**موسسه آموزش عالی آیندگان تنکابن**

***گزارش کارآموزی***

استاد راهنما :

**جناب اقای مهندس سروش**

تهیه کننده :

**زهرا احسانیان مفرد**

بهار 90

فهرست مطالب

مقدمه ............................................................................................................3

خلاصه کار اموزی ..........................................................................................4

شرکت سهامی بیمه اسیا .......................................................................................4

گذری بر تاریخچه بیمه در جهان و ایران ....................................................................5

اصطلاحات بیمه ای ..........................................................................................8

مفاهیم پایه ای برنامه نویسی سی شارپ .....................................................................15

توضیحات برنامه .............................................................................................26

 نتبجه گیری .....................................................................................................31

**مقدمه**

از زمانی که برنامه نویسی در لایه های میانی جامعه مجازی و در میان کاربران غیر حرفه ای و غالبا نا اشنا با عملکرد سیستم های کامپیوتری وارد شده و سیل برنامه های کاربردی کوچک که برای انجام امور ساده در میان کاربران استفاده می گردد رو به فزونی گرفت لزوم ساده ساده سازی فرایند تولید نرم افزارها احساس می گردید . در فضایی که تولید برنامه نیاز به تخصص بالا و شناخت کامل از عملکرد واحدهای سیستمی و سخت افزار رایانه را جلب می نمود(زبانهای سطح ماشینی ) پا به عرصه نهادند که دارای ساختاری ساده تر و زبانی نزدیک به زبان انسان بودند .

ورود اینگونه زبانهای برنامه نویسی به میدان تولید تولید نرم افزار انقلاب بزرگی در زمینه تولید نرم افزارهای کاربردی ایجاد نمود و ان را از کاری پیچیده و زمانبر به چرخه ای ساده و کم مشغله تبدیل نمود .

با گسترش روز افزون صنعت تولید نرم افزار در دنیا و گرایش صنایع مختلف به سوی استفاده از نرم افزار و ماشین به جای انسان در محدوده فعالیت خود همگام با افزایش تولید این سیستم ها ، پیچیدگی این سیستم ها نیز افزایش یافت . طراحی سیستم های پیچیده ای که هم از حیث ساختار برنامه سازی دارای لایه های متعدد بوده و هم هم از حیث امنیت نیازمند بکارگیری و همکاری تعداد زیادی برنامه نویس را جلب می نمود راه حل هایی بودند تا سازندگان انها از محتوای سیستم طراحی شده مطلع نگردند طراحان زبانهای برنامه نویسی را به سوی تولید زبانهایی سوق داد که امروزه به نام زبانهای برنامه نویسی شی گرا شناخته می شوند .

شرکت ( MICRO SYSTEM) SUN با معرفی زبان بسیار قدرتمند JAVA پا به عرصه این زبانها نهاد و در مقابل شرکت مایکروسافت با معرفی زبان C# به مقابله با این زبان پرداخت .

در برابر زبانهای برنامه نویسی ساخت یافته زبانهای شی گرا از قدرت و کاراییی بالاتری برخوردار می باشند ، در طراحی زبانهای شی گرا با تشریح یک شی در ساختار برنامه ئدر طول برنامه می توان از توانایی ها و و قابلیت های شی مورد نظر استفاده نمود . استفاده از اشیا در بدنه برنامه دارای دو مزیت عمده می باشد اول اینکه از طول برنامه کاسته شده و برنامه را ساده تر می کند و دوم اینکه این امکان را به طراحان سیستم میدهد تا با تقسیم سیستم به دچند زیر سیستم مجزا و اتصال انها در پایان کار امنیت انجام پروژه را افزایش دهد

***خلاصه کار اموزی***

اینجانب دوران کاراموزی را در سازمان بیمه اسیا گذرانده و در این دوران باکمک زبان سی شارپ سیستمی جهت ثبت و انجام پردازش بیمه نامه ها و همچنین ثبت و انجام پردازش های گوناگون بر روی پرونده های خسارتی طراحی و برنامه نویسی کردم .

در ادامه ابتدا به بیان تاریخچه بیمه و اصطلاحات بیمه ای می پردازیم و سپس اطلاعاتی پایه ای از زبان سی شارپ بیان می کنیم و نیز قطعاتی از کد برنامه را به همراه توضیح جهت نمونه می اوریم .

***شرکت سهامی بیمه اسیا***

شرکت بیمه اسیا در تیر ماه سال سال 1338 به منظور انجام معاملانت بیمه ای و عملیات بازرگانی تاسیس شد . این شرکت در همان تاریخ ذیل شماره 6795 در اداره ثبت شرکت ها به ثبت رسیده است بر اساس مصوبه شورای انقلاب اسلامی و لایحه قانونی ملی شدن موسسات بیمه و اعتباری مصوب سال 1358 شرکت سهامی بیمه اسیا از تاریخ 4/4/1358 ملی اعلام شد .

شرکت سهامی بیمه اسیا به منظور توسعه فعالیت های خود در سراسر کشور و تامین نیازهای بیمه ای هموطنان عزیز مبادرت به تشکیل سرپرست های یازده گانه نموده است که هریک ار انها نیز چند شعبه را تحت پوشش دارند .

در حال حاضر 11 نمایندگی حدود 90 شعبه و اداره پرداخت خسارت و افزون بر 2000 نمایندگی حقیقی و حقوقی در بیش از 275 شهر کشور شبکه فعالیت شرکت بیمه اسیا را تشکیل می دهند .

***گذری بر تاریخچه بیمه در جهان و ایران :***

**بيمه چگونه به وجود آمد :**

بيمه‌هاي غير زندگي پيشينه بلند مدتي دارند. نوعي بيمه‌هاي دريايي در حدود 3000 سال قبل مورد استفاده قرار گرفت. بيمه‌هاي زندگي نيز سابقه زيادي دارند. اين نوع بيمه‌ها نخستين‌بار هنگامي پديد آمد که سربازان رومي قسمتي از دستمزد خود را در صندوقي جمع‌آوري کردند تا چنانچه در جنگ کشته شدند آن پول به خانواده‌هايشان پرداخت شود.

**بيمه هاي غير زندگي :**

بيمه دريايي که امروز وجود دارد احتمالاً در حدود صده‌هاي يازدهم و دوازدهم ميلادي در منطقه‌اي در شمال ايتاليا به‌وجود آمده است. شخصي به نام لومباردز در صده‌هاي چهاردهم و پانزدهم انگليسي ها را با اين نوع فعاليت آشنا ساخت. مفاهيم بيمه‌گر و بيمه‌گري نخستين‌بار در بيمه دريايي مطرح شد. آن روزها هر بازرگاني که حاضر به تقبل بخشي از يک خطر بود نام خود را همراه با سهمي که از آن خطر قبول مي‌کرد در پايين صفحه‌اي که جزئيات خطر مزبور در آن درج شده بود مي‌نوشت. آن وقت مالکان کشتي‌ها و بازرگانان دريافتند که مي‌توانند کشتي‌ها را براي حمل اجناس سودآور اعزام کنند زيرا مي‌دانستند چنانچه کشتي دچار حادثه شود از اين طريق جبران خواهد شد. در سال 1666 آتش‌سوزي بسيار بزرگي در لندن روي داد که باعث به وجود آمدن بيمه آتش‌سوزي شد. معلوم نيست که نخستين بيمه آتش‌سوزي به صورت امروزي در چه زماني صادر شد اما گفته مي‌شود که نخستين شرکت بيمه آتش‌سوزي به‌نام اداره آتش در سال 1680 بنيان نهاده شد که بعدها فونيکس لقب گرفت.

**تاريخچه بيمه در ايران :**

بيمه به شکل سنتي آن به مفهوم وجود نوعي تعاون و همياري اجتماعي بمنظور سرشکن کردن زيان فرد و يا افراد معدود بين همه افراد کروه يا جامعه در ايران سابقه طولاني داشته و همواره مردم اين مرز و بوم با الهام از تعاليم اسلامي و فرهنگي خود براي کمک به جبران خسارتهاي ناخواسته‌اي که براي ديگر هموطنان و حتي مردم ديگر کشورها پيش مي‌آمد فعال و پيشگام بوده‌اند. با وجود اين بيمه شکل حرفه‌اي و امروزي آن براي اولين بار در سال 1269 هجري شمسي در کشور ما مطرح گرديده است.

در اين سال مذاکراتي بين دولت ايران و سفارت روس بعمل آمد و متعاقب آن امتياز فعاليت انحصاري در زمينه بيمه و حمل و نقل براي مدت 75 سال به يک تبعه روس بنام لازارپلياکف واگذار شد. با اين وجود نامبرده ظرف مهلت سه سالي که جهت آغاز فعاليت بيمه‌اي براي وي در نظر گرفته شده بود قادر به تأسيس شرکت بيمه مورد نظر نگرديد و به همين جهت اين امتياز از وي سلب شد. متعاقباً در سال 1289 هجري شمسي دو شرکت بيمه روسي به نامهاي نادژدا و کافکازمرکوري اقدام به تأسيس نمايندگي بيمه جهت بازديد و پرداخت خسارت در ايران نمودند.
آغاز فعاليت جدي در زمينه بيمه را مي‌توان سال 1310 هجري شمسي دانست. زيرا در اين سال قانون و نظامنامه راجع به ثبت شرکتها در ايران و به تصويب رسيد و متعاقب آن بسياري از شرکتهاي بيمه خارجي از جمله گستراخ- آليانس- ايگل استار- يورکشاير- رويال- ويکتوريا- ناسيونال سوئيس- فنيکس- اتحاد الوطني و ... اقدام به تأسيس شعبه يا نمايندگي در ايران نمودند.

گسترش فعاليت‌هاي شرکت‌هاي بيمه خارجي مسئولان کشور را متوجه ضرورت تأسيس يک شرکت بيمه ايراني کرد تأسيس و فعاليت رسمي خود را از اواسط آبان‌ماه همان سال آغاز شد.

تأسيس شرکت بيمه ايران را مي‌توان نقطه عطفي در تاريخ فعاليت بيمه شگور دانست زيرا از آن پس دولت با در اختيار داشتن تشکيلات اجرائي مناسب قادر به کنترل بازار و نظارت بر فعاليت مؤسسات بيمه خارجي گرديد. دو سال بعد از تأسيس شرکت سهامي بيمه ايران يعني در سال 1316 قانون بيمه در 36 ماده تدوين و به تصويب مجلس شوراي ملي رسيد. پس از آن نيز مقررات ديگري در جهت کنترل و نظارت بر فعاليت مؤسسات بيمه از طريق الزام آنها به واگذاري 25 درصد بيمه‌نامه‌هاي صادره بصورت اتکائي اجباري به شرکت سهامي بيمه ايران وضع شد و الزام به بيمه کردن کالاهاي وارداتي و صادراتي و اموال موجود در ايران و ايرانيان مقيم کشور نزد يکي از مؤسسات بيمه که در ايران به ثبت رسيده‌اند بر استحکام شرکتهاي بيمه افزود. شرکت سهامي ايران با حمايت دولت به فعاليت خود ادامه داد و اين حمايت منجر به تقويت نقش اين شرکت در بازار بيمه کشور و توقف تدريجي فعاليت شعب و نمايندگي‌هاي شرکتهاي بيمه خارجي گرديده بود بطوري که در سال 1318 بيش از 75 درصد از بازار بيمه‌اي کشور در اختيار شرکت بيمه ايران قرار گرفت و پنج شرکت بيمه خارجي که در آن‌زمان در ايران فعاليت مي‌کردند جمعاً موفق به کسب کمتر از 25 درصد از حق بيمه بازار شدند.

اين روند کماکان ادامه يافت تا آنکه در سال 1331 بر اساس مصوبه هيئت دولت کليه شرکتهاي بيمه خارجي موظف شدند جهت ادامه فعاليت خود در ايران مبلغ 250 هزار دلار به عنوان وديعه نزد بانک ملي ايران توديع نمايند و پس از آن نيز منافع ساليانه خود را تازماني‌که اين مبلغ به پانصد هزار دلار برسد بر آن بيفزايند. اين تصميم موجب تعطيل شدن کليه نمايندگي‌ها و شعب شرکتهاي بيمه خارجي در ايران به استثنا دو شرکت بيمه يورکشاير و اينگستراخ گرديد و شرايط را براي گسترش فعاليت شرکتهاي بيمه ايراني فراهم ساخت.

اولين شرکت بيمه خصوصي ايراني به نام شرق در سال 1329 هجري شمسي تأسيس گرديد. پس از آن تا سال 1343 به تدريج هفت شرکت بيمه خصوصي ديگر به نامهاي آريا- پارس- ملي- آسيا- البرز- البرز اميد و ساختمان و کار به ترتيب تأسيس و به فعاليت بيمه پرداختند.

همان‌طور که اشاره شد از سال 1316 کليه شرکتهاي بيمه موظف گرديدند 25 درصد از امور بيمه‌اي خود را به‌صورت اتکائي اجباري به شرکت بيمه ايران واگذار نمايند. اين واگذاري عمدتاً از طريق ارسال ليستهايي به نام بردرو که حاوي کليه اطلاعات راجع به بيمه‌نامه‌هاي صادره و خسارتهاي پرداخت شده توسط اين شرکتها بود انجام مي گرفت. بديهي است ارائه اطلاعات به شرکت بيمه رقيب هيچگاه نمي‌توانست مورد رضايت و علاقه شرکتهاي بيمه واگذارنده باشد. از سوي ديگر با افزايش تعداد شرکتهاي بيمه ضرورت اعمال نظارت بيشتر دولت بر اين صنعت و تدوين اصول و ضوابط استاندارد براي فعاليت‌هاي بيمه‌اي به منظور حفظ حقوق بيمه‌گذاران و بيمه‌شدگان احساس مي‌گرديد. به همين جهت در سال 1350 بيمه مرکزي ايران به منظور تحقق اهداف فوق تأسيس شد. در ماده 1 قانون تأسيس بيمه مرکزي و بيمه‌گري چنين آمده است :

به‌منظور تنظيم و تعميم و هدايت امر بيمه در ايران و حمايت بيمه‌گذاران و بيمه‌شدگان و صاحب حقوق آنها همچنين به منظور اعمال نظارت دولت بر اين فعاليت مؤسسه‌اي به نام بيمه مرکزي ايران طبق مقررات اين قانون به صورت شرکت سهامي تأسيس مي‌گردد.

اين قانون از دو بخش تشکيل شده است : در بخش اول - سازمان- ارکان- تشکيلات- وظايف و نحوه اداره بيمه مرکزي ايران تعيين گرديد و در بخش دوم ضوابط مربوط به نحوه تأسيس و فعاليت شرکتهاي بيمه و ادغام و انحلال و ورشکستگي آنها مشخص شده است.

تأسيس بيمه مرکزي ايران قوام بيشتري به صنعت بيمه کشور داد و از آن پس شورايعالي بيمه که يکي از ارکان بيمه مرکزي ايران مي‌باشد ضوابط و مقررات مختلفي را در رابطه با نحوه انجام عمليات بيمه در کشور و نرخ و شرايط انواع بيمه‌نامه تصويب نمود.

رشد سريع اقتصادي ناشي از افزايش قيمت نفت و تبع آن حجم سرمايه‌گذاري‌ها موجب توسعه بازار بيمه کشور در دهه 1350 گرديد و مجدداً شرکتهاي بيمه خارجي را علاقمند به سرمايه‌گذاري در ايران نمود.
بطوري که در سالهاي 1353 - 1354 چهار شرکت بيمه جديد به نامهاي تهران- دنا- حافظ- ايران و آميرکا با مشارکت سرمايه‌گذاران خارجي تأسيس گرديد و به ترتيب تعداد شرکتهاي بيمه خصوصي - ملي گرديد و اداره آنها به دولت سپرده شد. همچنين پروانه فعاليت نمايندگي شرکت‌هاي بيمه يورکشاير و اننگستراخ که تا اين سال فعاليت داشتند لغو گرديد.
در سالهاي 1360 و 1361 صدور بيمه‌نامه در ده شرکت بيمه ملي شده به نامهاي اميد- شرق- پارس- آريا- ساختمان و کار- تهران- حافظ- دانا- ملي و توانا متوقف گرديد و تنها سه شرکت بيمه ايران- آسيا- البرز به فعاليت جاري خود ادامه دادند.
در سال 1367 به موجب قانون اداره امور شرکتهاي بيمه مالکيت سهام شرکتهاي بيمه آسيا و البرز به دولت منتقل شد و با ادغام ده شرکت بيمه ديگر شرکت دولتي به نام بيمه دانا شکل گرفت تا منحصراً در زمينه بيمه اشخاص فعاليت نمايند.
در حال حاضر 5 شرکت دولتي بيمه ايران- آسيا- البرز- دانا و بيمه صادرات و سرمايه‌گذاري و شرکت‌هاي خصوصي بيمه ملت،توسعه،کارآفرين،پارسيان،سينا،رازي ،حافظ و سامان در زمينه بيمه فعاليت دارند.

 ***اصطلاحات بیمه ای***

AGENT
نماينده بيمه
شخصي است كه براي يك شركت بيمه ، به فروش بيمه نامه اقدام مي كند

ALL RISKS
تمام خطر
پوشش وسيع تري است كه در بيمه اموال ارائه مي شود و هر نوع زيان و صدمه اي راكه در بيمه نامه اوليه استثنا شده است نيز تحت پوشش قرار مي دهد

BROKER
دلال بيمه ; كارگزار
شخصي است كه طبق مقررات ، شرايط و مجوز فعاليت دلالي بيمه را دارد ومي تواند ضمن ارائه مشاوره به بيمه گذاران براي اخذ پوشش بيمه اي به هريك ازشركت هاي بيمه مراجعه كند

CLAIM
ادعاي خسارت
هنگامي كه بيمه گذار يا ذي نفع براساس شرايط بيمه نامه در پي دريافت خسارت ازبيمه گر است به آن ادعاي خسارت مي گويند (در زبان فارسي ، مترادف "خسارت"به كار مي رود

COMMISSION
كارمزد
درصدي از حق بيمه است كه شركت بيمه در ازاي فروش هر بيمه نامه به نماينده ياكارگزار مي پردازد

DOCUMENTARY CREDIT-LETTER OF CREDIT(L.C)
اعتبار اسنادي
به موجب اعتبار اسنادي بانك متعهد مي‌شود كه به دستور و بحساب خريدار كالا پس از دريافت مدارك معين، وجه كالا را به فروشنده بپردازد.

ENDORSEMENT
الحاقيه
نوشته اي است كه به بيمه نامه ضميمه مي شود و جزء جدايي ناپذير آن به حساب مي آيد. در واقع ، تغييراتي كه بعدا در بيمه نامه به وجود مي آيد، معمولا از طريق صدور الحاقيه انجام مي پذيرد

EXCESS OF LOSS TREATY
قرارداد اتكايي مازاد خسارت
در اين قرارداد چنانچه هر مورد خسارت از حد معيني تجاوز نمايد بيمه‌گر اتكايي نسبت به مازاد حد معين متعهد است خسارات كمتر از حد مذكور به عهده بيمه‌گر واگذارنده است.

FIRE
آتش‌سوزي
در بيمه آتش‌سوزي، آتش‌سوزي عبارت از آتشي است كه يا از يك منبع حرارتي غيرقابل كنترل سرچشمه گرٿته و يا منبع حرارتي معين كنترل شده‌اي را ترك نموده و با نيروي حرارتي خود گسترش و توسعه يابد.

GROUP LIFE INSURANCE
بيمه عمر گروهي
بيمه نامه اي است كه همزمان تعدادي از كاركنان يا كارگران يك سازمان را تحت پوشش قرار مي دهد. اين نوع بيمه معمولا در گروه هاي كارفرمايي و كارگري رايج است و كارفرما، كارگران را به ميزان چند برابر حقوق آنان ، بيمه عمر به شرط فوت مي كند

INSURED
بيمه شده
فردي كه با بيمه نامه تحت پوشش قرار گرفته است

INDEMNITY
اصل غرامت
 به موجب اصل غرامت كه بر بيمه‌هاي اشياء و مسئوليت حاكم است خسارت پرداختي توسط بيمه‌گر حداكثر به ميزان خسارت واقعي وارده به مورد بيمه در اثر وقوع خطر بيمه شده است. اصل غرامت مبتني بر ملاحظات فني، اخلاقي و رعايت نظم عمومي است. درياٿت خسارت بيش از ميزان واقعي بوسيله بيمه‌گذار يا ذينفع بيمه‌نامه علاوه بر آنكه هدٿ قرارداد بيمه نيست بلكه موجب سودجويي نيز مي‌گردد و از نظر اخلاقي و اجتماعي مذموم است. درواقع با درياٿت خسارت اضاٿي وضع مالي بيمه‌گذار پس از وقوع حادثه بهتر از وضع او در زمان قبل از حادثه مي‌شود و اين امر مي‌تواند علاوه بر آنكه انگيزه وي را در مراقبت از مورد بيمه و پيشگيري از وقوع حادثه كاهش دهد حتي موجب تشويق بيمه‌گذار به ايجاد خسارت عمدي نيز گردد. شايان توضيح اينكه اصل غرامت در بيمه‌هاي عمر و حادثه كاربرد ندارد.

INSURER
بيمه‌گر
 بيمه‌گر شخصي است حقوقي كه در مقابل درياٿت حق‌بيمه از بيمه‌گذار تعهد جبران خسارت و يا پرداخت وجه معيني را در صورت وقوع حادثه به عهده مي‌گيرد. در ماده 31 قانون تاسيس بيمه مركزي ايران و بيمه‌گري چنين آمده: عمليات بيمه در ايران به وسيله شركت‌هاي سهامي عام ايراني كه كليه سهام آنها با نام بوده و با رعايت قانون و طبق قانون تجارت به ثبت رسيده باشند انجام خواهد گرفت.

LIABILITY
مسؤوليت
منظور، مسؤوليت قانوني است كه در قبال عمل ناخواسته صدمه زدن به ديگر افراديا خسارت وارد آوردن به اموال آن ها براي شخص ايجاد مي شود

LOSS ASSESSOR
ارزياب خسارت
شخصي كه از طرف بيمه گذاران به مذاكره با بيمه گر درباره خسارت ادعا شده مي پردازد

POLICYHOLDER
بيمه گذار
شخص يا سازماني است كه بيمه گر براي او بيمه نامه صادر مي كند و معمولا منافع بيمه نامه به وي پرداخت مي شود

PREMIUM
حق بيمه
مبلغي است كه بيمه گذار براي بيمه به بيمه گر پرداخت مي كند

QUOTA - SHARE TREATY
قرارداد اتكايي مشاركت
در اين قرارداد بيمه‌گر اتكايي سهم ثابتي مثلاً 30 يا 50 درصد از مجموع بيمه‌نامه‌هاي صادره بيمه‌گر واگذارنده را در طول مدت معيني به عهده مي‌گيرد.

REINSURANCE
بيمه اتكايي
بيمه اتكايي پوششي است كه شركت هاي بيمه مي توانند با خريد آن ، همديگر را درقبال خسارت هاي بزرگ حمايت كنند

SURRENDER VALUE
ارزش بازخريد
 در بيمه‌نامه‌هاي عمر كه داراي ذخيره رياضي مي‌باشند (مثل بيمه‌هاي مختلط پس‌انداز و تمام عمر) چنانچه بيمه‌گذار پس از پرداخت مدتي حق بيمه (مثلاً حداقل شش ماه) مايل به ادامه قرارداد نباشد مي‌تواند تقاضاي بازخريد بيمه‌نامه را نمايد و مبلغي در حدود ذخيره رياضي دريافت كند. اصولاً حق بازخريد بيمه‌نامه عمر با بيمه‌گذار است. ولي چنانچه بيمه‌گذار از محل بيمه‌نامه عمر خود وام دريافت نموده باشد و اقساط خود را به بيمه‌گر پرداخت ننمايد، بيمه‌گر مي‌تواند راسا بيمه‌نامه را بازخريد و كل وام و سود متعلقه را از محل ارزش بازخريد برداشت نمايد.

THIRD PARTY
شخص ثالث
شخصي غير از بيمه گذار و بيمه گر است كه در يك حادثه تحت پوشش بيمه خسارت مي بيند

طرفین‌ قراردادبیمه:
 طرفین قرارداد بیمه عبارتند از:
1- بیمه‌گذار که‌ خریدار بیمه‌است
2- بیمه‌گر که ‌فروشنده‌ بیمه‌ است‌ ، یعنی‌ همان‌شرکت‌ بیمه‌

به‌ زبان‌ ساده‌ ، بیمه‌ چه ‌فایده‌هایی‌ دارد ؟
بیمه‌، افزون‌ بر رفع‌ تشویش‌ خاطر، دلهره‌ ونگرانی‌ ازحوادث‌ زیانبار احتمالی‌ آینده ‌، دارای‌ خصلت‌تعاون‌ و كمك‌به‌ همنوع‌ است‌.

آیا بیمه‌گذار می‌تواند اموال‌ خود را همزمان‌ برای‌یک‌ خطر به‌ نفع‌ خود نزد چند شرکت‌ بیمه‌ به‌ قیمت‌روز بیمه‌ کند ؟
خیر‌. بیمه‌ نباید موجب‌ سود و منفعت‌ بیمه‌گذار شود. البته‌بیمه‌مشترک‌ نزد چند شرکت‌ بیمه‌، به‌طوری‌ که‌ مجموع‌ارزش‌بیمه‌ شده‌ نزد همه‌ شرکتها از ارزش‌ واقعی‌ تجاوز نکندمجازاست‌. این‌ اصل‌ مخصوص‌ بیمه‌های‌ اموال‌ ومسؤولیت‌است‌ و در بیمه‌های‌ اشخاص‌ می‌توان‌ چندبیمه‌نامه‌ باسرمایه‌های‌ دلخواه‌ ، خریداری‌ کرد و ازمزایای‌آنها بهره‌مندشد.

چه ‌کسی‌ می‌تواند خریدار بیمه‌ باشد ؟
کسانی‌ که‌ مالک‌ یااستفاده‌ کننده‌ از اموال‌ منقول‌ و غیرمنقول‌باشند یا قانون مسؤولیت‌ خسارت‌ وارده‌ به‌ اشخاص‌ ثالث ‌را برعهده‌ داشته ‌باشند، دارای‌ نفع‌ بیمه‌پذیر هستند ومی‌توانند آن‌ را بیمه‌کنند.

تناوب خطرها به چه مفهوم است؟
خطر بیمه‌پذیر باید احتمالی و اتفاقی و درجه احتمال آن بین صفر و یک باشد. تناوب خطر به مفهوم آن است که خطر با شدت و ضعف مختلف در گذشته اتفاق افتاده و احتمال وقوع آن در آینده نیز وجود دارد. بیمه‌گر با استفاده از تجربه خسارتی گذشته می‌تواند بر اساس احتمالات، خطر را ارزیابی و حق بیمه را محاسبه کند. احتمالی بودن خطر مربوط به بیمه‌گذار است. بیمه‌گر با تکیه بر حساب احتمالات، نه تنها می‌‌تواند تعداد خسارت‌ها بلکه میزان خسارتی را که در یک دوره باید بپردازد برآورده کند. تنها چیزی که برای بیمه‌گر قابل پیش‌بینی نیست این است که اگر از میان جامعه بیمه‌گذاران قرار است به ۵۰ مورد خسارت پرداخت شود، مشخص نیست که این ۵۰ مورد چه کسانی هستند.

بیمه گر
بيمه‌گر شخصي است حقوقي كه در مقابل دريافت حق‌بيمه از بيمه‌گذار تعهد جبران خسارت و يا پرداخت وجه معيني را در صورت وقوع حادثه به عهده مي‌گيرد. در ماده 31 قانون تاسيس بيمه مركزي ايران و بيمه‌گري چنين آمده: عمليات بيمه در ايران به وسيله شركت‌هاي سهامي عام ايراني كه كليه سهام آنها با نام بوده و با رعايت قانون و طبق قانون تجارت به ثبت رسيده باشند انجام خواهد گرفت.

گروه‌های چهارگانه ریسک کدامند و کدام گروه‌ها از نظر بیمه‌گر برای بیمه شدن مطلوبند؟
گروه‌های چهارگانه ریسک عبارتند از:
الف‌ـ ریسک‌های خالص یا ایستا که ممکن است نتیجه زیان یا عدم زیان داشته باشند، مانند داشتن اتومبیل در مقابل خطرهائی که برای دارنده آن ممکن است داشته باشد.
ب‌ـ ریسک‌های سوداگرانه‌ که ممکن است نتیجه زیان، عدم زیان و سود داشته باشند، نظیر فعالیت‌های تجاری یعنی خرید به نیت فروش با قیمت بالاتر
پ‌ـ ریسک‌های خاص که در صورت تحقق، آثار زیانبار آن‌ها دامن‌گیر گروه کوچکی می‌شود؛ برای مثال، افراد یک خانواده که با اتومبیل خود مسافرت می‌کنند و در راه تصادف می‌کنند یا اعضاء تیم کوهنوردی که مسیر را گم می‌کنند و در طوفان و سرما گرفتار می‌شوند و صدمه می‌بینند.
ت‌ـ ریسک‌های عام که در صورت وقوع، تعداد زیادی از مردم دچار خسارت جانی و مالی می‌شوند، مانند سیل، طوفان و زلزله
از میان این چهارگروه ، ریسک‌های خالص و خاص بیمه‌پذیرند و مطلوب بیمه‌گران. زیرا که بیمه‌گر، هم قادر به ارزیابی عوامل مؤثر در ریسک و تعیین حق بیمه است و هم، خسارت وارده در حدی است که بیمه‌گر با مدیریت مالی مناسب توان پرداخت آنها را دارد. از دو گروه باقی‌مانده، ریسک‌های سوداگرانه به‌هیچ‌وجه بیمه‌پذیر نیست، زیرا که عوامل مؤثر در ریسک‌های سواگرانه به‌قدری زیاد است که بیمه‌گر نمی‌تواند همه این عوامل را شناسائی کند، در نتیجه نمی‌‌تواند حق بیمه را تعیین کند. گروه ریسک‌های عام از نظر فنی، بیمه‌پذیرند، زیرا که با تواتر و شدت متغیر، در گذشته اتفاق افتاده و احتمال وقوع آنها در آینده نیز وجود دارد. از آن‌جا که دامنه خسارت‌ها در صورت تحقق ریسک عام بسیار وسیع و توان مالی بیمه‌گر نیز برای پرداخت خسارت، محدود است، مطلوبیت چندانی برای بیمه‌گر ندارد. افزودن بر این، تأمین پوشش بیمه اتکائی برای ریسک‌های عام نیز به سادگی امکان‌پذیر نیست و حق بیمه سنگینی هم نیاز دارد. با این همه، بیمه‌گر تحت شرایطی خاص، برای بیمه‌گذاران متقاضی در مقابل دریافت حق بیمه مناسب آنها را بیمه می‌کند.

وظایف‌ بیمه‌گذار چیست‌ ؟
وظایف‌ بیمه‌گذار عبارت‌ است‌ از :
1-دادن‌اطلاعات‌ دقیق‌در پیشنهاد بیمه‌ تا بیمه‌گر بتواند حق‌ بیمه صحیح‌ را تعیین‌کند
2-پرداخت‌ به‌ موقع‌ حق‌ بیمه
3- حفاظت‌ از موردبیمه‌ درحد متعارف‌ بدون‌ توجه‌ به‌ وجودبیمه
4-اعلام‌تشدید خطر (اگر مورد بیمه‌ براثر عمل‌بیمه‌گذار یا عمل‌شخص‌ ثالث‌ تشدید شود مراتب‌ باید به‌اطلاع‌ بیمه‌گر برسدو با صدور الحاقیه‌ تأیید شود)
5-اعلام‌ بهنگام‌ وقوع‌حادثه‌ای‌ که‌ منجر به‌ خسارت‌ مورد تعهد بیمه‌گر است‌.

مدیریت ریسک چیست؟ و مدیر ریسک وظایف خود را چگونه انجام می‌دهد؟
مدیریت ریسک مفهومی جز پیش‌گیری و کاهش خطر نیست. وظیفه مدیر ریسک در رسیدن به هدف‌های مؤسسه، مکمکل وظایف اصلی مدیریت است تا آن موسسه بتواند با ظرفیت و توانائی‌های کامل فعالیت کند. به مفهوم دیگر، مدیریت ریسک هماهنگی و فعالیت امر تولید را در زمینه پتانسیل ریسک‌هائی که سازمان با آن روبه‌رو است مورد توجه قرار می‌دهد. مدیر ریسک وظایف خود را در چهار مرحله به شرح زیر انجام می‌دهد:
الف‌ـ شناسائی ریسک
ب‌ـ ارزیابی ریسک
پ‌ـ کنترل ریسک
ت‌ـ تأمین منبع مالی برای برخورد با ریسک

سه خطر استثنا در بیمه اموال
استثناها خطرهائی هستند که خسارت ناشی از آنها جزو تعهد بیمه‌گر نیست. در اغلب بیمه‌های اموال، سه‌خطر استثنا شده است:
1- خطر جنگ
2- خطر انرژی هسته‌ای و مواد رادیواکتیو
3- خطر برخورد امواج صوتی هواپیما‌هائی که با سرعتی فراتر از صوت پرواز می‌کنند

***مفاهیم پایه ای برنامه نویسی سی شارپ***

**Introduction to C#.Net**

سي شارپ در سال 2001 توسط شركت مايكروسافت به همراه بسته دات نت براي اولين بار مطرح و ارائه شد. كه بعد ها به عنوان يك زبان برنامه نويسي استاندارد توسط ECMA و ISO مورد تائيد قرار گرفت.
زبان برنامه نويسي سي شارپ توسط تيمي به مديريت [اندرز هايلزبرگ](http://csharptuning.blogfa.com/%C3%98%C2%A7%C3%99%C2%86%C3%98%C2%AF%C3%98%C2%B1%C3%98%C2%B2%20%C3%99%C2%87%C3%98%C2%A7%C3%99%C2%8A%C3%99%C2%84%C3%98%C2%B2%C3%98%C2%A8%C3%98%C2%B1%C3%9A%C2%AF) كه قبلا تجربه ارائه زبان هاي برنامه نويسي موفقي همچون توربو پاسكال و دلفي رو داشت ايجاد شد. هايلزبرگ سي شارپ را يك زبان برنامه نويسي شي گراء كه از زبان هاي برنامه نويسي بزرگ (همچون دلفي , جاوا و اسمال تاك) تاثير پذيري داشته معرفي مي كنه. Syntax زبان سي شارپ شبيه به ++C مي باشد و البته تفاوت هاي بسياري هم داره.

دات نت فريم ورك (NET Framework) يه بسته نرم افزاري شامل راه حل بسياري از نيازهاي توسعه نرم افزاري (Base Class Library) و همچنين امكان توسعه و اجراي برنامه هاي توليد شده براي اين فريم ورك مي باشد. برنامه هاي توليد شده براي دات نت فريم ورك توسط Common Language Runtime اجراء مي شوند.
**CLR** يا همان Common Language Runtime سرويس هايي مهمي از قبيل Garbage Collection , Memory Managment و Exception Handling رو ارائه مي دهد.

**Garbage Collection**: در زبان برنامه نويسي سي شارپ ديگر نگراني از جهت مديريت اشياء در حافظه وجود ندارد به اين معني كه وقتي شما شيء ايجاد مي كنيد بعد از اينكه ديگر از آن شي ء استفاده نكرديد آن شيء به طور اتوماتيك از حافظه پاك خواهد شد. اين كار توسط GC يا همان Garbage Collector انجام مي شود.
روش كار GC به اين ترتيب است كه تا موقعي كه Referenceي به يك object وجود داشته باشد آن شيء در حافظه باقي خواهد ماند اما در صورتيكه Referenceي به آن شيء وجود نداشته باشد بعد از يك بازه زماني نامشخص آن شيء به صورت اتوماتيك از حافظه پاك خواهد شد. در مورد نحوه عملكردش بعدا بيشتر توضيح ميدم.
يه نكته ديگري كه بايد بهش اشاره كنم اينه كه برخلاف زبان هاي برنامه نويسي ديگر وقتي كد شما در زبان برنامه نويسي سي شارپ (يا هريك از زبان هاي ديگر دات نت) كامپايل مي شود به يك زبان ديگري به نام Intermediate Language يا همان IL تبديل مي شود. و در موقعي كه درخواست براي اجراء آن داده مي شود توسط يك مكانيزمي به نام Just In Time Compiler كه در CLR موجود است به زبان خاص آن ماشين تبديل شده و اجراء مي گردد.



**كلاس(Class)**: در واقع يك ايده (Concept) يا ذهنيت مي باشد. مثلا يك نجار چه ذهنيتي نصبت به يك "ميز" دارد؟ خوب مسلما يه صفحه , چهار پايه و .... اين ذهنيتي است كه يك نجار نسبت به "ميز" دارد. اين دقيقا مفهوم كلاس است. يعني تا موقعي كه آن ميز ساخته نشه فقط يك ذهنيت يا در واقع كلاس است.
**شيء (Object):** اشياء موجوديت هايي هستند كه از كلاس ها (ذهنيت) ساخته مي شوند. باز براي مثال همان نجار وقتي ذهنيتش رو به واقعيت تبديل مي كنه در واقع يك ميز مي سازه يا يك Object (شيء) ايجاد كرده است.
حالا اگر يكم دقت كنيم مي بينيم كه توي دنياي واقعي هم همينطوره. مفهوم انسان يك كلاس است و وقتي يك نفر متولد مي شود يك شيء از آن كلاس داريم.

باز وقتي بيشتر دقت مي كنيم مي بينيم كه همه كلاس ها (ذهنيت ها) مي تونن يكسري مشخصات داشته باشند. مثلا انسان. مسلما هر انساني اسم داره , سن داره , رنگ مو , رنگ پوست و ... اين اطلاعات state يا خصوصيات هر فرد هستند. و البته هر انساني يكسري توانمندي داره يعني مي تونه يكسري كار انجام بده. مثلا راه بره , حرف بزنه , گوش كنه , بنويسه , .... اين ها رفتار ها يا behavoir هر فرد هستند.

حالا سوالي كه پيش مياد اينه كه آيا ميشه دو نفر با خصوصيات مشترك و رفتار مشترك رو يكي دونست؟
علي احمدي 33 ساله - علي احمدي 33 ساله؟

آيا به نظر شما اين دو يك نفر هستند!؟!

خير. پس هميشه يك چيز وجود داره كه ما ميتونيم دو نفر رو از هم تميز بديم: ما به اون مي گيم Identity (وسيله شناسايي). براي اين دو مي تونه شماره شناسنامشون باشه يا كد ملي يا ...

حالا به نظر شما براي كامپيوتر دو موجود كه داراي اطلاعات و رفتار هاي كاملا مشترك هستند چطور از هم تشخيص داده مي شوند؟ جواب محل قرار گيريشان در حافظه است.

**كلاس هاي پايه دات نت - dot Net Base Class Library**

بعد از اينكه مفهوم كلاس و شيء رو متوجه شديم. براي اطلاعتون بايد بگم كه دات نت فريم ورك حدود 2500 كلاس داره كه قبلا براي شما نوشته شده و شما بدون اينكه احتياجي به نوشتنشون داشته باشين خيلي راحت مي تونين از اين 2500 كلاس موجود استفاده كنين

**Namespace چيست؟**اگر يك مقدار فكر كنيم مي بينيم كه پيدا كردن يك كلاس بين 2500 كلاس پايه و احتمالا كلاس هايي كه خودتون نام گذاري شون مي كنين كار سختي خواهد بود!
Namespace ها در حقيقت اين امكان رو بشما مي دهند كه بتوانيد كلاس هاي خودتون رو دسته بندي كنين و هر كلاس رو داخل دسته مورد نظر خودشون قرار بدين. به عنوان مثال ما كلاسي داريم به نام SqlConnection و همينطور كلاس ديگري به نام SqlDataAdapter و .... براي همين يك Namespace به نام System.Data.SqlClient ايجاد شده كه كلاس هايي كه مربوط به System و كار با داده ها (Data) و در نهايت مختص Sql Server هستند رو داخل اين namespace قرار مي دهيم. توي اين تصوير تعدادي از namespace هاي عمومي دات نت نمايش داده شده اند.



اين نكته رو هم بياد داشته باشين كه اسم كامل يك كلاس شامل اسم به همراه نام namespace آن است:
System.Data.SqlClient.SqlConnection اسم كامل ( Qualified Name) كلاس SqlConnection مي باشد.

**using**
اما اگر قرار باشد براي استفاده از يك كلاس هميشه اسم كاملش رو بنويسيم كار ما خيلي سخت مي شود. براي همين شما ميتوانيد با يكبار نوشتن اسم Namespace آن كلاس (يا كلاس هايي كه مي خواهيد استفاده كنين) با استفاده از يك keyword به نام using از تكرار آن جلوگيري كنين.

به عنوان مثال من در يك مثال مي خواهم ده ها بار از كلاس Console و دستوراتش استفاده كنم. براي همين بالاي كدم يك بار using System; رو ذكر مي كنم تا از نوشتن كلمه System براي دفعات مكرر جلوگيري كنم:

;using System

برنامه نويسان سي شارپ دات نت معمولا از Visual Studio .NET براي توليد كد استفاده مي كنند. هرچند كه در اين مورد شما هيچ محدوديتي ندارين و مي تونين از هر ابزاري حتي Notepad براي توليد كد استفاده كنين.

**Visual Studio .NET 2005**نسخه 8 Visual Studio شركت مايكروسافت كه خيلي راحت مي تونين از بازار تهيه كنين و خوشبختانه (يا شايد هم متاسفانه) دو سه هزار تومان بيشتر قيمت نداره. البته به دليل قيمتي زيادي كه در بيرون از ايران داره مايكروسافت يك نسخه مجاني به نام Visual Studio 2005 Express Edition رو هم ارائه مي كنه كه مي تونين از لينك زير دانلودش كنين.
[Visual Studio 2005 Express Edition](http://www.microsoft.com/express/2005/)
**!** من توي توضيحاتم از نسخه Professional استفاده خواهم كرد.

بعد از اينكه [مراحل نصب](http://ramezanpour.spaces.live.com/blog/cns%21CE5255FECBD193D%21218.entry) تمام شد. از منوي Start --> All Programm --> Microsoft Visual Studio 2005 روي Microsoft Visual Studio 2005 كليك كنين تا پنجره مقابل براتون باز بشه. اين صفحه Start Page مي باشد.

اگر دقت كنين سمت چپ بالا دو تا گزينه Open و Create كه امكان ايجاد يا باز كردن پروژه يا وب سايت رو به شما مي دهد دارين. روي گزينه Create Project كليك كنين(مي تونين اين كار رو با استفاده از كليد هاي **Shift + Ctrl + N** هم انجام بدين. بهتون پيشنهاد مي كنم براي سريع تر شدن كارتون Shortcut ها رو ياد بگيرين.

در پنجره مقابل روي گزينه Visual CSharp كليك كنين تا انواع پروژه هايي كه مي شه با زبان سي شارپ توليد كرد رو بينين. در ابتدا دوره ما با پيروي از اصول مايكروسافت مثال هايمان در محيط Console Application كه يك محيطي شبيه با DOS و بدون طراحي UI مي باشد شروع خواهيم كرد. در ادامه وارد Windows و ... مي شويم.

در قسمت پائين صفحه داخل قسمت Name نام پروژه خود رو بنويسين و در قسمت Location مسير پروژه خود رو مشخص كنين و در نهايت اسم Solution رو وارد نمائيد. به صورت پيش فرض اسم Solution همان اسم پروژه شماست. دقت كنين كه در دات نت براي هر پروژه يك Folder ايجاد خواهد شد. البته در صورتيكه شما گزينه Create directory for solution رو هم تيك بزنين يك Folder هم براي Solution ايجاد خواهد شد.

**Solution چيست؟**
Solution يا به معناي فارسي راه حل بالاترین سطح دسته بندي در محيط ويژوال استوديو دات نت است. به اين معني كه يك Solution مي توانيد شما يك يا چند پروژه و هم اين طور فايل باشد. اصولا وظيفه Solution نگهداري اطلاعات روابط بين پروژه هاست كه يك فايل با پسوند sln مي باشد. و داراي تاثير در خروجي پروژه نخواهد بود. در ادامه دوره بيشتر در اين مورد توضيح خواهم داد.

در مقابل Project يا همان پروژه ها داراي خروجي مستقيم با توجه به نوعشان هستند. به عنوان مثال Console Application ها داراي خروجي با پسوند exe به معناي executable يا همان فايل هاي قابل اجراء مي باشند. در دات نت ما به خروجي هر پروژه فارغ از اينكه چه نوع فايلي است (exe يا dll) اسمبلي - Assembly مي گويم.

بعد از اينكه كليد OK رو زديد پروژه شما ايجاد مي شود و براي شما يك فايل به نام Programm.cs رو باز مي نمايد. اين فايل به صورت پيش فرض محلي است كه نرم افزار شما از داخل آن شروع خواهد شد. حالا فقط براي اينكه شروع كرده باشيم داخل اين فايل و مطابق شكل زير شروع به تايپ كردن نمائيد:
**Console.WriteLine("Hello World - C# Tuning!");**

و در نهايت براي اينكه نرم افزارتون رو بتونين تست كنين كافيه كه كليد Ctrl + F5 رو بزنين تا اين نتيجه رو بگيرين:
. در حقيقت كلاس Console داراي يك رفتار (Method) مي باشد كه امكان چاپ بر روي صفحه رو به ما مي دهد. اين متد WriteLine مي باشد.

پس اگر شما هر چيز ديگري داخل پرانتز و بين " " قرار دهيد همان را براي شما چاپ خواهد كرد.

**Method چيست؟**
در واقع Method ها همان رفتار هايي هستند كه ما از كلاس ها انتظار داريم. در مثال بالا نوشتن بر روي تصوير رو مي توانيم با استفاده از متد WriteLine بر روي كلاس Console استفاده كنيم.

**ايجاد متغيير ها در سي شارپ - Variables in CSharp**

براي ايجاد يك متغيير در سي شارپ بايد ابتدا نوع داده اي آن و سپس نام متغيير را وارد نمائيم و در انتها ; را تايپ كنيم:
کد:
int MyNumber = 1000;
Console.WriteLine(MyNumber);
در مثال بالا از نوع داده اي **int** كه يك نوع داده اي عددي است استفاده شده است. پس مي توانيم مقادير عددي رو داخل اين متغيير قرار دهيم. و مثل كد بالا با استفاده از WriteLine چاپش كنيم.

**نوع هاي داده اي در سي شارپ - C# DataTypes**

نوع هاي داده اي رو مي توان بسته به محل قرار گيريشان در حافظه به سه دسته تقسيم كرد:

Values Types

Reference Types

Pointer Types

اگرحافظه رو به دو قسمت Stack و Heap تقسيم كنيم. مقادير تمامي متغيير هاي نوع اول در حافظه Stack قرار مي گيرد و به همين جهت داراي رفتارهاي خاصي مي شود كه بيشتر توضيح خواهم داد. مقادير متغيير هاي نوع دوم در حافظه Heap قرار مي گيرند. در مورد Poiter Type ها صحبت نخواهيم كرد.

**Value Types:** همان طور كه گفتيم مهمترين خاصيت اين نوع متغيير ها قرارگيري مقادير آن ها در حافظه Stack مي باشد كه به همين دليل رفتارهاي خاصي خواهند داشت. اين نوع متغيير ها شامل : Primitive Types يا همان نوع هاي بدوي - Enum ها و Struct ها مي باشند.

Numeric Types , Boolean , Char نوع هايي هستند كه به آن ها Primitive Types مي گوييم. نوع هاي عددي شامل : int , double , decimal , long , short , byte و ... كه مقادير مختلفي از اعداد رو داخل خوشان نگه مي دارند.
Boolean ها متغيير هايي هستند كه فقط مقدار True و False به معني مثبت يا منفي رو داخل خوشان نگه مي دارند. و در نهايت Char ها همان كاراكتر ها هستند.
کد:
bool married = false;
int myNumber = 1000;
char c = 'c';
! دقت داشته باشين كه براي مقدار دهي char از ' يا همان Single Quotation استفاده كرديم. و براي bool فقط مقدار true يا false امكان پذير است.

گفتيم كه **ValueType** ها داراي رفتار خاصي هستند. اما چه رفتاري؟
وقتی یک متغییر از یکی از نوع های Value Type مثل int ایجاد می کنیم مقدار آن داخل حافظه Stack قرار می گیرد و وقتی از آن متغییر یک کپی میگیریم مقدار متغییر قبلی داخل متغییر جدید کپی می شود.
یعنی اگر مقدار متغییر دوم رو عوض کنیم تاثیری روی مقدار متغییر اول نخواهد گذاشت. این رفتار در مورد تمامی ValueType ها صدق می نماید:

وقتي مقدار متغيير دوم رو تغيير دهيم مقدار متغيير اول دست نخورده باقي ماند.

**Enumerations**فرض بفرمائيد كه شما مي خواهيد يك كلاس تعريف كنيد به نام انسان. وقتي خواص انسان رو تحت بررسي دارين به جنسيت مي رسيد. حالا مي خواهيم روي كلاس انسان يك متغيير براي نشان دادن جنسيت تعريف كنيم. به نظر شما جنسيت رو از چه نوعي بايد در نظر گرفت؟ bool يا int؟
اگر bool در نظر گرفتيم false به معني مرد خواهد بود يا true؟
اگر int در نظر گرفتيم چه عددي بيانگر مرد و چه عددي بيانگر زن مي باشد؟ و اگر كاربر شما عددي به غير اعداد انتخابي شما وارد كرد, چه طور؟

گاهي اوقات در توسعه نرم افزار ها ما به جنس (Type)هايي نياز داريم كه بتوانيم مقدار شان را محدود كنيم. مثلا همين جنسيت. با استفاده از Enumeration ها مي توانيم مقدار متغيير جنسيت رو به مرد يا زن محدود كنيم.
حالا مي بينيم كه چطور با استفاده از enum مي توانيم مقدار دهي يك متغيير را به مقادير محدودي نسبت دهيم. در دات هم مثال هاي متعددي از استفاده از enum ها وجود دارد. مثلا يك enum به نام Keys تمامي كليد هاي صفحه كليد رو در خود جاي داده است. يا enumي با نام ConnectionState وضعيت هاي امكان پذير يك Connection رو نشان مي دهد.

**كلاس ها - Classes**
قبل از اين تعريف كلاس رو با هم بررسي كرديم. اما حالا چطور مي توان يك كلاس توليد كرد. فرض بفرمائيد مفهومي مثل انسان رو مي خواهيم در غالب يك كلاس تعريف كنيم.

ابتدا يك پروژه جديد از نوع Console Application با يك نام دلخواه در مسير دلخواه تان ايجاد كنين. سپس يك فايل با نام Prgoramm.cs خواهيد داشت كه در مثال قبلي ديده بوديم. حالا بايد يك كلاس جديد به پروژه اضافه كنيم. براي اين كار بايد از منوي View گزينه Solution Explorer رو انتخاب كنيم يا مي توانيم از كليد هاي Alt + Ctrl + L استفاده نمائيم.

Solution Explorer در واقع يك نمايش مبنتي بر فايل از پروژه يا Solution شماست كه به صورت يك درخت واره (TreeView) مي باشد. راس آن Solution شماست و سپس پروژه و Properties و Reference و در نهايت كلاس ها و آيتم هاي ديگر پروژه شماست. روي Project يعني دومين آيتم از اين درخت واره راست كليك كرده و گزينه Add و بعد New Item رو كليك مي نمائيم. در پنجره اي كه باز مي شود يك Class را انتخاب و در قسمت پايين اسم فايل رو مشخص مي نمائيم. دقت كنين كه پسوند فايل هاي سي شارپ cs مي باشد. (براي اين مثال نام Person رو تايپ نمائيد.)
**!** نكته اي كه لازمه در نامگذاري فايل ها متدها و ... رعايت كنين اين است كه نام گذاري كلاس ها بهتر است به ترتيبي باشد كه حرف اول هر كلمه با حروف بزرگ (Upper Case) و ما بقي با حروف كوچك باشد. به عنوان مثال كلاس **Person** كه P با حروف بزرگ و ما بقي كوچك مي باشد. يا كلاس **SqlConnection** حروف اول هر كلمه يعني S و C با Upper case نوشته مي شوند.

وقتي اسم رو تايپ كردين و گزينه Add رو زديد. سپس يك فايل به Solution Explorer اضافه خواهد شد و همزمان براي شما نمايش داده مي شود. متن داخل فايل به اين ترتيب است:

کد:

***using System;******using System.Collections.Generic;******using System.Text;******namespace CSharpTuning.Samples******{******class Person******{******}******}***

قبل از اين در مورد كلمه using و namespace صحبت كرده ايم. همانطور كه مي دانيد يك namespace يك دسته بندي براي كلاس هاست. حالا ما داريم يك namespace به نام CSharpTuning.Samples براي مثال هاي اين وبلاگ توليد مي كنيم و كلاس Person را داخل اين namespace قرار مي دهيم. دقت كنين كه هر namespace داراي يك block از كد مي باشد كه با استفاده از علامت } شروع و بعد با { پايان مي يابد و هر آنچه كه داخل اين علامت باشد داخل آن namespace قرار خواهد گرفت.

**ايجاد كلاس در سي شارپ**براي ايجاد يك كلاس در سي شارپ شما بايد داخل يك namespace از كلمه كليدي class و سپس نام كلاس استفاده كنين. مثال با براي ايجاد كلاس Person داخل namespaceي به نام CSharptuning.Samples جمله فوق يعني class Person‍ رو به همراه يك block از كد براي اين كلاس ايجاد مي كنيم. دقت كنين كه شما تا به اينجا دو علامت } و دو علامت ‍{ ديده ايد يعني شما دو block از كد داريد كه يكي براي namespace و ديگري براي كلاس Person مي باشد. هر آنچه كه داخل {} علامت مربوط به كلاس Person قرار گيرد متعلق به كلاس Person است.

**فيلد ها - Fileds در سي شارپ**
تا ايجاي كار كلاس ما هنوز هيچ عضوي ندارد اما ما مي توانيم از اين كلاس استفاده كنيم. حالا بايد داخل اين كلاس اطلاعات مورد نظرمان را تعريف كنيم. من مي خواهم Field هاي Name و Age رو براي اين كلاس تعريف كنم. به اين معنا كه هر انسان (Person) داراي نام و سن مي باشد.

براي تعريف يك Field با داخل block كلاس مورد نظر ابتدا كلمه public و سپس نوع داده اي و سپس نام متغيير را وارد نمائيم. پس من داخل block كلاس Person اين دو جمله را تايپ مي نمايم.

کد:

***public string Name;
public int Age;***

پس تا اينجا كلاس ما بايد به اين شكل باشد:

حالا مي توانيم از اين كلاس شي ء بسازيم و به اشيايي كه ايجاد كرديم مقادير Name و Age رو ست كنيم.

**Ado.net**

به مجموعه کامپوننت هایی که برای دسترسی به داده های یک بانک اطلاعاتی در .net استفاده می شود ado.net گفته می شود ado.net برای دسترسی به داده ها از معماری غیر متصل استفاده می کند . معماری غیر متصل به این معناست که ابتدا برنامه به موتور بانک اطلاعاتی مورد نظر متصل شده ، داده های مورد نیاز خود را از بانک در یافت کرده و انها را در حافظه ی کامپیوتر ذخیره می کند . سپس برنامه از بانک اطلاعاتی قطع می شود و تغییرات مورد نظر خود را در حافظه انجام می دهد . هر زمان که لازم باشد تغییرات ایجاد شده در بانک اطلاعاتی ذخیره شوند ، برنامه یک اتصال جدید را به بانک اطلاعاتی ایجاد کرده و از طریق این اتصال تغییراتی را که در داده ها ایجاد کرده بود در جداول اصلی نیز ایجاد می کند . کامپوننت اصلی که داده ها ی در یافتی از بانک اطلاعاتی را درحافظه نگهداری می کند ، کامپوننت dataset است . این کامپوننت خود از چند کامپوننت دیگر مانند اشیایی از نوع datatable تشکیل شده است . بعد از اینکه داده ها در حافظه قرار گرفتند می توانید بین انها جستجو کنید ، دستورات select مورد نظر خود را روی انها اجرا کرده و انها را به این وسیله فیلتر کنید و یا تغییراتی را در این داده ها ایجاد کنید .

استفاده از معماری غیر همزمان مزایای زیادی دارد که مهمترین ان افزایش توانایی برنامه در سرویس دادن به چندین کاربر به صورت همزمان است . به عبارت دیگر می توانیم بگوییم با استفتده از این روش می توانیم تعداد افرادی که می توانند به صورت همزمان از برنامه استفاده کنند را از ده ها نفر به صدها نفر افزایش دهیم . دلیل این مورد نیز در اینست که در این روش برنامه ها فقط در مواقع مورد نیاز به بانک اطلاعاتی متصل می شوند و بعد از اجرای وظایف لازم اتصال خود را قطع می کنند ، به این ترتیب منلبع استفاده شده برای اتصال انها به بانک اطلاعاتی نیز ازاد شده و در اختیار کاربران دیگر قرار می گیرد

**فضای نام data :**

کلاسهای اصلی ado .net در فضای نام system.data قرار دارند . این فضای نام خود شامل چند فضای نام دیگر است که مهمترین هنها عبارتند از System.Data.OleDb و .System.Data.SqlClient فضای نام .System.Data.SqlClient شامل کلاسهایی است که برای دسترسی به بانک های اطلاعاتی ایجاد شده به وسیله SQL Server به کار می رود .فضای نام System.Data.OleDb نیز حاوی کلاسهایی است که برای دسترسی به بانک اطلاعاتی از نوع OLE ( مانند بانکهای اطلاعاتی access ) مورد استفاده قرار می گیرد . در فضای نام System.Data دو فضای نام دیگر نیز وجود دارد که عبارتند از .System.Data.Odbc \_ System.Data.OracleClient فضای نام OracleClien برای دسترسی به بانک های اطلاعاتی ایجاد شده به وسیله موتور بانک اطلاعاتی Oracle مورد استفاده قرار می گیرد .

کلاسهای موجود در این فضای نام نیز همانند کلاسهای موجود در فضای نام SqlClient برای دسترسی به بانک های اطلاعاتی از نوع oracle بهینه سازی شده اند . فضای نام odbc نیز حاوی کلاسهایی است که برای دسترسی به بانک های اطلاعاتی قدیمی از نوع ODBC که تکنولو ژی oledb را پشتیبانی نمی کنند ایجاد شده است .

کلاسهای موجود در این سرویس دهنده های بانک اطلاعاتی به حدی زیاد هستند که نمی توانیم تمام انها را در این قسمت معرفی کنیم با این وجود با تعدادی از مهمترین انها اشنا می شویم :

SqlConnection \_

SqlCommand \_

SqlDataAdapter \_

SqlParameter\_

به خاطر داشته باشید که این کلاسها فقط برای ارتباط زبا بانک اطلاعاتی sql server v مورد استفاده قرار می گیرند و نیز برای استفتده از این کلاسها باید فضای نام System.Data.SqlClient را با استفاده از راهنمای using به برنامه اضافه کرد تا لازم نباشد هر بار نام کامل انها را وارد کنیم .

using System.Data.SqlClient;

همچنین برای استفاده از کلاسهای پایه ای ado.net مانند DataView , و یا DataSet باید فضای نام system.data را نیز به برنامه اضافه کنیم بنابراین در ابتدای برنامه خود دستور زیر را وارد می کنیم

using System.Data;

***توضیحات برنامه :***

 در ابتدا طبق سخن گفته شده در قسمت قبل به ابتدای برنامه فضاهای نام مورد نیاز اضافه می گردند

انگاه برای به کار بردن دستور sqlconnection در تمامی فرم ها و نیز جلوگیری از تکرار دستور و اختصار در کد و زمان اجرای برنامه از یک class به نام class1 استفاده می کنیم و پارامتری را به صورت زیر تعریف نموده که در هر بار دستور connection مورد استفاده قرار گیرد :

public static string connection\_string = "Data Source=ZAHRA-VAIO\\SQL2008;database=bm;user id=sa;password=1; ";

**اتصال به بانک و انجام عملیات 4 گانه**

انگاه به طور مثال برای استفاه از اطلاعات جدول بیمه نامه و درج اطلاعات جدید در ان از دستورات زیر به ترتیب توضیح داده شده استفاده می شود :

try

 {

 SqlConnection objCon = new SqlConnection(Parameters.connection\_string);

 objCon.Open();

 using (SqlCommand cmd = new SqlCommand())

 {

 cmd.CommandType = CommandType.StoredProcedure;

 cmd.CommandText = "EXEC\_SQL";

 cmd.Connection = objCon;

 string SqlStr = "INSERT INTO sodor(bmgn,tsodor,tstart,tend,codsd,fk\_bmno,namanoo,moneyhag,takhfif,taahod) VALUES('" + textBox2.Text + "','" + maskedTextBox6.Text + "','" + maskedTextBox1.Text + "','" + maskedTextBox2.Text + "','" + textBox6.Text + "'," + comboBox1.SelectedValue + ",'" + textBox7.Text + "'," + maskedTextBox3.Text.Replace(",", "").Replace(" ", "") + "," + System.Convert.ToDecimal(maskedTextBox5.Text.Replace(",", "").Replace(" ", "")) + "," + maskedTextBox4.Text.Replace(",", "").Replace(" ", "") + ")";

 cmd.Parameters.Add(new SqlParameter("@SqlStr", SqlStr));

 cmd.ExecuteNonQuery();

 }

 string ss = "ایجاد رکورد با موفقیت انجام شد.";

 MessageBox.Show(ss);

 this.Hide();

 Form2 newa = new Form2();

 newa.ShowDialog(this);

 }

 catch (Exception ex)

 {

 MessageBox.Show("خطا در ایجاد رکورد");

 }

برای اینکه بتوانیم یک استثنای رخ داده را کنترل کنیم باید با استفاده از دستورات catch بلاکی ایجاد کنیم که بتواند شی پرتا ب شده برای ان استثنا را دریافت کند . اما بلاک catch به تنهایی نمی تواند ایجاد شود باید همراه با بلاک try تعریف شود برای تعریف بلاک try هم می توانیم از دستور try استفاده کنیم بدین ترتیب کدی که ممکن است موجب ایجاد استثنا شود را در یک بلاک try قرار می دهیم و سپس یک بلاک catch بعد از بلاک try می نویسیم تا در صورت بروز استثنا کنترل برنامه به این بلاک فرستاده شود

اصولا برای اینکه از اطلاعاتی select بگیریم یا عملیات insert delete update را انجام دهیم ابتدا یک connection با استفاده از objectconnection و تعریف پارامتر های ان در یک کلاس public تعریف می کنیم . انگاه از object ، sql command کمک می گیریم بدین صورت که یک شی به نام cmd از نوع sqlcommand تعریف می کنیم انگاه دستورات sql را از نوع stored procedure قرار می دهیم و در بانک اطلاعاتی sql یک stored procedure به نام EXEC\_SQL تعریف می کنیم که دارای تنها 3 خط برنامه می باشد و هدفش نیز این می باشد که یک رشته string را گرفته و ان را در داخل بانک اجرا می کند که این رشته یکی از عملیات درج ، حذف ، به روز رسانی و یا انتخاب می باشد بنابراین در کد برنامه command type را از نوع stored procedure قرار می دهیم و نیز command text را از نوع EXEC\_SQL تعریف می کنیم انگاه رشته مورد نظر را نوشته وتوسط دستور cmd.Parameters.Add به sql ارسال می کنیم انگاه یک adapter تعریف می کنیم که خود به command cmd متصل است و با استفاده از این adapter dataset ویا data table مورد نظر را پر می کنیم

اگر دستور مورد نظر select باشد از دستور fill در ادامه برنامه استفاده می کنیم تا بتوانیم اطلاعات را خوانده و مشاهده کنیم ولی اگر دستور مورد نظر insert ,updat , delete باشد از دستور cmd.ExecuteNonQuery();استفاده می کنیم که دستورات ارسال شده را در داخل بانک اجرا می کند که یک دستور اجرایی است و اطلاعاتی نمی خواند بلکه تنها وظیفه اجرای دستورات را داراست .

 **اجرای کنترل ها بر روی textbox**

برای اجرای کنترل ها و داشتن نظارت بر روی اطلاعات وارد شده در textbox ها از تریگر leave استفاده می کنیم که پس از ورود اطلاعات و به هنگام ترک textbox شروط و دستورات این تریگر را بررسی کرده و انها را اجرا می کند

به عنوان نمونه دستورات تریگر leave مربوط به maskedtextbox 1 را که تاریخ شروع را در بر دارد بررسی می کنیم :

private void maskedTextBox1\_Leave(object sender, EventArgs e)

 {

 try

 {

 if (maskedTextBox1.Text != " / /")

 {

 if (System.Convert.ToInt64(maskedTextBox1.Text.Substring(0, 4)) < 1300 || System.Convert.ToInt64(maskedTextBox1.Text.Substring(0, 4)) > 1450)

 {

 MessageBox.Show(" تاریخ باید در محدوده زمانی ما بین سالهای 1300 و 1450 قرار داشته باشد");

 maskedTextBox1.Text = "";

 maskedTextBox1.Focus();

 return;

 }

 CHECK\_DATETIME a = new CHECK\_DATETIME();

 if (a.CHECK\_DATE\_F(maskedTextBox1.Text) == false)

 {

 MessageBox.Show("خطا در ورود تاریخ");

 maskedTextBox1.Text = "";

 maskedTextBox1.Focus();

 return;

 }

 if (maskedTextBox1.Text != " / /" && maskedTextBox2.Text != " / /")

 {

 if (maskedTextBox1.Text.CompareTo(maskedTextBox2.Text) > 0)

 {

 MessageBox.Show(" تاریخ شروع باید کوچکتر ازتاریخ انقضا باشد .");

 maskedTextBox1.Text = "";

 maskedTextBox1.Focus();

 return;

 }

 }

 if (maskedTextBox1.Text != " / /" && maskedTextBox6.Text != " / /")

 {

 if (maskedTextBox1.Text.CompareTo(maskedTextBox6.Text) < 0)

 {

 MessageBox.Show("تاریخ شروع باید بزرگتر از تاریخ صدور باشد .");

 maskedTextBox1.Text = "";

 maskedTextBox1.Focus();

 return;

 }

 }

 }

 }

 catch (Exception ex)

 { }

در ابتدا اعداد وارد شده به کلاس شماره 2 پاس داده می شود و این کلاس بررسی می کند که ایا تاریخ وارد شده مجاز می باشد یا خیر و در صورت مجاز نبودن تاریخ پیغام خطا بازگردانده می شود و سپس با استفاده از دستور CompareTo مقایسه ای را ما بین دو textbox انجام میدهیم که حاصل مقایسه در این دستور سه عدد 0 و 1 و -1 می باشد .

***نتیجه گیری***

از انجایی که ثبت انواع بیمه نامه ها و نیز درج خسارت های احتمالی برای انواع بیمه نامه ها در سیستم سنتی موجبات به وجود امدن اشتباهات متعدد را فراهم اورده و نیز باعث گردیده تا عملیات پرداخت خسارت و پی گیری زمان فعالیت بیمه نامه ها با مشکلات متعدد مواجه گردد لذا استفاده از سیستم رایانه ای اشتباهات و مشکلات سیستم دستی را کاهش داده و نیز انجام انواع فعالیت ها را برای کاربران بهینه تر و اسان تر ساخته و مدیران نیز به راحتی می توانند عملیات نظارتی و همچنین عملیات اماری را به راحتی انجام دهند . و علاوه بر کاهش درصد خطا سرعت انجام عملیات گوناگون را نیز افزایش می دهد .

در نتیجه مسئولان و مدیران را در ارائه هرچه بهتر خدمات به مشتریان یاری می رساند .

استفاده از زبان c# برای ساخت این سیستم امنیت ان را بالا برده و نیز در مرحله طراحی قابلیت های جالب توجی را برای کاربر و طراح در نظر گرفته است .