3-Дәріс. Енгізу-шығару әдістері және C# тілінің операциялары

Дәрістің мақсаты:

- Мәліметтер типтерін (value/reference), сандық, логикалық, жолдық және т.б. түрлерін ажырату.
- Консольмен енгізу/шығаруды (Console.ReadLine/WriteLine), форматтауды қолдану.

Негізгі тақырыптар

1. Операторлар және приоритет: арифметикалық, салыстыру, логикалық, биттік, меншіктеу, тернарлық; унарлық/бинарлық.
2. Консольдік енгізу/шығару: Console.Write/WriteLine, Read/ReadLine; Convert/Parse арқылы түрлендіру.
3. Форматтау: интерполяция/орын-дәрежелік форматтау {0,10:fmt}, ақша бірлігі C, үлгілер #.## т.б.
4. Қарапайым бағдарлама құрылымы: namespace, class Program, static void Main(string[] args).

C# тілінде программалауды бастау. Кез келген программада айнымалыларды сипаттап, операторлар мен операцияларды жүзеге асырумен бірге мәліметтер пернелерден енгізіліп, экранға шығарылады. Мұндай стандартты енгізу-шығару құрылғылары консоль деп аталады.

C# тілінде енгізу-шығару операторлары жоқ, оның орнына стандартты объектілер қолданылады. Консольмен жұмыс істеу үшін Console класы пайдаланылады, ол System атаулар кеңістігінде анықталған. Осы кластың Write және WriteLine әдістері экранға мәлімет шығару кезінде қолданылған болатын. Төмендегі мысалда Console класының WriteLine әдісін қарастырайық:

```
using System;
namespace algashqy_sabaq {
  class Program  {
       static void Main(string[] args) {
         // "c# бағдарламалау тілінде программалауды бастаймыз!" сөзін экранға
шығару
          Console.WriteLine("c# бағдарламалау тілінде программалауды бастаймыз!");
          Console.ReadKey();
       }
  }
       }
```

Программа кодын әрбір жол бойынша қарастырып шығайық. Оның алғашқы төрт жолы автоматты түрде экранға программаның бастапқы үлгісі ретінде құрастырылып шығады. Олар **using** сөзінен басталып, алдын ала келісім бойынша, біздің жобамызға қандай кітапханалардың қосылатынын мәлімдейді.

Ары қарай біздің өз атаулар кеңістігіміз ашылады, оның аты осы ашқан жобамыздың атымен бірдей болады. Атаулар кеңістігі негізінде көлемді программалық жобалар үшін керек, ол идентификаторларды логикалық түрде топтау үшін қажет болады. Бір идентификатор бірнеше атаулар кеңістігінде, бір-бірінен тәуелсіз түрде қолданыла береді.

Келесі программа жолы: **class Program**. Мұнда алдымен C# тілінің объектіге бағытталған программалау (ОБП) тілдері тобынан екендігін еске түсірейік. ОБП ортасында барлық мәліметтер (айнымалылар) және оларды өңдейтін процедуралар мен функциялар (әдістер) кластарға біріктіріліп отырады. **Main** атты әдіс программаға бастапқы кіру нүктесі болып саналады, яғни программаны орындау осы **Main** әдісінда жазылған нұсқаулардан басталады. Ал // таңбаларынан басталып жазылған мәтін түсініктеме түріндегі комментарийлер.

Бұл мысалдағы негізі атқарылатын қызмет мәтінді консулға шығару. Ол үшін **Console.WriteLine** әдісі шақырылып, «c# бағдарламалау тілінде программалауды бастаймыз!» атты мәтінді экранға шығарады. Соңында **Console.ReadKey** – консулды ұстап тұратын әдіс.

Бұл біздің C# тілінде бағдарламалау пәні бойынша алғашқы программамыз. Бұл мысалда Console класының WriteLine әдісі арқылы мәліметтерді экранға шығаруды қарастырдық. Алда Console.WriteLine әдісі біздің барлық мысалдарда мәліметтерді шығаруға қолданылады. Сонымен бірге осы дәрісте Console класының мәліметтерді енгізу әдістерінде қарастырамыз. Оның алдында C# тілінің операцияларына тоқаталйық.

C# тілінде де өзге программалау тілдері сияқты қосу (+), азайту (-), көбейту (*), бөлу (/) 4 арифметикалық операция сандық типтегі мәліметтер үшін қолданылады. Алдыңғы үш операция бүтін сандық типке қолданылса нәтижесі де бүтін сандық тип болса, бөлу амалында нәтижені нақты сан типіне түрлендіру жағдайлары болады. Бұл арифметикалық операциядан бөлек тағы бір ең көп қолданатын операция **Қалдық табу** (%) *операциясы* көбінесе бүтін типтегі шамаларға қолданылады. Егер екі операнд та бүтін сан болса, нәтиже x-(x/y)*y формуласымен анықталады. Егер бір операнд нақты сан болса, нәтиже x-n*y (n – x-ті y-ке бөлгендегі бүтін сан) формуласымен анықталады. Мысалы:

```
using System;
namespace ConsoleApplication1
{  class Program
   {  static void Main()
      {  int  x = 11, y = 4;
      float  z = 4;
      Console.WriteLine(z * y);           // Нәтиже  16
      Console.WriteLine(z * 1e308);   // Нәтиже  шексіздік
      Console.WriteLine(x / y);           // Нәтиже  2
      Console.WriteLine(x / z);           // Нәтиже  2,75
      Console.WriteLine(x % y);          // Нәтиже  3
}}}
```

Осы және басқада C# тілінің негізгі операциялары олардың *приоритеттері* бойынша келесі кестелерде берілген. Операндтарының санына қарай олар унарлық, бинарлық және тернарлық болып бөлінеді.

Операция	**Қысқаша сипаттамасы**
++	1-ге арттыру
--	1-ге кеміту
~	разрядтар бойынша терістеу
!	логикалық терістеу
-	арифметикалық терістеу (унарлық минус)
+	унарлық плюс
new	жадыны бөлу
typeof	типті алу
checked	тексерілетін код
unchecked	тексерілмейтін код
(type)x	типті түрлендіру

1.4 – кесте. Унарлық (бір орынды) операциялар

Операция	**Қысқаша сипаттамасы**
/	бөлу
%	бөлгендегі қалдық
+	қосу
-	азайту
<<	солға ығыстыру

>>	оңға ығыстыру
<	аз
<=	аз немесе тең
>	көп
>=	көп немесе тең
is	типке жататындығын тексеру
as	типті келтіру
==	тең
!=	тең емес
? :	шартты операция (тернарлық)
*=	көбейтіп алып меншіктеу
/=	бөліп алып меншіктеу
%=	бөлгендегі қалдықты тауып меншіктеу
+=	қосып алып меншіктеу
-=	азайтып алып меншіктеу
<<=	солға ығыстырып меншіктеу
>>=	оңға ығыстырып меншіктеу
&=	разрядтар бойынша ЖӘНЕ операциясын орындап меншіктеу
|=	разрядтар бойынша НЕМЕСЕ операциясын орындап меншіктеу
^=	разрядтар бойынша арифметикалық НЕМЕСЕ операциясын орындап меншіктеу

1.5 – кесте. Бинарлық (екі орынды) және тернарлық (үш орынды) операциялар

1.4 және 1.5 – кестегі операциялар c# тіліндегі жиі қолданылатын операциялар болып табылады. Алдағы дәрістерде бұл операцияларды нақты қолдану барысында толық түсінетін боламыз.

Console класында мәліметтерді енгізу. Console класында мәліметтерді 3 енгізу әдісі (Read, ReadLine) анықталған. Read әдісі консольден бір символ енгізуді орындайды. Ол жеке символды int типінде қабылдайды. ReadLine әдісі string типтегі символдар тіркесін енгізуге арналған. Сондықтан сан енгізу екі кезең арқылы жүзеге асырылады:

1. Сандар сөз тіркесі ретінде пернелерден **string** типті айнымалыға енгізіледі.
2. Содан соң сөз тіркесі керекті сандық типке түрлендіріледі.

Түрлендіру **System** атаулар кеңістігінде анықталған арнайы **Convert** класы көмегімен немесе әрбір стандартты арифметикалық кластың құрамында бар **Parse** әдісімен орындалады.

Келесі программа қарапайым арифметикалық амалдарды әртүрлі бастапқы мәліметтер арқылы орындау операцияларынан тұрады.

1.2-мысал. a мен **b** бүтін сандарын қосу, азайту, көбейту мен бөлуді орындау

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ArifOperation1 {
  class Program {
    static void Main(string[] args) {
        int a, b, plus, minus;
        // мәліметтердісипаттау
        double del, mult;
        Console.Write("Input ia: ");
```

```
        // a сөз тіркесін оқып, оны бүтін санға түрлендіру
        a = int.Parse(Console.ReadLine());
        Console.Write("Input ib: ");   //b тіркесін енгізу
        b = int.Parse(Console.ReadLine());
        // есептеулерді орындау
        plus = a + b;
        minus = a - b;
        mult = a * b;
        del = a / b;
        // сандық мәндерді экранға шығару
        Console.WriteLine("plus = " + plus);
        Console.WriteLine("minus = " + minus);
        Console.WriteLine("del = " + del);
        Console.WriteLine("mult = " + mult);
        Console.ReadLine();
    }
}
        }
```

Бұл мысалда a және b бүтін типтегі айнымалыларды Console.ReadLine() әдісі арқылы пернетақтадан енгіздік. Біздің енгізген симолдарымыздың консул string типте қабылдайтындықтан Parse әдісін қолдаып a және b айнымалысының мәнін бүтін санға (int типіне) түрлендірдік. Console.WriteLine("plus = " + plus) мәліметті шығару үшін қолданған WriteLine әдісінде "plus = " тырнақшаның ішіндегі мәтіннің өзін ал plus айнымалысының мәнін шығарады. Бұл жердегі + белгісі екі тіркесті біріктіру қызметін атқарады. Ал программаның қалған бөлігі алдыңғы мысалдарда айтылған.

Форматтау спецификаторлары. Console класында мәліметтерді шығару барысында форматтау спецификаторлары қолданылады. Форматтау спецификаторы мынадай болады:

{argnum, width: fmt} мұндағы *argnum* – нөлден бастылып жазылатын шығарылатын аргумент нөмірі; *width* – шығару өрісінің (санның) ені; *fmt* – формат, *width* және *fmt* параметрлерін жазбай кетсе де болады.

Егер жақша ішінде формат спецификаторы кездессе, оның орнына кезектегі аргумент *argnum* параметрі жазылады. *width* және *fmt* параметрлерін көрсетпесек, ол тек келесі аргументті ғана белгілейді. Мысалы, {0} спецификаторы *arg0* аргументін, {1} спецификаторы – *arg1* аргументін, т.с.с. белгілейді.

Мысалы, келесі операторды орындасақ,

WriteLine("Aqpanda {0} nemese {1} kun bolady", 28, 29);

нәтиже мынадай болады:

Aqpanda 28 nemese 29 kun bolady

мұнда 28 **{0}** спецификаторы орнына, ал 29 – **{1}** спецификаторы орнына жазылады. Спецификатор аргументтің өзіне сәйкес орнын (28 және 29) білдіреді. Қостырнақшадан кейін **+** емес **,** қойылады.

Келесі жолда осы оператордың орналасатын сандардың енін көрсететін түрі келтірілген.

WriteLine("Aqpanda {0,10} nemese {1,5} kun bolady", 28, 29);

Мұның нәтижесі:

Aqpanda 28 nemese 29 kun bolady

Санның ені берілген мәннен аз болса, олар босорындармен толтырылады, *width* параметрі санның ең кіші енін береді, егер санның ені одан артық болса, ол толық шығарылатын болады. Аргументтер константа сияқты сандық мән түрінде емес, айнымалы немесе өрнек түрінде бола береді.

Көбінесе жылжымалы нүктелі сандар мен ондық сандар форматтауды талап етеді.

Мәліметтерді форматтаудың ең қарапайым түрі – **WriteLine()** әдісінде қолданылатын шаблонды (үлгіні) көрсету. Ол үшін . # символдары арқылы сан разрядтарын белгілейтін керекті форматтың ұлгісі беріледі.Оған қоса, цифрларды бөліп тұратын ондық нүктені (үтірді) көрсетуге де болады. Төменде 10-ды 3-ке бөлгендегі нәтиженің қажетті форматы көрсетілген.

Console.WriteLine("10/3 = {0:#.##}", 10.0/3.0);

Бұл оператор нәтижесі:

10/3 = 3.33

Бұл мысалдағы **#.##** шаблоны санның бөлшегін екі разрядпен беруді көрсетіп тұр, нүктенің сол жағындағы санның бүтін бөлігінің де енін беруге болады.

Келесі мысалдағы мынадай оператор:

Console.WriteLine("{0:###,###.##}", 123456.56);

нәтижесі:

123,456.56

Ақша мөлшерін белгілейтін сандарда форматтаудың C спецификаторын көрсету ұсынылады. Мысалы:

decimal balance;
balance = 12323.09m;
Console.WriteLine("Тквартал балансы = {0:C}", balance);

Мұның нәтижесі АҚШ долларымен көрсетіледі:

Квартал балансы = $12,323.0 9

Мысалы жеңілдіктер арқылы бағаны есептеу программасындағы нәтижені C форматымен жазу былай орындалады:

```
// Жергілікті валютада C форматы спецификаторын қолдану
using System;
class UseDecimal
{
  static void Main()
  {
    decimal price;
    decimal discount;
    decimal discounted_price;
    // бағаны жеңілдікпен есептеу
    price = 19.95m;
    discount = 0.15m;                    // жеңілдік нормасы  15%
    discounted_price = price - (price * discount);

    Console.WriteLine("Zhengildikpen berilgen baga: {0:C}",
                      discounted_price);
  }
}
```

```
Zhenildikpen berilgen baga: 16,96т.
```

Бақылау сұрақтары

1. float, double, decimal типтерінің айырмашылығы неде және әрқайсысы қайда тиімді?
2. Негізгі оператор топтарын (арифметикалық, салыстыру, логикалық, меншіктеу, тернарлық) атаңыз және приоритет туралы айтыңыз.
3. Консольден сан енгізудің екі кезеңдік үлгісін (ReadLine + Parse/Convert) түсіндіріңіз.

4. Форматтау спецификаторлары {index,width:format} қалай жұмыс істейді? Мысал келтіріңіз.
5. Visual Studio мен командалық жол арқылы компиляциялаудың айырмашылығы.
6. Қарапайым «енгізу–есептеу–шығару» бағдарламасының құрылымын сипаттаңыз.

Ұсынылатын әдебиеттер

1. Колдаев, В.Д. Основы алгоритмизации и программирования: Уч. пособие/ под ред. проф. Л.Г.Гагариной. -М.: ИД «ФОРУМ»: ИНФРА-М,2015.-416 с.
2. Бөрібаев Б. Компьютердің арифметикалық негіздері: Жоғары оқу орындары студенттеріне арналған оқу құралы. -Алматы: Қазақ университеті, 2009. -80 б.
3. Бөрібаев Б. Алгоритмдеу, мәліметтер құрылымы және программалау тілдері: оқулық. –Алматы: Қазақ университеті, 2012. -235 б.
4. Вирт Н. Алгоритмы и структуры данных: Пер. с англ. -М.: Мир, 2011.-360с.
5. Бөрібаев Б., Абдрахманова М. C# тілінде программалау (практикалық курс): оқу құралы –Алматы: Қазақ университеті, 2018. -258 б.
6. Troy Dimes. C# Programming for Beginners. Paperback – January 25, 2015.
7. Шилдт Г. C# 4.0: полное руководство. М.: ООО ”И.Д.Вильямс”, 2017. -1056 с.
8. Джуст В. Разработка обслуживаемых программ на языке C#. СПб.: Изд. дом «Питер», 2017.
9. Шарп Д. Microsoft Visual C#. Подроб. руководство. 8-е изд.С-Петербург: Изд. дом «Питер», 2016.
10. A. Troelsen, P. Japikse. Pro C# 7: With .NET and .NET Core. – Apress, 2017. -1372 p.