

10 дәріс. Интерфейстер. Интерфейстік сілтемелер, интерфейстік қасиеттер мен индексторлар. Интерфейстерді мұралау.

Дәрістің мақсаты: студенттерде интерфейстермен жұмыс істеу ерекшеліктері туралы түсініктерін көрсетуге қабілет қалыптастыру.

Осы дәрісті меңгеру нәтижесінде студенттер келесі қабілеттерге ие болады:

- Интерфейстерді абстрактілі кластардан ажырату;
- Интерфейсті мұралаудың ерекшеліктері туралы түсініктерін көрсету.

Объектіге бағытталған программалауда класс қызметті қалай орындау керектігін емес, нені атқару керектігін анықтау керек болады. Оның үстіне бұрын абстрақты әдісі атанған үлгі бола алады.

Демек, бірақ оны іске асыру әдісі Қайтаратын түрі айқындалады және қолтаңба әдісін берілмейді.

Ал туынды әдіс, оның әрбір сыныптағы өзінікі-дағы абстрактілі белгілі бір базалық сыныпта іске асыру қамтамасыз етілуге тиіс. Осылайша, абстрақты әдісі интерфейс, бірақ ол әдісін іске асыруды айқындайды. Әрине, абстрактілі сыныптар мен әдістері пайда болғаны белгілі, бірақ олардың негізіне салынған мүмкін принципті жақсы дамыған әрі. Жаңа кілт сөз көмегімен бөлу C # және оны іске асыру көзделген интерфейс сыныпты interface.

Абстрактілі интерфейстер синтаксистік тұрғысынан болып сыныптары. Бірақ бар интерфейске әдістерінің бірін де дене болмауы тиіс. Бұл дегеніңіз жаңа интерфейске ешқандай қатысы жоқ іске асыру беріледі. Атап айтқанда сол ғана көрсетіледі, бірақ ол онда қалай істеу керек болады. Интерфейсі сатылуы мүмкін болысымен айқындалатын болады, ол кез келген саны сынып бар. Бұдан басқа, кез келген саны бір сыныпта интерфейсдерді іске асырылуы мүмкін.

Интерфейс (яғни нақты өткізу) дененің осы интерфейске сипатталған әдістерді сыныпта іске асыру үшін берілуге тиіс. Іске асыру айқындау үшін әрбір класқа өз интерфейс бөлшектердің толық еркіндік беріледі. Демек, екі бірдей интерфейс сатылуы мүмкін сыныптарында әрқилы. Дегенмен олардың әрқайсысында бірдей әдістерін жиыны осы интерфейсіннің ұстануы тиіс. Ал оның қайда қалады, себебі осы объектілердің пайдаланылуы мүмкін кез келген осындай интерфейс объектілері осы екі менеджеріне арналған сынып коды, белгілі барлық біркелкі. C # бас интерфейстері қолдауы арқасында мүмкін полиморфизмді басты принципі толық көлемде іске асырылды: көп әдістерін бір интерфейс.

Кілт сөз interface интерфейс көмегімен жарияланады. Хабарландыру төмен оңайлатылған нысаны келтірілген интерфейс.

```
interface аты {
```

```
қайтарылатын_тип әдіс_аты1 (параметрлер_тізімі);
```

```
қайтарылатын_тип әдіс_аты2 (параметрлер_тізімі);
```

//.

```
қайтарылатын_тип әдіс_атыN (параметрлер_тізімі);
```

```
}
```

Онда аты - бұл нақты и - фейс атауы. Интерфейс әдістерін ғана пайдаланылады және оларды возвращаемый_тип хабарландыруда қолтаңба. Олар, шын мәнінде, абстрактілі әдістермен болып табылады. Жоғарыда, ешқандай интерфейске түсіндірді ретінде іске асыру мүмкін емес. Сондықтан әр сыныпта өзіне қосатын тиіс барлық әдістерді интерфейсіннің іске асырылды, бұл интерфейс. Сол интерфейске анықталмаған ашық деп есептеледі, сондықтан да көрсету керек емес әдістері өзінде оларға қатынасу анық.

Хабарландыру сыныбына арналған бірқатар мысал келтірілген сандар, генерациялайтын тізбекті интерфейс төмен.

```
public interface ISeries {
```

```
int GetNextO; void қалпына келтіру тәртібі бойынша қайтаруға
```

```
// келесі саны ();
```

```
// қайта іске қосу
```

```
void SetStart (int x); // бастапқы мәнді орнату
```

```
}
```

Интерфейске ISeries осы аты беріледі. I префиксі көрсетуге міндетті емес, бірақ бұл үшін интерфейс аты ажырата білу керек тәжірибеде қабылданған бағдарламалау интерфейсіннің атауында бір жылғы сынып бар. Сондықтан сыныптағы сатылуы мүмкін кез келген public ретінде жарияланады интерфейсі ISeries қандай болса да бағдарламалары.

Қасиеттері, сонымен қатар, басқа әдістерін интерфейсестерге көрсетуге индекстары мен оқиғалар. Мүшелері интерфейсестер деректер болмауы керек. Конструкторлар болмайды, сондай-ақ оларда деструкторы немесе операторные анықтауға болады. Бұдан басқа, static интерфейс мүшелерінің бірі ретінде жариялануы мүмкін де емес.

Интерфейстерді іске асыру

Қалай ғана белгілі болады, ол бір немесе бірнеше сыныптары интерфейсін сатылуы мүмкін. Интерфейс іске асыру үшін оның атын көрсету жеткілікті базалық Сынып атының кейін, осыған ұқсас сыныбы. Интерфейс төмен жалпы сыныпта іске асырудың нысаны келтірілген.

```
class имя_класса: имя_интерфемса {
```

```
Денесі // сынып
```

```
}
```

Онда имя_интерфейса сатылатын нақты аты - бұл интерфейс. Егер тіпті интерфейсін сыныпта іске асырылады, онда бұл толық жасалуға тиіс. Атап айтқанда, ішінара интерфейсін іске асыруға және тек бөліп-бөліп болмайды.

Сыныпта бірнеше интерфейсестерді іске асыруға жол беріледі. Бұл жағдайда барлық өткізетін интерфейсестер тізімін сыныпта үтір арқылы көрсетіледі. Мұрагер болады

базалық класы және сол уақытта бір немесе одан да көп сыныпта іске асыруға интерфейс. Мұндай жағдайда аты көрсетілуге тиіс, базалық сыныпты тізімін бөлетін үтір алдында интерфейстері.

Интерфейс, `public` ретінде іске асыратын әдістері жариялануға тиіс. Мәселе мынада, сондай-ақ осы әдістер интерфейске анықталмаған жобаланып жатыр, сондықтан оларды іске асыру ретінде өзінде ашық ашық болуы тиіс. Бұдан басқа, дәл сәйкес келуі тиіс және қолтаңба үлгісі мен сигнатураға возвращаемому интерфейс айқындау әдісін сатылатын Қайтаратын түрі көрсетілген.

Бұрын ұсынылған бағдарламалар интерфейсi іске асырылып жатқан төмен мысал келтірілген `ISeries`. Бұл бағдарламада `ByTwos` бірізді жинайтын класы құрылады, сандар, онда бірқатар кейін жасалған әрбір көп санды екіге өткен.

// `ISeries` интерфейсi іске асыруға, сынып `ByTwos`: `{Int iseries start; VAL INT;`

```
public ByTwos () {
    start = 0; val = 0;
}
public {val int GetNext () = 2; return val;
}
public void {қалпына келтіру (
    val = start;
}
public void SetStart (int x) {start = x; val = start;
```

Көріп отырғаныңыздай, `class` бойы `ByTwos` `ISeries` интерфейске айқындалатын үш әдісі іске асырылады. Тура келеді, себебі, бұл жоғары сыныпта ретінде түсіндірді интерфейсi ішінара сатуға болмайды.

Сынып коды, онда келтірілген `ByTwos` `ISeries` интерфейсi іске асырушы сынып қолдану төмен көрсетіп жатыр.

Сынып қолдану `ByTwos` // көрсету сататын интерфейс, `using System;`

```
{{static class SeriesDemo void Main (
    ByTwos ob = new ByTwos (); /
    for (int i = 0; i < 5; i)
        Console.WriteLine ("келесі саны тең" ob. GetNext ());
    Console.WriteLine ("XnСбросить"); ob. Қалпына келтіру ();
    for (int i = 0; i < 5; i)
        Console.WriteLine ("келесі саны тең" ob. GetNext ());
    Console.WriteLine ("XnНачать санынан бастап 100");
    ob.SetStart (100);
```

```

for (int i = 0; i < 5; i)
    Console.WriteLine ("келесі саны тең" ob. GetNext ());
}
}

```

Интерфейстік сілтемелерді қолдану

Айнымалы сілтеме түрін болса да бір түрлі ерсі көрінсе, бірақ жариялауға жол беріледі, яғни C # интерфейсного айнымалы мәнді сілтемелер интерфейс. Мұндай айнымалы сілтеме жасауы мүмкін кез келген объект, оның іске асыратын интерфейс. Нысан үшін оның нұсқасы осы нысанның интерфейсной сыныпта сілтеме арқылы өткізілген әдісін шақырған кезде орындалады.

Төменде келтірілген сақталуы тиіс бағдарламасын қолдану арқылы көрсетіп жатыр интерфейсной сілтемелер. Бұл бағдарламада айнымалы сілтемелер шақыру мақсатында объектілер үшін - екі сыныпты ByTwos және Primes интерфейсін әдістері пайдаланылады. Осы мысалда жиналған, айқын көрсетіледі және барлық бағдарламасының бөлігі үшін бірыңғай файл.

Көрсету интерфейсін // URL - using System;

Интерфейс, public interface ISeries // {анықтау

int GetNext (); void қалпына келтіру тәртібі бойынша қайтаруға // келесі саны (); // қайта іске қосу

void SetStart (int x); // бастапқы мәнді орнату

}

// // ISeries процестерін жүзеге асыру үшін бірқатар жүйелі түрде, онда көп санды екіге алдыңғы кейіннен сандардың әрбір // интерфейсін пайдалану өндіруге, сынып ByTwos: {Int iseries start; VAL INT;

val = start;

}

public void SetStart (int x) {start = x; val = start;

}

}

// // Интерфейсін іске асыру үшін пайдалануға ISeries өндіруге процесін жай сандар.

class Primes: {Int iseries start; VAL INT;

public Primes () {

start = 2; val = 2;

}

public int GetNextO {int i, j; bool isprime;

```

val;
for (i = val; i < 1 000 000; i) {
isprime = true; for (j = 2; i / <= j j; j) {
if ((i % j) = 0) {isprime = false; break;
}
}
if (isprime) {val = i; break;
}
}
return val;
}
public void {қалпына келтіру (
val = start;
}
public void SetStart (int x) {start = x; val = start;
}
}
{{static class SeriesDemo2 void Main (
ByTwos twoOb = new ByTwos ();
Primes primeOb = new Primes ();
ISeries ob;
for (int i = 0; i < 5; i) {
ob = twoOb;
Console.WriteLine ("келесі жұп сан сияқты" ob. GetNext ()); ob = primeOb;
Console.WriteLine ("келесі жай сан" "сияқты" ob. GetNext ());
}
}

```

Міне, осы бағдарламаның орындалуын қай нәтижеге алып келеді:

Келесі жұп сан сияқты жай сан - 4 - 5 жұп санды жай сан сияқты келесі 6 жұп сан сияқты 3
2 келесі сияқты келесі келесі жұп санды жай сан сияқты келесі жұп сан 10-ға тең 8 7 келесі
келесі жай сан сияқты сияқты сияқты келесі келесі жай сан сияқты 13 11

Main () айнымалы сілтеме жасау үшін ISeries интерфейсі ob дісінде жарияланады. Ол дегеніңіз, онда сілтеме нысан сақталуы мүмкін кез келген таптың интерфейсі іске асырушы ISeries. Бұл жағдайда ол мыналарға primeOb сыныптарын, оларда ISeries Primes ByTwos twoOb объектілері және сілтеме жасау үшін және тиісінше интерфейсі іске асырылды.

Интерфейстерді мұралау

Интерфейсін алу мүмкін бір басқа. Синтаксис бар осындай мұрагерлік интерфейстерді бірдей сынып бар. Бір кезде іске асырылады, онда барлық мүшелері екінші сыныпта интерфейс, наследующий тиіс сияқты мұрагерлік төменде келтірілген мысалда тізбекте айқындалған интерфейстерді іске асырылды.

Мысалы, мұрагерлік интерфейстерді // using System;

```
public interface IA {void Meth1 (); void Meth2 ();  
}
```

Енгізілген Meth1 Meth2 // жаңа әдістері () () және базалық интерфейсi

Ал тағы бір әдіс - туынды интерфейсi // қосылды Meth3 ().

```
public interface кеңейту: Void iA {Meth3 ();  
}
```

Осы класс // тиіс барлық әдістерді интерфейстерді іске асырылды IA және кеңейту.

```
class MyClass: {{Public void Meth1 кеңейту (  
Console.WriteLine ("іске асыруға Meth1 әдісі ().");  
}  
public void {Meth2 (  
Console.WriteLine ("іске асыруға Meth2 әдісі ().");  
}  
public void {Meth3 (  
Console.WriteLine ("іске асыруға Meth3 әдісі ().");  
}  
}
```

```
class IFExtend {  
static {void Main (  
MyClass ob = new MyClass ();  
ob.Meth1 (); ob. Meth2 (); ob. M ^ th3 ();
```

Мүдде үшін сынып бөлмесінен MyClass Meth1 әдісін () іске асыруға жойып көріңіз. Құрастыру кезінде әкеп соғатын болса, бұл қате. Бұдан бұрын, түсіндірді ретінде айқындалған, оның ішінде барлық әдістерді іске асырылды, бұл ретте сыныптағы тиіс кез келген, өткізетін интерфейсi бар және олар интерфейске другцх интерфейстерді ішінен мұраға қалдырылады..