

# ЖОЛДЫҚ АЙНЫМАЛЫЛАР

## 8.1 Символдық тип

Бағдарламалауда символдарды, сөздерді, сөйлемдерді, мәтіндерді өңдеумен байланысты көптеген есептер класы бар.

Осы мақсатта көптеген бағдарламалау тілдеріне символдармен және мәтіндермен, яғни `char` және `string` типтерімен жұмыс істейтін арнайы құралдар қосылған.

`C#` тілінде осы типтердің екеуі де бар. Олардың арасында тығыз байланыстың барын айта кету керек.

Кейбір авторлар `char` типіндегі айнымалы бір символдан тұратын жол деп есептейді. Басқа авторлар жолдық типтегі айнымалыларды `char` типіндегі айнымалылар массиві ретінде қарастырады.

`C#` тілінде символдармен жұмыс жасауға арналған арнайы `char` класы қолданылады, онда символдарды көрсетудің екі байттық жүйесі және осы символдармен жұмыс жасаудың көптеген әдістер жиыны қолданылады.

Символдарды көрсетудің екі байттық жүйесі Unicode кодтамасы деген атқа ие болды. Windows тобындағы ОЖ-де 2 байтты UTF-16LE кодтамасы қабылданған.

Барлық символдар топтастырылған және реттелген. Мысалы, 0 мен 9 арасындағы сандардың екілік кодтары өсу тәртібінде, қатарынан орналасқан. Алфавиттің а және z арасындағы үлкен және кіші әріптері де өсу тәртібінде, қатарынан орналасқан. Кириллицада қолданылатын "Ё" әрпі ғана өзгеше сипатталады.

Символдарды кодтаудың осы ерекшелігі мәтіндерді өңдеудің түрлі алгоритмдерін оңай құруға мүмкіндік береді, мысалы, алфавит бойынша сөздерді реттеу.

Unicode кодтамасымен берілетін алфавитте 65000 аса символдар бар болғандықтан, кодтардың көбі резервте сақталады және қолданылмайды.

`Char` класында апострофқа алынған символдармен берілген символды тұрақтылар және басқарушы символдар мен Unicode-тізбегі анықталған.

Бағдарламада символдарды қолдану үшін олар апострофқа алынады. Escape-тізбегінің коды ақпаратты экранға, принтерге, басқа да құрылғыларға шығару үшін пішімдеуді қолданады. Ол бір немесе бірнеше символдан тұрады, оның алдында кері слеш (\) жазылады.

Символды көрсету Unicode-тізбегінде кодтың көмегімен орындалады.

8.1-кестеде кейбір escape-тізбегі көрсетілген.

8.1-кесте – `C#` тіліндегі кейбір escape-тізбегінің мысалдары

<code>\n</code>	Жаңа жол символы
<code>\t</code>	Көлденең табуляция
<code>\v</code>	Вертикаль табуляция
<code>\b</code>	Бір позицияға шегіну символы
<code>\r</code>	Күймешені қайтару
<code>\f</code>	Бетті ауыстыру

\a	Дыбыстық сигнал (ескерту)
\\	Кері слеш символы
\'	Бір тырнақша символы
\"	Қос тырнақша символы
\?	Сұрақ белгісі символы
\0	Нөлдік символ (барлық биттер нөлге тең)

Ескерту, мәтін ішінде escape-тізбегін қарапайым символдар сияқты қолдануға болады.

Char класында символдармен жұмыс жасау үшін әдістердің үлкен жиыны бар. Көптеген әдістер статикалық болып келеді және объекттерді құруды қажет етпейді. Кейбір әдістер 8.2-кестеде көрсетілген.

### 8.2-кесте – кейбір char класының статикалық әдістері

Әдістер	Сипаттамасы
Double GetNumericValue(char c);	Символдың сандық кодын қайтарады. Егер символ болмаса, онда -1 қайтарылады.
UnicodeCategory GetUnicodeCategory (char c);	Барлық символдар категорияға бөлінген, сондықтан әдіс символ категориясын қайтарады.
bool IsControl(char c);	Егер c символы басқарушы болса, онда ол true мәнін қайтарады.
bool IsDigit(char c);	Егер c символы цифр болса, онда ол true мәнін қайтарады.
bool IsLetter(char c);	Егер c символы әріп болса, онда ол true мәнін қайтарады.
bool IsLetterOrDigit(char c);	Егер c символы цифр немесе әріп болса, онда ол true мәнін қайтарады.
bool IsLower(char c);	Егер c символы кіші әріп болса, онда ол true мәнін қайтарады.
bool IsNumber(char c);	Егер c символы сан болса, онда ол true мәнін қайтарады.
bool IsPunctuation(char c);	Егер c символы тыныс белгілерінің символы болса, онда ол true мәнін қайтарады.
bool IsSeparator(char c);	Егер c символы бөлгіштерге жатса, онда true мәні қайтарылады.
bool IsSurrogate(char c);	Егер c символы құрастырылған болса, онда true мәні қайтарылады.
bool IsSymbol(char c);	Егер c символ болса, онда true мәні қайтарылады.
bool IsUpper(char c);	Егер c символы бас әріп болса, онда true мәні қайтарылады.
bool IsWhiteSpace(char c);	Егер c символы бос орын болса, онда true мәні қайтарылады.

char ToLower(char c);	с символын кіші әріпке ауыстырады
String ToString(char c);	с символын жолға ауыстырады
char ToUpper(char c);	с символын бас әріпке ауыстырады

GetUnicodeCategory() әдісі char класына сәйкес (UnicodeCategory) категориясының бір мәнін қайтарады. Кейбір категориялардың сипаттамасы 8.3-кестеде көрсетілген.

8.3-кесте – Char класының символдар категориялары (UnicodeCategory).

Категория атауы	Символдың сипаттамасы
ClosePunctuation	Фигуралық, квадрат, дөңгелек жақша сияқты жұп символдарға арналған жабушы символ
Control	Басқарушы символ
CurrencySymbol	Ақша бірлігінің символы
DashPunctuation	Сызықшалар (тире, дефис)
DecimalDigitNumber	ондық сан
EnclosingMark	жабушы маркер
FinalQuotePunctuation	Дәйексөздің жабушы маркері (мысалы, жабушы тырнақша)
Format	Пішімдеуші символ
InitialQuotePunctuation	Дәйексөздің ашушы маркері
LetterNumber	Әріптік сан (мысалы, рим цифрлары)
LineSeparator	Жолдарды бөлгіш
LowerCaseLetter	Кіші әріптер
MathSymbol	Математикалық символ
ModifierLetter	Символ алдыңғы символдың модификаторы болып келеді
NonSpacingMark	Негізгі символды өзгертетін, бос орыннан өзге символ.
OpenPunctuation	Ашылатын жақшаны көрсететін символ (фигуралық, квадратты, дөңгелек)
OtherLetter	Әріп, бірақ кіші, үлкен әріптер емес.
OtherNotAssigned	Бұл символ Unicode-символы емес
OtherNumber	Ондық немесе әріптік емес сандық символ (мысалы, бөлшектің символы)
OtherPunctuation	Фигуралық, квадрат, дөңгелек жақша сияқты жұп символдарға арналған ашушы символ
OtherSymbol	Математикалық емес символ, ағымдағы символ немесе өзгертуші символы
ParagraphSeparator	Азат жолды бөлушісі
PrivateUse	Жиі қолданылатын символ (private-use)
SpaceSeparator	Пішімдеуші немесе басқарушы символ емес бөлуші
SpacingCombiningMark	Символды таңбалау орнының енін өзгертуші

	СИМВОЛ
TitlecaseLetter	Мәтіннің ең бірінші бас әрпі (азат жол маркері)
UppercaseLetter	Бас әріп

Айнымалылар мен char класына тиісті кейбір әдістер қолданылған мысалда алфавит символдарының Unicode кодына және керісінше түрлендірудің нұсқалары қарастырылған. Char класының кейбір әдістерін бас әріптерді кіші әріптерге ауыстыру үшін түрлендіру. Символ бойынша оның категориясының тұрақтысы.

Бағдарлама код:

```
using System;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main()
        { char i;
          int a1, a2;
          for (i = 'a'; i <= 'z'; i++)
          { Console.WriteLine("{0}-", i);
            Console.WriteLine("{0} ", (int)i);
          }
          Console.WriteLine();
          Console.WriteLine();
          a1 = (int)'A';
          a2 = (int)'Я';
          for (int j = a1; j <= a2; j++)
          {
              Console.WriteLine("{0}-", j);
              Console.WriteLine("{0} ", (char)j);
          }
          Console.WriteLine();
          Console.WriteLine();
          for (i = 'A'; i <= 'Я'; i++)
          {
              Console.WriteLine("{0}-", char.ToLower(i));
              Console.WriteLine("{0} ", (int)char.ToLower(i));
          }
          Console.WriteLine("\n\n");
          System.Globalization.UnicodeCategory cat1, cat2;
          cat1 = char.GetUnicodeCategory('A');
          cat2 = char.GetUnicodeCategory(';');
          Console.WriteLine("'A' - category {0}", cat1);
          Console.WriteLine("';' - category {0}", cat2);
          Console.WriteLine("'4' - category {0}",
char.GetUnicodeCategory('4'));
```

```

        Console.WriteLine("'+' - category {0}",
char.GetUnicodeCategory('+'));
        Console.WriteLine();
        Console.WriteLine("Enter pernesin basiniz");
        Console.ReadLine();
    }
}
}

```

### Бағдарлама жұмысы:

```

a-97 b-98 c-99 d-100 e-101 f-102 g-103 h-104 i-105 j-106
k-107 l-108 m-109 n-110 o-111 p-112 q-113 r-114 s-115 t-116
u-117 v-118 w-119 x-120 y-121 z-122

```

```

1040-А 1041-Б 1042-В 1043-Г 1044-Д 1045-Е 1046-Ж 1047-З
1048-И 1049-Й 1050-К 1051-Л 1052-М 1053-Н 1054-О 1055-П
1056-Р 1057-С 1058-Т 1059-У 1060-Ф 1061-Х 1062-Ц 1063-Ч
1064-Ш 1065-Щ 1066-Ъ 1067-Ы 1068-Ь 1069-Э 1070-Ю 1071-Я

```

```

a-1072 б-1073 в-1074 г-1075 д-1076 е-1077 ж-1078 з-1079
и-1080 й-1081 к-1082 л-1083 м-1084 н-1085 о-1086 п-1087 р-
1088 с-1089 т-1090 у-1091 ф-1092 х-1093 ц-1094 ч-1095 ш-1096
щ-1097 ъ-1098 ы-1099 ь-1100 э-1101 ю-1102 я-1103

```

```

'A' - category UppercaseLetter
';' - category OtherPunctuation
'4' - category DecimalDigitNumber
'+' - category MathSymbol

```

Enter pernesin basiniz

## 8.2 Жолдық айнымалы ұғымы

C# тілінде жолдармен жұмыс істеу үшін (мәтіндермен, символдардың кез келген бірігуі) сәйкес класымен анықталатын арнайы `string` типі қолданылады. `String` класы сілтемелік типке жатады және кез келген ұзындықтағы жолдық айнымалыларды анықтауға көмектеседі.

Барлық сілтемелік типтердегідей айнымалылардың мәні үйіндіде сақталады. Бірақ `string` типіндегі айнымалылардың ерекшелігі – олардың болжап болмайтын «еркін» ұзындығы. Сондықтан `string` типіндегі айнымалының мәнін өзгерту үйіндіде жаңа өзгертілген мәнді құру жолымен орындалады. Бұл орайда басқа айнымалылардың мәндері өзгермейді. Жолдық айнымалылармен жұмыс жасауда осы ерекшелікті ескеру керек.

C# тілінде өзгермейтін (`immutable`) класс ұғымы да бар. Осы кластар үшін объект мәнін өзгертуге болмайды. Бар объект негізінде әдістер жаңа объект құра

алады, бірақ алдыңғы объект мәні өзгертілмейді. Осындай өзгермейтін класқа `string` класы да жатады. Осы кластың бір де бір әдісі басқа объекттердің мәнін өзгерте алмайды. `String` класының әдістері мен қасиеттері 8.4 және 8.4-кестесінде көрсетілген.

8.4-кесте – `String` класы

Қасиет	Сипаттама
<code>public char this[ int Index] {get:};</code>	Бұл қасиет жолдан керекті символды алуға көмектеседі (тек оқу үшін)
<code>public int Length(){get:};</code>	Жолдың ұзындығын қайтарады (өзгертілмейді)

`Length` қасиеті `S` жолдық айнымалының «ұзындығын», яғни жолдық айнымалының символдар санын анықтауға көмектеседі, сондықтан бағдарламада жиі қолданылады.

Жол символына оның индексі бойынша қол жеткізу әрбір символ бойынша талдау жүргізген кезде жиі қолданылады.

Мәтін сөздерін сұрыптау немесе жеке жолдық айнымалыларды салыстыру үшін `Compare()` әдісін қолдану керек, мысалы, `s1` және `s2` айнымалыларын салыстыру былай орындалады:

```
if (Compare (s1, s2)==0)
```

Класс әдістерін қолдануды оқулықтың басқа бөлімдерінде қарастырамыз.

Жолдық айнымалыларды қолданудан бұрын оларды бағдарламада жариялауды үйрену керек.

`string` класының объекттері сілтемелік типтегі айнымалылар сияқты жарияланады - класс конструкторы айқын немесе айқын емес шақырылады.

8.5-кесте – `String` класының кейбір әдістері

Әдіс	Сипаттама
<code>public static int Compare (string S1, string S2);</code>	Екі жолды салыстырады және егер <code>S1 &lt; S2</code> болса, теріс сан қайтарылады. Егер <code>S1 == S2</code> болса, нөл саны, егер <code>S1 &gt; S2</code> болса, оң сан қайтарылады. Әдісті орындау барысында символдар регистрінің айырмашылығы, мезгіл мен ақша бірлігінің ұлттық жазылуы, т.б. ескерілмейді.
<code>public static int CompareOrdinal(string S1, string S2);</code>	Екі жолды символдарының сандық мәндерін салыстыру арқылы салыстырады
<code>public static string Concat (params object[]);</code>	Жолдарды біріктіреді
<code>public void CopyTo (int SourceInd, char[] Dest, int DestInd, int Count);</code>	<code>Dest</code> массивінде <code>DestInd</code> символынан бастап мынаны көшіреміз: жолдың <code>SourceInd</code> символынан бастап <code>Count</code> санынан аспайтын символдарды

public bool EndsWith (string S);	Егер жол S ішкі жолымен аяқталса, true қайтарылады.
public static string Format (string S, params object[]);	S жолын құрады
public int IndexOf(string S);	S ішкі жолы немесе символы табылса, оның бірінші индексі қайтарылады, егер табылмаса, теріс сан қайтарылады.
public static string Join (string S, params string[] ss);	ss жолдарын S бөлгіші арқылы бір ұзын жолға біріктіреді.
public static int LastIndexOf (string S);	S ішкі жолы немесе символы табылса, оның соңғы табылған индексі қайтарылады, егер табылмаса, теріс сан қайтарылады.
public string PaddLeft (int TotalWidth, char C);	Жолдың ұзындығы TotalWidth тең болғанша оны сол жағынан C символдарымен толықтырады.
public static string Remove (int StartIndex, int count);	Ағымдағы жолдан StartIndex символынан бастап count аспайтын символдарды жояды.
public string Replace (string S, string ss);	Ағымдағы жолдың S ішкі жолын ss түрлендіреді.
public string[] Split(params char[] delimiters);	Жолды delimiters шектеулеріне сәйкес сөздерге бөледі.
public bool StartsWith (string S);	Ағымдағы жол S ішкі жолынан басталса, true қайтарылады.
public string SubString ( int BegInd, int EndInd);	Ағымдағы жолдан BegInd және EndInd символдарымен шектелген ішкі жолды қайтарады.
public string ToLower();	Ағымдағы жолды кіші әріптермен жазып қайтарады.
public string Insert (int Ind. string S);	Ind индексінен бастап S ішкі жолын орналастырады.
public string Trim();	Бос орындары өшірілген жол қайтарылады.
public string TrimEnd();	Бос орындар жойылған алғашқы жолды қайтарады
public string TrimStart();	Бос орындар жойылған алғашқы жолды қайтарады
public string ToUpper();	Ағымдағы жолды үлкен әріптермен жазып қайтарады.

Жолдық айнымалыны жариялаған кезде конструктор айқын шақырылмайды.  
Мысалы:

```
string st1 = "Жол";
```

Жолдық айнымалыны `string` класының конструкторларын айқын шақырып жариялауға болады, мысалы: бірінші ұзындығы нөл болатын жолдық айнымалыны жариялау, ал одан кейін оған мәнді меншіктеу:

```
string st2 = new string(' ', 0);  
st2 = "Жол 2";
```

Жолдық айнымалыны жариялаудан кейін оны бағдарламада түрлі операцияларда қолдануға болады. Негізгі операциялар: меншіктеу, баламалылығын тексеру (`==` немесе `!=`), жолдарды тіркестіру (қосу).

Егер `st1` жолдық айнымалыға басқа `st2` жолдық айнымалы меншіктелсе, онда `st1` айнымалысы `st2` айнымалының мәні сақталған үйінді (жады аумағындағы адрес) сілтемесіне ие болады. Сонымен жолдық айнымалыларды меншіктеуде олардың мәндерінің орнына сілтемелер «көбейтіледі».

Басқа сілтемелік айнымалылардан өзге жолдық айнымалылардың баламалылығын тексеру кезінде олардың сілтемелері емес, мәні салыстырылады.

Тіркестіру операциясын орындау барысында (`Concat()` әдісі - + операциясы) бір жолдық айнымалының мәніне басқа жолдық айнымалының мәні қосылады. Сондықтан алынған мән бір топта жаңа сілтеме бойынша орналасады.

Жолдық айнымалылар қолданылатын мысалды қарастырайық:

```
using System;  
namespace ConsoleApplication1  
{ class Program  
  {  
    static void Main()  
    {  
      string a = "Tabigat ";  
      Console.WriteLine("a sozi = {0}", a);  
      string b = new string(' ', 0);  
      Console.WriteLine("b sozinin yzindigi = {0} символ",  
b.Length);  
      Console.WriteLine("b sozinin engiziniz");  
      b = Console.ReadLine();  
      Console.WriteLine("b sozi = {0}", b);  
      string c = a + b;  
      Console.WriteLine("Operazia zhymisi c = a + b; c = {0}",  
c);  
      Console.WriteLine("a sozinin yzindigi = {0} символ",  
a.Length);  
      Console.WriteLine("b sozinin yzindigi = {0} символ",  
b.Length);  
      Console.WriteLine("Olardin ortak yzindigi = {0} символ",  
c.Length);  
      Console.WriteLine("Operazia zhymisi if (string.Compare(a,  
b) >= 0) вывод a > b;");  
      Console.WriteLine("    else (вывод b >= a);");  
    }  
  }  
}
```



```

    if (string.Compare(a, b) >= 0) Console.WriteLine(" a >
b");
    else Console.WriteLine(" b >= a");
    c = a.Remove(1, 2);
    Console.WriteLine("c = a.Remove(1,2); c = {0}", c);
    Console.WriteLine("Enter pernesin basiniz");
    Console.ReadLine();
}
}
}

```

**Бағдарлама жұмысы:**

```

a sozi = Tabigat
b sozinin yzindigi = 0 СИМВОЛОВ
b sozinin engiziniz
Tabigat
b sozi = Tabigat
Operazia zhymisi c = a + b; c = Tabigat Tabigat
a sozinin yzindigi = 8 СИМВОЛОВ
b sozinin yzindigi = 7 СИМВОЛОВ
Olardin ortak yzindigi = 15 СИМВОЛОВ
Operazia zhymisi if (string.Compare(a, b) >= 0) вывод a
> b;
    else (вывод b >= a);
    a > b
    c = a.Remove(1,2); c = Tigat
    Enter pernesin basiniz

```

Бағдарламада жолдық айнымалыларды жариялаудың бірнеше жолдары, қосу операциясы арқылы айнымалыларды біріктіру әдісі, айнымалының «ұзындығын» анықтау үшін пайдаланылатын Length қасиеті қолданылады.

Бағдарламада Compare бүтін санды әдісі жолдық айнымалыларды салыстыру үшін қолданылған. Диалог режимінде b айнымалыға түрлі мәндерді беріп Compare әдісінің жұмысын тексеруге болады.

Ағымдағы айнымалыларды қолданатын әдістері a.Remove мысалында қарастырылған.

String класының қасиеті жолдық айнымалыны символдар массиві ретінде қарастырады, және әрбір символды оның индексі арқылы шақырады. Осы char типіндегі қасиет символды өзгертпей, тек қарап шығуға мүмкіндік береді.

Жолдық айнымалылармен жұмыс жасайтын әдістерді қолданатын бірнеше мысалдарды қарастырайық. Мәтіндермен жұмыс жасау барысында міндеттердің бірі мәтінді сөздер массиві түрінде көрсету болып табылады. Мәтіндермен жұмыс ітеу барысындағы мәселенің бірі мәтінді сөздер массив түрінде ұсыну. Осы мәселені шешудің екі жолын қарастырайық – мәтіннің әрбір символын талдау арқылы сөздерді бөлу және Join, Split арнайы әдістерін қолдану арқылы. Split динамикалық әдісі мәтінді элементтерге бөлуді орындайды. Join статикалық әдісі кері операцияны орындайды – элементтерден жолды құрайды.

8.1-есебі. Диалог режимінде 20 сөзден аспайтын мәтін енгізіледі. Барлық сөздерді жеке массивке орналастыру және шығару керек. Бір немесе бірнеше «бос орын» символымен бөлінген кез келген символдар жиыны сөз болып есептеледі.

Бағдарлама коды:

```
using System;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main()
        {
            int j, k;
            string[] clova;
            clova = new string[20];
            string str1 = new string(' ', 0);
            string b = new string(' ', 0);
            string str2 = new string(' ', 0);
            Console.WriteLine("Matindi engiziniz");
            str1 = Console.ReadLine();
            j=0;
            for (int i = 0; i < str1.Length; i++)
            {
                if (str1[i]!=' ')
                {
                    b = str1[i].ToString();
                    str2 = str2 + b;

                }
                else
                if (str2[0] != ' ')
                {
                    clova[j] = str2;
                    str2 = "";
                    j++;
                }
            }
            if (str2[0] != ' ')
            {
                clova[j] = str2;
                str2 = "";
                j++;
            }
            for (int i=0;i<j;i++)
            Console.WriteLine(" {0} soz = {1} ", i+1, clova[i]);
            Console.WriteLine("Enter pernesin basiniz");
            Console.ReadLine();
        }
    }
}
```

```
}  
}  
}
```

Бағдарлама жұмысы:

```
Matindi engiziniz  
20 sozden aspaitin matin engizildi  
1 soz = 20  
2 soz = sozden  
3 soz = aspaitin  
4 soz = matin  
5 soz = engizildi  
Enter pernesin basiniz
```

Осы бағдарламада мына үзінді

```
if (str2[0] != ' ' )  
{  
clova[j] = str2;  
str2 = "";  
j++;  
}
```

екі рет қолданылады. Мәтіннің соңында «бос орын» символы болмауы мүмкін, сондықтан `str2` жолдық айнымалының «бос орын» символына теңдігі тексеріледі. Егер `str2` айнымалы бос орынға тең болмаса, онда `clova[j]` массивіне соңғы сөз қосылады.

Бос орынды мәтінді тергеннен кейін арнайы жолмен қосуға болады, онда тексеруді орындамауға болады.

8.2-есепі. Жолда алгебралық өрнек жақшаларсыз жазылған. Мысалы,  $A \cdot B + A \cdot C \cdot D + C \cdot X$ . Өрнек диалог режимінде енгізіледі. Алгебралық өрнектегі көбейткіштер санын анықтау керек.

Есепті шешудің бір жолы – өрнекті қосылғыштар түрінде көрсету және әрбір қосылғышта көбейту символдарын есептеу. Егер қосылғышта көбейту символы бар болса, онда осы қосылғыштың көбейткіштер саны бірге артады, әйтпесе қосылғышта көбейткіштер жоқ. Барлық қосылғыштардың көбейткіштерін қосып, өрнектегі жалпы көбейткіштер санын табамыз. `Split` динамикалық әдісі мәтінді элементтерге – қосылғыштарға бөледі, бөлгіш ретінде ‘+’ символы қолданылады. Қосылғыштардан тұратын алгебралық өрнекті құрастыру `Join` әдісі арқылы орындалады. `Split` және `Join` әдістері жолдағы жеке сөздермен жұмыс жасауды жеңілдетеді.

Мысалы, `c:\Program Files\Microsoft Visual Studio .NET\readmy.html`. Мұнда жеке сөз болып диск, барлық каталогтар мен файл аттары есептеледі, ал сөзді анықтауда кері көлбеу сызық қолданылады. (жеке сөздер: `c:`, `Program Files`, `Microsoft Visual Studio .NET`, `readmy.html`).

Бағдарлама коды:

```
using System;  
namespace ConsoleApplication1
```

```

{
class Program
{
    static void Main()
    {
        int k, n;
        string[] kosilgish;
        string str1 = new string(' ', 0);
        Console.WriteLine("Algebralik ornekti engiziniz");
        str1 = Console.ReadLine();
        // cлагаемое массивінің өлшемі Split әдісі қайтаратын
        // массивтің өлшеміне сәйкес автоматты түрде анықталады
        kosilgish = str1.Split('+');
        n = 0;
        for (int i = 0; i < kosilgish.Length; i++)
        {
            Console.WriteLine("kosilgish[{0}] = {1}", i + 1,
kosilgish[i]);
            k = 0;
            for (int j = 0; j < kosilgish[i].Length; j++)
            if (kosilgish[i][j] == '*') k++;
            if (k != 0) n = n + k + 1;
        }
        Console.WriteLine("Kobeitkishter sani = {0}", n);
        //алгебралық өрнекті "құрастыруды" орындайық.
        string algebr = string.Join("+", kosilgish);
        Console.WriteLine("Jana ornek = {0}", algebr);
        Console.WriteLine("Enter pernesin basiniz");
        Console.ReadLine();
    }
}
}

```

#### Бағдарлама жұмысы:

```

Algebralik ornekti engiziniz
a * b * c + ab * d + abc + b * b
kosilgish[1]= a * b * c
kosilgish[2]= ab * d
kosilgish[3]= abc
kosilgish[4]= b * b
Kobeitkishter sani =7
Jana ornek = a * b * c + ab * d + abc + b * b
Enter pernesin basiniz

```

Split және Join әдістері тек бір бөлгішті қолданған кезде жақсы жұмыс істейді.

Егер талдау барысында бірнеше бөлгіштер берілсе, онда жолды құрастыру барысында қиындықтар туындауы мүмкін – жолды бастапқы күйдегі қалпына келтіруге болмайды, өйткені қолданылған бөлгіш түрі белгісіз болады.

### 8.3 Өзін-өзі тексеру сұрақтары

1 C# тілінде Unicode жазуы нені білдіреді?

2 C# тілінде escape-тізбегі не үшін қолданылады?

3 Көлденең табуляцияны қандай escape-тізбегі орындайды?

4 `string st1 = new string(' ', 0)`; жазбасы нені білдіреді?

5 Жолдық айнымалылармен жұмыс жасағанда `Split` динамикалық әдісі не үшін қолданылады?

6 C# тілінде `str.Length` қасиеті нені орындайды? Мұндағы `str` – жолдық айнымалы.

7 C# тілінде `string.Compare(str1, str2)` әдісі нені орындайды? Мұндағы `str1, str2` - жолдық айнымалылар.

8 C# тілінде `str1 = string.Concat(str2, str3)`; әдісі нені орындайды? Мұндағы `str1, str2, str3` – жолдық айнымалылар.

9 Бағдарламаның келесі үзіндісі нені шығарады:

```
string clovo = "abcdefg";
```

```
Console.WriteLine(" {0}", clovo.Length); ?
```

10 Бағдарламаның келесі үзіндісі нені шығарады?

```
char b;
```

```
int i, n;
```

```
string clovo = "abcadeafg";
```

```
b = clovo[0]; n = 0;
```

```
for (i = 0; i < clovo.Length; i++)
```

```
    if (clovo[i] == b) n++;
```

```
Console.WriteLine(" {0}", n); ?
```