1-і тақырыптың жалғасы. C# ТІЛІНІҢ ҚАРАПАЙЫМ ОПЕРАТОРЛАРЫ.

3-дәріс. C# ТІЛІНДЕГІ БАҒДАРЛАМА ҚҰРЫЛЫМЫ, ЕНГІЗУ–ШЫҒАРУ ОПЕРАТОРЛАРЫ

3.1 C# ТІЛІНДЕГІ БАҒДАРЛАМА ҚҰРЫЛЫМЫ

Есепті шешуге арналған атаулар кеңістігін анықтап алғаннан кейін бағдарлама класын сипаттау басталады.

Класс дегеніміз бұл деректерді (өрістерді) және функцияларды (әдістерді) біріктіретін, осы деректерді біртұтас өңдейтін – код үзіндісі (тип).

Анықтама бойынша кез келген бағдарлама деректер мен оларды өңдеу әдістері бар код болып есептеледі. Сондықтан C# тілінде түрлі есептерді шешетін бағдарлама кодтары бағдарлама кластарына орналастырылады. Класс сипаттамасы `class` қызметтік сөзінен кейін басталады, қызметтік сөзден кейін класс атауы, одан кейін фигуралық жақшаның ішіне деректер мен оларды өңдеу әдістері орналастырылады.

C# тіліндегі бағдарлама класының сипаттамасының ішінде `Main()` атауы бар әдіс орналасуы керек. Осы әдістен бастап бағдарламаның орындалуы басталады. Назар аударыңыз, C# тілінде бас және кіші әріптер ажыратылады.

Егер есеп қарапайым болса, онда барлық әдістер `Main()` әдісінде ғана жазыла алады. Егер есепті бөлшектеу қажет болса (яғни басқа кластармен немесе әдістермен орындалатын бөлек үзінділерге бөлу), онда бағдарлама класында осы үзінділерді тиісті түрде сипаттау керек, ал `Main()` әдісінде оларды іске қосу ретін көрсетуіңіз керек.

3.2 Бағдарлама мысалы

Бағдарлама кодын дайындау үшін компьютерде C# бағдарламау тілінің компиляторын орнату керек. Қазіргі уақытта C# тілінің бірнеше компиляторлары белгілі, мысалы, Visual Studio 2008 визуалды бағдарламалау ортасы, Turbo C# Explorer, т.б.

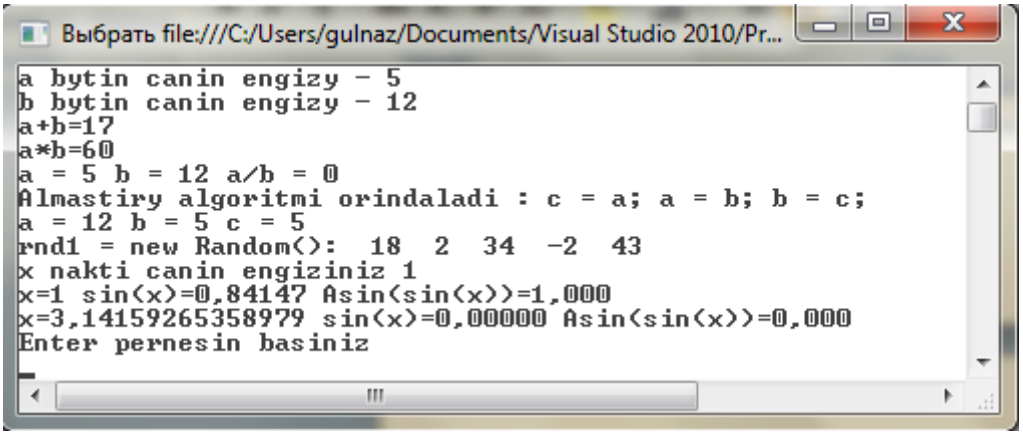
Әрбір компилятордың бағдарламаны дайындаған кезде ескеретін өз ерекшеліктері бар. Дәріс мазмұны Visual Studio 2008 визуалды бағдарламалау ортасында C# компиляторын қолдануға бағытталған.

Оқуға арналған бағдарлама кодын қарастырайық. Бағдарлама a және b бүтін айнымалылардың мәндерін енгізуге (диалогтық тәртіпте) және осы айнымалыларды пайдаланып, арифметикалық амалдарды орындауға мүмкіндік береді. Бағдарламада толық сипаттамасы келесі дәрістерде қарастырылатын кейбір операторлар бар.

```
using System;
namespace ConsoleApplication1
```

```
{
    class Program
    {
        static void Main(string[] args)
        {
            int a, b, c;
            double x, y, z;
            string buf;
            Console.Write("a bytin canin engizy - ");
            buf = Console.ReadLine();
            a = Convert.ToInt32(buf);
            Console.Write("b bytin canin engizy - ");
            buf = Console.ReadLine();
            b = Convert.ToInt32(buf);
            c = a + b;
            Console.WriteLine("a+b={0}", c);
            c = a * b;
            Console.WriteLine("a*b={0}", c);
            c = a / b;
            Console.WriteLine("a = {0} b = {1} a/b = {2}", a, b,
c);
            Console.WriteLine("Almastiry algoritmi orindaladi :
c = a; a = b; b = c;");
            c = a; a = b; b = c;
            Console.WriteLine("a = {0} b = {1} c = {2}", a, b,
c);
            Random rnd = new Random();
            Console.Write("rnd1 = new Random():");
            for (int i = 1; i <= 5; i++)
            {
                a = rnd.Next() % 101 - 50;
                Console.Write("  " + a.ToString());
            }
            Console.WriteLine();
            Console.Write("x nakti canin engiziniz ");
            buf = Console.ReadLine();
            x = Convert.ToDouble(buf);
            y = Math.Sin(x);
            z = Math.Asin(y);
            Console.WriteLine("x={0} sin(x)={1:F5}
Asin(sin(x))={2:F3}", x, y, z);
            x = Math.PI;
            y = Math.Sin(x);
            z = Math.Asin(y);
            Console.WriteLine("x={0} sin(x)={1:F5}
Asin(sin(x))={2:F3}", x, y, z);
            // Задержка рабочего экрана монитора
            Console.WriteLine("Enter pernesin basiniz");
            Console.ReadLine();
        }
    }
}
```

3.2-суретінде бағдарлама жұмысы көрсетілген.



3.2-суреті– Бағдарлама жұмысының нәтижесі «1-мысал»

Қалған мысалдарда бағдарлама жұмысын көрсету жұмыс терезесінің көшірмесімен ауысады, мысалы:

```
a bytin canin engizy - 5
b bytin canin engizy - 12
a+b=17
a*b=60
a = 5 b = 12 a/b = 0
Almastiry algoritmi orindaladi : c = a; a = b; b
= c;
a = 12 b = 5 c = 5
rnd1 = new Random(): 18 2 34 -2 43
x nakti canin engiziniz 1
x=1 sin(x)=0,84147 Asin(sin(x))=1,000
x=3,14159265358979                    sin(x)=0,00000
Asin(sin(x))=0,000
Enter pernesin basiniz
```

`Main()` әдісі `public` және `static` деген екі модификаторымен анықталған (көбінесе қол жеткізу спецификаторы деп аталады).

`public` - ашық модификаторы әдістің бағдарлама ішінде немесе сыртында басқа әдістерге қол жеткізімді екенін көрсетеді.

`Main()` әдісі `Program` класында орналасады және әдетте кластың әдістеріне класс типті айнымалыларды, яғни объектіні құрғаннан кейін ғана қол жеткізуге мүмкіндік бар. Бірақ, егер әдіс `static` модификаторымен жарияланса, онда оны «класс деңгейінде» қолдануға болады, яғни класс объектісін құрмай.

`Main()` әдісінің дөңгелек жақшаларының ішінде жолдар массивы түрінде берілген кіріс параметрлері анықталған, олар арқылы нұсқаулар бағдарламаның командалық жолынан әдіс іске қосылғанда жеткізіле алады.

Біздің бағдарламада бұл параметрлер қолданылмайды, сондықтан оларға назар аудармауға немесе өшіріп тастауға болады және `Main()` әдісінің қысқартылған түрде жазылуы қалтырылады.

Бағдарламада монитор экранындағы кідіріс бойынша түсініктеме берілген. Бағдарлама түсініктемесін `//` белгісімен бір жолға немесе `/* . . . . */` белгілері арқылы бірнеше жолдарға жазуға болады.

`Console.ReadLine();` нұсқауы (бағдарлама соңында) бағдарлама жұмысының нәтижесін көру үшін «бағдарламаның жұмыс терезесін тоқтату» амалын орындайды, сонымен қатар `Console.WriteLine("Для продолжения нажмите клавишу Enter");` нұсқауы «Для продолжения нажмите клавишу Enter» хабарламасын шығарады.

Қарастырылып отырған мысалды `Main()` функциясы үш негізгі әрекетті орындайды:

– монитор экранына әзіржауапты және бағдарлама жұмысының нәтижесін шығару;

– диалогтық режімде перне арқылы айнымалылардың мәндерін енгізу;

– кейбір арифметикалық өрнектерді есептеу және `c, x, y, z` айнымалыларына есептеулер нәтижелерін меншіктеу.

Әрбір әрекетті жеке қарастырайық.

3.3 Монитор экранына ақпаратты шығару

C# тілінде консольді қосымшалармен жұмыс жасау барысында монитор экранына шығару `Console.WriteLine()` және `Console.Write()` статикалық әдістері арқылы орындалады (дәлірек айтқанда `Console` класының `WriteLine()` және `Write()` статикалық әдістері).

Ақпаратты шығарғаннан кейін `Console.WriteLine()` әдісі терезе курсорын келесі жолдың басына көшіруді орындайды.

Ақпаратты шығарғаннан кейін `Console.Write()` әдісі көрсетілген соңғы символдан кейінгі орында терезе курсорын бірден қалдырады.

Ескеру керек, екі әдіс те монитор экранына ақпаратты символды түрде ғана шығара алады, яғни бүтін немесе нақты типтегі мәндерді монитор экранына шығармас бұрын жолдық типке түрлендіру керек. Айнымалылардың сандық мәнін жолдық мәнге түрлендіру осы әдістердің әрқайсысында орындаған, сонымен қатар түрлендіруді берілген форматта жүргізуге болады.

Әдетте әдістердің бірінші параметрі арқылы түсініктеме мәтіні беріледі, онда фигуралы жақша арқылы «толтырғыш» жазылады. «Толтырғыштар» санына шек жоқ және есептің талаптарымен ғана анықталады. Барлық «толтырғыштар» нөлден бастап нөмірленеді.

Толтырғыштың нөмірінен кейін қос нүкте арқылы шығару форматының спецификаторын көрсетуге болады.

Шығару бойынша әдістерде бірінші параметрден кейін және формат арқылы айнымалылардың атаулары көрсетіледі. Олардың мәндері шығарылатын мәтінде «толтырғыштардың» орнына тиісті форматта орналастырылады. Әрине, айнымалылыр мен «толтырғыштардың» саны бірдей болуы керек. Мысалы,

```
Console.WriteLine("a = {0} b = {1} a/b = {2}", a, b, c);
Console.WriteLine("x={0} sin(x)={1:F5} Asin(x)={2:F3}",x, y, z);
```

Бірінші жағдайда шығарылатын мәтінде `{0}, {1}` және `{2}` орнына `a, b` және `c` айнымалыларының мәндері пішімдеусіз жазылады. Екінші жағдайда `sin(x)` және `Asin(x)` өрнектерін шығару сандарды пішімдейтін `F` спецификаторы арқылы анықталады және ол үтірден кейін 5 және 3 дейінгі дәлдікпен сәйкесінше орындалады. Кейбір пішімдеу спецификаторлары 3.1 кестесінде көрсетілген.

3.1- кестесі – Сандарды пішімдеу спецификаторлары

Спецификатор	Қызметі
C немесе c	Сандарды ақшалай пішімде шығару үшін қолданылады
D немесе d	Ондық сандарды шығару үшін қолданылады. Символдан кейін үтірден кейін шығарылатын символдар санын көрсетуге болады.
E немесе e	Сандарды экспоненциалдық пішімде шығару үшін қолданылады
F немесе f	Нақты санды «белгіленген нүктесімен» шығарады. Символдан кейін үтірден кейін шығарылатын символдар санын көрсетуге болады.
G немесе g	Ортақ (general) пішім. Шығарылатын санның өлшеміне байланысты F немесе E пішімі қолданады.
N немесе n	Санды шығаруда мыңдықты бөлектеу үшін қолданады. Бөлектегіш ретінде үтір қолданылады.
X немесе x	Бүтін санды он алтылық пішімде шығарады. Егер пішім үлкен символмен анықталса, онда он алтылық пішімдегі барлық әріптер үлкен болады.

Монитор экранына ақпаратты шығарудың қалған жолдары материалды ары қарай оқу барысында қарастырылады.

3.4 Пернетақтадан деректерді енгізу

C# тілінде пернетақтадан деректерді енгізу үшін (диалог режимі немесе интерактивті режим) консольді, статистикалық `Console.ReadLine()`

және `Console.Read()` әдістері қолданылады. Осы әдістер бағдарламаның орындалуын тоқтатады және компьютер пернетақтасынан деректерді енгізуді күтеді. Ескеру керек, `Console.ReadLine()` әдісі `string` типті айнымалыны, ал `Console.Read()` әдісі `int` типті айнымалыны қайтарады. Екі әдісте де жаңғырық функциясы бар, енгізілген ақпаратты монитор экранына қайталайды.

Пернетақтадан енгізілген ақпарат жадының арнайы аймағына - буферіне жазылады. Буферге енгізудің аяқталуы Enter пернесін (клавиша) басу арқылы орындалады (сонымен бірге осы перне кодтары буферге жазылады).

`Console.ReadLine()` әдісінің нәтижесі жолдық айнымалының мәні болады (осы жағдайда buf айнымалысы). Енгізу аяқталғаннан кейін әдіс жады буферінің ішіндегісін өшіреді.

`Console.Read()`әдісі Консольді енгізуге бөлінетін жады буферінен тек бір ғана символды алады және оны бүтін санға түрлендіреді. Енгізу аяқталғаннан кейін әдіс жады буферінің ішіндегісін өшірмейді. Мысалы, бірінші бағдарлама үзіндісі.

```
Console.Write("a bytin canin engizy - ");
buf = Console.ReadLine();
a = Convert.ToInt32(buf);
Console.Write("b bytin canin engizy - ");
buf = Console.ReadLine();
b = Convert.ToInt32(buf);
c = a + b;
Console.WriteLine("a+b={0}", c);
```

мына кодқа ауыстырылса:

```
Console.Write("a bytin canin engizy - ");
a = Console.Read();
Console.Write("b bytin canin engizy - ");
buf = Console.ReadLine();
b = Convert.ToInt32(buf);
c = a + b;
```

«a bytin canin engizy - » сөйлемінен кейін 47 саны енгізілсе, a айнымалысына 52 мәні меншіктеледі (символдың коды - 4), b айнымалысына 7 саны, ал c қосындысының нәтижесі 59 болады. b айнымалысы үшін монитор экранында кідіріс болмайды, b айнымалысына консольда енгізуге арналған буфердің қалған бөлігі меншіктеледі.

C# тілінде бағдарламалауды меңгеру барысында Console.Read() әдісін қолдануға кеңес берілмейді (жолдық айнымалыларды оқуға дейін).

3.5 Тексеру үшін сұрақтар

3.5.1 Төменде көрсетілген нұсқалардың қайсысы C# тілінде клавиатурадан енгізуді қамтамасыз етеді?

A) Console.WriteLine(“Введите значение a”,a);
B) Console.WriteLine(a);
C) buf = Console.ReadLine(a);
D) buf = Console.ReadLine();
E) Console.ReadLine(a);

3.5.2 Төменде көрсетілген нұсқалардың қайсысы C# тілінде айнымалылардың мәндерін монитор экранына шығаруды қамтамасыз етеді?

A) buf = Console.WriteLine(a,b,c);
B) Console.WriteLine(a,b,c);
C) Console.WriteLine(" {0} {1} ", a, b);
D) Console.WriteLine(“Вывод значений ”,a,b,c);
E) Console.ReadLine(a,b,c);

3.5.3 using қызметтік сөзі не үшін қолданылады?
3.5.4 C# тілінде программаның негізгі әдісі қалай аталады?
3.5.5 айнымалы ұғымы?

1- тақырыптың жалғасы. C# ТІЛІНДЕГІ ҚАРАПАЙЫМ ОПЕРАТОРЛАР.

4-ДӘРІС. C# ТІЛІНДЕГІ МЕНШІКТЕУ ОПЕРАТОРЛАРЫ, МАТЕМАТИКАЛЫҚ ФУНКЦИЯЛАР

4.1 Меншіктеу операторы

C# бағдарламалау тілінде меншіктеу операторы ‘=’ символымен белгіленеді. Меншіктеу операторының жазылу пішімі мына сипатта болады:

айнымалы = өрнек;

Мысалы:

c = a + b;

‘=’ белгісінің сол жағында айнымалы орналасуы керек, ал ‘=’ белгісінің оң жағында айнымалылар, тұрақтылар, нақты мәндер немесе өрнектер жазылуы мүмкін. Өрнектер ретінде арифметикалық операциялар, әртүрлі функциялар, т.б. бола алады. Өрнектегі әрекеттердің орындалу реті солдан оңға қарай, бірақ жақшалар арқылы оны өзгертуге болады.

C# тілінде келесі арифметикалық операциялар қолданылады:

+ – қосу операциясы;

- – азайту операциясы;

* – көбейту операциясы;

/ – нақты типтегі айнымалылар үшін бөлу операциясы;

/ – бүтін типтегі айнымалылар үшін бүтін санды бөлу операциясы (қалдығы жойылады – бөлудің нәтижесін мысалдан көріңіз);

% – бүтін санды бөлу операциясынан қалдықты табу операциясы бүтін типтегі айнымалылар үшін қолданылады.

Мысалы,

7/2=3; 7.0/2.0=3.5; 6%2=0; 7%2=1;

C# тілінде аталған операциялардың кез келгені меншіктеу операциясының қысқартылған түрде жазылу мүмкіндігін береді:

айнымалы операция = өрнек;

Мысалы, егер а айнымалысы 5 тең болса, онда а += 6 жазбасы а айнымалысына 11 мәнін меншіктейді. ‘+=’ операциясы ‘=’ символының оң жағына жазылған өрнектің мәнін ‘+’ символының сол жағында орналасқан айнымалы мәніне қосуды орындайды. Қосу опрациясының нәтижесі ‘+’ символының сол жағында орналасқан айнымалыға меншіктеледі.

‘–’ , ‘ *’, ‘ /’, ‘ %’ операциялары үшін меншіктеу операторы жоғарыда сипатталған түрде орындалады.

Мысалы, егер а айнымалысы 5 тең болса, онда а %= 3; меншіктеу операторы орындалғаннан кейін а айнымалысы 2 тең болады .

Алгебралық өрнектің нәтижесін кейбір айнымалыға меншіктеу операциясын орындаған кезде типтердің айқын немесе айқындалмаған түрде түрленуі орындалады.

Егер алгебралық өрнектің операндтары әртүрлі типте болса, онда есептеулердің алдына автоматты түрде типтерді түрлендіру орындалады. Түрлендіру мәні мен дәлдігін сақтай отырып, қысқа типтерді ұзын типтерге ауыстыруды қамтамасыз ететін ереже бойынша орындалады, бірақ керісінше емес.

Автоматты түрде (айқындалмаған) түрлендіру ылғи орындалмауы мүмкін, мәні жоғалмаған жағдайда ғана орындалады.

Арифметикалық операциялар int типінен қысқа типтерде анықталмаған. Бұл дегеніміз, егер өрнекте sbyte, byte, short және ushort типтеріндегі ғана шамалар болса, онда операцияның орындалуының алдында олар int типіне түрлендіріледі. Сонымен, кез келген арифметикалық операциялардың нәтижесі int типінен кіші емес типте болады.

Бір типтен екінші типке айқындалмаған түрлендіру орындалмаса, онда бағдарламашы мына операция арқылы айқын түрлендіруді орындай алады.

(тип) өрнек.

Мысалы, бүтін типтегі айнымалыға нақты типтегі мән меншіктелсе, онда осы мәннің бөлшек бөлігі жойылады. Типтерге айқын түрлендіруді орындау үшін float немесе int нұсқауларын пайдалануға болады. Мысалы:

```
x = (float)(31*c - 16);
Console.WriteLine("x={0}", x);
//или
a = (int)(18.6/3.6 + 3.7);
Console.WriteLine("a={0}", a);
```

Типтерді айқын түрлендіруді бағдарламаға айқындылықты кіргізу үшін әр түрлі типтегі айнымалылары бар есептеулерде қолдану орынды, бірақ айқын түрлендіру есептеудің дәлдігін жоғалтумен байланысты екенін ескеру керек.

C# тілінде инкремент (айнымалыны бір шамасына өсіру) және декремент (айнымалыны бір шамасына кеміту) операциялары үшін қысқартылған жазбалар қолданылады, яғни меншіктеу операциялары.

```
айнымалы = айнымалы + 1;
мына жазбамен ауыстырылады
айнымалы ++;  ,
меншіктеу операциясы
айнымалы = айнымалы - 1;
мына жазбамен ауыстырылады
айнымалы --; .
```

Мысалы, егер а айнымалысының мәні 5-ке тең болса, онда меншіктеу а++; операторын орындағаннан кейін а айнымалысының мәні 6-ға тең болады. Егер енді а--; меншіктеу операторы орындалса, онда а айнымалысының мәні 5-ке тең болады.

C# тілінде инкремент және декремент операциялар жазбасының префиксті пішіні рұқсат етілген, мысалы ++a немесе --a, бұл жазбалар жоғарыда қарастрылған постфиксті пішімдегі жазбалардан өзгеше.

Жазбаның префиксті пішімі біріншіден инкремент немесе декремент операцияларының орындалуын, ал содан кейін айнымалыны пайдалануды талап етеді. Постфиксті пішімдегі жазулар біріншіден айнымалыны, одан кейін инкремент немесе декремент операцияларының орындалуына рұқсат береді. Мысалы, a = b++; өрнегін келесі екі оператормен ауыстыруға болады: a = b; b = b + 1;. C# тілінде a = ++b; өрнегі келесі екі операторға балама болады: b = b + 1; a = b;.

C# тілінің осындай ерекшеліктерін бағдарлама кодын жазу кезінде ескеру керек.

C# тілінің басымдылықтары бойынша реттелген негізгі операцияларының тізімі ұсынылған әдебиетте келтірілген.

Қарастырылған бірінші бағдарламаның жолдарының бірінде

c = a; a = b; b = c;

өте маңызды алгоритм жазылған - алмасу алгоритмі . Бұл алгоритм a, b және c айнымалыларының мәндерін алмастыруға мүмкіндік береді. Осы алгоритм басқа да күрделі алгоритмдерде қолданылады, мысалы сандарды сұрыптау немесе тізімді алфавиттік ретте пішімдеу, т.б және осы алгоритмді түсіну өте маңызды.

4.2 C# тілінің стандартты математикалық функциялары

Кез келген бағдарламалау тілдері сияқты C# тілінде Math класына тиісті (System атаулар кеңістігінде), бағдарлама кодын жазғанда пайдалануға болатын стандартты математикалық функциялар (әдістер) жиыны бар.

Барлық әдістер public және static модификаторларымен жарияланады, сондықтан олар Math класының объектісін алдын ала құрмай-ақ бағдарламаның кез келген орнынан қол жетімді. 2.2-кестеде Math класының негізгі әдістері көрсетілген.

Әдістерді қайта жүктеу дегеніміз – әр түрлі деректер типі үшін атауы бойынша бірдей бірнеше әдістің қолданылуы. Мысалы, Max() әдісіне нақты немесе бүтін сандарды беруге болады.

2.2-кесте – Math класының негізгі әдістері

Әдіс	Сипаттамасы
double Abs(double d);	Аргумент модулін қайтарады.
double Acos(double d);	Арккосинус бойынша радианда бұрышты қайтарады.
double Asin(double d);	Арксинус бойынша радианда бұрышты қайтарады.
double Atan(double d);	Арктангенс бойынша радианда бұрышты қайтарады.

Long BigMul(int x, int y);	Екі 32-разрядты санның көбейтіндісін қайтарады.
double Ceiling(double d);	Аргументке тең немесе одан үлкен ең кіші бүтін санды қайтарады.
double Cos(double d);	D бұрышының косинусын радианмен қайтарады.
double Cosh(double d);	D бұрышының гиперболалық косинусын радианмен қайтарады.
int DivRen(int a, int b, out int R);	Екі бүтін сандарды бөлудің нәтижесін және шығу параметрі ретіндегі R қалдығын қайтарады.
E	2,71828182845905
double Exp(double d);	d дәрежелі E қайтарылады.
double Floor(double d);	Берілген санға тең немесе одан кіші ең үлкен бүтін санды қайтарады.
double IEEERemainder (double a, double b);	a санын b санына бөлу нәтижесінің қалдығын қайтарады.
double Log(double d);	d санының натурал логарифмін қайтарады. Артық жүктелген әдісте екінші параметр болып логарифм негізі жіберіледі.
double Log10(double d);	d санының ондық логарифмін қайтарады.
double Max(double a, double b));	Екі санның ең үлкенін қайтарады. Артық жүктелген әдіс.
double Min(double a, double b);	Екі санның ең кішісін қайтарады. Артық жүктелген әдіс.
PI	3,14159265358979
double Pow(double a, double b);	a санының b дәрежесін қайтарады.
double Round(double a);	A санын бүтін санға — дөңгелектеу. Артық жүктелген әдіс.
int Sign(double a);	a санының нөлден кіші, нөлге тең немесе одан үлкен болуына байланысты -1, 0 немесе +1 қайтарады. Артық жүктелген әдіс.

2.2-кестенің жалғасы

Әдіс	Сипаттамасы
double Sin(double a);	Радианда a бұрышының синусын қайтарады
double Sinh(double a);	Радианда a бұрышының гиперболалық синусын қайтарады
double Sqrt(double a);	a-ның квадрат түбірін қайтарады.
double Tan(double a);	Радианда a бұрышының тангенсін қайтарады.
double Tanh(double a);	Радианда a бұрышының гиперболалық тангенсін қайтарады

Бағдарламаны ретке келтіру процесінде деректерді енгізуді немесе кейбір процестерді үлгілеуді жеңілдету үшін көптеген бағдарламашылар белгілі бір диапазонда тең қалыпты бөлінген, бүтін немесе нақты псевдокездейсоқ сандар тізбектілігінің генераторын қолданады. Осындай тізбектілікті құратын әдістер Random класында орналасқан.

Класс негізгі мән ретінде кейбір бастапқы санды қолданады, оған разрядтарды араластыру алгоритмы қолданылады және осындай жолмен алынған санды кезекті кездейсоқ сан ретінде қайтарады. Сонымен қатар келесі кездейсоқ санды өндіру үшін осы сан негізгі сан болып есептеледі. Сонымен, кезекті санның араластыру алгоритмі және бастапқы мәні толық анықталатындықтан, псевдокездейсоқ сандар тізбектілігі өндіріліп шығады.

2.3-кесте –Random класының кейбір әдістері

Әдіс	Сипаттамасы
Public virtual int Next();	Кезекті псевдокездейсоқ санды қайтарады. Әдістің артық жүктелген нұсқаларында тудырылатын сандардың ең жоғарғы мәнін немесе олардың мәндерінің диапазонын көрсетуге болады.
Public virtual void NextBytes (byte[] buffer);	buffer айнымалысын псевдокездейсоқ мәндері бар байттармен толтырады.
Public virtual double NextDouble();	0.0-ден 1.0 дейінгі диапазондағы нақты псевдокездейсоқ санды қайтарады.

Класста екі конструктор бар: параметрсіз және int параметрлі. Біріншісі бастапқы мән ретінде ағымдағы күн мен уақытты, екіншісі бастапқы, негізгі мәнді қабылдайды. Сонымен, бірінші конструктор қайталанбайтын сандар сериясын, екіншісі - бірдей сандар сериясын шығара алады.

Ескерту, кластың әдісі тұрақты емес, сондықтан оларды пайдалану үшін rnd класының объектісі міндетті түрде құрылады. Мысалы:

```
Random rnd = new Random();
 Console.Write("  " + (rnd.Next()%101).ToString());
 Console.WriteLine();
```

Бұл үзінді 0-ден 100 дейінгі аралықта псевдокездейсоқ бүтін санды қалыптастырады. Бұл мүмкін, өйткені rnd.Next() әдісі 0-ден ең жоғарғы деңгейге дейін псевдокездейсоқ бүтін санды қалыптастырады. 101-ге бөлуден қалатын қалдық 0-ден 100 дейінгі аралықта болады. a = rnd.Next() % 101 - 50; - жазбасы -50-ден 50 дейінгі аралықта псевдокездейсоқ бүтін санды қалыптастырады.

Талдау жасау үшін жалған кездейсоқ бүтін санды қалыптастырудың барлық жағдайлары қарастырылады, 0 – бұл жағдайда а айнымалысының мәні минус 50, ал 100 болғанда - а айнымалысының мәні 50-ге тең.

а айнымалысының барлық қалған мәндері осы шеткі мәндер арасында болады.

Стандартты математикалық функцияларды қолдану кез келген күрделі алгебралық өрнектерді бағдарламалауға мүмкіндік береді.

4.3 Мәтіндік файлдармен жұмыс

Бағдарламаны тестілеу кезеңінде бағдарлама жұмысын әр түрлі кірістік деректердің мәндері бойынша тексеруге тура келеді. Осындай «тестілеу» мәндер арнайы «кірістік» деректердің мәтіндік файлына жазылады және бағдарламаның орындалуы үшін кезекпен ұсынылады. Кейіннен тексеру мен талдау жұмыстарын жүргізу үшін бағдарлама жұмысының нәтижесі көбінесе «шығыстық» файлға жазылады.

Осы мақсаттарды жүзеге асыру үшін C# тілінде мәтіндік файлдармен жұмыс жасау қарастырылған - input.txt атты «кірістік» файл және output.txt атты «шығыстық» файл. Файлдар бағдарлама файлдары орналасқан каталогта құрылуы тиіс, бастапқы орны — ...\ConsoleApplication1\bin\Debug..

Мысал ретінде бағдарламаны тестілеу процесін қарастырайық, онда нақты санды бөлу нәтижелері қарастырылады.

Бағдарлама коды:

```
using System;
using System.IO;
using System.Text;
namespace ConsoleApplication1
{
   class Program
   {
      static void Main()
      {
         double z,y,x;
         StreamReader f1 = new StreamReader("input.txt");
         StreamWriter f = new StreamWriter("output.txt");
         string buf = f1.ReadLine();
         z = Convert.ToDouble(buf);
         buf = f1.ReadLine();
```

```
            x = Convert.ToDouble(buf);
            f.WriteLine("z = {0} x = {1} ", z, x);
            y = z / x;
            f.WriteLine(" z / x = {0} ", y);
            y = x / z;
            f.WriteLine(" x / z = {0} ", y);
            y = -z / 0;
            f.WriteLine(" - z / 0 = {0} ", y);
            y = -0.0 / 0.0;
            f.WriteLine("-0.0 / 0.0 = {0} ", y);
            y = 0.0 / 0.0;
            f.WriteLine(" 0.0 / 0.0 = {0} ", y);
            f1.Close();
            f.Close();
        }
    }
}
```

input.txt «кірістік» файлында 123 саны жазылған, осы сан z айнымалысына меншіктеледі. Екінші сан берілмегендіктен x айнымалысына нөл меншіктеледі.

Бағдарлама жұмысының нәтижесі output.txt шығыстық, мәтіндік файлда орналасады, нәтижені «блокнот» арқылы көруге болады:

```
z = 123 x = 0
z / x = бесконечность
x / z = 0
- z / 0 = -бесконечность
-0.0 / 0.0 = NaN
0.0 / 0.0 = NaN
```

Бағдарлама жұмысының нәтижесін толығырақ қарастырайық. Біз нөлге бөлу жағдайын жасадық, бұл жағдай көптеген бағдарламалау тілдерінде бағдарлама жұмысын үзеді.

C# тілінде нақты сандар үшін үш нұсқа қарастырылған (бағдарламаның есептеу нәтижелері үшін) – Infinity, NegativeInfinity және NaN. Алғашқы екеуі математикадан белгілі - шексіздік және теріс шексіздік. Үшінші NaN (Not a Number) мәні нәтиже нақты сан болмағанда немесе бағдарлама нәтижені анықтай алмаған жағдайда орын алады.

Осы мәндердің пайда болу жағдайларын қарастырайық. Егер көбейту немесе бөлу операциясының орындалуы барысында модуль бойынша нәтиже ең жоғарғы мүмкін саннан артық болса, онда нәтиженің таңбасына қарай мәні шексіздікке немесе теріс шексіздікке ие болады. Осы мәндерді анықтайтын double және float типтерінің тұрақтылары бар. Нақты сандар мен шексіздік арасында қосу, алу және көбейту операцияларын орындағанда нәтиже шексіздік мәніне ие болады, нәтиже теріс таңбада болуы мүмкін. Нақты санды шексіздікке бөлгенде нәтиже нөлге тең болады.

Егер шексіздік шексіздікке бөлінсе немесе нөл шексіздікке көбейтілсе, онда нәтиже NaN болады. Операция орындалу нәтижесі нақты сан болмаса, мысалы, теріс сан квадратының түбірін шығару кезінде, онда нәтиже жоғарыдағыдай NaN болады. Егер операциялар ішінде NaN қолданылса, онда нәтиже NaN болады.

Қарастырылған мысалдың екінші мақсаты - мәтіндік файлдармен жұмыс. Осы жұмыстың негізгі кезеңдерін қарастырайық.

Біріншіден, арнайы атаулар кеңістігін қосу керек:

using System.IO,

файлдың енгізу-шығару жұмысына жауап береді.

Екіншіден, файлдың айнымалыларын жариялау керек – файлмен жұмыс жасау үшін объектілер құру:

(StreamReader f1 = new StreamReader("input.txt");

StreamWriter f = new StreamWriter("output.txt");),

Оларға магнитті дискідегі файл аты сәйкес келеді

Үшіншіден, магнитті дисктегі мәтіндік ақпаратты жазу мен оқу операцияларын орындау (енгізу және шығару).

Төртіншіден, алынған мәтіндік ақпаратты сәйкес типтердің мәндеріне түрлендіру операцияларын орындау.

Бесіншіден, мәтіндік файлмен жұмысты орындап болғаннан кейін оны жабу керек.

f1.Close(); f.Close();

4.4 Өзін-өзі тексеру сұрақтары

4.4.1 Төмендегі нұсқалардың қайсысы меншіктеу операторы?

A) a*c = b;
B) a = c + 1;
C) a*x + b := 0;
D) –y = y;
E) a := b;.

4.4.2 Төмендегі үзіндінің орындалуы нәтижесінде қандай сандар шығарылады? Енгізілетін сан x = 1,2.

```
double x;
Console.Write("Введите значение x ");
buf = Console.ReadLine();
x = Convert.ToDouble(buf);
x = x - 1.2;
x = Math.Sqrt(x + 4) * x + 1;
Console.WriteLine("x= {0} ", x);
```

A) 0;
B) 1.2;
C) -1;

D) 2;
E) 1;

4.4.3 Сандарды шығару формат қалай сипатталады?
4.4.4 10 % 3 өрнегі неге тең болады?
4.4.5 10 / 3 өрнегі неге тең болады?