شود.

اعلان Namespace به سیستم اعلان می نماید که شما از توابع کتابخانه ای System جهت اجرای برنامه ها خود استفاده می نمایید. دستوراتی مانند WriteLine و ReadLine جزو توابع کتابخانه ای System می باشند. اغلب دستورات و توابع مهم و کلیدی استفاده از کنسول ورودی/خروجی در این کتابخانه می باشد. در صورتیکه در ابتدای برنامه از using System استفاده نکنید، باید در ابتدای هر یک از دستورات برنامه که مربوط این کتابخانه است، از کلمه System استفاده نمایید.

## تاریخچه سی شارپ

سی شارپ (#C) زبانی شیء گرا و سطح بالا (high level) از خانوادهٔ زبان‌های چارچوب دات نت شرکت مایکروسافت است. زبان C#، یک زبان برنامه نویسی چند الگویی است و منظم شده مدل های تابعی، امری، عمومی، شی گرا و جز گرا می باشد. این زبان توسط ماکروسافت و جزئی از دات نت به وجود آمد و بعداً استانداردهای ECMA و ISO را نیز در بر گرفت. C# یکی از ۴۴ زبان برنامه نویسی ای است که توسط Common Language Runtime از .NET Framework پشتیبانی می شوند و در همه جا به وسیله Microsoft Visual Studio شناخته می شود.



نمایه 4- عکس جلد کتاب آموزش سی شارپ

این زبان برپایه سادگی، مدرن بودن، همه منظوره و شی گرا بودن ساخته شد. Anders Hejlsberg، طراح زبان برنامه نویسی دلفی، سرپرستی تیم طراحان زبان C# را بر عهده داشت. این زبان دارای دستوری شی گرا مشابه C++ است و به شدت از زبان های جاوا و دلفی تاثیر پذیرفته است. در ابتدا نام این زبان COOL بود که مخفف C like Object Oriented Language بود، هر چند در جولای ۲۰۰۰، زمانی که ماکروسافت پروژه را عمومی اعلام کرد، اسم آن به C# تغییر پیدا کرد. آخرین نسخه آن نسخه ۳ است که از ویزوال استدیو ۲۰۰۸ استفاده می کند. نسخه بعدی ۴ است که در دست ساخت است.

## اهداف طراحی زبان

استاندارد ECMA این اهداف طراحی زبان را برای C# بر آورده می سازد:

سی شارپ یک زبان برنامه سازی ساده، مدرن، برای اهداف عمومی و شی گرا است.

به دلیل اهمیت داشتن موضوع نیرومندی و دوام و بهره وری برنامه نویس، زبان دارای چک کننده Strong Type، چک کننده مرزهای آرایه، تشخیص حالتی که یک متغیر مقدار دهی اولیه نشده است، قابلیت انتقال کدها و Garbage Collection خودکار است.

این زبان برای استفاده در اجزای توسعه نرم افزار برای دستیابی به مزایای سیستم های توزیعی در نظر گرفته شده است.

قابلیت انتقال برنامه نویس بسیار مهم است، خصوصاً برای آن دسته از برنامه نویسانی که با زبان های C و ++C آشنا هستند.

پشتیبانی از این زبان برای بین المللی شدن بسیار مهم است.

زبان سی شارپ برای نوشتن برنامه ها برای سیستم های تعبیه شده و میزبان در نظر گرفته شده است، از سیستم عامل های پیچیده بسیار بزرگ گرفته تا توابع اختصاصی بسیار کوچک.

هر چند برنامه های نوشته شده با C# طوری هستند که از لحاظ حافظه و پردازنده مورد نیاز مقرون به صرفه باشند، ولی خود زبان از لحاظ اندازه و کارایی به خوبی زبان های C و اسمبلی نیست.

## تاریخچه

در سال ۱۹۹۹، شرکت سان اجازه استفاده از زبان برنامه نویسی جاوا را در اختیار ماکروسافت قرار داد تا در سیستم عامل خود از آن استفاده کند. جاوا در اصل به هیچ پلت فرم یا سیستم عاملی وابسته نبود، ولی ماکروسافت برخی از مفاد قرار داد را زیر پا گذاشت و قابلیت مستقل از سیستم عامل بودن جاوا را از آن برداشت. شرکت سان پرونده ای علیه ماکروسافت درست کرد و ماکروسافت مجبور شد تا زبان شی گرای جدیدی با کامپایل جدید که به ++C شبیه بود را درست کند. در طول ساخت دات نت، کلاس های کتابخانه ای با زبان و کامپایلر SMC نوشته شدند. در سال ۱۹۹۹ آندرس هلزبرگ گروهی را برای طراحی زبانی جدید تشکیل داد که در آن زمان نامش Cool بود و همانند C بود با خواص شی گرایی. ماکروسافت در نظر داشت اسم این زبان را تا آخر Cool قرار دهد، ولی به دلیل مناسب نبودن برای اهداف تجاری این کار را نکرد. در ارائه و معرفی رسمی .NET در PDC در سال ۲۰۰۰ این زبان به C# سی شارپ تغییر نام یافت و کتابخانه کلاس ها و runtime در ASP.NET به C# منتقل شدند. مدیر و سرپرست طراحان در ماکروسافت آندرس هلزبرگ بود که تجربه قبلی او در طراحی Framework و زبان های برنامه سازی ++C, Delphi, Borland

Turbo Pascal, Visual C به آسانی در دستورالعمل‌های سی شارپ قابل رویت است و به همان خوبی در هسته CLR.

## ویژگی‌ها

برخی از تفاوت‌های زبان سی شارپ با C و C++ عبارتند از:

هیچ تابع یا متغیر سراسری (Global) وجود ندارد، تمام متدها و اعضا بایستی در داخل کلاس‌ها تعریف شوند. این امر ممکن است، هر چند برای استفاده از متغیرها و توابع عمومی باید از متدها و متغیرها در کلاس‌های عمومی استفاده کرد.

متغیرهای عمومی، برخلاف C و C++، نمی‌توانند بلاک‌های پیوستی را در بر بگیرند.

سی شارپ دارای یک نوع داده بولی است (bool). برخی از عبارات‌ها مانند while و دستور شرطی if // که شرطی هستند، نیازمند یک عبارت نوع بولی هستند. همان‌طور که C++ نیز دارای نوع داده بولی است، این نوع داده به راحتی میتواند به یا از Integerها تبدیل شود، و عبارتی مانند (دستور شرطی if(a // نیازمند این امر است که a از یک نوع قابل تبدیل به bool یا اشاره گر باشد. کامپایلر سی شارپ برنامه نویس را در این شرایط مجبور به استفاده از عباراتی می‌کند که به درستی یک مقدار bool را برمیگردانند. بنابراین دستوری مانند (دستور شرطی if(a = b // باعث بروز خطا می‌شوند. (به جای = بایستی از == استفاده شود)

در سی شارپ، اشاره گرهای به حافظه بایستی فقط در داخل بلوکهای unsafe استفاده شوند و برنامه در این حالت برای اجرا نیاز به اجازه از کاربر دارد. بیشتر دسترسی‌های از طریق شی امن است که یا همیشه در حال اشاره به شی صحیح موجود است یا یک مقدار Null دارد. اشاره گری به شی به درد نخور یا بلاک حافظه رندم غیر ممکن است. اشاره گر نا امن می‌تواند به نمونه‌ای از value-type، آرایه، رشته یا بلاکی که حافظه به آن داده شده است اشاره نماید. کدی که به عنوان نا امن علامت نخورده باشد، هنوز می‌تواند اشاره گرها را از سیستم بازیابی یا در آن ذخیره کند ولی نمی‌تواند مرجع جدیدی به آنها اختصاص دهد.

حافظه ساماندهی شده نمی‌تواند صریحاً آزاد شود، ولی به طور خودکار به عنوان به درد نخور تلقی می‌شود. انتخاب آدرس‌های به درد نخور حافظه نفوذ ناپذیر است. هم چنین C# با استفاده از عبارات، پشتیبانی مستقیمی از پایان اجباری می‌کند (پشتیبانی از اصطلاح Resource Acquisition Is Initialization).

وراثت چندگانه از کلاس‌ها در این زبان پشتیبانی نمی‌شود. البته یک کلاس امکان ارث بری از تعداد نامحدود واسطه‌ها را دارد. پشتیبانی نکردن از وراثت چندگانه به دلیل اهداف معماری این زبان در CLI و برای جلوگیری از پیچیدگی است.

سی شارپ بسیار typesafe تر از C++ است. تنها تبدیلات ضمنی مثل تبدیل نوع داده کوچکتر به بزرگتر یا تبدیل نوع مشتق شده به نوع پایه به طور پیش فرض و بدون خطا صورت می‌پذیرد. هیچ تبدیل ضمنی‌ای میان Booleanها و Integerها وجود ندارد و هر تبدیل user-defined بایستی به صراحت با یکی از

کلمات explicit یا implicit نشانه گذاری شود. تبدیل b به a در حالتی که a یک Integer و b یک double باشد در زبان ++C مجاز است اما در سی شارپ به یک خطای زمان کامپایل منجر می شود (بایستی به صورت explicit تعریف شود) اعضای Enumeration در داخل محدوده شخصی خود قرار دارند. C# قابلیت syntactic sugar را برای توابع متداول، اکسسور ها و ماجول های کسول شده در یک کلاس به صورت ویژگی ها قرار داده است. اکسسورها که خاصیت نیز گفته می شوند در زبان سی شارپ قادر به کنترل دسترسی اعضا و معتبرسازی داده ها هستند. تمام انواع بازتابی (Reflection) و بازیابی (Recovery) قابل استفاده است. در حال حاضر (۳ جوان ۲۰۰۸) دارای ۷۷ کلمه رزرو شده است.

### سیستم یکپارچه شده

سی شارپ دارای یک سیستم نوع یکپارچه است که به آن CTS می گویند. این بدان معناست که تمام انواع، شامل موارد اصلی مانند Integer ها، مشتق شده از System.Object هستند. به عنوان مثال، هر نوع یک متد به نام ToString() را به ارث می برد. بخاطر کارائی، انواع اولیه (و انواع مقداری) به طور داخلی فضایی برای آنها بر روی پشته در نظر گرفته می شود.

### انواع داده

CTS داده ها را به دو نوع تقسیم می کند:

- نوع مقداری
- نوع مرجعی

انواع داده ای توده ساده ای از داده می باشند. نمونه های انواع داده ای نه حویت مرجعی دارند و نه مفاهیم مقایسه مراجع را. برای مقایسه برابری یا عدم برابری انواع داده ای، خود مقدار داده ها را با یکدیگر مقایسه می کنیم مگر اینکه عملگر های مشابه دوباره تعریف شده باشند. مقادیر داده های مرجعی همیشه یک مقدار پیش فرض دارند و همیشه می توانند ایجاد و یا کپی شوند. یکی دیگر از محدودیت های انواع داده ای این است که آنها نمی توانند از یکدیگر مشتق شوند (ولی میتوانند اشتراکاتی داشته باشند) و هم چنین نمی توانند در سازنده مقدار دهی اولیه شوند. مثالی از انواع داده ای، بعضی از انواع اولیه مانند int و float و char و System.DateTime می باشند. در مقابل، انواع مرجعی مفهوم تعریف مرجعی را دارند (که در آن هر نمونه از نوع مرجع، به طور ذاتی از دیگر نمونه ها جدا می شود، حتی اگر داده هر دو نمونه یکی باشد). این دقیقاً نمونه مشابه مقایسه تساوی یا عدم تساوی داده های مرجعی است، که در آن آزمایش برای مرجع ها از داده ای ها سریع تر است. در کل نه همیشه امکان تعریف نمونه مرجعی وجود دارد و نه امکان کپی یا نمایش مقادیر مقایسه دو نمونه. ولی به هر حال انواع مرجعی خاص می توانند این اعمال را از طریق سازنده های عمومی یا

اجرای واسطه‌های مشابه (مثل ICloneable یا IComparable) انجام دهند. نمونه‌هایی از انواع مرجعی، اشیاء، System.String و System.Array می‌باشند. هر دو نوع داده قابلیت انعطاف توسط تعریف به وسیله کاربر را دارند.

## Boxing و EnBoxing

Boxing عمل تبدیل مقدار نوع داده‌ای به نوع مرجع مشابه آن می‌باشد.

مثال:

```
int foo = 42;           // Value type...
object bar = foo;       // foo is boxed to bar.
```

UnBoxing عمل تبدیل نوع مرجع به نوع داده‌ای می‌باشد. مثال:

```
int foo = 42;           // Value type.
object bar = foo;       // foo is boxed to bar.
int foo2 = (int)bar;    // Unboxed back to value type.
```

سی شارپ به برنامه‌نویس با استفاده از کلمه کلیدی Struct اجازه می‌دهد تا انواع مقداری User-defined را ایجاد کند. از دیدگاه برنامه‌نویسی، آنها کلاس‌های سبک وزن به نظر می‌رسند. برخلاف کلاس‌ها (که بر روی heap قرار می‌گیرند) و شبیه به انواع اولیه استاندارد مانند انواع مقداری Structها نیز بر روی پشته قرار می‌گیرند. آنها همچنین می‌توانند قسمتی از یک شیء باشند، یا در یک آرایه مرتب شوند، بدون حافظه غیر مستقیمی که به طور معمول برای انواع کلاس تخصیص می‌یابد.

## ویژگی‌های جدید در C# 2.0

ویژگی‌های جدید در C# .NET SDK 2.0 (مطابق با سومین ویرایش استاندارد ECMA-334):

### کلاسهای partial

کلاس‌های Partial اجازه اجرای کلاس‌ها از بیش از یک سورس فایل را می‌دهند. این امر اجازه می‌دهد تا کلاس‌های بسیار بزرگ را قطعه قطعه کنیم و همچنین برای زمانی که برخی قسمت‌های یک کلاس به طور خودکار تولید می‌شوند مفید است.

file1.cs:

```
public partial class MyClass
{
    public MyClass()
    {
        // implementation
    }
}
```



```
}
```

file2.cs:

```
public partial class MyClass
{
    public void SomeMethod()
    {
        // implementation
    }
}
```

## Generic ها

generic ها یا نوع های پارامتری شده یا چندریختی های پارامتری یک ویژگی جدید NET ۲,۰ است که به وسیله سی شارپ پشتیبانی می شود. برخلاف Template های سی پلاس پلاس، در این انواع به جای اینکه نمونه سازی توسط کامپایلر انجام شود، در زمان اجرا صورت می گیرد، بنابراین می توانند چند زبانه باشند در حالی که C++ نمی تواند. آنها دارای ویژگی هایی هستند که به طور مستقیم توسط Template های C++ پشتیبانی نمی شوند مانند نوع محدودیت ها در پارامترهای Generic با استفاده از رابط ها (Interface). سی شارپ از پارامترهای Generic بدون نوع پشتیبانی نمی کند. برخلاف generic های جاوا، generic های دات نت برای پارامتری کردن انواع داده ای در اشیا ماشین مجازی CLI، از مفاهیم شی گرای استفاده می کنند که اجازه بهینه سازی و حفاظت انواع اطلاعات را می دهد.

## کلاس های static

کلاس ها به صورت Static قابل تعریف نیستند مگر اینکه تمام اعضای آنها Static باشند. که این امر بسیار شبیه به مفهوم مدل در زبانهای رویه ای است. (زبان رویه ای : یک زبان برنامه نویسی که در آن عنصر اصلی برنامه نویسی یک زیربرنامه است. مانند زبانهای C، پاسکال و...)

## یک شکل جدید از تکرار کننده با استفاده از سازنده توابع

یک شکل جدید از iterator (تکرار کننده)، با استفاده از ساختار yield return بسیار شبیه به yield زبان Python.

```
// Method that takes an iterable input (possibly an array) and returns all
even numbers.
public static IEnumerable<int> GetEven(IEnumerable<int> numbers)
{
    foreach (int i in numbers)
    {
        // دستور شرطی if (i % 2 == 0) yield return i;
    }
}
```

## Delegate های ناشناس

Delegate های ناشناس که عملکردهای محدودی را در C# به وجود می آورند. کد کنار بدنه Deletage ناشناس، دسترسی کامل برای خواندن یا نوشتن در متغیرهای عمومی، پارامترهای توابع و اعضای کلاسهای دارای محدوده Deletage را دارد ولی پارامترهای out و ref را پشتیبانی نمی کند. برای مثال :

```
int SumOfArrayElements(int[] array)
{
    int sum = 0;
    Array.ForEach(
        array,
        delegate(int x)
        {
            sum += x;
        }
    );
    return sum;
}
```

## contravariance Delegate covariance and

تبدیل گروه های متد به نوع Deletage در برگشت دارای covariant و در انواع پارامترها دارای contravariant هستند.

اکسسورهای یک خاصیت (set و get) می توانند دارای سطح دسترسی متفاوتی باشند.

مثال :

```
string status = string.Empty; // حذف مقادیر متنی

public string Status
{
    get { return status; } // anyone can get value of this
    property,
    protected set { status = value; } // but only derived classes can
    change it
}
```

نکته مهم : سطح دسترسی خاصیت نمی تواند بالاتر از سطح دسترسی اکسسورها باشد. به عنوان مثال private بودن خاصیت و public بودن اکسسور باعث بروز خطا می شود.

## نوع داده Nullable

نوع داده Nullable (که با یک علامت سوال قابل تشخیص است : `int? i = null`) اجازه تخصیص مقدار null را برای انواع داده ای می دهد. این امر باعث بهبودی فعل و انفعال با پایگاه داده SQL می شود. در این حالت نوع ستونی INTEGER NULL در SQL به طور مستقیم به `int?` در سی شارپ تبدیل می شود. داده های Nullable در آخرین لحظات آگوست ۲۰۰۵ اضافه شدند چند هفته متنده به اتمام کار اداری و برای بهبود زبان. متغیر Null در حقیقت خالی نیست، بلکه نمونه ای است از `struct Nullable<T>` با ویژگی HasValue مساوی false. وقتی در برنامه قرار می گیرد، خود به خود نمونه خالی در آن قرار می گیرد، نه

مقدار خود آن، در نتیجه اشاره گر مقصد همیشه غیر Null می باشد، حتی برای مقادیر Null. کد زیر نضص بالا را مشخص می کند:

```
int? i = null;
object o = i;
// دستور شرطی if (o == null)
    Console.WriteLine(«Correct behaviour - runtime version from September
2005 or later»);
else
    Console.WriteLine(«Incorrect behaviour - pre-release runtime (from
before September 2005)»);
```

وقتی درون شی ای کپی می شود، نمونه Nullable به صورت تشریفاتی در آن قرار می گیرد و در نتیجه مقادیر و منابع Null با هم برابر می شوند. در گذشته این خاصیت دارای مجادله بود تا زمانی که علاوه بر NET2، به هسته CLR نیز مجهز شد و همه تکنولوژی ها نظیر C# و VB و SQL Server 2008 و Visual Studio 2010 را شامل شد.

## کدهای مربوط به فرم ورود به سیستم

نمایه 5 - عکس فرم ورود به سیستم گل فروشی

```
namespace FlowerShop
{
    public partial class frmLogin : Form // علامت ارث بری از کلاس اصلی فرم
    {
        public frmLogin()
        {
            InitializeComponent(); // این متد در هنگام قبل از لود فرم فراخوانی شده و باعث می شود که اجزا و اشیاء
            استفاده شده بر روی فرم ظاهر و فعال شوند
        }

        private void frmLogin_Load(object sender, EventArgs e) // این متد در هنگام لود فرم اجرا می
        شود
        {
            // ثبت رخداد // Class_tblLogs LogReg = new Class_tblLogs(this.Text + " فعال
            شد.");
            LogReg.UDF_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می شود
        }

        private void frmLogin_FormClosed(object sender, FormClosedEventArgs e) // این متد
        در زمان بسته شدن فرم اجرا می شود
        {
            // ثبت رخداد // Class_tblLogs LogReg = new Class_tblLogs(this.Text + " بسته
            شد.");
            LogReg.UDF_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می شود
            Application.Exit(); // بسته شدن تمام برنامه
        }

        private void Btn_Login_Click(object sender, EventArgs e) // رخداد کلیک
        {
            Class_tblUsers LogInUsr = new Class_tblUsers(Txt_UserName.Text.Trim(),
            Txt_Password.Text.Trim());
            if (LogInUsr.UDF_LoginUser() == true) // بررسی اینکه آیا اطلاعات کاربری صحیح بوده یا خیر
            {
                Txt_UserName.Text = String.Empty; // حذف مقادیر متنی
                Txt_Password.Text = String.Empty; // حذف مقادیر متنی
                this.Hide();
                // ثبت رخداد // Class_tblLogs LogReg = new Class_tblLogs(this.Text + " به ورود
                شد.");
                LogReg.UDF_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می
                شود

                frmMainMdi frm = new frmMainMdi();
                frm.ShowDialog();
            }
            else
            {
                // نمایش پیغام // MessageBox.Show("نام کاربری یا کلمه عبور صحیح نیست");
            }
        }

        private void Btn_Cancel_Click(object sender, EventArgs e) // رخداد کلیک
        {
            // ...
        }
    }
}
```

```

        درخواست " + this.Text + " ثبت رخداد // Class_tblLogs LogReg = new Class_tblLogs(
    ); // شد لغو ورود
    LogReg.UDF_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می شود
    Application.Exit(); // بسته شدن تمام برنامه
}

private void Txt_UserName_KeyDown(object sender, KeyEventArgs e)
{
    بررسی اینکه فیلدهای نام
    کاربری و کلمه عبور خالی نباشند
    if (Txt_UserName.Text != "" && Txt_Password.Text != "")
    {
        بررسی فشرده شدن کلید اینتر //
        if (int.Parse(e.KeyValue.ToString()) == 13)
        {
            اجرای متد لاگین
            Btn_Login.PerformClick();
        }
    }
}

private void Txt_Password_KeyDown(object sender, KeyEventArgs e)
{
    بررسی اینکه فیلدهای نام
    کاربری و کلمه عبور خالی نباشند
    if (Txt_UserName.Text != "" && Txt_Password.Text != "")
    {
        بررسی فشرده شدن کلید اینتر //
        if (int.Parse(e.KeyValue.ToString()) == 13)
        {
            اجرای متد لاگین
            Btn_Login.PerformClick();
        }
    }
}
}
}
}

```

## کدهای مربوط به فرم اصلی سیستم گل فروشی



نمایه 6 - فرم اصلی سیستم گل فروشی

```
namespace FlowerShop
{
    public partial class frmMainMdi : Form // علامت ارث بری از کلاس اصلی فرم
    {
        frmFoodCategory frm2; // ایجاد شی از کلاس فرمهای برنامه
        frmFoodUnits frm3; // ایجاد شی از کلاس فرمهای برنامه
        frmDefineFoods frm4; // ایجاد شی از کلاس فرمهای برنامه
        frmDefineCustomer frm5; // ایجاد شی از کلاس فرمهای برنامه
        frmEditUsers frm6; // ایجاد شی از کلاس فرمهای برنامه
        frmFactor frm7; // ایجاد شی از کلاس فرمهای برنامه
        frmMojodi frm8; // ایجاد شی از کلاس فرمهای برنامه
        frmCalculator frm9; // ایجاد شی از کلاس فرمهای برنامه

        public frmMainMdi()
        {
            InitializeComponent(); // این متد در هنگام قبل از لود فرم فراخوانی شده و باعث می شود که اجزا و اشیاء استفاده شده بر روی فرم ظاهر و فعال شوند
        }

        private void frmMainMdi_Load(object sender, EventArgs e) // این متد در هنگام لود فرم اجرا می شود
        {
            فعال // ثبت رخداد // Class_tblLogs LogReg = new Class_tblLogs(this.Text + "
            گریدید");
            LogReg.UDF_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می شود
            toolStripStatusLabel.Text = " + : با شده وارد";
            Class_GlobalVar.Str_LoggedUserName.Trim();
        }

        private void frmMainMdi_FormClosed(object sender, FormClosedEventArgs e) // این متد در زمان بسته شدن فرم اجرا می شود
        {
            ثبت رخداد // Class_tblLogs LogReg = new Class_tblLogs(this.Text + "
            شد بسته");
        }
    }
}
```

```

        LogReg.UDF_SaveLogToDb()); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می شود
        Application.Exit(); // بسته شدن تمام برنامه
    }

    private void Mnu_File_Exit_Click(object sender, EventArgs e) // رخداد کلیک
    {
        this.Close(); // بسته شدن شی
    }

    private void Mnu_Reports_Logs_Click(object sender, EventArgs e) // رخداد کلیک
    {
        if (frm1 == null || frm1.IsDisposed) // بررسی اینکه آیا فرم قبلا باز شده یا خیر
        {
            frm1 = new frmRptLogs();
            frm1.MdiParent = this; // تخصیص فرم والد به فرم فرزند
            frm1.Show(); // نمایش شی
        }
        else
        {
            frm1.Activate(); // فعال سازی آبجکت
        }
    }

    private void Mnu_File_FoodCategu_Click(object sender, EventArgs e) // رخداد کلیک
    {
        if (frm2 == null || frm2.IsDisposed) // بررسی اینکه آیا فرم قبلا باز شده یا خیر
        {
            frm2 = new frmFoodCategory();
            frm2.MdiParent = this; // تخصیص فرم والد به فرم فرزند
            frm2.Show(); // نمایش شی
        }
        else
        {
            frm2.Activate(); // فعال سازی آبجکت
        }
    }

    private void Mnu_File_FoodUnit_Click(object sender, EventArgs e) // رخداد کلیک
    {
        if (frm3 == null || frm3.IsDisposed) // بررسی اینکه آیا فرم قبلا باز شده یا خیر
        {
            frm3 = new frmFoodUnits();
            frm3.MdiParent = this; // تخصیص فرم والد به فرم فرزند
            frm3.Show(); // نمایش شی
        }
        else
        {
            frm3.Activate(); // فعال سازی آبجکت
        }
    }

    private void Mnu_File_FoodDef_Click(object sender, EventArgs e) // رخداد کلیک
    {
        if (frm4 == null || frm4.IsDisposed) // بررسی اینکه آیا فرم قبلا باز شده یا خیر
        {
            frm4 = new frmDefineFoods();
            frm4.MdiParent = this; // تخصیص فرم والد به فرم فرزند
            frm4.Show(); // نمایش شی
        }
        else
        {
            frm4.Activate(); // فعال سازی آبجکت
        }
    }

```

```

    }
}

private void customerToolStripMenuItem_Click(object sender, EventArgs e) //
رخداد کلیک
{
    if (frm5 == null || frm5.IsDisposed) // بررسی اینکه آیا فرم قبلاً باز شده یا خیر
    {
        frm5 = new frmDefineCustomer();
        frm5.MdiParent = this; // تخصیص فرم والد به فرم فرزند
        frm5.Show(); // نمایش شئی
    }
    else
    {
        frm5.Activate(); // فعال سازی آبجکت
    }
}

private void mnuSecurity_DefineUsers_Click(object sender, EventArgs e) // رخداد
کلیک
{
    if (frm6 == null || frm6.IsDisposed) // بررسی اینکه آیا فرم قبلاً باز شده یا خیر
    {
        frm6 = new frmEditUsers();
        frm6.MdiParent = this; // تخصیص فرم والد به فرم فرزند
        frm6.Show(); // نمایش شئی
    }
    else
    {
        frm6.Activate(); // فعال سازی آبجکت
    }
}

private void Mnu_File_Factor_Click(object sender, EventArgs e) // رخداد کلیک
{
    if (frm7 == null || frm7.IsDisposed) // بررسی اینکه آیا فرم قبلاً باز شده یا خیر
    {
        frm7 = new frmFactor();
        frm7.MdiParent = this; // تخصیص فرم والد به فرم فرزند
        frm7.Show(); // نمایش شئی
    }
    else
    {
        frm7.Activate(); // فعال سازی آبجکت
    }
}

private void Mnu_File_InsertFood_Click(object sender, EventArgs e) // رخداد کلیک
{
    if (frm8 == null || frm8.IsDisposed) // بررسی اینکه آیا فرم قبلاً باز شده یا خیر
    {
        frm8 = new frmMojodi();
        frm8.MdiParent = this; // تخصیص فرم والد به فرم فرزند
        frm8.Show(); // نمایش شئی
    }
    else
    {
        frm8.Activate(); // فعال سازی آبجکت
    }
}

```



```

private void Mnu_Help_About_Click(object sender, EventArgs e) // رخداد کلیک
{
    // نمایش پیغام // MessageBox.Show("1 نسخه - گل فروشی برنامه");
}

private void Mnu_Help_Calculator_Click(object sender, EventArgs e) // رخداد کلیک
{
    if (frm9 == null || frm9.IsDisposed) // بررسی اینکه آیا فرم قبلا باز شده یا خیر
    {
        frm9 = new frmCalculator();
        frm9.MdiParent = this; // تخصیص فرم والد به فرم فرزند
        frm9.Show(); // نمایش شیء
    }
    else
    {
        frm9.Activate(); // فعال سازی آبجکت
    }
}

private void Mnu_Reports_PrintFactor_Click(object sender, EventArgs e) // رخداد
کلیک
{
    if (frm10 == null || frm10.IsDisposed) // بررسی اینکه آیا فرم قبلا باز شده یا خیر
    {
        frm10 = new frmRptFactorPrint();
        frm10.MdiParent = this; // تخصیص فرم والد به فرم فرزند
        frm10.Show(); // نمایش شیء
    }
    else
    {
        frm10.Activate(); // فعال سازی آبجکت
    }
}

private void Mnu_Reports_DailySale_Click(object sender, EventArgs e) // رخداد
کلیک
{
    if (frm11 == null || frm11.IsDisposed) // بررسی اینکه آیا فرم قبلا باز شده یا خیر
    {
        frm11 = new frmRptTotalSaleRate();
        frm11.MdiParent = this; // تخصیص فرم والد به فرم فرزند
        frm11.Show(); // نمایش شیء
    }
    else
    {
        frm11.Activate(); // فعال سازی آبجکت
    }
}
}
}

```

## کدهای مربوط به فرم ثبت دسته بندی گل

کد دسته بندی	عنوان دسته بندی گل
۷	گل قارچی
۶	گل ریشه پیازی
۵	گل دو ساله
۴	گل چند ساله
۳	گل ماشین عروس
۲	گل مصنوعی
۱	گل طبیعی

عنوان دسته بندی:

جستجو کردن   حذف کردن   ویرایش کردن   اضافه کردن

نمایه 7 - فرم دسته بندی گل

```
namespace FlowerShop
{
    public partial class frmFoodCategory : Form // علامت ارث بری از کلاس اصلی فرم
    {
        public frmFoodCategory()
        {
            InitializeComponent(); // این متد در هنگام قبل از لود فرم فراخوانی شده و باعث می شود که اجزا و اشیاء
            // استفاده شده بر روی فرم ظاهر و فعال شوند
        }

        private void btnAdd_Click(object sender, EventArgs e) // رخداد کلیک
        {
            Txt_AssetCategory.Text = String.Empty; // حذف مقادیر متنی
            btnAdd.SendToBack(); // نمایش کلید در پشت سایر آبجکتها
            Btn_Save.BringToFront(); // نمایش کلید در بالای سایر آبجکتها
            // نمایش پیغام
            MessageBox.Show("نمایید وارد را اطلاعات لطفا");
        }

        private void frmFoodCategory_Load(object sender, EventArgs e) // این متد در هنگام لود
        // فرم اجرا می شود
        {
            // ثبت رخداد
            Class_tblLogs LogReg = new Class_tblLogs(this.Text + " فعال
            ").گریدید";
            LogReg.UDF_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می شود
            UDF_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو
        }

        private void UDF_GrdFill() // توسط این متد اطلاعات فراخوانی شده در گرید ویو نمایش داده می شود
        {
            LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد شیء از
            // کلاس لینک جهت اتصال به پایگاه داده
            var Query = from Row in db.tblFoodCategories
                        orderby Row.FoodCategoryID descending
                        select new
                        {
                            Row.FoodCategoryID,

```

```

        Row.FoodCategoryDescription
    };
    dataGridView1.DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو
}

private void frmFoodCategory_FormClosed(object sender, FormClosedEventArgs e)
// این متد در زمان بسته شدن فرم اجرا می شود
{
    ثبت رخداد // Class_tblLogs LogReg = new Class_tblLogs(this.Text + " شد بسته.");
    LogReg.UDF_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می شود
}

private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)
// این متد در هنگام کلیک بر روی سطرهای گرید ویو انجام می شود و مقادیر مربوط به محتویات هر سلول را بر می گرداند
{
    Int32 selectedCellCount =
dataGridView1.GetCellCount(DataGridViewElementStates.Selected);

    if (selectedCellCount > 0)
    {
        if (dataGridView1.AreAllCellsSelected(true))
        {
            // مایش پیغام
Cells");
        }
        else
        {
            System.Text.StringBuilder sb = new System.Text.StringBuilder();
            for (int i = 0; i < selectedCellCount; i++)
            {
                Txt_CategoryID.Text =
dataGridView1.Rows[int.Parse(dataGridView1.SelectedCells[i].RowIndex.ToString())].Cells[0].Value.ToString();
                Txt_AssetCategory.Text =
dataGridView1.Rows[int.Parse(dataGridView1.SelectedCells[i].RowIndex.ToString())].Cells[1].Value.ToString();
            }
        }
    }
}

private void Btn_Save_Click(object sender, EventArgs e) // رخداد کلیک
{
    بررسی اینکه آیا فیلد متنی خالیست یا خیر
    {
        // مایش پیغام
        return; // خروج از متد
    }
    else
    {
        LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد
        شی از کلاس لینک جهت اتصال به پایگاه داده
        tblFoodCategory FoodCtg = new tblFoodCategory();
        FoodCtg.FoodCategoryDescription = Txt_AssetCategory.Text.Trim(); // دریافت مقدار از فیلد متنی
        db.tblFoodCategories.InsertOnSubmit(FoodCtg);
        db.SubmitChanges(); // ثبت تغییرات در پایگاه داده
        UDF_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو
        btnAdd.BringToFront(); // نمایش کلید در بالای سایر آبجکتها
        Btn_Save.SendToBack(); // نمایش کلید در پشت سایر آبجکتها
        // مایش پیغام
    }
}

```

```

        Txt_AssetCategory.Text = "";
        Txt_CategoryID.Text = "";
    }

    private void Btn_Edit_Click(object sender, EventArgs e) // رخداد کلیک
    {
        if (Txt_CategoryID.Text == "" || Txt_AssetCategory.Text == "") // بررسی اینکه
        آیا فیلد متنی خالیست یا خیر
        {
            MessageBox.Show("انتخاب را عنوان لطفا");
            return; // خروج از متد
        }

        DialogResult dr = MessageBox.Show("اطمینان رکورد ویرایش عمل به نسبت آیا",
        "ویرایش", "خطا", MessageBoxButtons.OKCancel, MessageBoxIcon.Warning);
        if (dr == DialogResult.OK)
        {
            LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد
            شی از کلاس لینک جهت اتصال به پایگاه داده
            tblFoodCategory FoodCtg = db.tblFoodCategories.Single(n =>
            n.FoodCategoryID == Decimal.Parse(Txt_CategoryID.Text.Trim()));
            FoodCtg.FoodCategoryDescription = Txt_AssetCategory.Text.Trim();//
            دریافت مقدار از فیلد متنی
            db.SubmitChanges(); // ثبت تغییرات در پایگاه داده
            Class_tblLogs LogReg = new Class_tblLogs(this.Text + " رکورد
            " + Txt_AssetCategory.Text.Trim() + " -ID= " + Txt_CategoryID.Text.Trim() + " ویرایش");
            LogReg.UDF_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می
            شود
            UDF_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو
        }
        Txt_AssetCategory.Text = "";
        Txt_CategoryID.Text = "";
    }

    private void Btn_Delete_Click(object sender, EventArgs e) // رخداد کلیک
    {
        if (Txt_CategoryID.Text == "" || Txt_AssetCategory.Text == "") // بررسی اینکه
        آیا فیلد متنی خالیست یا خیر
        {
            MessageBox.Show("انتخاب را عنوان لطفا");
            return; // خروج از متد
        }

        DialogResult dr = MessageBox.Show("اطمینان رکورد حذف عمل به نسبت آیا",
        "حذف", "خطا", MessageBoxButtons.OKCancel, MessageBoxIcon.Warning);
        if (dr == DialogResult.OK)
        {
            LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد
            شی از کلاس لینک جهت اتصال به پایگاه داده
            tblFoodCategory FoodCtg = db.tblFoodCategories.Single(n =>
            n.FoodCategoryID == Decimal.Parse(Txt_CategoryID.Text.Trim()));
            db.tblFoodCategories.DeleteOnSubmit(FoodCtg);
            db.SubmitChanges(); // ثبت تغییرات در پایگاه داده
            Class_tblLogs LogReg = new Class_tblLogs(this.Text + " رکورد
            " + Txt_AssetCategory.Text.Trim() + " -ID= " + Txt_CategoryID.Text.Trim() + " حذف");
            LogReg.UDF_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می
            شود
            UDF_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو
        }
        Txt_AssetCategory.Text = "";
    }

```

```

        Txt_CategoryID.Text = "";
    }

    private void Btn_Search_Click(object sender, EventArgs e) // رخداد کلیک
    {
        if (Txt_AssetCategory.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر
        {
            // نمایش پیغام
            MessageBox.Show("نمایشید وارد جستجو برای را عنوان از بخشی یا عنوان لطفا");
            return; // خروج از متد
        }
        LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد شی از
        // کلاس لینک جهت اتصال به پایگاه داده
        var Query = from Row in db.tblFoodCategories
                    where SqlMethods.Like(Row.FoodCategoryDescription, "%" +
        Txt_AssetCategory.Text.Trim() + "%")
                    select new
                    {
                        Row.FoodCategoryID,
                        Row.FoodCategoryDescription
                    };
        dataGridView1.DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو
        // عملیات - " + this.Text + " ثبت رخداد
        Class_tblLogs LogReg = new Class_tblLogs(this.Text + " جستجو");
        LogReg.UDF_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می شود
        if (Query.Count() <= 0)
        {
            Txt_AssetCategory.Text = "";
            Btn_Refresh.BringToFront(); // نمایش کلید در بالای سایر آبجکتها
            Btn_Search.SendToBack(); // نمایش کلید در پشت سایر آبجکتها
            // نمایش پیغام
            MessageBox.Show("نشده یافته رکوردی");
        }
    }

    private void Btn_Refresh_Click(object sender, EventArgs e) // رخداد کلیک
    {
        UDF_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو
        Btn_Refresh.SendToBack(); // نمایش کلید در پشت سایر آبجکتها
        Btn_Search.BringToFront(); // نمایش کلید در بالای سایر آبجکتها
    }
}

```

## کدهای مربوط به فرم ثبت واحد شمارش گل

کد واحد شمارش	شرح واحد شمارش
۵	بسته
۴	گلدان
۳	گروه
۲	دسته
۱	شاخه

واحد شمارش:

جستجو کردن    حذف کردن    ویرایش کردن    اضافه کردن

نمایه ۸ - فرم ثبت واحد شمارش گل

```
namespace FlowerShop
{
    public partial class frmFoodUnits : Form // علامت ارث بری از کلاس اصلی فرم
    {
        public frmFoodUnits()
        {
            InitializeComponent(); // این متد در هنگام قبل از لود فرم فراخوانی شده و باعث می شود که اجزا و اشیاء
            // استفاده شده بر روی فرم ظاهر و فعال شوند
        }

        private void frmFoodUnits_Load(object sender, EventArgs e) // این متد در هنگام لود فرم
        // اجرا می شود
        {
            فعال // ثبت رخداد // Class_tblLogs LogReg = new Class_tblLogs(this.Text + ".");
            LogReg.UDF_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می شود
            UDF_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو
        }

        private void frmFoodUnits_FormClosed(object sender, FormClosedEventArgs e) //
        // این متد در زمان بسته شدن فرم اجرا می شود
        {
            بسته // ثبت رخداد // Class_tblLogs LogReg = new Class_tblLogs(this.Text + ".");
            LogReg.UDF_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می شود
        }

        private void btnAdd_Click(object sender, EventArgs e) // رخداد کلیک
        {
            Txt_AssetUnit.Text = String.Empty; // حذف مقادیر متنی
            btnAdd.SendToBack(); // نمایش کلید در پشت سایر آبجکتها
            Btn_Save.BringToFront(); // نمایش کلید در بالای سایر آبجکتها
            // نمایش پیغام // MessageBox.Show("اطلاعات وارد را لطفاً");
        }

        private void UDF_GrdFill() // توسط این متد اطلاعات فراخوانی شده در گرید ویو نمایش داده می شود
        {

```

```

        LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد شی از
        کلاس لینک جهت اتصال به پایگاه داده
        var Query = from Row in db.tblFoodUnits
                     orderby Row.FoodUnitID descending
                     select new
                     {
                         Row.FoodUnitID,
                         Row.FoodUnitDescription
                     };
        dataGridView1.DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو
    }

    private void Btn_Save_Click(object sender, EventArgs e) // رخداد کلیک
    {
        if (Txt_AssetUnit.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر
        {
            مایش پیغام // MessageBox.Show("الزامی عنوان ورود");
            return; // خروج از متد
        }
        LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد شی از
        کلاس لینک جهت اتصال به پایگاه داده
        tblFoodUnit FoodUnit = new tblFoodUnit();
        FoodUnit.FoodUnitDescription = Txt_AssetUnit.Text.Trim(); // دریافت مقدار از فیلد متنی
        db.tblFoodUnits.InsertOnSubmit(FoodUnit);
        db.SubmitChanges(); // ثبت تغییرات در پایگاه داده
        UDF_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو
        Txt_UnitID.Text = "";
        Txt_AssetUnit.Text = "";
    }

    private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e) // این متد در هنگام کلیک بر روی سطرهای گرید ویو انجام می شود و مقادیر مربوط به محتویات هر سلول را بر می گرداند
    {
        Int32 selectedCellCount =
        dataGridView1.GetCellCount(DataGridViewElementStates.Selected);

        if (selectedCellCount > 0)
        {
            if (dataGridView1.AreAllCellsSelected(true))
            {
                مایش پیغام // MessageBox.Show("All cells are selected", "Selected
                Cells");
            }
            else
            {
                System.Text.StringBuilder sb = new System.Text.StringBuilder();
                for (int i = 0; i < selectedCellCount; i++)
                {
                    Txt_UnitID.Text =
                    dataGridView1.Rows[int.Parse(dataGridView1.SelectedCells[i].RowIndex.ToString())].Cells[0].Value.ToString();
                    Txt_AssetUnit.Text =
                    dataGridView1.Rows[int.Parse(dataGridView1.SelectedCells[i].RowIndex.ToString())].Cells[1].Value.ToString();
                }
            }
        }
    }

    private void Btn_Edit_Click(object sender, EventArgs e) // رخداد کلیک
    {

```

```

        if (Txt_UnitID.Text == "" || Txt_AssetUnit.Text == "") // بررسی اینکه آیا فیلد متنی
        خالیست یا خیر
        {
            // انتخاب را عنوان لطفا
            MessageBox.Show("انتخاب را عنوان لطفا");
            return; // خروج از متد
        }

        DialogResult dr = // مایش پیغام
        // مایش پیغام // MessageBoxButtons.OKCancel, "ویرایش", "خطا", "دارید
        MessageBoxIcon.Warning);
        if (dr == DialogResult.OK)
        {
            LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد
            شی از کلاس لینک جهت اتصال به پایگاه داده
            tblFoodUnit FoodUnit = db.tblFoodUnits.Single(n => n.FoodUnitID ==
            int.Parse(Txt_UnitID.Text.Trim()));
            FoodUnit.FoodUnitDescription = Txt_AssetUnit.Text.Trim(); // دریافت مقدار از
            فیلد متنی
            db.SubmitChanges(); // ثبت تغییرات در پایگاه داده
            ثبت رخداد // Class_tblLogs LogReg = new Class_tblLogs(this.Text + "
            رکورد " + Txt_AssetUnit.Text.Trim() + " -ID= " + Txt_UnitID.Text.Trim() + "
            ویرایش");
            LogReg.UDF_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می
            شود
            UDF_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو
            Txt_UnitID.Text = "";
            Txt_AssetUnit.Text = "";
        }
    }

    private void Btn_Delete_Click(object sender, EventArgs e) // رخداد کلیک
    {
        if (Txt_UnitID.Text == "" || Txt_AssetUnit.Text == "") // بررسی اینکه آیا فیلد متنی
        خالیست یا خیر
        {
            // انتخاب را عنوان لطفا
            MessageBox.Show("انتخاب را عنوان لطفا");
            return; // خروج از متد
        }

        DialogResult dr = // مایش پیغام
        // مایش پیغام // MessageBoxButtons.OKCancel, "حذف", "خطا", "دارید
        MessageBoxIcon.Warning);
        if (dr == DialogResult.OK)
        {
            ثبت رخداد // Class_tblLogs LogReg = new Class_tblLogs(this.Text + "
            رکورد " + Txt_AssetUnit.Text.Trim() + " -ID= " + Txt_UnitID.Text.Trim() + "
            حذف");
            LogReg.UDF_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می
            شود
            LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد
            شی از کلاس لینک جهت اتصال به پایگاه داده
            tblFoodUnit FoodUnit = db.tblFoodUnits.Single(n => n.FoodUnitID ==
            int.Parse(Txt_UnitID.Text.Trim()));
            db.tblFoodUnits.DeleteOnSubmit(FoodUnit);
            db.SubmitChanges(); // ثبت تغییرات در پایگاه داده
            UDF_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو
        }
        Txt_UnitID.Text = "";
        Txt_AssetUnit.Text = "";
    }

    private void Btn_Search_Click(object sender, EventArgs e) // رخداد کلیک
    {
        if (Txt_AssetUnit.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر
    
```



```

{
    مایش پیغام // MessageBox.Show("نمائید وارد جستجو برای را عنوان از بخشی یا عنوان لطفا");
    return; // خروج از متد
}
LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد شی از
کلاس لینک جهت اتصال به پایگاه داده
var Query = from Row in db.tblFoodUnits
              where SqlMethods.Like(Row.FoodUnitDescription, "%" +
Txt_AssetUnit.Text.Trim() + "%")
              select new
              {
                  Row.FoodUnitID,
                  Row.FoodUnitDescription
              };
dataGridView1.DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو
عملیات - " // Class_tblLogs LogReg = new Class_tblLogs(this.Text + "
ثبت رخداد
جستجو");
LogReg.UDF_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می شود
if (Query.Count() <= 0)
{
    Txt_AssetUnit.Text = "";
    Btn_Refresh.BringToFront(); // نمایش کلید در بالای سایر آبجکتها
    Btn_Search.SendToBack(); // نمایش کلید در پشت سایر آبجکتها
    مایش پیغام // MessageBox.Show("نشده یافته رکوردی");
}
}

private void Btn_Refresh_Click(object sender, EventArgs e) // رخداد کلیک
{
    فراخوانی متد نمایش رکوردها در گرید ویو // UDF_GrdFill();
    نمایش کلید در پشت سایر آبجکتها // Btn_Refresh.SendToBack();
    نمایش کلید در بالای سایر آبجکتها // Btn_Search.BringToFront();
}
}
}

```

## کدهای مربوط به فرم تعریف عنوان گل

فرم تعریف گل جدید

کد	عنوان	دسته بندی	واحد شمارش
۱۶	گل کاغذی	گل چند ساله	گروه
۱۵	بنجامین	گل چند ساله	گروه
۱۳	شب بو	گل مصنوعی	شاخه
۱۱	حسن یوسف	گل قارچی	دسته
۱۰	پتوس	گل قارچی	دسته
۹	مریم	گل قارچی	دسته
۸	شفایق	گل طبیعی	شاخه
۷	لاله	گل طبیعی	شاخه
۵	آزالیا	گل قارچی	دسته
۳	فیلندرون	گل ماشین عروس	شاخه
۲	اطلسی	گل مصنوعی	شاخه
۱	رز	گل طبیعی	شاخه

عنوان:

واحد شمارش:

دسته بندی:

جستجو کردن  حذف کردن  ویرایش کردن  اضافه کردن

نمایه ۹ - فرم ثبت عنوان گل

```
namespace FlowerShop
{
    public partial class frmDefineFoods : Form // علامت ارث بری از کلاس اصلی فرم
    {
        public frmDefineFoods()
        {
            InitializeComponent(); // این متد در هنگام قبل از لود فرم فراخوانی شده و باعث می شود که اجزا و اشیاء
            // استفاده شده بر روی فرم ظاهر و فعال شوند
        }

        private void frmDefineFoods_Load(object sender, EventArgs e) // این متد در هنگام لود فرم
        اجرا می شود
        {
            فعال // ثبت رخداد // Class_tblLogs LogReg = new Class_tblLogs(this.Text + "
            ".گردید");
            LogReg.UDF_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می شود
            UDF_CmbFill();
            UDF_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو
        }

        private void frmDefineFoods_FormClosed(object sender, FormClosedEventArgs e)
        // این متد در زمان بسته شدن فرم اجرا می شود
        {
            ".شد بسته" // ثبت رخداد // Class_tblLogs LogReg = new Class_tblLogs(this.Text + "
            با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می شود
            LogReg.UDF_SaveLogToDb());
        }

        private void UDF_CmbFill()
        {
            LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد شیء از
            // کلاس لینک جهت اتصال به پایگاه داده
            var Query1 = from Row in db.tblFoodUnits
                orderby Row.FoodUnitID descending
                select new
                {
                    Row.FoodUnitID,
                }
            }
        }
    }
}
```

```

        Row.FoodUnitDescription
    };
    var Query2 = from Row in db.tblFoodCategories
        orderby Row.FoodCategoryID descending
        select new
        {
            Row.FoodCategoryID,
            Row.FoodCategoryDescription
        };

    Cmb_FoodUnit.DisplayMember = "FoodUnitDescription";
    Cmb_FoodUnit.ValueMember = "FoodUnitID";

    Cmb_FoodCategory.DisplayMember = "FoodCategoryDescription";
    Cmb_FoodCategory.ValueMember = "FoodCategoryID";

    Cmb_FoodUnit.DataSource = Query1;
    Cmb_FoodCategory.DataSource = Query2;
}

private void UDF_GrdFill() // توسط این متد اطلاعات فراخوانی شده در گرید ویو نمایش داده می شود
{
    LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد شی از
    کلاس لینک جهت اتصال به پایگاه داده
    var Query = from Row in db.tblFoodsDescriptions
        orderby Row.FoodID descending
        select new
        {
            Row.FoodID,
            Row.FoodDescription,
            Row.tblFoodCategory.FoodCategoryDescription,
            Row.tblFoodUnit.FoodUnitDescription
        };
    dataGridView1.DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو
}

private void btnAdd_Click(object sender, EventArgs e) // رخداد کلیک
{
    Txt_FoodDescription.Text = String.Empty; // حذف مقادیر متنی
    btnAdd.SendToBack(); // نمایش کلید در پشت سایر آبجکتها
    Btn_Save.BringToFront(); // نمایش کلید در بالای سایر آبجکتها
    Txt_FoodDescription.Clear();
    MessageBox.Show("نمایید وارد را اطلاعات لطفا"); // مایش پیغام
}

private void Btn_Save_Click(object sender, EventArgs e) // رخداد کلیک
{
    if (Txt_FoodDescription.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر
    {
        MessageBox.Show("باید می الزامی عنوان ورود"); // مایش پیغام
        return; // خروج از متد
    }

    LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد شی از
    کلاس لینک جهت اتصال به پایگاه داده
    tblFoodsDescription FoodDs = new tblFoodsDescription();

    FoodDs.FoodDescription = Txt_FoodDescription.Text.Trim(); // دریافت مقدار از فیلد
    متن
    FoodDs.FoodCategoryID =
    Decimal.Parse(Cmb_FoodCategory.SelectedValue.ToString());
    FoodDs.FoodUnitID = int.Parse(Cmb_FoodUnit.SelectedValue.ToString());
}

```

```

db.tblFoodsDescriptions.InsertOnSubmit(FoodDs);
db.SubmitChanges(); // ثبت تغییرات در پایگاه داده

UDF_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو
UDF_CmbFill();
btnAdd.BringToFront(); // نمایش کلید در بالای سایر آبجکتها
Btn_Save.SendToBack(); // نمایش کلید در پشت سایر آبجکتها
Txt_FoodDescription.Clear();
مایش پیغام // MessageBox.Show("گرددید درج جدید رکورد");
}

private void Btn_Edit_Click(object sender, EventArgs e) // رخداد کلیک
{
    بررسی اینکه آیا // if (Txt_FoodDescription.Text == "" || Txt_FoodID.Text == "")
    فیلد متنی خالیست یا خیر
    {
        مایش پیغام // MessageBox.Show("فرمانید انتخاب را عنوان لطفا");
        خروج از متد // return;
    }
    if (Cmb_FoodUnit.Text == "" || Cmb_FoodCategory.Text == "")
    {
        مایش پیغام // MessageBox.Show("باید می الزامی شمارش واحد و بندی دسته انتخاب");
        خروج از متد // return;
    }

    DialogResult dr = مایش پیغام // MessageBox.Show("اطمینان رکورد ویرایش عمل به نسبت آیا")
    مایش پیغام // MessageBoxButtons.OKCancel, مایش پیغام "ویرایش اخطار", "دارید"
    MessageBoxIcon.Warning);
    if (dr == DialogResult.OK)
    {
        LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد
        شی از کلاس لینک جهت اتصال به پایگاه داده
        tblFoodsDescription FoodDs = db.tblFoodsDescriptions.Single(n =>
        n.FoodID == int.Parse(Txt_FoodID.Text.Trim()));

        دریافت مقدار از // FoodDs.FoodDescription = Txt_FoodDescription.Text.Trim();
        فیلد متنی

        FoodDs.FoodUnitID = int.Parse(Cmb_FoodUnit.SelectedValue.ToString());
        FoodDs.FoodCategoryID =
        Decimal.Parse(Cmb_FoodCategory.SelectedValue.ToString());

        db.SubmitChanges(); // ثبت تغییرات در پایگاه داده
        ثبت رخداد // Class_tblLogs LogReg = new Class_tblLogs(this.Text + " رکورد"
        " + Txt_FoodDescription.Text.Trim() + " -ID= " + Txt_FoodID.Text.Trim() + " ویرایش");
        LogReg.UDF_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می شود
        شود

        Txt_FoodID.Clear();
        Txt_FoodDescription.Clear();
        UDF_CmbFill();
        UDF_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو
    }
}

private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)
این متد در هنگام کلیک بر روی سطرهای گرید ویو انجام می شود و مقادیر مربوط به محتویات هر سلول را بر می گرداند
{
    Int32 selectedCellCount =
    dataGridView1.GetCellCount(DataGridViewElementStates.Selected);

    if (selectedCellCount > 0)

```

```

{
    if (dataGridView1.AreAllCellsSelected(true))
    {
        // مایش پیغام
        Cells");
    }
    else
    {
        System.Text.StringBuilder sb = new System.Text.StringBuilder();
        for (int i = 0; i < selectedCellCount; i++)
        {
            Txt_FoodID.Text =
dataGridView1.Rows[int.Parse(dataGridView1.SelectedCells[i].RowIndex.ToString())].Cells[0].Value.ToString();
            Txt_FoodDescription.Text =
dataGridView1.Rows[int.Parse(dataGridView1.SelectedCells[i].RowIndex.ToString())].Cells[1].Value.ToString();
        }
        UDF_SelectCmbValue(int.Parse(Txt_FoodID.Text.Trim()));
    }
}

private void UDF_SelectCmbValue(int int_FoodUD)
{
    LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد شی از
    // کلاس لینک جهت اتصال به پایگاه داده
    tblFoodsDescription FoodDs = db.tblFoodsDescriptions.Single(n => n.FoodID
== int_FoodUD);
    Cmb_FoodUnit.SelectedValue = FoodDs.FoodUnitID;
    Cmb_FoodCategory.SelectedValue = FoodDs.FoodCategoryID;
}

private void Btn_Delete_Click(object sender, EventArgs e) // رخداد کلیک
{
    if (Txt_FoodDescription.Text == "" || Txt_FoodID.Text == "") // بررسی اینکه آیا
    // فیلد متنی خالیست یا خیر
    {
        // مایش پیغام
        MessageBox.Show("فرمانید انتخاب را عنوان لطفا.");
        return; // خروج از متد
    }

    DialogResult dr = // مایش پیغام
    // مایش پیغام
    // مایش پیغام
    MessageBoxButtons.OKCancel,
    MessageBoxIcon.Warning);
    if (dr == DialogResult.OK)
    {
        LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد
        // شی از کلاس لینک جهت اتصال به پایگاه داده
        tblFoodsDescription FoodDs = db.tblFoodsDescriptions.Single(n =>
n.FoodID == int.Parse(Txt_FoodID.Text.Trim()));

        // ثبت رخداد
        Class_tblLogs LogReg = new Class_tblLogs(this.Text + " رکورد "
+ Txt_FoodDescription.Text.Trim() + " -ID= " + Txt_FoodID.Text.Trim() + " حذف");
        LogReg.UDF_SaveLogToDb(); // ثبت تغییرات در پایگاه داده می
        // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می
        // شود

        db.tblFoodsDescriptions.DeleteOnSubmit(FoodDs);
        db.SubmitChanges(); // ثبت تغییرات در پایگاه داده

        Txt_FoodID.Clear();
        Txt_FoodDescription.Clear();
    }
}

```

```

        UDF_CmbFill();
        UDF_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو
    }
}

private void Btn_Search_Click(object sender, EventArgs e) // رخداد کلیک
{
    if (Txt_FoodDescription.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر
    {
        // نمایش پیغام
        MessageBox.Show("نماینده وارد جستجو برای را عنوان از بخشی یا عنوان لطفا");
        return; // خروج از متد
    }
    LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد شیء از
    // کلاس لینک جهت اتصال به پایگاه داده
    var Query = from Row in db.tblFoodsDescriptions
        where SqlMethods.Like(Row.FoodDescription, "%" +
            Txt_FoodDescription.Text.Trim() + "%")
        select new
        {
            Row.FoodID,
            Row.FoodDescription,
            Row.tblFoodCategory.FoodCategoryDescription,
            Row.tblFoodUnit.FoodUnitDescription
        };
    dataGridView1.DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو
    // عملیات - ثبت رخداد
    Class_tblLogs LogReg = new Class_tblLogs(this.Text + " جستجو");
    LogReg.UDF_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به پایگاه داده ثبت می شود
    if (Query.Count() <= 0)
    {
        Txt_FoodDescription.Text = "";
        Txt_FoodID.Clear();
        Btn_Refresh.BringToFront(); // نمایش کلید در بالای سایر آجکتها
        Btn_Search.SendToBack(); // نمایش کلید در پشت سایر آجکتها
        // نمایش پیغام
        MessageBox.Show("نشده یافته رکوردی");
    }
}

private void Btn_Refresh_Click(object sender, EventArgs e) // رخداد کلیک
{
    UDF_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو
    UDF_CmbFill();
    Btn_Refresh.SendToBack(); // نمایش کلید در پشت سایر آجکتها
    Btn_Search.BringToFront(); // نمایش کلید در بالای سایر آجکتها
    Txt_FoodDescription.Clear();
    Txt_FoodID.Clear();
}
}
}

```

## کدهای مربوط به فرم ثبت موجودی گل

فرم موجودی

شماره ردیف	کد گل	نام گل	تاریخ ورود	تعداد گل	قیمت واحد
۴۴	۱۱	حسن یوسف	۱۳۹۸/۰۵/۲۳	۵	۱۰۰۰۰,۰۰۰
۴۳	۱۳	شب بو	۱۳۹۸/۰۵/۲۳	۴	۳۷۰۰۰,۰۰۰
۴۲	۱۶	گل کاغذی	۱۳۹۸/۰۵/۲۳	۳	۲۳۰۰۰,۰۰۰
۳۹	۸	شفایق	۱۳۹۸/۰۵/۲۱	۶	۱۴۷۰۰,۰۰۰
۳۸	۱۰	پتوس	۱۳۹۸/۰۵/۲۱	۲	۱۵۰۰,۰۰۰
۳۷	۵	آزالیا	۱۳۹۸/۰۵/۲۱	۱	۲۵۰۰,۰۰۰
۳۶	۱	رز	۱۳۹۸/۰۵/۲۱	۵	۱۲۰۰۰,۰۰۰

عنوان گل: گل کاغذی  
تعداد:  
قیمت واحد:

جستجو عنوان: جستجو کردن حذف کردن ویرایش کردن اضافه کردن

نمایه 10 - ثبت موجودی گل

```
namespace FlowerShop
{
    public partial class frmMojodi : Form // علامت ارث بری از کلاس اصلی فرم
    {
        public frmMojodi()
        {
            InitializeComponent(); // این متد در هنگام قبل از لود فرم فراخوانی شده و باعث می شود که اجزا و اشیاء
            // استفاده شده بر روی فرم ظاهر و فعال شوند
        }

        private void UDF_GrdFill() // توسط این متد اطلاعات فراخوانی شده در گرید ویو نمایش داده می شود
        {
            LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد شی از
            // کلاس لینک جهت اتصال به پایگاه داده
            var Query = from Row in db.tblMojodis
                        orderby Row.RowMojodi descending
                        select new
                        {
                            Row.RowMojodi,
                            Row.FoodID,
                            Row.tblFoodsDescription.FoodDescription,
                            Row.EnterDate,
                            Row.NumberOfFood
                        };
            Grd_Mojodi.DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو
        }

        private void UDF_CmbFill()
        {
            LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد شی از
            // کلاس لینک جهت اتصال به پایگاه داده
            var Query1 = from Row in db.tblFoodsDescriptions
                        orderby Row.FoodID descending
                        select new
                        {

```

```

        Row.FoodID,
        Row.FoodDescription
    };

    Cmb_FoodDes.DisplayMember = "FoodDescription";
    Cmb_FoodDes.ValueMember = "FoodID";
    Cmb_FoodDes.DataSource = Query1;
}

private void btnAdd_Click(object sender, EventArgs e) // رخداد کلیک
{
    Txt_FoodUnit.Text = String.Empty; // حذف مقادیر متنی
    btnAdd.SendToBack(); // نمایش کلید در پشت سایر آبجکتها
    Btn_Save.BringToFront(); // نمایش کلید در بالای سایر آبجکتها
    Txt_FoodUnit.Clear();
    MessageBox.Show("نمایید وارد را گل موجودی لطفا"); // نمایش پیغام
}

private void Btn_Save_Click(object sender, EventArgs e) // رخداد کلیک
{
    if (Txt_FoodUnit.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر
    {
        MessageBox.Show("باید می الزامی عنوان ورود"); // نمایش پیغام
        return; // خروج از متد
    }

    LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد شی از
    // اتصال به پایگاه داده
    tblMojodi mojodi = new tblMojodi();
    Class_ShamsiTools shamsi = new Class_ShamsiTools();
    shamsi.UDF_GetCurrentShamsiDate();

    mojodi.FoodID = int.Parse(Cmb_FoodDes.SelectedValue.ToString());
    mojodi.EnterDate = shamsi.CurrentShamsiDate;
    mojodi.NumberOfFood = float.Parse(Txt_FoodUnit.Text.Trim());

    db.tblMojodis.InsertOnSubmit(mojodi);
    db.SubmitChanges(); // ثبت تغییرات در پایگاه داده

    UDF_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو
    UDF_CmbFill();
    btnAdd.BringToFront(); // نمایش کلید در بالای سایر آبجکتها
    Btn_Save.SendToBack(); // نمایش کلید در پشت سایر آبجکتها
    Txt_FoodUnit.Clear();
    MessageBox.Show(".گردید درج جدید رکورد"); // نمایش پیغام
}

private void frmMojodi_Load(object sender, EventArgs e) // این متد در هنگام لود فرم اجرا
    می شود
{
    Class_tblLogs LogReg = new Class_tblLogs(this.Text + " فعال // ثبت رخداد
    ".گردید");

    LogReg.UDF_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می شود
    UDF_CmbFill();
    UDF_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو
}

private void Grd_Mojodi_CellClick(object sender, DataGridViewCellEventArgs
e) // این متد در هنگام کلیک بر روی سطرهای گرید ویو انجام می شود و مقادیر مربوط به محتویات هر سلول را بر می گرداند

```



```

{
    Int32 selectedCellCount = Grd_Mojodi.
GetCellCount(DataGridViewElementStates.Selected);

    if (selectedCellCount > 0)
    {
        if (Grd_Mojodi.AreAllCellsSelected(true))
        {
            مایش پیغام // MessageBox.Show("All cells are selected", "Selected
Cells");
        }
        else
        {
            System.Text.StringBuilder sb = new System.Text.StringBuilder();
            for (int i = 0; i < selectedCellCount; i++)
            {
                Txt_FoodID.Text =
Grd_Mojodi.Rows[int.Parse(Grd_Mojodi.SelectedCells[i].RowIndex.ToString())].Cells[1].V
alue.ToString();
                Txt_FoodUnit.Text =
Grd_Mojodi.Rows[int.Parse(Grd_Mojodi.SelectedCells[i].RowIndex.ToString())].Cells[4].V
alue.ToString();
                Cmb_FoodDes.SelectedValue =
int.Parse(Grd_Mojodi.Rows[int.Parse(Grd_Mojodi.SelectedCells[i].RowIndex.ToString())].
Cells[1].Value.ToString());
            }
        }
    }
}

private void Btn_Edit_Click(object sender, EventArgs e) // رخداد کلیک
{
    if (Txt_FoodID.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر
    {
        مایش پیغام // MessageBox.Show("فرمانید انتخاب را عنوان از بخشی یا عنوان لطفا");
        return; // خروج از متد
    }

    if (Cmb_FoodDes.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر
    {
        مایش پیغام // MessageBox.Show("فرمانید انتخاب را عنوان از بخشی یا عنوان لطفا");
        return; // خروج از متد
    }

    DialogResult dr = مایش پیغام // MessageBox.Show("اطمینان رکورد ویرایش عمل به نسبت آیا", "دارید
    مایش پیغام // MessageBoxButtons.OKCancel, مایش پیغام //
    MessageBoxIcon.Warning);
    if (dr == DialogResult.OK)
    {
        LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد
        شی از کلاس لینک جهت اتصال به پایگاه داده
        tblMojodi mogodi = new tblMojodi();
        Class_ShamsiTools shamsi = new Class_ShamsiTools();
        shamsi.UDF_GetCurrentShamsiDate();
        mogodi.FoodID = int.Parse(Cmb_FoodDes.SelectedValue.ToString());
        mogodi.EnterDate = shamsi.CurrentShmasiDate;
        mogodi.NumberOfFood = float.Parse(Txt_FoodUnit.Text.Trim());

        db.SubmitChanges(); // ثبت تغییرات در پایگاه داده
    }
}

```

```

        رکورد // Class_tblLogs LogReg = new Class_tblLogs(this.Text + "
" + Txt_FoodUnit.Text.Trim() + " -ID= " + Txt_FoodID.Text.Trim() + "ویرایش");
        LogReg.UDF_SaveLogToDb(); // با استفاده از این متد رکداد ارسال شده به کلاس در پایگاه داده ثبت می
        شود

        Txt_FoodID.Clear();
        Txt_FoodUnit.Clear();
        UDF_CmbFill();
        UDF_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو
    }
}

private void Btn_Delete_Click(object sender, EventArgs e) // رکداد کلیک
{
    بررسی اینکه آیا فیلد متنی خالیست یا خیر
    {
        مایش پیام // MessageBox.Show("فرمانید انتخاب را عنوان لطفا");
        خروج از متد // return;
    }

    DialogResult dr = مایش پیام // MessageBox.Show("اطمینان رکورد حذف عمل به نسبت آیا");
    مایش پیام // MessageBoxButtons.OKCancel, مایش پیام, "حذف اخطار", "دارید";
    MessageBoxIcon.Warning);
    if (dr == DialogResult.OK)
    {
        ایجاد LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); //
        شی از کلاس لینک جهت اتصال به پایگاه داده
        tblFoodsDescription FoodDs = db.tblFoodsDescriptions.Single(n =>
        n.FoodID == int.Parse(Txt_FoodID.Text.Trim()));

        رکورد // Class_tblLogs LogReg = new Class_tblLogs(this.Text + "
" + Cmb_FoodDes.SelectedItem.ToString() + " -ID= " + Txt_FoodID.Text.Trim() + "حذف");
        LogReg.UDF_SaveLogToDb(); // با استفاده از این متد رکداد ارسال شده به کلاس در پایگاه داده ثبت می
        شود

        db.tblFoodsDescriptions.DeleteOnSubmit(FoodDs);
        db.SubmitChanges(); // ثبت تغییرات در پایگاه داده

        Txt_FoodID.Clear();
        Txt_FoodUnit.Clear();
        UDF_CmbFill();
        UDF_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو
    }
}

private void Btn_Search_Click(object sender, EventArgs e) // رکداد کلیک
{
    بررسی اینکه آیا فیلد متنی خالیست یا خیر
    {
        مایش پیام // MessageBox.Show("نمائید وارد جستجو برای را عنوان از بخشی یا عنوان لطفا");
        خروج از متد // return;
    }

    ایجاد شی از LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); //
    کلاس لینک جهت اتصال به پایگاه داده
    var Query = from Row in db.tblMojodis
        where SqlMethods.Like(Row.tblFoodsDescription.FoodDescription
        , "%" + Txt_Search.Text.Trim() + "%")
        select new
        {
            Row.RowMojodi,
            Row.FoodID,
            Row.tblFoodsDescription.FoodDescription,

```

```

        Row.EnterDate,
        Row.NumberOfFood
    };
    Grd_Mojodi.DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو
    عملیات - " // Class_tblLogs LogReg = new Class_tblLogs(this.Text + " ثبت رخداد
    جستجو");
    LogReg.UDF_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می شود
    if (Query.Count() <= 0)
    {
        Txt_FoodUnit.Text = "";
        Txt_FoodID.Clear();
        Btn_Refresh.BringToFront(); // نمایش کلید در بالای سایر آجکتها
        Btn_Search.SendToBack(); // نمایش کلید در پشت سایر آجکتها
        مایش پیغام // MessageBox.Show("نشده رکوردی");
    }
}

private void Btn_Refresh_Click(object sender, EventArgs e) // رخداد کلیک
{
    UDF_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو
    UDF_CmbFill();
    Btn_Refresh.SendToBack(); // نمایش کلید در پشت سایر آجکتها
    Btn_Search.BringToFront(); // نمایش کلید در بالای سایر آجکتها
    Txt_FoodUnit.Clear();
    Txt_FoodID.Clear();
}
}
}

```

## کدهای مربوط به فرم ثبت مشخصات مشتری

کد مشتری	نام مشتری	نام خانوادگی	تاریخ عضویت	تلفن	تلفن همراه	آدرس
۱۰۰۰۷	مهرداد	شهرامی	۱۳۹۸/۰۲/۲۳	۲۴۳	۲۲۴	ایران درب دوم
۱۰۰۰۶	شهرام	عبدلی	۱۳۹۸/۰۲/۰۴	۱۲۳	۰۹۳۸	ایران - اصفهان
۱۰۰۰۰	حسین	حسینی	۱۳۹۷/۱۲/۰۹	۱۲۳	۰۹۳۶	ایران....

فرم تعریف مشتری

نام:

نام خانوادگی:

تلفن:

تلفن همراه:

آدرس:

وضعیت فعال: ☐

جستجو:

نام خانوادگی ☐ نام ☐ کد مشتری ☐

افزودن  ویرایش  حذف  جستجو

نمایه 11 - فرم ثبت مشخصات مشتری

```
namespace FlowerShop
{
    public partial class frmDefineCustomer : Form // علامت ارث بری از کلاس اصلی فرم
    {
        public frmDefineCustomer()
        {
            InitializeComponent(); // این متد در هنگام قبل از لود فرم فراخوانی شده و باعث می شود که اجزا و اشیاء
            // استفاده شده بر روی فرم ظاهر و فعال شوند
        }
        private void UDF_TextBoxEmpty()
        {
            Txt_CustomerID.Clear();
            Txt_Name.Clear();
            Txt_Family.Clear();
            Txt_Tell.Clear();
            Txt_Mobail.Clear();
            Txt_Address.Clear();
            Chb_IsActive.Checked = false;
        }

        private void UDF_GrdFill() // توسط این متد اطلاعات فراخوانی شده در گرید ویو نمایش داده می شود
        {
            LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد شیء از
            // کلاس لینک جهت اتصال به پایگاه داده
            var Query = from Row in db.tblCustomers
                        orderby Row.CustomerID descending
                        select new
                        {
                            Row.CustomerID,
                            Row.CustomerName,
                            Row.CustomerFamily,

```

```

        Row.DateOzviatSH,
        Row.CustomerTell,
        Row.CustomerMobail,
        Row.CustomerAddress,
        Row.CustomerIsActive
    };
    Grd_Customer.DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو
}

private void Btn_DeleteCustomer_Click(object sender, EventArgs e) // رخداد کلیک
{
    if (Txt_CustomerID.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر
    {
        // مایش پیغام // MessageBox.Show("فرمائید انتخاب را نظر مورد مشتری لطفا");
        return; // خروج از متد
    }

    DialogResult dr = // مایش پیغام // MessageBox.Show("اطمینان رکورد حذف عمل به نسبت آیا",
    // مایش پیغام // MessageBoxButtons.OKCancel, // مایش پیغام //
    MessageBoxIcon.Warning);
    if (dr == DialogResult.OK)
    {
        LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد
        شی از کلاس لینک جهت اتصال به پایگاه داده

        tblCustomer Customer = db.tblCustomers.Single(n => n.CustomerID ==
        decimal.Parse(Txt_CustomerID.Text.Trim()));

        // ثبت رخداد // Class_tblLogs LogReg = new Class_tblLogs(this.Text + " رکورد
        " + Txt_CustomerID.Text.Trim() + "-ID" + Txt_CustomerID.Text.Trim() + "حذف");
        LogReg.UDF_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می
        شود

        db.tblCustomers.DeleteOnSubmit(Customer);
        db.SubmitChanges(); // ثبت تغییرات در پایگاه داده

        UDF_TextBoxEmpty();
        UDF_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو
    }
}

private void frmDefineCustomer_Load(object sender, EventArgs e) // این متد در هنگام
لود فرم اجرا می شود
{
    // ثبت رخداد // Class_tblLogs LogReg = new Class_tblLogs(this.Text + " فعال
    ".گردید");
    LogReg.UDF_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می شود
    UDF_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو
}

private void btnAdd_Click(object sender, EventArgs e) // رخداد کلیک
{
    Txt_Name.Text = String.Empty; // حذف مقادیر متنی
    Txt_Family.Text = String.Empty; // حذف مقادیر متنی
    Txt_Tell.Text = String.Empty; // حذف مقادیر متنی
    Txt_Mobail.Text = String.Empty; // حذف مقادیر متنی
    Txt_Address.Text = String.Empty; // حذف مقادیر متنی
    btnAdd.SendToBack(); // نمایش کلید در پشت سایر آبجکتها
    Btn_Save.BringToFront(); // نمایش کلید در بالای سایر آبجکتها
    Txt_Family.Clear();
}

```

```

        // نمایش پیغام
    }

    private void Btn_Save_Click(object sender, EventArgs e) // رخداد کلیک
    {
        if (Txt_Name.Text == "" || Txt_Family.Text == "" || Txt_Tell.Text == "" ||
            Txt_Mobail.Text == "" || Txt_Address.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر
        {
            // نمایش پیغام
            return; // خروج از متد
        }

        LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد شیء از
        // کلاس لینک جهت اتصال به پایگاه داده

        tblCustomer Customer = new tblCustomer();
        Class_ShamsiTools shamsi = new Class_ShamsiTools();
        shamsi.UDF_GetCurrentShamsiDate();

        Customer.CustomerName = Txt_Name.Text.Trim(); // دریافت مقدار از فیلد متنی
        Customer.CustomerFamily = Txt_Family.Text.Trim(); // دریافت مقدار از فیلد متنی
        Customer.DateOzviatSH = shamsi.CurrentShmasiDate;
        Customer.CustomerTell = Txt_Tell.Text.Trim(); // دریافت مقدار از فیلد متنی
        Customer.CustomerMobail = Txt_Mobail.Text.Trim(); // دریافت مقدار از فیلد متنی
        Customer.CustomerAddress = Txt_Address.Text.Trim(); // دریافت مقدار از فیلد متنی
        Customer.CustomerIsActive = Chb_IsActive.Checked;

        db.tblCustomers.InsertOnSubmit(Customer);
        db.SubmitChanges(); // ثبت تغییرات در پایگاه داده

        UDF_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو

        btnAdd.BringToFront(); // نمایش کلید در بالای سایر آبجکتها
        Btn_Save.SendToBack(); // نمایش کلید در پشت سایر آبجکتها

        UDF_TextBoxEmpty();

        // نمایش پیغام
    }

    private void Grd_Customer_CellClick(object sender, DataGridViewCellEventArgs e) // این متد در هنگام کلیک بر روی سطرهای گرید ویو انجام می شود و مقادیر مربوط به محتویات هر سلول را بر می گرداند
    {
        Int32 selectedCellCount = Grd_Customer.
        GetCellCount(DataGridViewElementStates.Selected);

        if (selectedCellCount > 0)
        {
            if (Grd_Customer.AreAllCellsSelected(true))
            {
                // نمایش پیغام
                Cells");
            }
            else
            {
                System.Text.StringBuilder sb = new System.Text.StringBuilder();
                for (int i = 0; i < selectedCellCount; i++)
                {
                    Boolean Bln_Check;

```

```

        Txt_CustomerID.Text =
Grd_Customer.Rows[int.Parse(Grd_Customer.SelectedCells[i].RowIndex.ToString())].Cells[
0].Value.ToString();
        Txt_Name.Text =
Grd_Customer.Rows[int.Parse(Grd_Customer.SelectedCells[i].RowIndex.ToString())].Cells[
1].Value.ToString();
        Txt_Family.Text =
Grd_Customer.Rows[int.Parse(Grd_Customer.SelectedCells[i].RowIndex.ToString())].Cells[
2].Value.ToString();
        Txt_Tell.Text =
Grd_Customer.Rows[int.Parse(Grd_Customer.SelectedCells[i].RowIndex.ToString())].Cells[
4].Value.ToString();
        Txt_Mobail.Text =
Grd_Customer.Rows[int.Parse(Grd_Customer.SelectedCells[i].RowIndex.ToString())].Cells[
5].Value.ToString();
        Txt_Address.Text =
Grd_Customer.Rows[int.Parse(Grd_Customer.SelectedCells[i].RowIndex.ToString())].Cells[
6].Value.ToString();
        Bln_Check =
Boolean.Parse(Grd_Customer.Rows[int.Parse(Grd_Customer.SelectedCells[i].RowIndex.ToStr
ing())].Cells[7].Value.ToString());
        if (Bln_Check == true)
        {
            Chb_IsActive.Checked = true;
        }
        else if (Bln_Check == false)
        {
            Chb_IsActive.Checked = false;
        }
    }
}
}

private void Btn_Edit_Click(object sender, EventArgs e) // رخداد کلیک
{
    if (Txt_CustomerID.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر
    {
        مایش پیغام // MessageBox.Show("لطفاً مورد مشتری را انتخاب کنید");
        return; // خروج از متد
    }

    DialogResult dr = مایش پیغام // MessageBox.Show("اطمینان رکورد ویرایش عمل به نسبت آیا")
    مایش پیغام // MessageBoxButtons.OKCancel, مایش پیغام // MessageBoxIcon.Warning);
    if (dr == DialogResult.OK)
    {
        LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد
        شی از کلاس لینک جهت اتصال به پایگاه داده
        tblCustomer Customer = db.tblCustomers.Single(n => n.CustomerID ==
Decimal.Parse(Txt_CustomerID.Text.Trim()));

        Customer.CustomerName = Txt_Name.Text.Trim(); // دریافت مقدار از فیلد متنی
        Customer.CustomerFamily = Txt_Family.Text.Trim(); // دریافت مقدار از فیلد متنی
        Customer.CustomerTell = Txt_Tell.Text.Trim(); // دریافت مقدار از فیلد متنی
        Customer.CustomerMobail = Txt_Mobail.Text.Trim(); // دریافت مقدار از فیلد متنی
        Customer.CustomerAddress = Txt_Address.Text.Trim(); // دریافت مقدار از فیلد متنی
        Customer.CustomerIsActive = Chb_IsActive.Checked;
        db.SubmitChanges(); // ثبت تغییرات در پایگاه داده
    }
}

```

```

        رکورد // Class_tblLogs LogReg = new Class_tblLogs(this.Text + "
" + Txt_Name.Text.Trim() + Txt_Family.Text.Trim() + " -ID= " +
Txt_CustomerID.Text.Trim() + "ویرایش");
        LogReg.UDF_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می
        شود
        UDF_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو
        UDF_TextBoxEmpty();
    }
}

private void Btn_SearchCustomer_Click(object sender, EventArgs e) // رخداد کلیک
{
    if (Txt_Search.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر
    {
        مایش پیغام // MessageBox.Show("لطفاً جستجو فیلد را پر کنید");
        return; // خروج از متد
    }

    if (Rdo_CustomerID.Checked == true)
    {
        LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد
        شیء از کلاس لینک جهت اتصال به پایگاه داده
        var Query = from Row in db.tblCustomers
                     where Row.CustomerID ==
Decimal.Parse(Txt_Search.Text.Trim())
                     select new
                     {
                         Row.CustomerID,
                         Row.CustomerName,
                         Row.CustomerFamily,
                         Row.DateOzviatSH,
                         Row.CustomerTell,
                         Row.CustomerMobail,
                         Row.CustomerAddress,
                         Row.CustomerIsActive
                     };
        Grd_Customer.DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو
        رکورد // Class_tblLogs LogReg = new Class_tblLogs(this.Text + " -
        عملیات");
        LogReg.UDF_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می
        شود
        if (Query.Count() <= 0)
        {
            Txt_Search.Text = "";
            button3.BringToFront(); // نمایش کلید در بالای سایر آبجکتها
            Btn_SearchCustomer.SendToBack(); // نمایش کلید در پشت سایر آبجکتها
            مایش پیغام // MessageBox.Show("رکوردی یافت نشد");
        }
        else if (Rdo_CustomerName.Checked == true)
        {
            LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد
            شیء از کلاس لینک جهت اتصال به پایگاه داده
            var Query = from Row in db.tblCustomers
                         where SqlMethods.Like(Row.CustomerName , "%"
+Txt_Search.Text.Trim() + "%")
                         select new
                         {
                             Row.CustomerID,
                             Row.CustomerName,
                             Row.CustomerFamily,

```



```

        Row.DateOzviatSH,
        Row.CustomerTell,
        Row.CustomerMobail,
        Row.CustomerAddress,
        Row.CustomerIsActive
    };
    Grd_Customer.DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو
    // ثبت رخداد // Class_tblLogs LogReg = new Class_tblLogs(this.Text + " -
    LogReg.UDF_SaveLogToDb()); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می
    شود
    if (Query.Count() <= 0)
    {
        Txt_Search.Text = "";
        button3.BringToFront(); // نمایش کلید در بالای سایر آبجکتها
        Btn_SearchCustomer.SendToBack(); // نمایش کلید در پشت سایر آبجکتها
        // نمایش پیغام // MessageBox.Show("نشد یافته رکوردی");
    }
    else if (Rdo_CustomerFamily.Checked == true)
    {
        LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد
        شی از کلاس لینک جهت اتصال به پایگاه داده
        var Query = from Row in db.tblCustomers
                     where SqlMethods.Like(Row.CustomerFamily, "%" +
        Txt_Search.Text.Trim() + "%")
                     select new
                     {
                         Row.CustomerID,
                         Row.CustomerName,
                         Row.CustomerFamily,
                         Row.DateOzviatSH,
                         Row.CustomerTell,
                         Row.CustomerMobail,
                         Row.CustomerAddress,
                         Row.CustomerIsActive
                     };
        Grd_Customer.DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو
        // ثبت رخداد // Class_tblLogs LogReg = new Class_tblLogs(this.Text + " -
        LogReg.UDF_SaveLogToDb()); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می
        شود
        if (Query.Count() <= 0)
        {
            Txt_Search.Text = "";
            button3.BringToFront(); // نمایش کلید در بالای سایر آبجکتها
            Btn_SearchCustomer.SendToBack(); // نمایش کلید در پشت سایر آبجکتها
            // نمایش پیغام // MessageBox.Show("نشد یافته رکوردی");
        }
    }
}

private void button3_Click(object sender, EventArgs e) // رخداد کلیک
{
    UDF_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو
    button3.SendToBack(); // نمایش کلید در پشت سایر آبجکتها
    Btn_SearchCustomer.BringToFront(); // نمایش کلید در بالای سایر آبجکتها
}
}
}

```

## کدهای مربوط به فرم ثبت فاکتور فروش

نمایه 12 - فرم ایجاد فاکتور فروش

```
namespace FlowerShop
{
    public partial class frmFactor : Form // علامت ارث بری از کلاس اصلی فرم
    {
        frmSearchCustomer frm001;
        DataTable FactDt = new DataTable();

        public frmFactor()
        {
            InitializeComponent(); // این متد در هنگام قبل از لود فرم فراخوانی شده و باعث می شود که اجزا و اشیاء
            // استفاده شده بر روی فرم ظاهر و فعال شوند
        }

        // اول گریدویو در گل های گروه مقادیر درج
        private void UDF_GrdCategoryFill()
        {
            LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد شیء از
            // کلاس لینک جهت اتصال به پایگاه داده
            var Query = from Row in db.tblFoodCategories
                        orderby Row.FoodCategoryID ascending
                        select new
                        {
                            Row.FoodCategoryID,
                            Row.FoodCategoryDescription
                        };
            Grd_FoodCategory.DataSource = Query; // در گرید ویو
        }

        // دوم ویوی گرید در کالا گروه از استفاده با گروه هر در گل مقادیر درج
        private void UDF_CatID(Decimal Dec_CategoryID)
        {
            LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد شیء از
            // کلاس لینک جهت اتصال به پایگاه داده
        }
    }
}
```

```

var Query = from Row in db.tblFoodsDescriptions
            where Row.FoodCategoryID == Dec_CategoryID
            select new
            {
                Row.FoodID,
                Row.FoodDescription
            };
DataTable dt = new DataTable();
dt.Columns.Add("FoodID", typeof(int));
dt.Columns.Add("FoodDescription", typeof(String));
dt.Columns.Add("FoodStock", typeof(float));

foreach (var Row in Query)
{
    dt.Rows.Add(new Object[] { Row.FoodID, Row.FoodDescription,
UDF_GetStackRate(Row.FoodID) });
}

Grd_FoodDescription.DataSource = dt;
}

/// <summary>
/// گل موجودی دریافت جهت
/// </summary>
/// <param name="int_FoodID"></param>
/// <returns name="int_Stock"></returns>
private float UDF_GetStackRate(int int_FoodID)
{
    float flt_Stock = 0;
    LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد شی از
    کلاس لینک جهت اتصال به پایگاه داده
    try
    {
        xtblMandehGaza tblStock = db.xtblMandehGazas.Single(n => n.FoodsID ==
int_FoodID);
        flt_Stock = float.Parse(tblStock.MandehGaza.ToString());
    }
    catch
    {
        flt_Stock = 0;
    }
    return flt_Stock;
}

private void Grd_FoodCategory_CellClick(object sender,
DataGridViewCellEventArgs e)// این متد در هنگام کلیک بر روی سطرهای گزیده ویرو انجام می شود و مقادیر مربوط به محتویات
هر سلول را بر می گرداند
{
    UDF_TXTFactorClear();
    Int32 selectedCellCount =
Grd_FoodCategory.GetCellCount(DataGridViewElementStates.Selected);

    if (selectedCellCount > 0)
    {
        if (Grd_FoodCategory.AreAllCellsSelected(true))
        {
            // مایش پیغام
Cells");
        }
        else
        {
            System.Text.StringBuilder sb = new System.Text.StringBuilder();

```

```

        for (int i = 0; i < selectedCellCount; i++)
        {
            Decimal Dec_CategoryID =
            Decimal.Parse(Grd_FoodCategory.Rows[int.Parse(Grd_FoodCategory.SelectedCells[i].RowIndex.ToString())].Cells[0].Value.ToString());
            UDF_CatID(Dec_CategoryID);
        }
    }
}

این متد در هنگام لود فرم اجرا می شود
private void frmFactor_Load(object sender, EventArgs e) //
{
    فعال // Class_tblLogs LogReg = new Class_tblLogs(this.Text + "
    ثبت رخداد
    گردد");
    LogReg.UDF_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می شود
    UDF_GrdCategoryFill();
}

//private void PassData(object sender, int number)
//{
//    //Sms_Menu.Enabled = (Boolean)sender;
//    // مایش پیغام // MessageBox.Show(number.ToString());
//}

این متد در هنگام کلیک بر روی سطرهای گزیده و یو انجام می شود و مقادیر مربوط به محتویات
private void Grd_FoodDescription_CellClick(object sender,
DataGridViewCellEventArgs e)
{
    Int32 selectedCellCount =
    Grd_FoodDescription.GetCellCount(DataGridViewElementStates.Selected);
    UDF_TXTFactorClear();
    if (selectedCellCount > 0)
    {
        if (Grd_FoodDescription.AreAllCellsSelected(true))
        {
            مایش پیغام // MessageBox.Show("All cells are selected", "Selected
            Cells");
        }
        else
        {
            System.Text.StringBuilder sb = new System.Text.StringBuilder();
            for (int i = 0; i < selectedCellCount; i++)
            {
                Txt_FoodCode.Text =
                Grd_FoodDescription.Rows[int.Parse(Grd_FoodDescription.SelectedCells[i].RowIndex.ToString())].Cells[0].Value.ToString();
                Txt_FoodName.Text =
                Grd_FoodDescription.Rows[int.Parse(Grd_FoodDescription.SelectedCells[i].RowIndex.ToString())].Cells[1].Value.ToString();
                Txt_Stock.Text =
                Grd_FoodDescription.Rows[int.Parse(Grd_FoodDescription.SelectedCells[i].RowIndex.ToString())].Cells[2].Value.ToString();
                Txt_TedadeDarkhast.Text = Txt_FeeTotal.Text =
                Txt_UnitPrice.Text = Txt_Tajhfif.Text = "0";
            }
        }
    }
}

```

```

private Decimal UDF_GetFoodINFO(int int_FoodID)
{
    Decimal Dec_FoodPrice = 0;
    LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد شی از
    کلاس لینک جهت اتصال به پایگاه داده

    return Dec_FoodPrice;
}

private void UDF_TXTFactorClear()
{
    Txt_FoodCode.Text = "";
    Txt_FoodName.Text = "";
    Txt_Stock.Text = "";
    Txt_TedadeDarkhast.Text = "";
    Txt_UnitPrice.Text = "";
    Txt_Tajhfif.Text = "";
    Txt_FeeTotal.Text = "";
}

private void Txt_TedadeDarkhast_KeyPress(object sender, KeyPressEventArgs e)
{
    int ascii = Convert.ToInt16(e.KeyChar);
    if (ascii >= 48 && ascii <= 57)
    {
        e.Handled = false;
    }
    else
    {
        e.Handled = true;
    }
}

private void Txt_Tajhfif_KeyPress(object sender, KeyPressEventArgs e)
{
    int ascii = Convert.ToInt16(e.KeyChar);
    if (ascii >= 48 && ascii <= 57)
    {
        e.Handled = false;
    }
    else
    {
        e.Handled = true;
    }
}

private void Txt_UnitPrice_KeyPress(object sender, KeyPressEventArgs e)
{
    int ascii = Convert.ToInt16(e.KeyChar);
    if (ascii >= 48 && ascii <= 57)
    {
        e.Handled = false;
    }
    else
    {
        e.Handled = true;
    }
}

private void Btn_AddToList_Click(object sender, EventArgs e) // رخداد کلیک
{
    if (Txt_FoodName.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر

```

```

{
    مایش پیغام // MessageBox.Show("است نشده انتخاب گل عنوان");
    return; // خروج از متد
}
if (Decimal.Parse(Txt_UnitPrice.Text.Trim()) == 0 ||
Decimal.Parse(Txt_TedadeDarkhast.Text.Trim()) == 0)
{
    مایش پیغام // MessageBox.Show("باشد می الزامی واحد قیمت , تعداد مقادیر ورود");
    return; // خروج از متد
}
if (Decimal.Parse(Txt_TedadeDarkhast.Text.Trim()) >
Decimal.Parse(Txt_Stock.Text.Trim()))
{
    مایش پیغام // MessageBox.Show("باشد می موجودی مقدار از بیش شده انتخاب مقدار");
    return; // خروج از متد
}

if (FactDt.Rows.Count <= 0 && FactDt.Columns.Count <= 0)
{
    FactDt.Columns.Add("FoodID", typeof(int));
    FactDt.Columns.Add("FoodName", typeof(string));
    FactDt.Columns.Add("FoodQuantity", typeof(float));
    FactDt.Columns.Add("FoodUnitPrice", typeof(Decimal));
    FactDt.Columns.Add("FoodDiscount", typeof(Decimal));
    FactDt.Columns.Add("FoodTotalPrice", typeof(Decimal));

    FactDt.Rows.Add(int.Parse(Txt_FoodCode.Text.Trim()),
    Txt_FoodName.Text.Trim(), float.Parse(Txt_TedadeDarkhast.Text.Trim())
    , Decimal.Parse(Txt_UnitPrice.Text.Trim()) ,
    Decimal.Parse(Txt_Tajhfif.Text.Trim()), Decimal.Parse(Txt_FeeTotal.Text.Trim()));

    Grd_Factor.DataSource = FactDt;
}
else
{
    int int_CntFound = 0;
    float flt_darkhast = 0;
    foreach (DataRow Row in FactDt.Rows)
    {
        if (int.Parse(Row[0].ToString().Trim()) ==
int.Parse(Txt_FoodCode.Text.Trim()))
        {
            int_CntFound++;
            flt_darkhast = float.Parse(Row[2].ToString());
        }
    }
    if (int_CntFound == 0) // شود می درج فاکتور در نباشد تکراری کلای که صورتی در
    {
        FactDt.Rows.Add(int.Parse(Txt_FoodCode.Text.Trim()),
        Txt_FoodName.Text.Trim(), Decimal.Parse(Txt_TedadeDarkhast.Text.Trim())
        , Decimal.Parse(Txt_UnitPrice.Text.Trim()) ,
        Decimal.Parse(Txt_Tajhfif.Text.Trim()), Decimal.Parse(Txt_FeeTotal.Text.Trim()));
        Grd_Factor.DataSource = FactDt;
    }
    else
    {
        float flt_Mojoodi2 =
UDF_GetStackRate(int.Parse(Txt_FoodCode.Text.Trim()));
        //flt_Mojoodi2 += flt_darkhast;
        if (flt_Mojoodi2 < (flt_darkhast +
float.Parse(Txt_TedadeDarkhast.Text.Trim())))
        {

```

```

        // مایش پیغام // MessageBox.Show("باشد نمی کافی موجودی مقدار");
    }
    else
    {
        foreach (DataRow Row in FactDt.Rows) // به لیست به تکراری اقلام درج
        {
            if (int.Parse(Row[0].ToString().Trim()) ==
int.Parse(Txt_FoodCode.Text.Trim()))
            {
                int_CntFound++;
                float flt_Quantity = float.Parse(Row[2].ToString());
                flt_Quantity +=
float.Parse(Txt_TedadeDarkhast.Text.Trim());
                Decimal Dec_UnitPrice =
Decimal.Parse(Row[3].ToString());
                Decimal Dec_Takhfif =
(Decimal.Parse(Row[4].ToString()) + Decimal.Parse(Txt_Tajhfif.Text.Trim()));
                Decimal Dec_TotalPrice = ((Dec_UnitPrice *
Decimal.Parse(flt_Quantity.ToString())) - Dec_Takhfif);

                Row[2] = flt_Quantity.ToString();
                Row[3] = Dec_UnitPrice.ToString();
                Row[4] = Dec_Takhfif.ToString();
                Row[5] = Dec_TotalPrice.ToString();
            }
        }
    }
}

private void button1_Click(object sender, EventArgs e) // رخداد کلیک
{
    if (frm001 == null || frm001.IsDisposed) // بررسی اینکه آیا فرم قبلا باز شده یا خیر
    {
        frm001 = new frmSearchCustomer();
        frm001.passControl = new frmSearchCustomer.PassControl1(PassData);
        frm001.ShowDialog();
    }
    else
    {
        frm001.Dispose();
        frm001 = new frmSearchCustomer();
        frm001.passControl = new frmSearchCustomer.PassControl1(PassData);
        frm001.ShowDialog();
    }
}

private void PassData(object sender, int number)
{
    Boolean Bln_Add_Detail = (Boolean)sender;
    int int_Details_Count = number;
    if (BlN_Add_Detail == true)
    {
        Txt_CustomerCode.Text = number.ToString();
    }
    else
    {
        // مایش پیغام // MessageBox.Show("است نشده انتخاب مشتری کد");
    }
}

```

```

private void Txt_CustomerCode_TextChanged(object sender, EventArgs e)
{
    LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد شی از
    کلاس لینک جهت اتصال به پایگاه داده
    try
    {
        tblCustomer tblCmr = db.tblCustomers.Single(n => n.CustomerID ==
        Decimal.Parse(Txt_CustomerCode.Text.Trim()));
        Txt_NameSname.Text = tblCmr.CustomerName + " " +
        tblCmr.CustomerFamily;
        Txt_Phone.Text = tblCmr.CustomerTell;
        Txt_Mobile.Text = tblCmr.CustomerMobail;
        Txt_Address.Text = tblCmr.CustomerAddress;
    }
    catch
    {
    }
}

private void Txt_TedadeDarkhast_TextChanged(object sender, EventArgs e)
{
    try
    {
        Txt_FeeTotal.Text = ((Decimal.Parse(Txt_TedadeDarkhast.Text.Trim()) *
        Decimal.Parse(Txt_UnitPrice.Text.Trim())) -
        Decimal.Parse(Txt_Tajhfif.Text.Trim())).ToString();
    }
    catch
    { }
}

private void Txt_UnitPrice_TextChanged(object sender, EventArgs e)
{
    try
    {
        Txt_FeeTotal.Text = ((Decimal.Parse(Txt_TedadeDarkhast.Text.Trim()) *
        Decimal.Parse(Txt_UnitPrice.Text.Trim())) -
        Decimal.Parse(Txt_Tajhfif.Text.Trim())).ToString();
    }
    catch
    { }
}

private void Txt_Tajhfif_TextChanged(object sender, EventArgs e)
{
    try
    {
        Txt_FeeTotal.Text = ((Decimal.Parse(Txt_TedadeDarkhast.Text.Trim()) *
        Decimal.Parse(Txt_UnitPrice.Text.Trim())) -
        Decimal.Parse(Txt_Tajhfif.Text.Trim())).ToString();
    }
    catch
    { }
}

private void Btn_Delete_Click(object sender, EventArgs e) // رخداد کلیک
{
    Int32 selectedCellCount =
    Grd_Factor.GetCellCount(DataGridViewElementStates.Selected);
    int int_FoodID = -5;
    for (int i = 0; i < selectedCellCount; i++)

```



```

        {
            int_FoodID =
int.Parse(Grd_Factor.Rows[int.Parse(Grd_Factor.SelectedCells[i].RowIndex.ToString())].
Cells[0].Value.ToString());
        }
        if (int_FoodID == -5)
        {
            مایش پیغام // MessageBox.Show("است نشده انتخاب حذف برای فاکتور از آیتی");
            return; // خروج از متد
        }
        for (int i = FactDt.Rows.Count - 1; i >= 0; i--)
        {
            DataRow dr = FactDt.Rows[i];
            if (int.Parse(dr["FoodID"].ToString()) == int_FoodID)
            {
                dr.Delete();
            }
        }
        Grd_Factor.DataSource = FactDt;
    }

private void Btn_Close_Click(object sender, EventArgs e) // رخداد کلیک
{
    if (Txt_CustomerCode.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر
    {
        مایش پیغام // MessageBox.Show("فرمائید انتخاب را مشتری کد لطفا");
        return; // خروج از متد
    }
    GrPr_Mohasebeh.Enabled = true;
    Grpr_Customer.Enabled = false;
    Grpr_Factor.Enabled = false;
    Decimal Dec_TotalPrice = 0;
    Decimal Dec_TotalTakhfif = 0;
    foreach (DataRow Row in FactDt.Rows)
    {
        Dec_TotalPrice += Decimal.Parse(Row[5].ToString());
        Dec_TotalTakhfif += Decimal.Parse(Row[4].ToString());
    }

    Lbl_Total_Price.Text = Dec_TotalPrice.ToString() + " ریال ";
    Lbl_Total_Takhfif.Text = Dec_TotalTakhfif.ToString() + " ریال ";
    Lbl_Total_Aglam.Text = FactDt.Rows.Count.ToString() + " مورد ";
}

private void btnAdd_Click(object sender, EventArgs e) // رخداد کلیک
{
    GrPr_Mohasebeh.Enabled = false;
    Grpr_Customer.Enabled = true;
    Grpr_Factor.Enabled = true;
    Lbl_Total_Aglam.Text = "0";
    Lbl_Total_Price.Text = "0";
    Lbl_Total_Takhfif.Text = "0";
}

private void Btn_Save_Click(object sender, EventArgs e) // رخداد کلیک
{
    LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد شی از
    کلاس لینک جهت اتصال به پایگاه داده
    tblFactorID tblFact = new tblFactorID();
    Class_ShamsiTools shDate = new Class_ShamsiTools();
    shDate.UDF_GetCurrentShamsiDate();
}

```

```

tblFact.CustomerID = Decimal.Parse(Txt_CustomerCode.Text.Trim());
tblFact.FactorShDate = shDate.CurrentShmasiDate;

db.tblFactorIDs.InsertOnSubmit(tblFact);
db.SubmitChanges(); // ثبت تغییرات در پایگاه داده

tblFactorDetail tblFID;
foreach (DataRow Row in FactDt.Rows)
{
    tblFID = new tblFactorDetail();
    tblFID.FactorID = tblFact.FactorID;
    tblFID.FoodID = int.Parse(Row[0].ToString());
    tblFID.FoodQuantity = float.Parse(Row[2].ToString());
    tblFID.UnitPrice = Decimal.Parse(Row[3].ToString());
    tblFID.Takhfif = Decimal.Parse(Row[4].ToString());
    tblFID.TaxRate = Decimal.Parse(Row[5].ToString());

    db.tblFactorDetails.InsertOnSubmit(tblFID);
    db.SubmitChanges(); // ثبت تغییرات در پایگاه داده
}

Lbl_Total_Aglam.Text = "0";
Lbl_Total_Price.Text = "0";
Lbl_Total_Takhfif.Text = "0";
GrPr_Mohasebeh.Enabled = false;
Grpr_Customer.Enabled = true;
Grpr_Factor.Enabled = true;
FactDt.Rows.Clear();

for (int i = FactDt.Rows.Count - 1; i >= 0; i--)
{
    DataRow dr = FactDt.Rows[i];
    {
        dr.Delete();
    }
}

Grd_Factor.DataSource = FactDt;

// نمایش پیغام
MessageBox.Show("شدند درج داده پایگاه در موارد کلیه");
UDF_TXTFactorClear();
UDF_TXTCustomerClear();
}

private void UDF_TXTCustomerClear()
{
    Txt_CustomerCode.Text = "";
    Txt_NameSname.Text = "";
    Txt_Phone.Text = "";
    Txt_Mobile.Text = "";
    Txt_Address.Text = "";
}
}
}

```

## کدهای مربوط به فرم ثبت کاربران سیستم

کد کاربر	نام کاربری	کلمه عبور	نام	نام خانوادگی	نوع کاربر	تلفن	تلفن همراه
۶	omid	۱۲۳۴	امید	امیدوار	منتشی	۶۶۳	۰۹۳۷۱۳۴۰۹۸
۳	hamid	hamid	hamid	hamid	منتشی	۳۳	۰۹۳۵
۲	reza	reza	رضا	رضایی	منتشی	۲۲۹	۰۹۳۷
۱	admin	admin	admin	admin	مدیر	۱۱۱	۰۹۳۶

جستجو کردن

حذف کردن

ویرایش کردن

افزافه کردن

منتشی

نوع کاربر:

تلفن:

تلفن همراه:

جستجو:

نام کاربری:

کلمه عبور:

نام:

نام خانوادگی:

غیر فعال شدن:

نمایه 13 - فرم مربوط به کاربران سیستم

```
namespace FlowerShop
{
    public partial class frmEditUsers : Form // علامت ارث بری از کلاس اصلی فرم
    {
        public frmEditUsers()
        {
            InitializeComponent(); // این متد در هنگام قبل از لود فرم فراخوانی شده و باعث می شود که اجزا و اشیاء استفاده شده بر روی فرم ظاهر و فعال شوند
        }

        private void UDF_ClearTextBox()
        {
            Txt_UseID.Clear();
            Txt_UserName.Clear();
            Txt_Password.Clear();
            Txt_Name.Clear();
            Txt_Family.Clear();
            Txt_Tell.Clear();
            Txt_Mobail.Clear();
            Chb_BanBit.Checked = false;
        }

        private void UDF_CmbFill()
        {
            LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد شی از
            // کلاس لینک جهت اتصال به پایگاه داده
            var Query1 = from Row in db.tblUserTypes
                          orderby Row.UserTypeID descending
                          where Row.UserTypeID != 3
                          select new
                          {
                              Row.UserTypeID,
                              Row.UsersTypeDescription
                          };

            Cmb_UserType.DisplayMember = "UsersTypeDescription";
        }
    }
}
```

```

Cmb_UserType.ValueMember = "UserID";

Cmb_UserType.DataSource = Query1;
}

private void UDF_GrdFill() // توسط این متد اطلاعات فراخوانی شده در گرید ویو نمایش داده می شود
{
    LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد شی از
    کلاس لینک جهت اتصال به پایگاه داده
    var Query = from Row in db.tblUsers
        orderby Row.UsersID descending
        where Row.UserID != 3 select new
        {
            Row.UserID,
            Row.UserName,
            Row.UsersPass,
            Row.UserFName,
            Row.UsersSName,
            Row.tblUserType.UsersTypeDescription,
            Row.UserTell,
            Row.UserMobail,
            Row.UserBanBit
        };
    dataGridView1.DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو
}

private void Btn_Save_Click(object sender, EventArgs e) // رخداد کلیک
{
    if (Txt_UserName.Text == "" || Txt_Password.Text == "" || Txt_Name.Text ==
    "" || Txt_Family.Text == "" || Txt_Tell.Text == "" || Txt_Mobail.Text == "") // بررسی
    اینکه آیا فیلد متنی خالیست یا خیر
    {
        MessageBox.Show("باید می الزامی شده خواسته موارد ورود");
        return; // خروج از متد
    }

    LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد شی از
    کلاس لینک جهت اتصال به پایگاه داده
    tblUser user = new tblUser();

    user.UserName = Txt_UserName.Text.Trim(); // دریافت مقدار از فیلد متنی
    user.UsersPass = Txt_Password.Text.Trim(); // دریافت مقدار از فیلد متنی
    user.UserFName = Txt_Name.Text.Trim(); // دریافت مقدار از فیلد متنی
    user.UsersSName = Txt_Family.Text.Trim(); // دریافت مقدار از فیلد متنی
    user.UserID = int.Parse(Cmb_UserType.SelectedValue.ToString());
    user.UserTell = Txt_Tell.Text.Trim(); // دریافت مقدار از فیلد متنی
    user.UserMobail = Txt_Mobail.Text.Trim(); // دریافت مقدار از فیلد متنی
    user.UserBanBit = Chb_BanBit.Checked;

    db.tblUsers.InsertOnSubmit(user);
    db.SubmitChanges(); // ثبت تغییرات در پایگاه داده

    UDF_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو
    UDF_CmbFill();

    btnAdd.BringToFront(); // نمایش کلید در بالای سایر آبجکتها
    Btn_Save.SendToBack(); // نمایش کلید در پشت سایر آبجکتها
    UDF_ClearTextBox();

    MessageBox.Show("گرددید درج جدید رکورد");
}

```

```

    }

    private void btnAdd_Click_1(object sender, EventArgs e)
    {
        Txt_UseID.Text = String.Empty; // حذف مقادیر متنی
        Txt_UserName.Text = String.Empty; // حذف مقادیر متنی
        Txt_Password.Text = String.Empty; // حذف مقادیر متنی
        Txt_Name.Text = String.Empty; // حذف مقادیر متنی
        Txt_Family.Text = String.Empty; // حذف مقادیر متنی
        Txt_Tell.Text = String.Empty; // حذف مقادیر متنی
        Txt_Mobail.Text = String.Empty; // حذف مقادیر متنی
        Chb_BanBit.Checked = false;

        btnAdd.SendToBack(); // نمایش کلید در پشت سایر آبجکتها
        Btn_Save.BringToFront(); // نمایش کلید در بالای سایر آبجکتها
        UDF_ClearTextBox();
        // نمایش پیغام // MessageBox.Show("اطلاعات لطفا");
    }

    private void Cmb_UserType_SelectedIndexChanged(object sender, EventArgs e)
    {
    }

    private void frmEditUsers_Load(object sender, EventArgs e) // این متد در هنگام لود فرم
    اجرا می شود
    {
        // ثبت رخداد // Class_tblLogs LogReg = new Class_tblLogs(this.Text + " فعال
        // گرید");
        LogReg.UDF_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می شود
        UDF_CmbFill();
        UDF_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو
    }

    private void Btn_Delete_Click(object sender, EventArgs e) // رخداد کلیک
    {
        if (Txt_UseID.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر
        {
            // نمایش پیغام // MessageBox.Show("فرم انتخاب را نظر مورد کاربر لطفا");
            return; // خروج از متد
        }

        DialogResult dr = // نمایش پیغام // MessageBox.Show("اطمینان رکورد حذف عمل به نسبت آیا");
        // نمایش پیغام // MessageBoxButtons.OKCancel, // نمایش پیغام "حذف اخطار", "دارید
        MessageBoxIcon.Warning);
        if (dr == DialogResult.OK)
        {
            LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد
            شی از کلاس لینک جهت اتصال به پایگاه داده

            tblUser user = db.tblUsers.Single(n => n.UsersID ==
            int.Parse(Txt_UseID.Text.Trim()));

            // ثبت رخداد // Class_tblLogs LogReg = new Class_tblLogs(this.Text + " رکورد
            " + Txt_UseID.Text.Trim() + " -ID= " + Txt_UseID.Text.Trim() + " حذف");
            LogReg.UDF_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می
            شود

            db.tblUsers.DeleteOnSubmit(user);
            db.SubmitChanges(); // ثبت تغییرات در پایگاه داده
        }
    }

```

```

        UDF_ClearTextBox();
        UDF_CmbFill();
        UDF_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو
    }
}

private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e) // این متد در هنگام کلیک بر روی سطرهای گرید ویو انجام می شود و مقادیر مربوط به محتویات هر سلول را بر می گرداند
{
    Int32 selectedCellCount =
dataGridView1.GetCellCount(DataGridViewElementStates.Selected);

    if (selectedCellCount > 0)
    {
        if (dataGridView1.AreAllCellsSelected(true))
        {
            // مایش پیغام // MessageBox.Show("All cells are selected", "Selected
Cells");
        }
        else
        {
            System.Text.StringBuilder sb = new System.Text.StringBuilder();
            for (int i = 0; i < selectedCellCount; i++)
            {
                Boolean Bln_Chk;
                Txt_UseID.Text =
dataGridView1.Rows[int.Parse(dataGridView1.SelectedCells[i].RowIndex.ToString())].Cells[0].Value.ToString();
                Txt_UserName.Text =
dataGridView1.Rows[int.Parse(dataGridView1.SelectedCells[i].RowIndex.ToString())].Cells[1].Value.ToString();
                Txt_Password.Text =
dataGridView1.Rows[int.Parse(dataGridView1.SelectedCells[i].RowIndex.ToString())].Cells[2].Value.ToString();
                Txt_Name.Text =
dataGridView1.Rows[int.Parse(dataGridView1.SelectedCells[i].RowIndex.ToString())].Cells[3].Value.ToString();
                Txt_Family.Text =
dataGridView1.Rows[int.Parse(dataGridView1.SelectedCells[i].RowIndex.ToString())].Cells[4].Value.ToString();
                Txt_Tell.Text =
dataGridView1.Rows[int.Parse(dataGridView1.SelectedCells[i].RowIndex.ToString())].Cells[6].Value.ToString();
                Txt_Mobail.Text =
dataGridView1.Rows[int.Parse(dataGridView1.SelectedCells[i].RowIndex.ToString())].Cells[7].Value.ToString();
                Bln_Chk =
Boolean.Parse(dataGridView1.Rows[int.Parse(dataGridView1.SelectedCells[i].RowIndex.ToString())].Cells[8].Value.ToString());

                if (BlN_Chk == true)
                {
                    Chb_BanBit.Checked = true;
                }
                else if (BlN_Chk == false)
                {
                    Chb_BanBit.Checked = false;
                }
            }
        }
    }
}

```

```

        UDF_SelectCmbValue(int.Parse(Txt_UseID.Text.Trim()));
    }
}

private void UDF_SelectCmbValue(int int_UserID)
{
    LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد شی از
    کلاس لینک جهت اتصال به پایگاه داده
    tblUser user = db.tblUsers.Single(n => n.UsersID == int_UserID);
    Cmb_UserType.SelectedValue = user.UserTypeID;
}

private void Btn_Edit_Click(object sender, EventArgs e) // رخداد کلیک
{
    if (Txt_UseID.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر
    {
        مایش پیغام // MessageBox.Show("فرماتید انتخاب را نظر مورد کاربر لطفا");
        return; // خروج از متد
    }
    if (Cmb_UserType.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر
    {
        مایش پیغام // MessageBox.Show("باشد می الزامی کاربر نوع انتخاب");
        return; // خروج از متد
    }

    DialogResult dr = مایش پیغام // MessageBox.Show("اطمینان رکورد ویرایش عمل به نسبت آیا");
    مایش پیغام // MessageBoxButtons.OKCancel, مایش پیغام "ویرایش اخطار", "دارید"
    MessageBoxIcon.Warning);
    if (dr == DialogResult.OK)
    {
        LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد
        شی از کلاس لینک جهت اتصال به پایگاه داده
        tblUser user = db.tblUsers.Single(n => n.UsersID ==
int.Parse(Txt_UseID.Text.Trim()));

        user.UsersName = Txt_UserName.Text.Trim(); // دریافت مقدار از فیلد متنی
        user.UsersPass = Txt_Password.Text.Trim(); // دریافت مقدار از فیلد متنی
        user.UserFName = Txt_Name.Text.Trim(); // دریافت مقدار از فیلد متنی
        user.UsersSName = Txt_Family.Text.Trim(); // دریافت مقدار از فیلد متنی
        user.UserTypeID = int.Parse(Cmb_UserType.SelectedValue.ToString());
        user.UserTell = Txt_Tell.Text.Trim(); // دریافت مقدار از فیلد متنی
        user.UserMobail = Txt_Mobail.Text.Trim(); // دریافت مقدار از فیلد متنی
        user.UserBanBit = Chb_BanBit.Checked;

        db.SubmitChanges(); // ثبت تغییرات در پایگاه داده
        رکورد " // Class_tblLogs LogReg = new Class_tblLogs(this.Text + "
" + Txt_UserName.Text.Trim() + " -ID= " + Txt_UseID.Text.Trim() + "ویرایش");
        LogReg.UDF_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می
        شود

        UDF_ClearTextBox();
        UDF_CmbFill();
        UDF_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو
    }
}

private void Btn_Search_Click(object sender, EventArgs e) // رخداد کلیک
{
    if (Txt_Search.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر
    {
        مایش پیغام // MessageBox.Show("نمائید وارد را جستجو فیلد لطفا");
    }
}

```

```

        return; // خروج از متد
    }

    if (Rdo_UserID.Checked == true)
    {
        LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد
        شی از کلاس لینک جهت اتصال به پایگاه داده
        var Query = from Row in db.tblUsers
                     where Row.UsersID == int.Parse(Txt_Search.Text.Trim())
                     select new
                     {
                         Row.UsersID,
                         Row.UserName,
                         Row.UsersPass,
                         Row.UserFName,
                         Row.UsersSName,
                         Row.tblUserType.UsersTypeDescription,
                         Row.UserTell,
                         Row.UserMobail,
                         Row.UserBanBit
                     };
        dataGridView1.DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو
        // ثبت رخداد // Class_tblLogs LogReg = new Class_tblLogs(this.Text + " -
        LogReg.UDF_SaveLogToDb()); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می
        شود
        if (Query.Count() <= 0)
        {
            Txt_Search.Text = "";
            Btn_Save.BringToFront(); // نمایش کلید در بالای سایر آبجکتها
            Btn_Search.SendToBack(); // نمایش کلید در پشت سایر آبجکتها
            // نمایش پیغام // MessageBox.Show("نشد یافته رکوردی");
        }
    }
    else if (Rdo_UserName.Checked == true)
    {
        LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد
        شی از کلاس لینک جهت اتصال به پایگاه داده
        var Query = from Row in db.tblUsers
                     where SqlMethods.Like(Row.UserName, "%" +
        Txt_Search.Text.Trim() + "%")
                     select new
                     {
                         Row.UsersID,
                         Row.UserName,
                         Row.UsersPass,
                         Row.UserFName,
                         Row.UsersSName,
                         Row.tblUserType.UsersTypeDescription,
                         Row.UserTell,
                         Row.UserMobail,
                         Row.UserBanBit
                     };
        dataGridView1.DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو
        // ثبت رخداد // Class_tblLogs LogReg = new Class_tblLogs(this.Text + " -
        LogReg.UDF_SaveLogToDb()); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می
        شود
        if (Query.Count() <= 0)
        {
            Txt_Search.Text = "";

```



```

        Btn_Save.BringToFront(); // نمایش کلید در بالای سایر آجکتها
        Btn_Search.SendToBack(); // نمایش کلید در پشت سایر آجکتها
        // نمایش پیغام // MessageBox.Show("نشد یافته رکوردی");
    }
}
else if (Rdo_Family.Checked == true)
{
    LSC_FlowerShopDataContext db = new LSC_FlowerShopDataContext(); // ایجاد
    شی از کلاس لینک جهت اتصال به پایگاه داده
    var Query = from Row in db.tblUsers
                where SqlMethods.Like(Row.UsersSName, "%" +
    Txt_Search.Text.Trim() + "%")
                select new
                {
                    Row.UsersID,
                    Row.UserName,
                    Row.UsersPass,
                    Row.UserFName,
                    Row.UsersSName,
                    Row.tblUserType.UsersTypeDescription,
                    Row.UserTell,
                    Row.UserMobail,
                    Row.UserBanBit
                };
    dataGridView1.DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو
    // ثبت رخداد // Class_tblLogs LogReg = new Class_tblLogs(this.Text + " -
    ("جستجو عملیات";
    LogReg.UDF_SaveLogToDb()); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت می
    شود
    if (Query.Count() <= 0)
    {
        Txt_Search.Text = "";
        Btn_Save.BringToFront(); // نمایش کلید در بالای سایر آجکتها
        Btn_Search.SendToBack(); // نمایش کلید در پشت سایر آجکتها
        // نمایش پیغام // MessageBox.Show("نشد یافته رکوردی");
    }
}
}

private void Btn_Refresh_Click(object sender, EventArgs e) // رخداد کلیک
{
    UDF_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو
    UDF_CmbFill();
    Btn_Refresh.SendToBack(); // نمایش کلید در پشت سایر آجکتها
    Btn_Search.BringToFront(); // نمایش کلید در بالای سایر آجکتها
    UDF_ClearTextBox();
}
}
}

```

## چاپ فاکتور

شماره فاکتور: ۲۴ - نام و نام خانوادگی: مهرداد شهرامی - تاریخ فاکتور: ۱۳۹۸/۰۵/۱۱

شماره فاکتور:

چاپ فاکتور

تاریخ	مشتری	نام گل	تعداد	قیمت واحد	تخفیف
۱۳۹۸/۰۵/۱۱	مهرداد شهرامی	شقایق	۳	۵۰۱۴	۱۰۰۰
۱۳۹۸/۰۵/۱۱	مهرداد شهرامی	شب بو	۱	۹۵۶۸	۰
۱۳۹۸/۰۵/۱۱	مهرداد شهرامی	فیلندرون	۱	۵۶۹۳	۰
۱۳۹۸/۰۵/۱۱	مهرداد شهرامی	بنجامین	۲	۵۴۰۸	۰
۱۳۹۸/۰۵/۱۱	مهرداد شهرامی	آزالیا	۷	۴۴۶۱	۳۵۰۰
۱۳۹۸/۰۵/۱۱	مهرداد شهرامی	مریم	۵	۱۱۳۳۲	۲۰۰۰
۱۳۹۸/۰۵/۱۱	مهرداد شهرامی	پتوس	۱۰	۱۲۸۲۳	۰
۱۳۹۸/۰۵/۱۱	مهرداد شهرامی	حسن یوسف	۲	۴۸۰۰	۱۲۰۰

نام مشتری: مهرداد شهرامی تاریخ فاکتور: ۱۳۹۸/۰۵/۱۱

**فاکتور فروش**

ردیف	نام گل	قیمت واحد	تعداد	تخفیف	مبلغ کل
۱	شقایق	۵۰۱۴	۳	۱۰۰۰	۱۴۰۰۴۲
۲	شب بو	۹۵۶۸	۱	۰	۹۵۶۸
۳	فیلندرون	۵۶۹۳	۱	۰	۵۶۹۳
۴	بنجامین	۵۴۰۸	۲	۰	۱۰۸۱۶
۵	آزالیا	۴۴۶۱	۷	۳۵۰۰	۲۷۷۲۷
۶	مریم	۱۱۳۳۲	۵	۲۰۰۰	۵۴۶۶۰
۷	پتوس	۱۲۸۲۳	۱۰	۰	۱۲۸۲۳۰
۸	حسن یوسف	۴۸۰۰	۲	۱۲۰۰	۸۴۰۰
جمع کل:					۲۵۹۰۱۳۶

namespace FlowerShop

```
{
    public partial class frmFactorPrint : Form
    {
        public frmFactorPrint()
        {
            InitializeComponent();
        }

        private void frmFactorPrint_Load(object sender, EventArgs e)
        {
            UDF_GetData();
        }
    }
}
```

```

private void UDF_GetData()// واكشی دیتا
{
    this.xtblDropFactorsTableAdapter.Fill(this.dataset.xtblDropFactors);
    this.xtblFactorRptTableAdapter.FillByFactorID(this.dataset.xtblFactorRpt,
int.Parse(comboBox1.SelectedValue.ToString()));
}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    this.xtblFactorRptTableAdapter.FillByFactorID(this.dataset.xtblFactorRpt,
int.Parse(comboBox1.SelectedValue.ToString()));
}

private void Btn_AddToList_Click(object sender, EventArgs e)// نمایش فاکتور
{
    StiReport Report = new StiReport();
    Report = (StiReport)stiReport1.Clone();
    Report.RegData(this.dataset.xtblFactorRpt);
    Report.Compile();
    Report.Show();
}
}
}

```

## **فصل چهارم: جمع بندی ، نتیجه گیری و پیشنهادات**

## جمع بندی

سیستم ارائه شده نمونه ای از صدها مورد از سیستمهای نرم افزاری است که به صورت تحت ویندوز سازی شده است. زبان برنامه نویسی مورد استفاده نیز یک زبان برنامه نویسی شی گرا با عنوان C# می باشد که به علت وجود منابع آموزشی بسیار زیاد نسبت به سایر زبان های شی گرا، از این زبان برنامه نویسی به عنوان زبان برنامه سازی Base استفاده شده است.

C# یکی از آن زبان هایی است که برخلاف C، امکانات شی گرایی را در بطن وجودش پیاده سازی کرده است. برنامه نویسان میتوانند با استفاده از C# به صورت قدرتمندی اقدام به تولید برنامه های شی گرا نمایند. اما به خاطر داشته باشید وجود تفکر شی گرایی به هنگام برنامه نویسی مهم تر از وجود امکانات شی گرایی در آن زبان خاص است.

حالا که با مفهوم شی گرایی در دنیای حقیقی آشنا شدید می توانید با دنبال کردن این سری پروژه ها، تمام این مفاهیم را با استفاده از C# در دنیای برنامه نویسی هم پیاده سازی کنید. در این پروژه سعی شده است تا جدیدترین روش های شی گرایی در C# بررسی شود. به همین خاطر ما از کلاس های سبک جدید C# برای آموزش استفاده کرده ایم که به مراتب قابلیت های بالاتری نسبت به کلاس های کلاسیک C# دارند.

برای اتصال به پایگاه داده از تکنولوژی ADO.NET و LINQ استفاده خواهد شد. البته موتور پایگاه داده SQL Server 2005 می باشد که یک موتور بسیار قوی با مدیریت بسیار عالی برای دسترسی به داده است. این مورد تنها زمانی مشخص می گردد که داده ها در حجم بالا میان Server و Client در حال رد و بدل شدن هستند.

## نتیجه گیری

پس از پیاده سازی این طرح نتایج حاصله از بررسی های به عمل آمده در مورد تکنولوژیهای برنامه نویسی این نتیجه حاصل شد که همیشه کوتاهترین راه بهترین راه برای رسیدن به هدف نیست. به عنوان مثال هنگام استفاده از ابزار آماده برای دسترسی به داده ها اگر چه عملیات برنامه نویسی کاهش یافت ، ولی مشکل اینجا بود که در هنگام کار با داده ها زمان واکنشی و نمایش داده ها به میزان قابل توجهی افزایش یافته بود و این مساله در طولانی مدت باعث برخی مشکلات خواهد شد.

## پیشنهادهای

متأسفانه در ایران به علت عدم وجود قانون کپی رایت ، نمی توان در این زمینه پیشنهادات چندانی را ارائه نمود. لیکن برای بهتر شدن کار ، برخی مسائل به شرح ذیل ارائه می گردد که در صورت ارائه باید قبلاً زیرساختهای آن به نحو احسن پیاده سازی شده باشد:

- ۱- برنامه برای برنامه نویس چه سودی می تواند داشته باشد
- ۲- آیا پروژه می تواند دارای سرمایه گذار باشد یا نه
- ۳- آیا با توجه نبود قانون برای جلوگیری از دزدی نرم افزار می تواند قفل مناسبی برای نرم افزار تهیه نمود
- ۴- برای برنامه در ابتدا طرح و نقشه تهیه نمود (تجزیه و تحلیل اساسی)
- ۵- در مشتریان ایجاد نیاز نمود
- ۶- به مشتری مزایای این گونه طرح ها را گوشزد نمود
- ۷- حداقل امکان پروژه با تکنولوژیهای مختلف ساگاری داشته باشد

- ۸- به گونه ای برای جلوگیری از دزدی نرم افزار هزینه نرم افزار در فروش به تعداد بالا گرفته شود و نه در فروش به صورت تکی و آن هم به یک مشتری خاص
- ۹- افرادی که قرار است بر روی پروژه کار کنند قبلاً مشخص شده باشند و از افراد قابل اطمینان باشند
- ۱۰- برنامه نویسان در ابعادی کوچکتر از ذینفع باشند. چراکه این مساله خود موجب دلخوشی و افزایش بازدهی کار آنها خواهد شد.

## منابع

### کتابها

۱. بولتن آموزشی برنامه نویسی نوشته آقای مهندس اسدی - انتشارات امیر کبیر - چاپ سوم - بهار ۸۶
۲. کتاب آموزش سی شارپ نوشته مهندس شایان - انتشارات جمیل - چاپ پنجم - تابستان ۹۰
۳. کتاب طراحی پایگاه داده نوشته مهندس رانکوهی - انتشارات جهاد دانشگاهی - چاپ چهارم - بهار ۸۳
۴. کتاب طراحی پایگاه داده ها نوشته مهندس ساها کیان - انتشارات جهاد دانشگاهی - چاپ سوم - سال ۸۸
۵. کتاب آموزش C# نوشته مهندس زاهد - انتشارات توحید - چاپ سوم - سال ۸۸

### وب سایتها

<http://support.microsoft.com>  
<http://www.codeproject.com>  
<http://www.barnamenevis.org>  
<http://www. www.w3schools.com>  
<http://forum.asp.net>  
<http://forum.persiantools.com>  
<http://web.shaar.com>  
<http://www.farsiebook.com>  
<http://www.zytrax.com>  
<http://www.regular-expressions.info>  
<http://www.grymoire.com>  
<http://www.developer.apple.com>