**13 дәріс. Аластамаларды өзіндік генерациялау. Аластамалардың туынды кластары.**

**Дәрістің мақсаты:** студенттерде аластамалардың туынды кластарын құру ерекшеліктері туралы түсініктерін көрсетуге қабілет қалыптастыру.

Осы дәрісті меңгеру нәтижесінде студенттер келесі қабілеттерге ие болады:

- Аластамаларды өзіндік генерациялау;
- Туынды аластамалар кластрын құру.

Туынды аластама кластарын алу

Кірістірілген қомақты бөлігін қамтитын өзгеретіндігіне қарамастан барынша таралған қателерді болдырмау ерекше жағдайлардың ғана шектелмейді өңдеу бағдарламалық C # Осы қателерге. Шын мәнінде күшті жақтарын C # тәсілдің бірі болып қабылданған ерекше жағдайлардың мынада, сонымен қатар, осы тілді пайдаланушы анықтайтын ерекшеліктерді пайдалануға жол беріледі, яғни өңдеуге қойылатын тұрады кім бастап арналған программалайды #. Атап айтқанда осындай арнайы қателерді өндеу үшін пайдалануға болады, ал олар өз шығару коды құрылады өте оңай. Бұл үшін жылғы сыныпты анықтауға жеткілікті сыныбы, туынды Exception. Енді бір ғана жұмыс түрлерінің жүйесі ретінде пайдалану үшін оларды іске асыруға міндетті түрде мұндай сыныптарында бірдеңе мүлде жеткілікті ерекшеліктер.

ЕСКЕРТПЕ

Өткен жылғы сыныпты еді, өйткені бұл иерархия сыныптарын алып тастау үшін арнайы құрылған туынды ретінде Application.Exception бастапқыда ерекшеліктер қолданбалы сипаттағы құрсауланған. Бірақ енді бұны істеудің орнына Microsoft корпорациясы, ал осы жылғы сыныпты шығару алуға ұсыным жасамаса, туынды Exception. Дәл осы себеп бойынша осы кітапта бұл тәсіл және қаралады.

Сыныптар құрылатын сипаттары мен әдістері, олар үшін белгілі бір сыныпта Exception алуға және қол жетімді пайдаланушы автоматты түрде жаңартылады. Әрине, кез келген ерекшеліктер осы мүшелердің Exception құрылатын сыныптарда сынып қайта анықтауға болады.

"Қоспау" сыныбы үшін конструкторлар сүйеді, әдетте, онда барлық кезде меншікті құрылады, онда мүмкіндігінше сыныпта айқындалған Exception. Жай арнайы сыныптарында қол жеткізу қиын емес, себебі бұл үшін қолайлы тиісті сынып бере кілт сөзді Exception конструкторға пайдалана отырып, осы ерекшеліктер жеткілікті дәлелдер base. Бірақ іс жүзінде формальды түрде конструкторлар ғана беру керек, олар бағдарламада пайдаланылады.

Мысал қарастырайық, онда арнайы үлгідегі шығару бағдарламасын пайдаланылады. Мысалы, әбден жиым, индекстелетін температура шегінде - 5 27 дейін жарамды болып есептеледі. Ал егер индексі шегінен шығып, онда бұл қатені өңдеу үшін арнайы ауыспалы RangeArray жиымның сыныпта анықталды. Мұндай ауыспалы жүгінген әрбір операциядан кейін кодында массивке использовавшем RangeArray сыныбы, нарықтық және тексерді. Әрине, мұндай көзқарас "икемсіз" және насырға шабатын қателерді өңдеуге қойылатын

қосымша қателерге. Төменде келтірілген сақталуы тиіс жақсартқан сынып RangeArray нұсқадағы қателерді өңдеу шекарасын бұзу

Жиымның ілтипатты астам көмегімен орындалады және сенімді әдіспен арнайы шығархатын қиыс жағдайлар.

Қателерді өңдеу үшін пайдалануға қойылатын арнайы алып тастау / / // массивке сынышты жүгінген кезде RangeArray.

using System;

Rangearray // сыныбына арналған қиыс жағдай жасау. class RangeArrayException: Exception {

/* Барлық конструкторлар сынышты іске асыруға Exception. Жай ғана мұндай конструкторлар құрастырушы іске асырады базалық қажет.

Ешқандай қосымша әрекеттерді болдырмау сыныбы RangeArrayException болғандықтан, онда ештеңе Exception қосады сыныбы қойылатын талап етілмейді. */

public RangeArrayException (): base () {}

public RangeArrayException (string str): base (str) {}

public RangeArrayException (

string str, Exception inner): base (str, inner) protected RangeArrayException ({}

System.Runtime.Serialization.SerializationInfo si,

System.Runtime.Serialization.StreamingContext sc): base (si, sc) {}

// ToStringO RangeArrayException шығару сыныбы үшін алдын ала анықтау әдісі. public override string ToStringO {return Message;

}

}

Жетілдірілген нұсқасы сынышты // RangeArray. class RangeArray {

// Жабық деректер.

int [] a; Сілтеме // базалық жиым int lowerBound; неғұрлым аз индексі int // upperBound; Ең көп // индексі

Автоматты түрде // тек оқуға арналған сипат {int Length Length сатылатын және түсінікті, public get; private set; }

Жиым // жүктелген шарт бойынша салу (int low, int high) {high public RangeArray мөлшеріне;

if (high < = low) {

New RangeArrayException throw ("жоғарғы жол астылық аз емес.");

}

a = new int [high - low];

Length = high - low;

lowerBound = low; upperBound = - high;

}

Бұл индекстеуші // сыныбына арналған RangeArray. public {int this [int index]

Бұл аксессор // get. get {

if (ok (index)) {

return a [index - lowerBound];

{} else

New RangeArrayException throw ("қате шекарасын бұзған.");

}

}

Бұл аксессор // set. set {

if (ok (index)) {

a [index - lowerBound] = value;

}

else throw new RangeArrayException ("қате шекарасын бұзған."); }

}

/ / // Логикалық true мәнін қайтаруға, егер бекітілген шектерде private bool ok (int index) орналасқан, {индексі

if (index > = lowerbound & Index < = upperbound) return TRUE; return false;

}

}

{{{Try тыс, сынып RangeArrayDemo static void Main () / / // көрсету жиым қолдану өз бетінше бастап тапсырма берған индекстеу

RangeArray ra = new RangeArray (-5, 5);

RangeArray ra2 = new RangeArray (1, 10);

i

Нысанды пайдалану ретінде // га жиым.

Console.WriteLine ("жиым га ұзындығы: және - га. Length); for (int i = - 5; i < = 5; i) ra [i] = i;

Console.Write ("жиым мазмұнын ra: "); for (int i = - 5; i < = 5; i)

Console.Write (ra [i] "");

Console.WriteLine ("\n");

```
Нысанды пайдалану ретінде // ra2 жиым.
Console.WriteLine ("жиым ұзындығы ra2: "ra2.Length); for (int i = 1; i < = 10; i) ra2 [i] = i;
Console.Write ("жиым ұзындығы ra2: "); for (int i = 1; i < = 10; i)
Console.Write (ra2 [i] "");
Console.WriteLine ("\n");
{} catch (RangeArrayException exc
Console.WriteLine (exc);
}
Ал енді өңдеуді // көрсету кейбір қателер.
Console.WriteLine ("шығару қателері шекарасын бұзған.");
Құрастырғышты try // дұрыс пайдалануға берілген {
RangeArray ra3 = new RangeArray (100 - 10); // Қате!
{} catch (RangeArrayException exc
Console.WriteLine (exc);
}
{
// Дұрыс пайдалануға берілген индексі, try
RangeArray ra3 = new RangeArray(100, -10); // Қате!
} catch (RangeArrayException exc) {
Console.WriteLine(exc);
}
// Дұрыс пайдалануға берілген индексі, try
RangeArray ra3 = new RangeArray(-2, 2);
for(int i = -2; i <= 2; i++) ra3[i] = i;
Console.Write("ra3 жиымы: ");
for (int i = -2; i <= 10; i++)
Console.Write(ra3[i] + " ");
} catch (RangeArrayException exc) {
Console.WriteLine(exc);
}
}
}
```

Осы бағдарламаны орындау кейін келесі нәтиже болып шығады.

Жиым га ұзындығы: 11

Жиым мазмұнын га: Ұзындығы - 5 4 3 2 1012345 жиым га2: 10

Жиым мазмұнын га2: 12345678910

Қате шығару шекарасын бұзу.

Жоғарғы жол астылық аз емес.

Жиым мазмұнын га3: - 2-1012 шекарасын бұзу қатесі.

Жиым шекарасын бұзған кезде қате пайда болды, сынып RangeArrayException түріндегі нысан RangeArray дайындайды. Келесі орындарға RangeArray үш сыныпта осылай болуы мүмкін: Жаңа аксессоре индекстегіш аксессоре құрастырушыда сыныпты, set және get индекстегіш RangeArray. "Қоспау" түрінің нысандарын тиіс деп ұстап алу үшін осы сияқты, мүмкін оның RangeArray құрастырылған және қол жетімді try блогын көрсетған жоғарыда нақтыланады. Қателер туралы хабарлау, енді бірі ретінде пайдалана отырып әрекет етеді, сондықтан ол RangeArray сыныбы үшін арнайы алып тастау мүмкін қателерді өндеу механизміне бағдарламасында деректер түрлерінің толық интеграцияланған кіріктірілген жаңа C #, табылатын.

Жай ғана өз дәлелдерін береді, бірақ оның орнына олар жоқ екенін ескеріңіз конструкторларды сыныпты денесіндегі ерекшеліктер RangeArrayException операторлары Exception кілт сөз base пайдалана отырып, қандай да бір класы. Бұрын ретінде түсіндірді жағдайларда фирмалардың толықтырады, бүкіл Exception конструкторларға сыныпты ерекшеліктер құру үдерісі базалық сыныпты туынды сыныбы функциялары тапсыруға болады. "Қоспау" сыныбы толықтыруға міндетті түрде бір жылғы сыныпты мұра етхатын тиіс функцияларды туынды ғой мүлде Exception.

Жоғарыда келтірілген бастап одан әрі кітап оқуға көшу көріңіз бұрын аздап поэкспериментировать ұсынады. Атап айтқанда, Түсініктеме қылу үшін алдын ала анықтау әдісін () және бақылаулармен нәтижелерімен ToString көріңіз. Бұдан басқа, қараңызшы, бұл ретте стандартты құралдарымен де хабар әдепкі құрастырғышты пайдалана отырып, адамның қолына алып тастау жасап көріңіз.