

## ҚҰРЫЛЫМДАР

### 10.1 Құрылымдар туралы мағлұмат

Көптеген бағдарламалау тілдерінде массивтерден кейін деректерді сипаттаудың келесі түрі - құрылымдар қолданылады, ол әр түрлі типтегі айнымалыларды біріктіретін деректер типі ретінде қарастырылады.

C# тілінде құрылым дегеніміз - класс типіндегі деректер сипаттамасының форма, бірақ кейбір шектеулері бар.

Сонымен қатар құрылымда класс сияқты әр түрлі типтегі деректер өрісімен қатар осы өрістермен жұмыс жасайтын әр түрлі әдістер, конструкторлар бола алады.

Құрылымда класқа қарағанда өзі бағынатын немесе өзіне бағынышты құрылым болмайды, бірақ Object класының әдістеріне ие.

Құрылымда біріктірілген айнымалылардың саны мен құрамы компьютер жадысының көлемімен ғана шектеледі. Әдетте құрылымда айнымалылар объект немесе процесске сәйкес біріктіріледі. Мысалы, тауар сипаттамасы, адам, технологиялық процесс, биржадағы ставкалар динамикасы туралы деректер, т.б.

Құрылым келесі пішімде анықталады:

```
[атрибуттар][спецификаторлар] struct құрылым_аты [:интерфейстер]
    { құрылым_денесі }
```

мұнда,

атрибуттар – құрылым туралы қосымша ақпаратты береді;

спецификаторлар – әдетте құрылымды қолдану шарттарын анықтайды;

интерфейстер – құрылым пайдалана алатын базалық кластар (өрістері жоқ):

құрылым денесі – құрылым элементтерінің құрамын анықтайды.

Құрылымды жариялау класты жариялауға өте ұқсас, бірақ екі айырмашылық бар: class сөзінің орнына struct сөзі қолданылып, міндетті емес интерфейстер тізімі көрсетіледі.

Құрылым денесіне айнымалылар және олардың типтері жазылады. Айнымалы әдетте құрылым өрісі болып аталады. Мысалы, студент туралы жазбаны дайындағанда бойы, салмағы, аяқ киім өлшемі, көзінің түсі сияқты сипаттамалар керек болуы мүмкін. Осы сипаттамаларды құрылымда жазайық:

```
struct student
{
    public string name, cvet_gl;
    public int rost, ves, god_roj;
    public float raz_ob;
};
student styd = new student();
```

Осы мысалда жазба (styd) типіндегі бір айнымалы және жазба өрістеріне арналған түрлі типтегі алты айнымалы жарияланған:

styd.name – жолдық айнымалы;

styd.rost, styd.ves, styd.god\_roj – бүтін типтегі айнымалы;

styd.raz\_ob – нақты типтегі айнымалы;

styd.cvet\_gl – жолдық айнымалы.

Бағдарламалауда нақты жазба өрістерін қолданғанда осылай құралған атауды (айнымалы атауы, нүкте символы, өріс атауы) қолдану керек. Мысалы, оларға мәндерді меншіктеу:

```
styd.name = " Makanov";
styd.cvet_gl = "kara";
styd.rost = 173;
styd.ves = 68;
styd.raz_ob = 27.5;
styd.god_roj = 1986;
```

Келесі мысалды қарастырайық. Студент типіндегі жазбаны анықтау, мәндерді енгізу, студент туралы мәліметтерді шығару керек.

Бағдарлама коды:

```
using System;
namespace ConsoleApplication1
{
    class Program
    {
        struct student
        {
            public string name, cvet_gl;
            public int rost, ves, god_roj;
            public double raz_ob;
        };

        static void Main()
        {
            student styd = new student();
            Console.WriteLine("Stydenttin ati {0}", styd.name);
            Console.WriteLine("Kozinin tysi {0}", styd.cvet_gl);
            Console.WriteLine("boi yzindigi {0}", styd.rost);
            Console.WriteLine("tygan jili {0}", styd.god_roj);
            Console.WriteLine("sakmagi {0}", styd.ves);
            Console.WriteLine("Kiim olshemi {0}", styd.raz_ob);
            Console.WriteLine();
            styd.name = "Makanov";
            styd.cvet_gl = "kara";
            styd.rost = 173;
            styd.ves = 68;
            styd.raz_ob = 27.5;
            styd.god_roj = 1986;
            Console.WriteLine("Stydenttin ati {0}", styd.name);
            Console.WriteLine("Kozinin tysi {0}", styd.cvet_gl);
            Console.WriteLine("boi yzindigi {0}", styd.rost);
            Console.WriteLine("tygan jili {0}", styd.god_roj);
            Console.WriteLine("sakmagi {0}", styd.ves);
```

```
Console.WriteLine("Kiim olshemi {0}", styd.raz_ob);
Console.WriteLine();
Console.WriteLine("Enter pernesin basiniz");
Console.ReadLine();
}
}
}
```

Бағдарлама жұмысы:

```
Stydenttin ati
Kozinin tysi
boi yzindigi 0
tygan jili 0
sakmagi 0
Kiim olshemi 0
```

```
Stydenttin ati Makanov
Kozinin tysi kara
boi yzindigi 173
tygan jili 1986
sakmagi 68
Kiim olshemi 27,5
```

```
Enter pernesin basiniz
```

Жазба өрістерін экранға шығару өрістерге «нөлдік» мәндер меншіктелетінін көрсетеді.

Деректерді жазбаға біріктіру оларды деректердің басқа құрылымдарында қолдануға мүмкіндік (мысалы, массивте, файлдарда) береді.

## 10.2 Құрылым массивін қолдану

Компьютер жадысында ақпаратты көрсетудің жиі қолданылатын формаларының бірі - жазбалар массиві түрінде деректерді ұйымдастыру. 10.1 есебін мысалға ала отырып деректерді өңдейтін, жазбалар массиві түрінде көрсетілген кейбір алгоритмдерді қарастырайық.

10.1-есеп. Студенттер туралы кездейсоқ 20 жазбалар массивін құру. Әрбір жазба студенттің тегі (алты кездейсоқ әріп, біріншісі бас әріп) мен "Физика. Тарих. Матем. Бағдарл. Информ." пәндерінің бес кездейсоқ бағаларынан тұрады.

Құрылған жазбалар массивін шығару. Әрбір студент бойынша бағалар қосындысын есептейтін және шығаратын өңдеу жұмысын орындау керек.

Емтиханнан ең үлкен және ең кіші бағалар алған студенттерді монитор экранына шығару керек.

Кодтар кестесінде бас әріптер (кириллица) 1040 – 1071 нөмерлерінде, ал кіші әріптер 1072 – 1102 аралығында орналасады. Сондықтан студент тегін жазғанда бірінші әріп бірінші диапазоннан, ал қалған бес әріп екінші диапазоннан алынады.

### Бағдарлама коды:

```
using System;
namespace ConsoleApplication1
{
    class Program
    {
        public struct student
        {
            public string fio;
            public int[] ocen;
        };
        public static student[] styd = new student[20];
        public static void sozd()
        {
            char c;
            string s;
            int k, j = 0;
            char[] buka = new char[6];

            Random rd = new Random();
            for (int i = 0; i < 20; i++)
            {
                styd[i].ocen = new int[6];
                j = rd.Next(32) + 1040;
                buka[0] = (char)(j);
                s = buka[0].ToString();
                for (int m = 1; m < 6; m++)
                {
                    j = rd.Next(32) + 1072;
                    buka[m] = (char)(j);
                    s = s + buka[m].ToString();
                    k = rd.Next(4) + 2;
                    styd[i].ocen[m - 1] = k;
                }
                styd[i].fio = s;
            }
            Console.WriteLine("Studentter tizimi kyrildi");
        }
        public static void printz()
        {
            Console.Write("Nomer \t");
            Console.Write("Ati-zhoni \t");
            Console.Write("Fizika \t");
            Console.Write("Tarih. \t");
            Console.Write("Matem. \t");
            Console.Write("Bagdarl. \t");
        }
    }
}
```

```

Console.Write("Informat. \t");
Console.WriteLine();
for (int i = 0; i < 20; i++)
{
Console.Write("{0}\t{1}\t", i + 1, styd[i].fio);
for (int m = 0; m < 5; m++)
    Console.Write("{0}\t", styd[i].ocen[m]);
Console.WriteLine();
}
Console.WriteLine();
}
public static void stat()
{
    int k, j;
    Console.Write("ΦΙΟ \t");
    Console.WriteLine("Ball kosindisi");
    for (int i = 0; i < 20; i++)
    {
        k = 0;
        for (j = 0; j < 5; j++)
            k = k + styd[i].ocen[j];
        Console.WriteLine("{0} {1}", styd[i].fio, k);
        styd[i].ocen[5] = k;
    }
    Console.WriteLine();
}

public static void maxmin()
{
    int k = 0, j = 0;
    int maxi = -100;
    int mini = 150;
    for (int i = 0; i < 20; i++)
    {
        if (maxi < styd[i].ocen[5])
            { maxi = styd[i].ocen[5]; j = i; }
        if (mini > styd[i].ocen[5])
            { mini = styd[i].ocen[5]; k = i; }
    }
    Console.WriteLine();
    Console.WriteLine("{0} - Maksimal baldi {1} aldi", maxi,
    styd[j].fio);
    Console.WriteLine("{0} - Manimal baldi {1} aldi", mini,
    styd[k].fio);
}
static void Main(string[] args)

```

```

{ sozd();
  printz();
  stat();
  maxmin();
  Console.ReadLine();
}
}
}

```

**Бағдарлама жұмысы:**

Studentter tizimi kyrildi

Nomer Ati-zhoni Fizika Tarih. Matem. Bagdarl. Informat.

1	Юаьоюю	2	2	4	2	4
2	Тфютьэ	3	3	3	2	3
3	Прилюс	5	5	3	4	4
4	Блшвйъ	3	4	4	4	5
5	Алъпам	5	3	4	3	5
6	Фсбфтэ	4	5	2	2	2
7	Яшфылф	2	4	3	4	3
8	Чппюов	3	5	5	4	2
9	Бчзждц	5	4	5	2	4
10	Алъщчх	2	5	4	2	3
11	Дътфцс	2	3	2	4	5
12	Щьыжоу	3	3	2	4	5
13	Жондщю	2	2	2	5	4
14	Стеъюа	5	2	3	3	5
15	Гтслхг	2	5	3	4	4
16	Лфппфу	4	5	2	5	5
17	Йкэджк	4	3	2	4	5
18	Злзжмй	4	4	4	3	3
19	Щзтььд	3	4	5	4	3
20	Дйъвпр	5	2	3	3	3

ФИО Ball kosindisi

Юаьоюю	14
Тфютьэ	14
Прилюс	21
Блшвйъ	20
Алъпам	20
Фсбфтэ	15
Яшфылф	16
Чппюов	19
Бчзждц	20
Алъщчх	16
Дътфцс	16
Щьыжоу	17

Жондщю 15  
Стееьюа 18  
Гтслхг 18  
Лфппфу 21  
Йкэджк 18  
Злзжмй 18  
Щзтььд 19  
Дйъвпр 16

21 - Maximal baldi Прилюс aldi

14 - Manimal baldi Юаьоюю aldi

Бағдарламада әрбір студент бағасының қосындысын сақтау үшін массивте қосымша бағана қолданылған.

### 10.3 C# тіліндегі тізім

Бағдарламаларды жазғанда әдетте бір бірімен байланысқан атаулары бар бірнеше тұрақтыларды анықтау қажет болады.

Ол үшін деректердің түгенделген типін қолдану ыңғайлы, оның барлық мүмкін мәндері бүтін санды тұрақтылар тізімімен анықталады, мысалы:

```
enum Кемпірқосақ { Қызыл, Тоқсары, Сары, Жасыл, Көк,  
Күлгін };
```

Жоғарыда келтірілген мысалда әрбір тұрақтыға еш әрекетсіз `int` типіндегі, 0-ден басталатын реттілік мәндер меншіктеледі, бірақ басқа да мәндерді меншіктеуге болады, мысалы:

```
enum Numer { Қызыл = 2, Тоқсары, Сары, Жасыл = 10, Көк  
};
```

Оранжевый және Сары тұрақтыларына 3 және 4 мәндері, Көк тұрақтысына 11 мәні меншіктеледі.

Әрбір тізім ішіндегі тұрақтылардың атаулары бірегей болуы керек, ал мәндері бірдей бола алады.

Тізімнің тұрақтылардан артықшылығы - байланыстырылған тұрақтылардың көрнекі болуында, сонымен қатар компилятор типтерді тексеруді орындайды, ал құру ортасы тұрақтылардың мүмкін мәндерін көрсетеді, тізім ретінде көрсетеді

Тізімді жазу пішімі:

```
[атрибуттар] [спецификаторлар] enum тізім_атауы  
[:базалық_тип ]  
тізім_денесі [ ; ]
```

Тізім спецификаторларының мағынасы класс үшін қолданылатын спецификаторлардың мағынасымен бірдей, сонымен қатар тек `new`, `public`, `protected`, `internal` және `private` спецификаторлары ғана рұқсат етілген.

Базалық тип – тізімді құрайтын элементтер типі. Еш әрекетсіз `int` типі қолданылады, бірақ типті бүтінсанды типтердің (`char` типінен өзге) ішінен таңдап, айқын түрде анықтауға болады: `byte`, `sbyte`, `short`, `ushort`, `int`, `uint`, `long`, `ulong`.

Тізім денесі тұрақтылар атауларынан тұрады, оның әрбіріне мәнді меншіктеуге болады. Егер мән көрсетілмесе, онда ол мән алдында тұрған тұрақтының мәніне бір қосылып есептеледі. Тұрақтылар еш әрекетсіз `public` спецификаторында болады.

Тізімдік типтегі айнымалылармен арифметикалық операцияларды (+, -, ++, - -), разряд бойынша логикалық операцияларды (^, &, |, ~) орындауға, оларды қатынас операциялары (<, <=, >, >=, ==, !=) арқылы салыстыруға және байтпен (`sizeof`) көлемін есептеуге болады.

Бүтін санды өрнектерде және меншіктеу операцияларында тізімдік типтегі айнымалыларды қолданғанда типті айқын түрлендіру керек.

Мысал (тізім туралы мысал [2] алынған):

```
using System;
using System.Collections.Generic;
using System.Text;
namespace ConsoleApplication1
{
    struct Боец
    {
        public enum Воинское_Звание
        {
            Рядовой, Сержант, Лейтенант, Майор, Полковник, Генерал
        }
        public string Фамилия;
        public Воинское_Звание Звание;
    }
    class Program
    {
        static void Main()
        {
            Боец x;
            x.Фамилия = "Иванов";
            x.Звание = Боец.Воинское_Звание.Рядовой;
            for (int i = 1980; i < 2011; i = i + 5)
            {
                if (x.Звание <= Боец.Воинское_Звание.Генерал)
                {
                    Console.WriteLine("Shili: {0} {1} {2}", i, x.Звание,
x.Фамилия);
                    x.Звание++;
                }
            }
            Console.WriteLine("Enter pernesin basiniz");
        }
    }
}
```



```
Console.ReadLine();  
}  
}  
}
```

Бағдарлама жұмысы :

```
Shili: 1980 Рядовой Иванов  
Shili: 1985 Сержант Иванов  
Shili: 1990 Лейтенант Иванов  
Shili: 1995 Майор Иванов  
Shili: 2000 Полковник Иванов  
Shili: 2005 Генерал Иванов  
Enter pernesin basiniz
```

## 10.4 C# тіліндегі файлдар

Кез келген бағдарламалау тілінде файлдармен жұмыс жасау құралдары бар. Көптеген бағдарламалар жұмысты файлдардан деректерді оқудан бастайды және нәтижелерді файлға жазумен аяқтайды

C# тілінде файлдармен жұмыс жасауға арналған бірнеше класс бар.

Белгілі бір кластар тобы (олар `System.IO` атаулар кеңістігінде орналасқан) файлдармен деректер бірлігі сияқты жұмыс істейді. Олардың міндетіне каталогтармен (файлдар папкасы) жұмыс жасау, файлдардың орнын ауыстыру, олардың атауларын өзгерту, көшіру, файл атрибуттары, файлдарды ұсыну кестесімен (FAT) жұмыс жасау кіреді. Осы топты шартты түрде файлдарды өңдеу класы деп атауға болады.

Windows жүйесінде (сонымен қоса C# тілінде) түрлі құрылғылар (оперативтік жад, монитор, дискілік жад, желілер) арасында деректерді жеткізу үшін деректер ағыны (stream) тұжырымдамасы қолданылады.

Деректердің екілік, символдық, жолдық ағынымен анықталған файлдармен жұмыс жасауға көмектесетін арнайы кластар бар. Осы кластар да `System.IO` кеңістігінде анықталған.

C# тілінде құрылымдар, кластар түрінде көрсетілген деректермен жұмыс жасау үшін сериализация ұғымына топтастырылған арнайы класс жиынтығы бар.

Сериализация ұғымын объекттерді, жазбаларды байттар тізбектілігіне түрлендіру процесімен түсіндіруге болады (файлға жазылатын ақпарат байттар тізбектілігімен берілуі керек). Кері процесс - байттар тізбектілігінен объекттерді, жазбаларды қалпына келтіру десериализация деп аталады.

Кейде сериализация дегеніміз – объекттің ағымдағы жағдайын жадыда немесе ақпарат құрылғысында сақтау, ал кері үдеріс - ақпарат құрылғысынан деректерді оқу десериализация деп аталады.

Сериализация мен десериализация процесстерін басқаратын класс `System.Runtime.Serialization` атаулар кеңістігінде орналасады.

## 10.5 Файлдармен жұмыс жасағанда сериализация мен десериализацияны

## ҚОЛДАНУ МЫСАЛЫ

Студенттер туралы есепте сериализация мен десериализация процесстерін қолдану мысалын қарастырайық.

10.2-есеп. Диалог режимінде студенттердің саны 20-дан аспайтын массивті құру керек. Әрбір жазбада студенттің тегі, топ атауы, емтихан балы, жасы жазылады. Монитор экранында жазбалар массивін қарап шығуды алдын ала қарастыру керек.

Жазбалар массивін монитор экранында көрсетуді, файлға жазбалар массивін сериализациялау және файлдан жазбалар массивін десериализациялау процесстерін қолдануды, файлды қайтадан қарап шығуды қарастыру керек. Бағдарламада менюді қолдану керек.

Есептің шартына сәйкес 4 өрісі бар құрылымды құрайық: студенттің аты, топ атауы, емтихан балы, студент жасы.

```
using System;
using System.Collections.Generic;
using System.Text;
using System.IO;
using System.Runtime.Serialization.Formatters.Binary;
namespace ConsoleApplication1
{
    [Serializable]
    struct Stydent
    {
        public String Name; //студенттің аты
        public String Grup; // тобы
        public int Ball; // балл
        public int Let; // жасы
    }
    class Program
    {
        public static Stydent[] Styds = new Stydent[20];
        public static int kol = 0;
        public static void sozd()
        {
            int b;
            string buf;
            Console.WriteLine("Toptagi stydentter sani <= 20 !!");
            buf = Console.ReadLine();
            kol = Convert.ToInt32(buf);
            for (int i = 0; i < kol; i++)
            {
                Console.WriteLine("{0}-stydenttin atin engiziniz", i + 1);
                buf = Console.ReadLine();
                Styds[i].Name = buf;
            }
        }
    }
}
```

```

    Console.WriteLine("{0}-stydenttin tobin engiziniz", i +
1);
    buf = Console.ReadLine();
    Styds[i].Grup = buf;
    Console.WriteLine("{0}-stydenttin balin engiziniz", i +
1);
    buf = Console.ReadLine();
    b = Convert.ToInt32(buf);
    Styds[i].Ball = b;
    Console.WriteLine("{0}-stydenttin shasin engiziniz", i +
1);
    buf = Console.ReadLine();
    b = Convert.ToInt32(buf);
    Styds[i].Let = b;
}
}
public static void zapf()
{
    //сериализация үшін ағынды құрамыз:
    FileStream StreamOut = new FileStream("Sytds.dat",
        FileMode.Create, FileAccess.Write);
    //екілік пішімді қолданамыз:
    BinaryFormatter fmt = new BinaryFormatter();
    fmt.Serialize(StreamOut, Styds); // объектерге
сериализациялау
    StreamOut.Close(); // ағынды жабамыз
}
public static void printm()
{
    Console.WriteLine("{0,20}, {1, 20}, {2, 10}, {3, 10}",
        "Ati", "Tobi", "Ball", "Shasi");
    foreach (Stydent T in Styds)
    {
        if (T.Ball > 0)
            Console.WriteLine("{0,20}, {1, 20}, {2, 10}, {3, 10}",
                T.Name, T.Grup, T.Ball, T.Let);
    }
}
public static void vvod()
{
    int i = 0;
    //десериализация үшін ағынды құрамыз:
    FileStream StreamIn = new FileStream("Sytds.dat",
        FileMode.Open, FileAccess.Read);
    // екілік пішімді қолданамыз:
    BinaryFormatter fmt = new BinaryFormatter();

```

```

    Styds =
(Stydent[]) fmt.Deserialize(StreamIn); //Десериализациялау
    kol = 0;
    while (Styds[i].Ball > 0) { kol++; i++; }
    StreamIn.Close(); // ағынды жабамыз
}
public static void Main()
{
    int k = 0;
    string buf;
    while (k < 6)
    {
        Console.WriteLine("1 - Massivti kyry");
        Console.WriteLine("2 - Massivti failga shazy");
        Console.WriteLine("3 - Massivti shigary");
        Console.WriteLine("4 - faildan massivti oky");
        Console.WriteLine("5 - Massivti shigary");
        Console.WriteLine("6 - Shigy");
        Console.WriteLine("Menu punktterin tandaniz");
        buf = Console.ReadLine();
        k = Convert.ToInt32(buf);
        switch (k)
        {
            case 1: sozd(); break;
            case 2: zapf(); break;
            case 3: printm(); break;
            case 4: vvod(); break;
            case 5: printm(); break;
            default: break;
        }
    }
}
}
}
}
}
}

```

#### **Бағдарлама жұмысы:**

```

1 - Massivti kyry
2 - Massivti failga shazy
3 - Massivti shigary
4 - faildan massivti oky
5 - Massivti shigary
6 - Shigy
Menu punktterin tandaniz
1
Toptagi stydentter sani <= 20 !!
3
1-stydenttin atin engiziniz

```

Бедаш Дмитрий

1-styudenttin tobin engiziniz

09-ВТ-1

1-styudenttin balin engiziniz

90

1-styudenttin shasin engiziniz

18

2-styudenttin atin engiziniz

Жумашев Ержегит

2-styudenttin tobin engiziniz

09-ВТ-1

2-styudenttin balin engiziniz

85

2-styudenttin shasin engiziniz

19

3-styudenttin atin engiziniz

Аскарова Дина

3-styudenttin tobin engiziniz

09-ИС-1

3-styudenttin balin engiziniz

89

3-styudenttin shasin engiziniz

18

1 - Massivti kyry

2 - Massivti failga shazy

3 - Massivti shigary

4 - faildan massivti oky

5 - Massivti shigary

6 - Shigy

Menu punktterin tandaniz

2

1 - Massivti kyry

2 - Massivti failga shazy

3 - Massivti shigary

4 - faildan massivti oky

5 - Massivti shigary

6 - Shigy

Menu punktterin tandaniz

3

Ati, Tobi, Ball, Shasi

Бедаш Дмитрий, 09-ВТ-1, 90, 18

Жумашев Ержегит, 09-ВТ-1, 85, 19

Аскарова Дина, 09-ИС-1, 89, 18

1 - Massivti kyry

2 - Massivti failga shazy

3 - Massivti shigary

4 - faildan massivti oky  
5 - Massivti shigary  
6 - Shigy  
Menu punktterin tandaniz  
6

Бағдарламаны екінші рет іске қосқанда:

1 - Massivti kyry  
2 - Massivti failga shazy  
3 - Massivti shigary  
4 - faildan massivti oky  
5 - Massivti shigary  
6 - Shigy

Menu punktterin tandaniz

4

1 - Massivti kyry  
2 - Massivti failga shazy  
3 - Massivti shigary  
4 - faildan massivti oky  
5 - Massivti shigary  
6 - Shigy

Menu punktterin tandaniz

5

Ati, Tobi, Ball, Shasi

Бедаш Дмитрий, 09-ВТ-1, 90, 18

Жумашев Ержегит, 09-ВТ-1, 85, 19

Аскарова Дина, 09-ИС-1, 89, 18

1 - Massivti kyry  
2 - Massivti failga shazy  
3 - Massivti shigary  
4 - faildan massivti oky  
5 - Massivti shigary  
6 - Shigy

Menu punktterin tandaniz

Бағдарламада есептің шартына сәйкес төрт әдіс қолданылған. (студенттер массивін шығару әдісі екі рет қолданылған).

## 10.6 Өзін-өзі тексеру сұрақтары

- 1 C# тілінде құрылымды жариялағанда қандай қызметтік сөз қолданылады?
- 2 struct типіндегі айнымалылар әдетте қалай аталады?
- 3 Жазбада қандай айнымалылар біріктіріле алады?
- 4 Жазбаның ішінде айнымалылар қалай аталады?
- 5 Құрылымның қандай мәндік және сілтемелік типі бар?

6 Жазба өрісінің атауы қалай анықталады?

7 C# тілінде тізімдік типті жариялағанда қандай қызметтік сөз қолданылады?

8 Тізімдік тип нені біріктіреді?

9 C# тілінде құрылымдармен, кластармен жұмыс істегенде сериализацияны қалай түсінесіз?

10 C# тілінде құрылымдармен, кластармен жұмыс істегенде десериализацияны қалай түсінесіз?

