

7. КЛАСС ҰҒЫМЫ

7.1 Класс ұғымы

C# тілі объекті-бағытталған бағдарламалау тілі болып табылады. Оның негізгі ұғымы - класс. C# тілін оқу барысында барлық мысалдарда біз класс типіндегі құрылымдарды қолданған болатынбыз. Осы бөлімде тек қана кластар қарастырылады.

C# тілінде бағдарламалауға арналған кітаптарда келтірген анықтамалардан бастайық.

Фаронов В.В. анықтамасы бойынша класс алдында class қызметтік сөзі тұратын код үзіндісі арқылы анықталады [1]: «Класс дегеніміз – деректер типі, яғни кластың нақты даналары - объекттерді дайындалатын «схема».

Павловская Т.А. [2] анықтамасы: «Класс дегеніміз кластың даналары деп аталатын нақты объекттер жиынтығының сипаттамалары мен әрекеттерін анықтайтын жалпылама ұғым».

Ескеретін жағдай, C# тілі пайда болуына дейін ОББ бар болған және класс ұғымы бұрыннан қолданылады. Әдебиетте кездескен ең қысқа анықтаманы келтірейік: «Кластар дегеніміз – бағдарлама жасаушы анықтайтын тип».

Осы анықтамада кластың өте маңызды ерекшелігі көрсетілген – ол – массивтерге, жазуларға немесе құрылымдарға қарағанда жаңа деректер типі. Бірақ, бағдарламашы анықтайтын кез келген тип класс бола бермейді. Класты анықтаған кезде екінші бір маңызды ерекшелігі - класс құрамының болуы, кластың қысқа түрде жазылуы және оның оңай есте сақталуы.

Класс дегеніміз – өрістерден, әдістерден және оқиғалардан тұратын деректер типі.

Деректер типі дегеніміз – класс данасы деп аталатын көптеген объекттердің қасиеттері мен әрекеттерін сипаттайтын семантикалық бірлік.

Семантикалық класс класс өрістері, класс әдістері мен оқиғаларының сипаттамасы деп аталатын деректер сипаттамасын ұсынады.

Кейбір авторлар модуль болып келетін кластарды жеке топтарға бөледі, мысалы, басқару элементтерінің класы. Ондай кластардың қосымша қызметтері бар. Олар жоба құрылымының жеке архитектуралық бірлігі болып келеді.

Кластың жазба пішімін қарастырайық. Класс жазбасының пішімінде class қызметті сөзінен кейін оның атауы және одан кейін фигуралы жақшаларда класс денесі жазылады. Бұл класс сипаттамасының ең қысқа құрамы болып келеді. Кластың жалпы сипаттамасы мына пішімде болады (міндетті емес элементтер квадратты жақшаларда көрсетілген):

```
[ атрибуттар ] [ спецификаторлар ]  
class кластың_атауы [ : түп тегі ]  
{ кластың_денесі } ,
```

мұнда

атрибуторлар – класс туралы қосымша мәліметті береді;

спецификаторлар – класс құрамына қол жеткізу шарттарын анықтайды.

тип тегі (родители) – базалық кластар;

класс денесі – класс элементтерінің құрамын анықтайды.

Класты жариялауда мүмкін спецификаторлар: `abstract`, `sealed` және `protected`. Олар туралы мұрагерлікті қарастырған кезде толығырақ айтылатын болады. `Private`, `public`, `static` және `internal` спецификаторлары бағдарлама үшін кластың қолжететімділігін анықтайды. `Private` спецификаторы кластың көрінуін толық жабады, ал `public` спецификаторы класты бағдарламаның кез келген үзіндісіне көрсетеді (қол жететіндікті анықтайды). Негізінде класта `internal` қол жеткізу спецификаторы болады. Класс құрылымда анықталған және онда қол жетімді болады. `Static` спецификаторы осы класқа тиісті айнымалыны (класс объектісін) құрмай-ақ класс және оның элементтерін пайдалануға мүмкіндік береді.

Барлық спецификаторларды класта немесе оның жеке мүшелерінде қолдануға болады, мысалы, өрістерде, әдістерде.

Класты сипаттау пішімінің кейбір міндетті емес элементтерін біз келесі бөлімдерде қарастыратын боламыз.

Класс дегеніміз – белгілі бір мәндермен «толтырылатын» үлгі, яғни класс типіндегі айнымалы – класс данасын әзірлеуге арналған деректер типі.

Бағдарламада түрлі мәндерді бере отырып біз кластың әр түрлі объектілерді көретін боламыз, бірақ кластың типі өзгермейді.

Класты атаулар кеңістігінің ішінде немесе басқа кластың ішінде сипаттауға болады. Соңғы жағдай бойынша класты қабаттасқан класс деп атайды.

`C#` тілінде класс сілтемелік тип болып келеді және класс объектісін компьютер жадысында орналастыру үшін `new` операторын қолдану керек.

7.2 Класс құрамы

Класс денесінде деректер, әдістер және оқиғалар өңдеуіштері болуы мүмкін. Кластың осы құрамдас бөліктерін әдетте класс элементтері деп атайды.

Кластың негізгі элементтерін және олардың қызметтерін қарастырайық:

– класс тұрақтыларында өзгермейтін мәндер;

– класс өрістері (класс айнымалыларының типтері мен атаулары);

– класс әдістері, класс деректерімен жұмыс жасауға арналған, белгілі бір атауы бар бағдарлама кодының үзіндісі;

– класс қасиеттері дегеніміз - кластарға өз өрістерінің мәндерін бір-бірімен алмасуға (оқуға немесе жазуға) мүмкіндік беретін әдістердің жиыны;

– класс конструкторы дегеніміз – класс объектілерін құруға және класс өрістеріне мәндерді меншіктеуге арналған кластың арнайы әдістері;

– класс деструкторы дегеніміз – объектке бөлінген ресурстарды босату кезінде әрекеттердің тәртібін анықтайды;

– класс оқиғалары дегеніміз – кластың пайдаланушы әрекеттеріне немесе бағдарламадағы белгілі бір өзгерістерге жауап қайтаруға көмектесетін арнайы әдістер;

– деректер типі, мысалы, тізімдер, құрылымдар, кластар, делегаттар, интерфейстер.

– индекстарлар дегеніміз – класс деректерінің элементтеріне қол жеткізу құралы;

– операциялар дегеніміз – класс объекттеріне арналған, операциялардың белгілері арқылы орындалатын арнайы әрекеттер.

Класс деректері болып константа немесе класс айнымалылары (өрістер) болуы мүмкін. Класта деректерді жариялаған кезде әдетте оған қол жеткізу спецификаторы көрсетіледі, мысалы,

```
private int a;
```

Класс деректерін жариялау кезінде оларды жалпы жазу пішімі мынандай болады:

```
[ атрибуттар ] [ спецификаторлар ]  
[ const ] тип атауы [= бастапқы_мәні].
```

Әдетте класс деректері «бағдарлама үшін жабық болады» - private спецификаторы қолданылады. Егер деректердің алдына public спецификатор жазылса, онда олар «бағдарламада» қолжетімді болады.

Еш әрекетсіз деректер мен әдістер үшін private спецификаторы қолданылады.

Объект дегеніміз – класс типіндегі айнымалы, оны құрған кезде компьютер жадында класс элементтерінің мәндері сақталатын жеке аймақ бөлінеді.

Алайда, класта барлық объекттеріне ортақ, бір данадағы статикалық элементтер болуы мүмкін. статикалық деректерді класс деректері деп жиі атайды, ал қалғандарын - класс данасының деректері, яғни объекттер.

Кластың кейбір элементтеріне (әдістеріне) және өрістеріне қол жеткізуге объект құрылғаннан кейін ғана мүмкін болады. Егер рұқсат берілген болса, оларды қолдану үшін нүкте қолданылады, мысалы, stud объектіндегі name өрісіне қол жеткізу үшін былай жазамыз: stud.name = “Иванов”;

Объект үшін кластың әдісін шақыруға болады, мысалы, stud.poisk(a);, мұнда poisk(int a) – класс әдісі, stud – құрылған объект.

Синтаксис бойынша класта ішкі класс болуы мүмкін. Осындай жағдай жиі кездеспейді. Ішкі класты өзін туындатқан сыртқы класта және оның ұрпақтарында қолдануға болады. Ішкі кластарда әдетте private немесе protected қол жеткізу модификаторы болады.

7.3 Класс әдістері

Әдіс дегеніміз – кластың деректері және әдістерімен жұмыс істеуге арналған кластың атауы бар функционалдық элемент. Әдістер класқа қолдануға болатын әрекеттер жиынын анықтайды (кластың жұмысын анықтайды). Әдіс тек бір рет сипаталады және ол кластың түрлі объектітері үшін бірнеше рет шақырылуы мүмкін.

Класс әдістерінің жалпы жазылу пішімі мына түрде болады:

```
[ атрибуттар ] [ спецификаторлар ] әдіс типі  
әдістің атауы ( [ параметрлер ] )  
    {әдістің денесі}
```

Мысалы,

```
static void Main(string[] args)  
{ }
```

Ең жиі кездесетін спецификаторлар - private, public және static.

Private спецификаторымен жарияланған кластың кез келген әдісі тек осы класс әдістеріне ғана қолжетімді болады.

Public спецификаторы әдісті бағдарламаның кез келген бөлігінде қолдануға мүмкіндік береді.

Static спецификаторы арқылы әдісті класс объектісін құрмай-ақ «класс деңгейінде» қолдануға болады. Ол өте маңызды, өйткені біз статикалық әдістерді жиі пайдаланатын боламыз.

Қосымшада класс объектісін құрмай-ақ қолжетімді әдісті класс конструкторы қамтамасыз етеді(ол объектіні құрады).

Басқа әдістерге қол жеткізу кластың объектісін құрғаннан кейін ғана мүмкін.

Егер спецификатор көрсетілмесе класс әдісінде private спецификаторы қолданылады.

Әдіс типі қосымшада анықталған кез келген типте немесе C# тілінің стандартты типінде немесе void – типсіз болуы мүмкін. Мысалы:

```
int kol(int a) { ... }  
public double sym(out float r) { ... }  
public void poisk(ref float s) { ... }  
public int funkcij( int a, out int b, params int[] c) { ... }
```

Егер әдіс типі көрсетілген болса (void типінен өзге), онда әдіс денесінің соңғы операторы болып әдіс жұмысының нәтижесін қайтаратын return операторы болады. Бұл ретте әдісті айнымалыға меншіктеу немесе операторларда өрнек ретінде пайдалану керек. Осындай әдістер функциялар деп аталады.

Егер әдістің алдында void типі көрсетілсе, онда әдіс өз жұмысының нәтижесін return операторы арқылы қайтармауы керек (әдістің денесінде return операторы болмайды). Әдетте бұл әдісті процедура деп атайды. Әдістің атауы – бағдарламашы белгілейтін идентификатор. Әдіс атауының (атының)

мағынасы оның жұмысына байланысты болуы керек, мысалы, `sum`, `max`, `poisk`, т.б.

Әдіс және бағдарлама арасында деректермен алмасу үшін әдістің параметрлері (формалды параметрлер) қолданылады. Әдетте әдістің параметрлерін әдісті «күйге келтіру» құралы деп атайды.

C# тілінде әдістердің келесі параметрлері бар:

– мәндерді анықтайтын параметрлер (мәндік параметрлер, яғни әдіс қабылдайтын кіріс параметрлер);

– шығыстық параметрлер (`out` қызметтік сөзімен белгіленеді);

– сілтемелік параметрлер (`ref` қызметтік сөзімен белгіленеді);

– массивті параметр (`params` қызметтік сөзімен белгіленеді).

Мәндерді анықтайтын параметрлерде қызметтік сөз болмайды.

Класс әдістерінің параметрлері үтірлер арқылы бөлінеді. Әдісте массив параметрі біреу және параметрлер тізімінде соңғы болуы керек.

Егер әдісте мәндерді анықтайтын параметрлер жарияланса, онда бұл әдістің кейбір айнымалылар көшірмелерін өз құзырына алғандығын көрсетеді. Әдіс осы көшірмелердің мәнін өзгерте алады, бірақ олардың түпнұсқасы (бағдарламада) өзгермеген қалыпта қалады. Әдістің жұмысы аяқталғаннан кейін мәндерді анықтайтын параметрлер компьютер жадысынан жойылады.

Әдістің шығыстық параметрлері қосымшаға нәтижелерді жеткізу үшін арналған. Әдістің денесіндегі шығыстық параметрлерге кейбір мәндер меншіктелуі тиіс, әйтпесе қосымша компиляциясы кезінде қате кеткені туралы хабар шығады.

Егер әдісте сілтемелік параметрлер жарияланған болса, онда әдіс сәйкес айнымалылардың адресін өз құзырына алады және оларды өз алгоритмі бойынша қолдана алады (жаңа мәндерді жаза және оқи алады).

Әдістегі жарияланған массив параметрі нақты айнымалылардың кез келген санымен жұмыс жасауға арналған. Сонымен қатар `params` қызметтік сөзінен кейін тұрған формалды параметр кез келген өлшемді деректер массивімен сәйкестікке келтіріледі.

Сонымен, әдіске параметрлер арқылы керекті мәліметтерді (мәндерді анықтайтын параметрлер және сілтемелік параметрлер) жіберуге болады және әдіс өз жұмысының нәтижелерін қайтара алады (шығыстық параметрлер және сілтемелік параметрлер).

Әдіс денесінде кейбір алгоритмді орындайтын бағдарлама кодының үзіндісі бар. Бұл ретте әдіс формалды параметрлермен бірге әрекеттер үлгісі ретінде қолданылады. Бағдарламада формалды параметрлердің орнына нақты айнымалылар қолданылуы керек, нақты параметрлер мен әдістің әрекеттер үлгісі нақты айнымалылар үшін қолданылады.

7.4 Объект құрылымы

Класс типіндегі айнымалы объект деп аталады. Объект айнымалы болғандықтан оған компьютер жадысынан орын бөлінеді.

Объект бойынша жадыда нақтылы не сақталатынын қарастырайық.

Класс деректерінің барлық өрістерінің мәндері сақталады.

Объектті дайындаған кезде автоматты түрде құрылатын this арнайы өрісі (сілтеме бойынша параметр) объект адресін сақтайды.

Объект және класс әдістерінің байланысы this параметрі арқылы жүзеге асады. Кластың әрбір әдісі ағымдағы объект элементтерімен жұмыс жасау үшін this параметрін тікелей қолдана алады. this ағымдағы объектке (ағымдағы уақытта бағдарлама жұмысындағы объект) үнемі сәйкес болғандықтан, класс әдістері ағымдағы объект элементтерімен жұмыс жасайды.

This нұсқаушы пайдалану класс әдістерінің көшірмесін әрбір объект үшін жасамауға мүмкіндік береді. Сонымен класс әдістері әрбір объект үшін тираждалмайды.

7.5 Қосымшаны қуру мысалы

Ұшбұрыш класын құру бойынша мысалды қарастырайық (бірінші бөлімдегі мысалдың негізінде).

Визуалды бағдарламалау кезеңінде Toolbox терезесінен көшірілген стандартты басқару элементтерін қолданатын боламыз: тұрақты мәтін (Label), енгізу өрісі немесе редакциялау терезесі (TextBox) және командалық батырма (Button).

Form1.cs файлының коды:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
namespace WindowsApplication1
{
    public partial class Form1 : Form
    {
        public class treyg
        {
            private int a, b, c, p;
            public string ss;
            public void vvod(int sa, int sb, int sc)
            {
                if (sa > 0 && sb > 0 && sc > 0)
                {
                    if (sa + sb > sc && sa + sc > sb && sb + sc > sa)
                    {
                        a = sa; b = sb; c = sc;
                    }
                }
            }
        }
    }
}
```

```

        p = a + b + c;
        ss = "Үшбұрыш мысалы = " + p.ToString();
    }
    else
        ss = "Үшбұрыштың бір қабырғасы қалған екі қабырғаның қосындысынан үлкен. Мәндерді қайта енгізіңіз";
    }
    else
        ss = "Үшбұрыштың бір қабырғасының ұзындығы 0-ден кіші! Мәндерді қайта енгізіңіз";
    }
}
public Form1()
{ InitializeComponent();
}
private void button1_Click(object sender, EventArgs e)
{
    int A, B, C;
    treyg t = new treyg();
    A = Convert.ToInt32(textBox1.Text);
    B = Convert.ToInt32(textBox2.Text);
    C = Convert.ToInt32(textBox3.Text);
    t.vvod(A, B, C);
    textBox4.Text = t.ss;
}
}
}

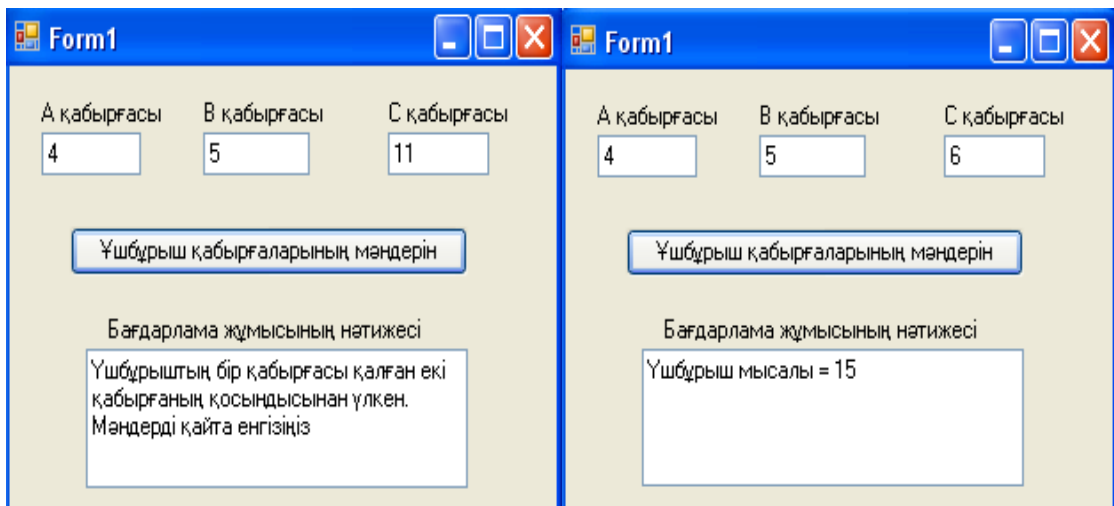
```

Кластың кейбір элементтерін және олардың бағдарламада қолданылуын қарастырайық.

class treyg класының деректерімен және әдістерімен жұмыс орындау үшін осы кластың объектісін– t айнымалысын құру керек.

```
treyg t = new treyg();
```

Кластың private int a, b, c, p; деректері жабық болады. Класс деректерінің элементтеріне оның әдістері арқылы ғана қол жеткізуге болады.



7.1-сурет – «Треугольник Класс» қосымшасының терезесі

Мысалы, егер `t` объектісін құрғаннан кейін деректер элементіне `b` (`t.b = 3;`) жаңа мәнді меншіктейтін болсақ, онда бұл әрекет қате кеткені туралы хабарды шығарады, өйткені класс деректерінің элементтеріне тікелей байланысуға `private` қол жеткізу спецификаторы рұқсат бермейді.

`treug` класында екі әдіс қолданылады – конструктор және үшбұрыш қабырғаларының мәндерін енгізу әдісі.

Конструктор өрістерінің мәндері «нөлдік» болатын объектіні құрайды.

`treug` класының өрістерінің мәндерін анықтау `public void vvod(int sa, int sb, int sc)` арқылы орындалады, әдістегі нақтылы параметрлері ретінде диалог режимінде енгізілген `A`, `B`, `C` айнымалыларының мәндері беріледі.

Бағдарламада үшбұрыш қабырғаларының мәндерін дұрыс енгізбеу және сәйкес түсініктемелерді экранға шығару жағдайлары қарастырылған. Бірақ мәтіндік өрістер бос болып, «Үшбұрыш қабырғаларының мәндерін енгізу» батырмасын басуға қатысты қорғаныс қарастырылмаған.

7.6 Өрістерге қол жеткізу

Әрбір өрісте қол жеткізу модификаторы болады, олардың мүмкін мәндері: `public`, `private`, `protected`, `internal`. `protected` және `internal` атрибуттары бірге қолданылуы мүмкін

`Private` модификаторы. Әдетте егер ешқандай модификатор көрсетілмесе, онда `Private` модификаторы қолданылады. Ол өрістерді басқа барлық кластардан жабық ұстайды және кластың өз әдістеріне ғана тікелей қол жеткізуге рұқсат береді (оқу, жазу).

Ескерту, барлық өрістер кластың барлық әдістеріне қол жетімді болады. Олар кластың әдістері үшін ауқымды ақпарат болып табылады, олармен барлық әдістер жұмыс істейді, яғни өрістерден керекті ақпаратты алады және оларды өзгертеді.

`Protected` модификаторы. Бұл модификатор мұрагер кластарға өрістерді ашық ұстайды. Егер `A` класы модификаторы `protected` болатын өрісті жарияласа, онда `A` класының мұрагері – `B` класының әдістері `A` класының өрістерін мұраға алады және осы өрістермен тікелей жұмыс істей алады.

`Internal` модификаторы. Бұл модификатор бойынша ынтымақтас кластарға өрістер ашық болады. Егер `A` және `B` кластары бір құрылымға (сборка) – бір жобаға тиісті болса, онда бұл кластар ынтымақтас кластар деп аталады. Егер `A` класы белгілі бір өрісі `internal` модификаторымен жарияласа, онда `A` класының клиенті болып келетін ынтымақтас `B` класының әдістері ондай өріспен тікелей жұмыс жасай алады.

`Protected` және `internal` атрибуттарының құрамдастырымы (комбинация). Бұл құрамдастырым мұрагер немесе ынтымақтас кластарға өрістерді ашық ұстайды. Егер ынтымақтас кластар болып келетін мұрагерлерге ғана қолжетімді болуы үшін өрістерге қол жеткізуге қатаң

шектеу қою керек болса, онда кластың өзін `internal` модификаторымен, ал сәйкес өрісті `protected` модификатором жариялау керек. Егер өрістер класс әдістеріне ғана қолжетімді болса, онда олар `private` қол жеткізу модификаторымен бірге жарияланады. Бұндай өрістер жабық өрістер деп аталады, бірақ әдетте олардың кейбірі басқа кластарға ашық болуы керек болады.

Егер А класының кейбір өрістері А класының мұрагері болып келетін В класының әдістеріне қолжетімді болу керек болса, онда осы өрістерді `protected` модификаторымен жариялау қажет. Бұндай өрістер қорғалған өрістер деп аталады. Егер кейбір өрістер А класына ынтымақтас В1, В2, т.б. кластарының әдістеріне қол жетімді болу керек болса, онда ол өрістерде `internal` модификаторын пайдалану қажет, ал барлық ынтымақтас кластарын бір жобаға (`assembly`) орналастыру керек. Ондай өрістер ынтымақтас өрістер деп аталады. Сонымен, егер кейбір өрістер кез келген В класының (В класына А класы қолжетімді) әдістеріне қолжетімді болу керек болса, онда осы өрістерде `public` модификаторы болуы керек. Мұндай өрістер ашық және көпшілік қолды өрістер деп аталады.

7.7 Өзін-өзі тексеру сұрақтары

- 1 Класс ұғымы
- 2 Класс қасиеттері туралы ұғым?
- 3 Класс конструкторы туралы ұғым?
- 4 Класс деструкторы туралы ұғым?
- 5 Класс оқиғасы туралы ұғым?
- 6 Класс индексаторы туралы ұғым?
- 7 Объектің `this` өрісінің міндеті?
- 8 Класс типіндегі айнымалы қалай аталады?
- 9 Класс сипаттамасында `static` қызметтік сөзі нені білдіреді ?
- 10 Класс деректерінің сипаттамасында `public` қызметтік сөзі нені білдіреді?