

7 дәріс. Ноль мән қабылдайтын типтер. Бөлшектенген кластар және бөлшектенген әдістер. Кеңейтуші әдістер.

Дәрістің мақсаты: студенттерде ноль мән қабылдайтын және бөлшектенген типтердің, кеңейтуші әдістердің қызметі және оларды пайдалану ерекшеліктері туралы түсініктерін көрсетуге қабілет қалыптастыру.

Осы дәрісті меңгеру нәтижесінде студенттер келесі қабілеттерге ие болады:

- Ноль мән қабылдайтын типтердің қызметі және оларды пайдалану ерекшеліктері туралы түсініктерін көрсету;
- Бөлшектенген типтер мен әдістердің қызметі және оларды пайдалану ерекшеліктері туралы түсініктерін көрсету;
- Кеңейтуші әдістердің қызметі және оларды пайдалану ерекшеліктері туралы түсініктерін көрсету.

Ноль мән қабылдайтын типтер

Ағымдағы қалдықпен шот-фактуралар және Тапсырыс берушінің мекен-жайына, нөмірі, сәйкестендіру нөмірлері тапсырыс берушілердің атаулы жазба деректер базасын шотыңызда сақталады. Осындай жағдайда тапсырыс берушінің толық емес деректер нүктесі құрылды, онда мүмкін осы бір немесе бірнеше өріс инициалданған. Мысалы, жай ғана пайдаланылмаған болса және бұл жағдайда, тапсырыс беруші өнім каталогы шот-фактураның нөмірі, демек, оны өріс сұрау қажет болмайды.

Өңдеу үшін мәндерді немесе жай ғана өріс немесе өрістер қолдануға тура келді, олар өздерінің көрсетпеген бұрын пайдаланылмайтын пайдаланылады қосымша өрістер жоқ. Сөз жоқ, олай болмаған жағдайда, егер олар өздерінің маңызы бар жағдайда ғана жарамды тосылып жатыр, бірақ әрқашан осылай болады, ол орнына маңызы бар жарамсыз болып жатыр. Ал қолмен енгізу және өңдеу өрісі пайдаланылады, жоқ па, бірақ олардың барлық жағдайларда көрсететін қосымша өрістерді жарамды деген шаруа аз емес. Осы екі түрі көп қиыншылықтар ноль мән беруге мүмкіндік береді.

Ноль мән қабылдайтын типтерді қолдану негіздері

Ол ерекше ноль мән қабылдайтын типтер нұсқасы бір түрінің маңызы бар құрылым. Айқындалатын басқа базалық ноль мән қабылдайтын типтер мәндердің түрі бос мәндерге сақтауға мүмкіндік береді (бос). Демек, осындай ауқым бар сандарды ұсыну мен сипаттамаларын және оның негізгі түрі ретінде ноль мән қабылдайтын типтер. Ол қосымша мүмкіндік береді, яғни айнымалы мәнін көрсетуге маңызы осы түрдегі баптандырылмаған. System ноль мән қабылдайтын типтер объектілері болып табылады. T - түрі маңызы бар, онда nullable < T > ноль мән қабылдайтын тип болмауға тиіс.

Ноль мән қабылдайтын типтер екі тәсілмен көрсетілуі мүмкін. Біріншіден, мүмкін, аттар кеңістігіндегі түрінің нысандарын Nullable < T > белгілі бір System, әбден айқын түрде жарияланды. Мәселен, өзгермелі үлгідегі bool int және төменде келтірілген мысалда ноль мән қабылдайтын типтер құрылады.

```
System.Nullable < int > Count;
```

```
System.Nullable < bool > done;
```

Екіншіден, ноль мән қабылдайтын типтер жиі жарияланады және қысқа болуы және сондықтан көрсете отырып тәсілімен пайдаланылатын таңбаның? базалық атындағы кейін түрі. Төменде келтірілген сақталуы тиіс және одан да көп таралған түрі int bool ноль мән қабылдайтын айнымалы хабарландыру тәсілі арқылы көрсетіп жатыр.

```
int? count; bool? done;
```

Әдетте төмендегідей түрлері ноль мән қабылдайтын нысан кодында ноль мән қабылдайтын типтер құрылатын кезде қолданылады.

```
int? count = null;
```

Бұл жолда айнымалы мәні анық инициалданатын кодын count бос (бос). Қабылданған Ереже бұл толық сәйкес келеді: Айнымалы пайдалануға, оған бұрын тағайындау қажет мән. Бұл жағдайда екенін білдіреді берілетін айнымалы мәні анықталмаған.

Бөлшектенген типтер

C # құрылымының немесе екі немесе одан да көп бөліктерге бөлу 2,0 нұсқасынан бастап, сыныбы, олардың әрқайсысының сақтай отырып жеке файлда интерфейс айқындау мүмкіндігі пайда болды. Бұл кілт сөз partial мәтінмәндік көмегімен жасалады. Осы бөлімдерге бірге алғанда құрастыру кезінде барлық бағдарламалары.

Егер partial өзгерткіші бөлшектенген тип жасау үшін пайдаланылады, онда ол мынадай жалпы нысанын қабылдайды:

```
partial тип_аты түрі {/ /.
```

Класс аты білдіреді, онда тип_аты интерфейс, құрылымының немесе орайда бөлінетін бөліктерге. Әр бөлігі бөлшектенген үлгідегі partial модификаторымен бірге көрсетілуі тиіс.

Қарапайым мысал қарастырайық, үш кезеңге бөлу координаттары бар бөлшектенген класс, XY файл. Файл мазмұнын қайтарылған бірінші.

```
{partial class XY
public XY (int int a, b) {
X = a;
Y = b;
}
}
```

Бұдан әрі екінші файлдың мазмұнын керек.

```
{partial class XY
public {int X get; set; }
}
```

Файл мазмұнын, ақыры, үшінші.

```
{partial class XY
public {int Y get; set; }
}
```

Бастапқы мәтін файл класс XY көрсетіп жатыр бағдарламасын қолдану төменде келтірілген сақталуы тиіс.

```
Көрсету ішінара
// айқындау қажет.
using System;
class Test {
static {void Main (
XY xy = new XY (1, 2);
Console.WriteLine (xy.X xy. Y);
}
}
```

Пайдалану үшін класпен XY компиляцияны қосу қажет ету үшін оның барлық файлдар. Мәселен, егер XY файлдарды сыныпты ху1 деп аталады. cs, ху2. cs және ху3. cs класы бар, ал Test файлында test. Пәрмен жолында енгізу үшін жеткілікті болса, онда оның CS құрастыру хабарлайды.

```
csc test.cs ху1.cs ху2.cs ху3.cs
```

Және ең соңғы ескертпе: Жаңа C # жинақталған сыныптар ішінара жол беріледі. Бірақ оның екі жақ сәйкес болуы керек, бұл хабарландыруда түрінің тағайындауын әрбір осындай сынып қалған бөліктерінде көрсетіледі.

Бөлшектенген әдістер

Алдыңғы Бөлімде түрлендіргішті partial класы ретінде түсіндірді көмегімен ішінара түрін жасауға болады. C # және ішінара әдісін пайдалану мүмкіндігін жасау үшін деректер пайда болды, бұл 3,0 нұсқасынан бастап өзгерткіші элементінде ішінара түрі. Оның бір бөлігіндегі жарияланады, ал басқа ішінара әдісі іске асырылады. Демек, оны сатудан немесе құрылымы ішінара сыныпта түрлендіргішті көмегімен хабарландыру partial әдісін бөлу керек болады.

Әдісін частичного басты ерекшелігі мынада, оны жүзеге асыру талап етілмейді! Егер іске асырылмай отыр, онда оның барлық шақырулар басқа бөлігіндегі сыныбынан немесе құрылымын ішінара әдісі үнсіз ескермеуде. Бұл қосымша міндетті функцияларын айқындауға мүмкіндік береді, бірақ ол болмаса да, ізденіп алу қажет. Егер бұл функцияларды іске асырылмаған болса, онда олар жай ғана ескермеуде.

Төменде келтірілген, онда кеңейтілген нұсқасы алдыңғы бағдарламаның ішінара құрылады Show әдісі (). Бұл әдіс басқа әдіспен іске қосылады ShowXY (). Барлық бөлігі ыңғайлы болу үшін бір файлда ұсынылды, бірақ олар көрсеткендей, алдыңғы Бөлімде көрсетілген жекелеген фацлам сынып XY бөлінуі мүмкін.

// Көрсету ішінара әдісін қолдану.

```
using System;
{partial class XY
public XY (int int a, b) {
X = a;
Y = b;
}
Void // ішінара жариялау әдісі, partial Show ();
}
{partial class XY
public {int X get; set; }
Ішінара сатуға // partial Show () void {әдіс
Console.WriteLine (" {0}, {1}", X, Y);
}
}
{partial class XY
public {int Y get; set; }
Ішінара әдісі, public void тудыруы // {ShowXY (
Show ();
}
}
class Test {
static {void Main (
XY xy = new XY (1, 2); xy. ShowXY ();
}
}
```

Екінші сыныпты жарияланады, ал оның бір бөлігін XY Show () бір бөлігі екенін ескеріңіз әдісі іске асырылады. Бұл дегеніңіз осы әдісті іске асыруда координаттар әдісі () ішінен ShowXY Show кезде X пен Y мәндері шығарылады, онда бұл әдісін шақыру шынында да нақты іске қосылады. () бар зардаптар: Бірақ егер Show (), онда оның координаттары X пен Y мәндерінің шығару әдісін іске асыруға ShowXY әдісін шақыру ештеңеге әкел меңді. () ішінен Түсініктеме қылу.

Ішінара әдістеріне бірқатар тән мынадай шектеулер. Void түрінің мәні қайтаруы керек. Оларда түрлендіргіштерді виртуалды мүмкін емес және олар қол жеткізу мүмкін емес. Сонымен қатар оларда out параметрлерін пайдалану мүмкін емес.