Қазақстан Республикасының Білім және Ғылым Министрлігі
Д.Серікбаев атындағы Шығыс Қазақстан Мемлекеттік Техникалық университеті

В.Л. Никифоров

АЛГОРИТМДЕР, МӘЛІМЕТТЕР ҚҰРЫЛЫМЫ ЖӘНЕ БАҒДАРЛАМАЛАУ

5В070300 «Ақпараттық жүйелер», 5В070400 «Есептеу техникасы және программалық қамтамасыз ету», 5В070500 «Математикалық және компьютерлік моделдеу» мамандықтарының студенттеріне зертханалық жұмысқа, СӨЖ және СОӨЖ арналған әдістемелік нұсқаулар

Оқу формасы : күндізгі

Өскемен қаласы
2015

ӘДК 681.3 (06)

Никифоров В.Л. Алгоритмдер, мәліметтер құрылымы және бағдарламалау: Әдістемелік нұсқаулық C# тілін терең меңгеру үшін 5В070400 «Есептеу техникасы және программалық қамтамасыз ету» және 5В070500 «Математикалық және компьютерлік моделдеу» мамандықтарына арнап жасалды / ШҚМТУ.- Өскемен, 2015.- 61 б.

Әдістемелік нұсқаулық теориялық мәліметті, көптеп есептерді шешу мысалдарын, СӨЖ-на арналған жеке тапсырма нұсқаларын және сұрақтарды, СОӨЖ зертханалық жұмыс кезінде істеген есеп-хатты қорғауға арналған деректерді қамтиды. Материалдар «Алгоритмдер, деректер құрылымы және бағдарламалау» пәнінен теориялық курсты игеруге көмектеседі, C# тіліндегі консоль қосымшасының Visual Studio.NET 2010 ортасында бағдарлама құруға практикалық дағдылану.

Ақпараттық технологиялар және энергетика факультетінің әдістемелік коммисиясымен бекітілді.

Хаттама № 2015г.

МАЗМҰНЫ

КІРІСПЕ

Осы нұсқаулықтарды әзірлеу кезінде алдыңғы жылдары оқыту тәжірибесі «Алгоритмдеу және программалау тілдері» және «Алгоритмдік тілдерде бағдарламалау» қолданылған.

Нұсқаулық теориялық материалды, көп көлемде есептерді шешу мысалдарын, жеке тапсырмалар бойынша СӨЖ нұсқалары және сұрақтарды, зертханалық жұмысты қорғау үшін СОӨЖ тапсырмаларын қамтиды.

Студенттердің өздік жұмысына арналған нұсқаулықта әрбір пән үшін дәріс материалын алдын ала зерттеу ұсынылады. Зертханалық жұмысқа арналған нұсқаулықты оқу компьютермен орындау кезінде практикалық игеру үшін қажет. Алдын ала дайындықсыз тақырыптың тест сұрақтарына және үй тапсырмасын орындау қиынға соғады .

Осы әдістемелік нұсқаулықпен жұмыс жасай отырып, Сіз оқып жатқан тақырыпқа арналған тест сұрақтарына жауап беру керексіз. Егер кейбір сұрақтар қиындық туғызса, онда Сіз теориялық материалды тақырыптың тиісті бөлімінен қайталауыңыз керек. Егер сізге теориялық материалды түсіну қиын болса, онда сіз СОӨЖ сағатында мұғалімнің қосымша сабағына келуіңіз керек.

Тест сұрақтарына жауап бергеннен кейін оқып жатқан тақырыптың үй тапсырмасын орындауға кірісіңіз. Бірінші қадамда Сіз үй тапсырмасының талаптарын түсініп және оның орындалу алгоритмін әзірлеуіңіз керек. Екінші қадамда Сіз бағдарламаны әзірлеуіңіз керек, компьютерде іске қосып және оған тестілеу жүргізу қажет. Аяқталған үй тапсырмасын мұғалімге көрсету үшін дайындау керек.

Үй тапсырмасын табысты аяқтағаннан кейін оқып жатқан пән модулінің жеке тапсырмасын орындауға тырысыңыз.

Тізімделген әрекеттер ретін әрбір оқылатын пән тақырыбы үшін өз бетімен орындау керек!

Осы әдістемелік нұсқаулық лекция материалдарын толықтырады, бірақ пәннің барлық аспектілерінің сипаттамасы болып табылмайды. Сондықтан студенттерге әдебиеттер тізімі ұсынылған, олар оны пәннің материалын тереңірек игеру үшін пайдалана алады.

ТАҚЫРЫП 1 C# БАҒДАРЛАМАЛАУ ТІЛІНІҢ ҚАРАПАЙЫМ ОПЕРАТОРЛАРЫ

1.1 Бірінші тақырыптың мақсаты

Консоль қосымшаларын құру үшін Visual Studio.NET ортасының негіздерімен танысу және практикалық дағды алу. C# тілінің қарапайым операторларын қолданып бағдарламалау тілін игеру.

1.2 Теориялық мәліметтер

1.2.1 Кіріспе

C # тілі программалау тілдері саласында ең танымал тіл болып табылады. Бұл 21-ші ғасырда құрылған бірінші бағдарламалау тілі болып табылады. Тіл халықаралық қоғамдастық тарапынан танылаған. 2006 жылдың маусым айында , Стандарттау жөніндегі еуропалық қауымдастық тіл стандартының төртінші нұсқасын бекітті: Standard ECMA-334 C# LanguageSpecifications, 4-th edition.

C# тілін жасаушылар тобын басқарушы Microsoft қызметкері Андреас Хейлсберг. Ол программистер әлемінде Microsoft келгенге дейін атақты болды. Хейлсберг Delphi визуалды бағдарламалау тілінің жетекші жасақтаушы болды.

C# тілінің келесі негізгі артылықшылықтар белгілеуге болады:

– C # тілі Framework.Net жақтауымен қатар құрылып,дамытылған және оның барлық мүмкіндіктерін толық ескереді;

– C # тілі толық объектілі - бағдарланған тіл болып табылады;

– Framework.Net жақтауының арқасында бағдарламашылар C# тілін виртуалды машинада жұмыс істеу артықшылығын алады;

– Framework.Net C # тілінің қолданбаларының түрлерін қолдайды ;

– іске асыру , сенімді және тиімді код құрылысын үйлестіре отырып, C # тілі табысты ықпал ететін маңызды фактор болып табылады .

1.2.2 C# тілінің типтер жүйесі

C# тілінің стандартына келесі негізгі типтердің жиыны кіреді:

– логикалық типтер (bool);

– символдық типтер (char);

– бүтін типтер. Бүтін типтер келесі үш өлшемдердің біреуі бола алады - short, int, long, signed немесе unsigned дискрипторымен бірге, онда мәнді түсіндіре көрсетілген - белгісі жоқ немесе бар ;

– нақты типтер. Бұл типтер келесі үш өлшемдердің біреуі бола алады - float, double, long double;

– void типі, ақпараттың жоқ болуды белгілеу үшін қолданылады.

Тіл түрлерді жобалауға мүмкіндік береді:

– көрсеткіштер (мысалы, int* - айнымалы типіне көрсеткіш int);

– сілтемелер (мысалы, double& - айнамалы типіне сілтеме double);

– массивтер (мысалы, char[] – элемент типінің массиві char).

Тіл пайдаланушы анықтаған түрлерді жобалауға мүмкіндік береді :

– анықталған түрлер (enum) нақты жиынтық мәндерін ұсынуға арналған;

– құрылым (struct);

– кластар (class).

C# тілінің негізгі типтерінің сипаттамасы 1.1 кестесінде көрсетілген.

1.1 Кесте C# тілінің деректерінің негізгі типтері

Типтің аты	Жүйелік тип	Мәні	Өлшемі
bool	System.Boolean	true, false	8 бит
sbyte	System.SByte	[-128, 127]	Белгімен, 8-бит
byte	System.Byte	[0, 255]	Белгісіз, 8-бит
short	System.Short	[-32768, 32767]	Белгімен , 16-бит
ushort	System.UShort	[0, 65535]	Белгісіз , 16-бит
int	System.Int32	$[-2^{31}, 2^{31}]$	Белгімен , 32-бит
uint	System.UInt32	$[0, 2^{32}]$	Белгісіз , 32-бит
long	System.Int64	$[-2^{63}, 2^{63}]$	Белгімен , 64-бит
ulong	System.UInt64	$\approx (0, 2^{64})$	Белгісіз , 64-бит
float	System.Single	$[10^{-45}, 10^{38}]$	7 сан
double	System.Double	$[10^{-324}, 10^{308}]$	15-16 сан
decimal	System.Decimal	$[10^{-28}, 10^{28}]$	28-29 айтарлықтай сандар
char	System.Char	[U+0000, U+ffff]	16-бит Unicode символ
string	System.String	Unicode символдардан тұратын жол	

1.2.3 Өрнектер

Өрнектер операндтар, тұрақтылар, айнымалылар, біріккен операция белгілері бар функциялар мен жақшалардан құралады. Өрнекті есептеу кезінде оның мәні мен түрі анықталады. Бұл өрнек сипаттамалары операндтар мәні және түрімен анықталады және өрнек ішіне кіретін есептеу қағидаларымен де. Ереже қойылады:

– операциялар басымдылығымен (операциялар үшін бір басымдылығын қолдану тәртібі – солдан оңға қарай немесе оңнан солға қарай);

– операндтар түрінің түрленуі және шамадан тыс жүктелген операцияларды іске асыру;

– орындалу операциясының нәтижесінің түрі мен мәні операндқа берілген мәнмен анықталған.

Операцияның басымдылығы мен орындау тәртібі кестеде келтірілген 1.2.

Кесте 1.2 C#-та операцияның басымдылығы мен орындау тәртібі

Басымдық	Категория	Операция	Реті
0	Негізгі	x.y, f(x), a[x], x++, x--, new	→
1	Унарлы	+, -, !, ++x, --x, (T)x	→
2	Мультипликативтік	*, /, %	→
3	Қосымша	+, -	→
4	Ауысым	<< ,>>	→
5	Қарым-қатынас түрі тексеру	<, >, <=, >=, is, as	→
6	Эквиваленттік	==, !=	→
7	Логикалық ЖӘНЕ (AND)	&	→
8	Логикалық НЕМЕСЕ (OR)	|	→
9	Шартты логикалық ЖӘНЕ	&&	→
10	Логикалық шартты немесе	||	→
11	Шартты өрнек	? :	←
12	Меншіктеу	=, *=, /=, %=, +=, -=	←

Кез келген жақшаға алынған өрнек жоғары басымдылық алады және есептелген болуы тиіс. Қажетті нақты жағдайда жақшалар өрнекті есептеудің стандартты тәртібін өзгертуге және орнатуға мүмкіндік береді. Күрделі өрнектерде жақша қою пайдалы, себебі стандартты тәртіп талап етілген тәртіппен сәйкес келді.

1.2.4 Типтердің түрленуі

Операция кастинга – операндтарды бір типке келтіру. Өрнекті есептеу барысында операнд типтерінің түрленуін орындау қажеттігі туындауы мүмкін. Мүмкіндігінше бұл түрлендіру автоматты түрде орындалады, бағдарламалаушы үшін емес. Бірақ айқын емес түрлендіру шектелген, өйткені қауіпсіз ғана болуы мүмкін. Қауіпті түрлендіруді қашан орындай екендігін бағдарламалаушы анық білуі тиіс. Тапсырма типінің түрленуінің мүмкіндігі операцияны қолдану, ол да кастинг деп аталады. Бұл унарлы операция келесі синтаксиспен:

(тип) өрнек;

Мұнда жақша ішінде тип көрсетілген, жақша сыртында тұрған өрнекті қою керек. Операция кастингін тек арифметикалық түрдің ішінде ғана қолдану керек. Оның көмегімен бір арифметикалық типті басқа бір типке келтіруге болады, бірақ бүтін типті логикалық bool типіне келтіруге болмайды.

Типке келтіру мысалын қарастырайық:

```
byteb1 = 1, b2 = 2, b3;
//b3 = b1 + b2;
b3 = (byte)(b1 + b2);
```

Бұл мысалда byte типті екі айнымалыны қосу керек, бір қарағаннан, ешқандай типке келтіру керек жоқ сияқты, қорытындыда да тип byte,

тағайындау операторлардың сол жағында келісілген. Алайда, бұл қысқа нөмірлерге ешқандай қосу операция бар екені қарапайым себеппен көрсетілген. Қосуды іске асыру int типінен басталады. Сондықтан қосуды орындау алдында екі операндта int типіне алмастырылады, қосу қорындысыда типі int болады, өрнекке мәнді ауыстыру кезінде byte типті айнымалыға ауысу кезінде компиляция периондында қате туындайды. Сол себепті екінші жолдағы оператор коды оқылмайды. Программист өрнекті byte типіне келтіруі керек , ол кодтың үшінші жолында келтірілген.

Келесі үзінді кодында типке келтіру мысалы көрсетілген:

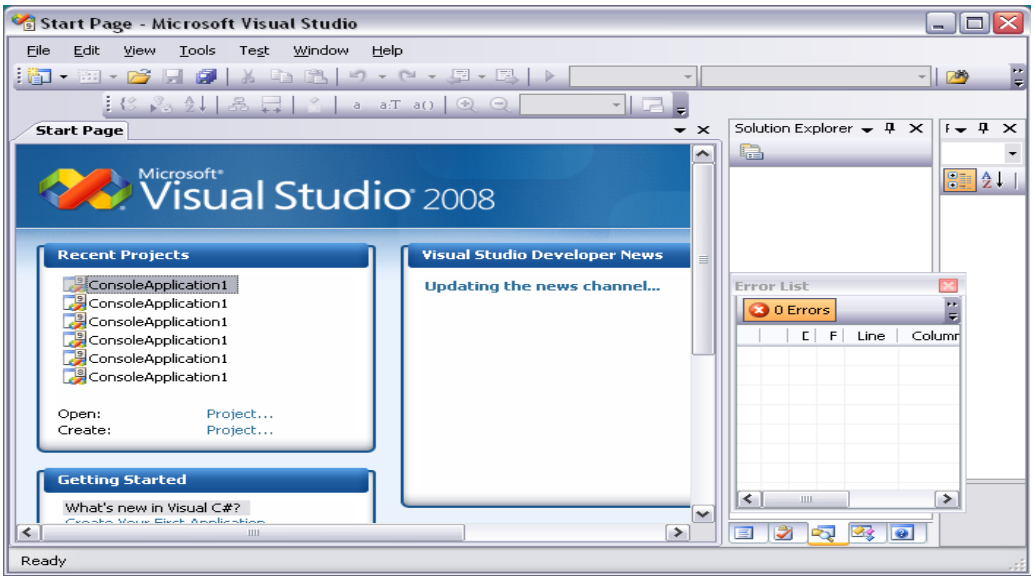
```
 int i1,i2;
i1 = -40;
i2 = (int)(1.8 * i1) + 32;
```

Бірініші операнд типінде көбейту қорытындысының типі double. Көбейтуді орындамас бұрын, қорытынды int типіне келтіріледі. Бірінші көбейтуге келтірген соң (int)(1.8 * i1) көбейту нақты сандармен орындалады қорытынды int типті болады, i2 айнымалысына келтіру үшін ешқандай өзгерту қажет емес. Егер операндтағы келтіруді алып тастаса, онда компиляция этапында қателік туындайды.

1.3 Зертханалық жұмысты орындау мысалы

a және b айнымалыларының орын ауыстыратын бағдарлама жазу. A айнымалысының бастапқы мәні 0-ден 20-ға дейінгі диапазонда кездейсоқ түрде құрылады, ал b айнымалысының мәні диалог режімінде енгізіледі. Екі айнымалының орны ауысқаннан кейін шешімін қарау. Бағдарламаны консоль қосымшасында орындау керек және жұмыс үстеліне орыналастыру қажет.

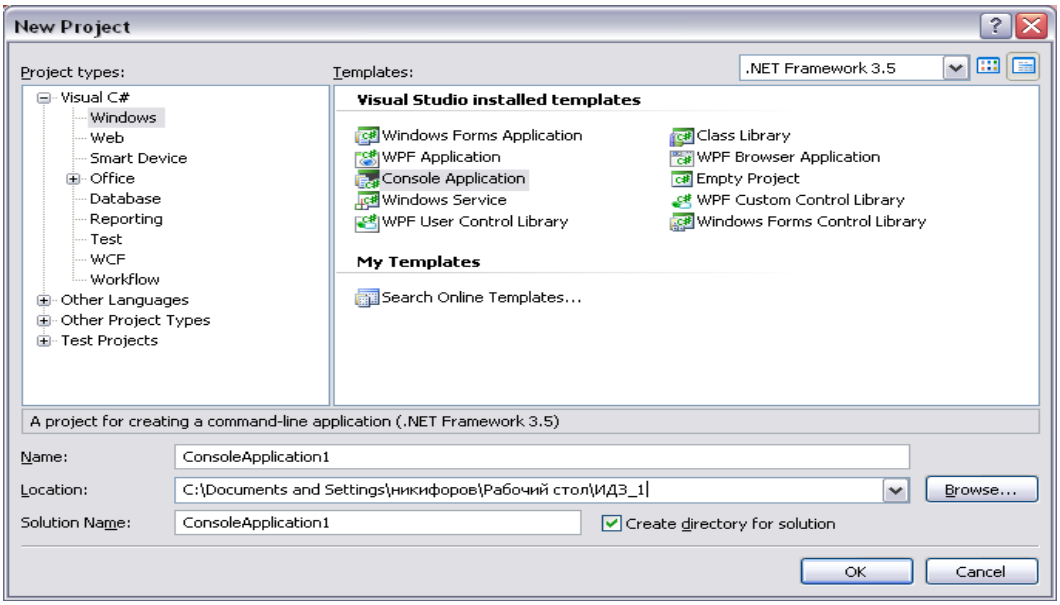
Visual Studio.NET бағдарламалау ортасын іске қосамыз (1.1 суретке қара).



1.1 Сурет – 1 терезе – Visual Studio.NET жүктеу

1.1 суреттің терезесінде File/New/Projekt… командаларды таңдаймыз

Жаңа терезе ашылады (New Project), онда жасақталатын жобаның типін таңдауға болады (1.2 суретті қара).



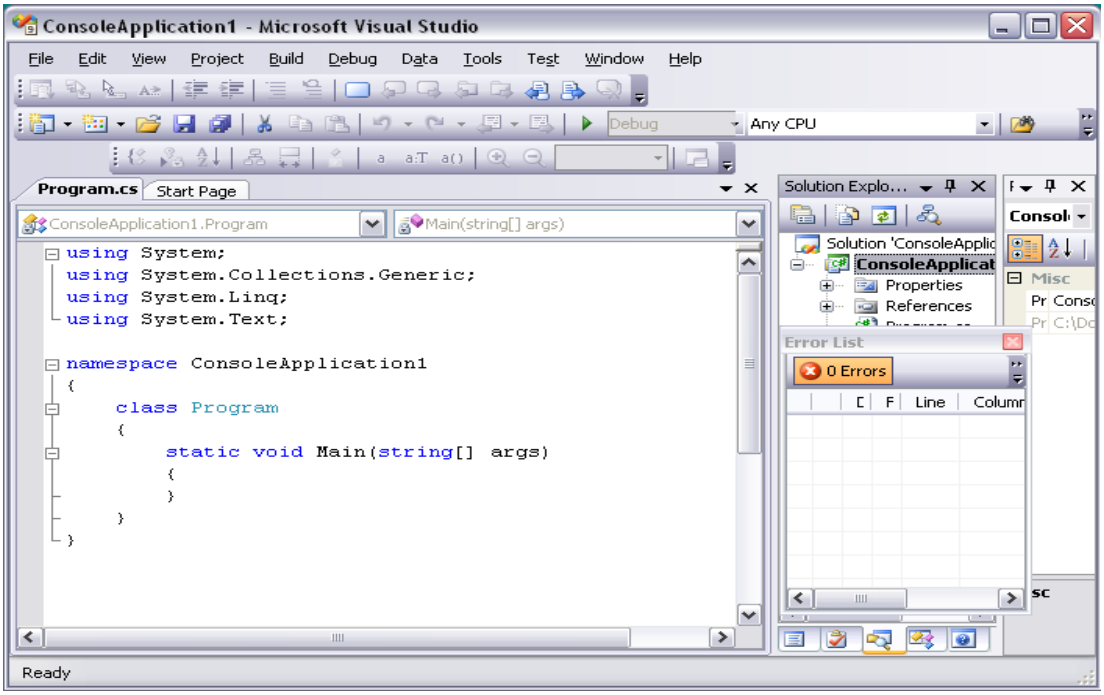
1.2 сурет – 2 Терезе – жаңа жобаны құру

2 терезеде

Калған терезе настройкаларын өзгермеуге болады.

2 терезеде жұмыстың типін ConsoleApplication таңдаймыз және и сақтау орынын көрсетеміз, ол жерде жұмыстың құрылған файлдары сақталады (Рабочий стол папка ИДЗ_1). Терезенің басқа баптауларын өзгертпеуге болады.

ОК батырманы басамыз және келесі 3 терезеге көшеміз.



1.3 сурет – 3 Терезе Visual Studio.NET ортасы

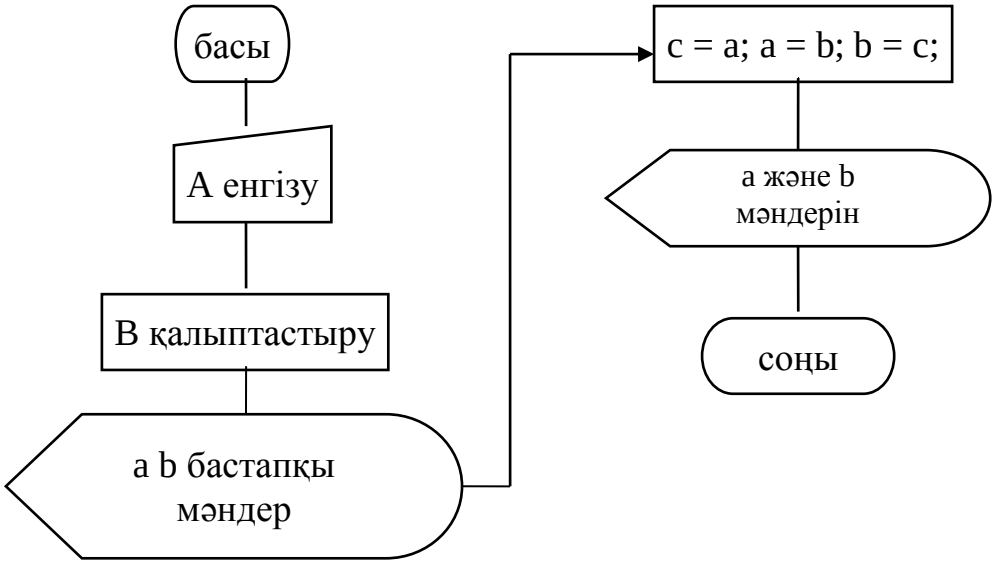
3 терезеде Program.cs редакторының бетінде есепті шешеудің бағдарламалық коды жазылады.

Есепті шығару алгоритмі өте қарапайым:

- a айнымалының мәнін диалог режимінде енгізуді ұйымдастыру керек;
- кездейсоқ түрде берілген диапазонда b айнымалысының мәнін көрсету;
- айнымалылардың кіріс мәндерін шығару;
- айнымалылардың мәндерін алмастыру;
- a және b айнымалыларының жаңа мәндерін басу.

Көрсетілген алгоритімді орындау үшін алдыңғы екі дәріс материалын оқу керек.

Өздік жұмыстың келесі деңгейі есепті шешу алгоритімінің құрылымдық схемасын сызу.



1.4 сурет – Есепті шешу алгоритмнің құрылымдық схемасы

Есепті шешу алгоритмнің құрылымдық схемасын қолдана отырып, бағдарламалау кодын құраймыз:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
 class Program
  {
   static void Main()
    {
     int a, b, c;
     string buf;
     //Диалог режимінде a айнымалының мәнін енгізу
     Console.Write("a-нің бүтін мәнін енгізіңіз");
     buf = Console.ReadLine();
     a = Convert.ToInt32(buf);
     // b айнымалының мәнін кездейсоқ түрде қалыптастыру
     Random rnd = new Random();
     b = rnd.Next() % 21;
     //Бастапқы мәндерді баспаға шығару
     Console.WriteLine("a және b айнымалылардың бастапқы
мәндері:");
     Console.WriteLine("a={0} b={1}", a, b);
     //Ауыстыру
     c = a; a = b; b = c;
     // Ауыстырудан кейінгі a және b айнымалылардың мәндерін
баспаға шығару
```

```
      Console.WriteLine("a және b айнымалылардың жаңа мәндері:");
      Console.WriteLine("a={0} b={1}", a, b);
      // Задержка рабочего экрана монитора
      Console.WriteLine("Жалғастыру үшін Enter пернесін
басыңыз");
      Console.ReadLine();
     }
   }
}
```

Бағдарлама жұмысы:

а-нің бүтін мәнін енгізіңіз 35
а және b айнымалылардың бастапқы мәндері:
a=35 b=13
а және b айнымалылардың жаңа мәндері:
a=13 b=35
Жалғастыру үшін Enter пернесін басыңыз

Зертханалық жұмыс бойынша есепхатты дайындау кезінде келесі құрылымды және реттілікті ұстану керек:

– титул беті;
– зертханалық жұмыстың атауы;
– зертханалық жұмыстың мақсаты;
– зертханалық жұмыстың жеке тапсырмасы;
– жеке тапсырма бойынша қысқаша түсініктемелер (керек болса есеп шығару алгоритмнің құрылымдық схемасы);
– жеке тапсырманың тиісті бағдарламалық код;
– бағдарлама жұмысының нәтижесі;
– қорытынды.

Титул беті есеп-хаттың бірініші беті және ақпарат көзі болып табылады , құжатты іздеу үшін қажет. Сондықтан , ол министрлік атауын , университеттің атауын , бөлім атауын , пәннің атауы, модуль атауын , зертханалық жұмысты орындаған студент аты-жөнін, және зертханалық жұмысты қабылдаған оқытушының аты-жөні көрсетіледі. Титул бетінің төменгі жағында қай жерде және қашан орындалғаны көрсетіледі, мысалы, Өскемен 2010 жыл.

Титул беті есеп-хаттың жалпы бет санына кіреді, бірақ титул бетінде беттің нөмірі қойылмайды.

Зертханалық жұмыс атауы зертханалық жұмысты орындауға арналған нұсқаулық ішіндегі атауымен сәйкес келуі керек, ол жиі модуль атауымен сәйкес келеді.

Зертханалық жұмыс мақсаты пән модулі мақсатының қысқаша сипаттамасы болып табылады.

Зертханалық жұмыстағы жеке тапсырма жеке тапсырманың толық мәтінін құрайды, зертханалық жұмыс бойынша үй тапсырмасын орындағаннан кейін ғана мұғалімнен алады.

Жеке тапсырмалар бойынша қысқаша түсініктемелер жеке тапсырмаларды орындау алгоритмін сипаттайды. Егер қажет болса,алгоритімнің құрылымдық схемасы немесе егжей-тегжейлі ауызша сипаттамасы қолданылады.

Жеке жұмысты орындауға қажетті код бағдарлама кодының толық мәтінін құрайды, немесе бағдарлама кодының үзіндісін, бағдарламалау ортасына арналған студент өзі бағдарламалаған, онсыз орындаған жұмысты түсіндіру мүмкін емес.

Бағдарламаның нәтижелері әдетте оның барлық режимдерде бағдарлама терезесінің көшірмелерін қамтиды .

Алынған нәтижелер , әдетте, зертханалық жұмыс нәтижесін көрсетеді.

Есеп-хат мәтінінің беттері А4 пішіміне сәйкес болуы керек.

Есеп-хатты басып шығару машиналық жолмен немесе компьютер құрылығысын пайдаланып басып шығару немесе графикалық жолмен ақ беттің бір жағына басып шығару керек. Қаріп түрі - TimesNewRoman, негізгі қаріп өлшемі - № 14, рұқсат етіледі № 12. Негізгі интервал -1, рұқсат етіледі -1,5.

Есеп-хат мәтінін келесідей басу керек: сол, жоғарғы, төменгі және оң – 20 мм.

Шегінісі кем дегенде төрт ондық басталады және бір құжат аясында бірдей болуы керек.

Есеп-хатты орындау тәсіліне қарамастан, мәтіннің басылғаны, суреттер, кестелер және қосымшалар сапасы айқын талаптарға сәйкес болуы керек.

Есеп-хатта белгілі бір сөздер, формулалар тек қара сиямен жазылуы тиіс, жазылған мәтін тығыздығы негізгі мәтіннің тығыздығына жақын болуы тиіс.

Қателіктер мен графикалық дәлсіздіктер қара сиямен, суретті машинамен немесе қолмен ақ бояумен тазалануы керек.

Есеп-хат беттері зақымданған, дақтар және белгілер толық жойылуға рұқсат етілмейді.

Есеп-хаттағы бөлімдер нүктесіз араб цифрларымен нөмірленген және абзацтан жазылған болуы тиіс. Бөлімшелер әрбір бөлімге сәйкес нөмірленуі тиіс. Бөлім нөмері саны бөлінген бөлім мен бөлімшелерден тұрады. Қосалқы нөмер аяғында нүкте қойылмайды. Бөлімшелердің ретінде бөлімдер бір немесе бірнеше элементтерден тұруы мүмкін.

Құжатта бөлім атаулары жолдың ортасында нүктесіз орналастырылады және бас әріппен астын сызбай және арнайы белгілемей жазылады.

Құжаттың бөлімшелер атауы жолдың ортасында кіші әріппен, бас әріппен бастап қою қаріппен жазылады.

Бөлімнің қосалқы бөлімдерді жоқ болса , онда элементтерді нөмірлеу сол бөлімде шегінде болуы тиіс, және элемент саны нүктемен бөлінген бөлім нөмірінен және элементтен тұруы тиіс .

Бөлімнің қосалқы бөлімдер бар болса , тармақтарды нөмірлеу әрбір бөлімшені және элемент саны шегінде болуы тиіс сандардан тұрады , тармақ және нүктелермен бөлінген болуы тиіс.

Бүкіл мәтінде нөмірлеуді сақтай отырып, есеп-хатты араб сандаырмен нөмірлеу керек.

Есеп-хаттың беттер нөмері беттің жоғарғы жағында ортасында соңында нүктесіз жазылуы керек.

Есеп-хат беттері қыстырғышпен қыстырылады немесе файлға салынады (степлер қолданбаңыз).

1.4 Зертханалық жұмыс бойынша үй тапсырмасы

Кесіндінің ұзындығын есептеу бағдарламасын жазу керек. Кесінді ұштарының координаттары - A(x1, y1) және B(x2, y2). A нүктесі координаттарының мәні диалог режімінде енгізіледі, ал B нүктесін минус 100-ден плюс 100-ге дейінгі аралықта кездейсоқ түрде беріледі. Бағдарлама жұмысының нәтижесін монитор экранына шығаруды қарастыру керек.

1.5 СӨЖ арналған жеке тапсырмалары

1.5.1 Өрнекті есептейтін программаны жазыңыз:

$Y=3*(X+1)^3+4*(X-1)+2$.

X мәнің диалог режімінде енгізіледі. Бір-бірден қарастыру әдісін қолданып (методом перебора) Y нөлге тең болатындай X мәнін табыңыз. Бағдарлама жұмысының нәтижесін монитор экранына шығаруды қарастыру керек.

1.5.2 Қосындыны есептейтін программаны жазыңыз:

$S=1+1/2+1/3+1/4+1/5$.

Бағдарлама жұмысының нәтижесін монитор экранына шығаруды қарастыру керек.

1.5.3 A және B айнымалыларының мәндерін алмастыруды орындайтын алгоритмі бар программаны жазыңыз (қосымша c айнымалысы қолданылмауы тиіс). Айнымалы мәндерін диалог режімінде енгізіледі. Бағдарлама жұмысының нәтижесін монитор экранына шығаруды қарастыру керек.

1.5.4 Диалог режімінде енгізілген нақты санның бүтін және бөлшек бөліктерін бөлетін программаны жазыңыз және оларды монитор экранына бөлек шығарыңыз.

1.5.5 Үшбұрыштың ауданын есептейтін программаны жазыңыз, үшбұрыштың A, B, C жақтары диалог режімінде енгізіледі.

1.5.6 Үш кесіндіден тұратын сынық кесінді ұзындығының есептейтін программаны жазыңыз. Кесінді үштарының координаттары A(x,y), B(x,y), C(x,y) және D(x,y). Координаттардың мәндері 0-ден 100 дейінгі аралықтан кездейсоқ түрде алынады. Бағдарлама жұмысының нәтижесін монитор экранына шығаруды қарастыру керек.

1.5.7 Өрнекті есептейтін программаны жазыңыз: $C = A^2 + \sqrt{(A + LnB)/2}$. A және B айнымалыларының мәндері диалог режімінде енгізіледі.

1.5.8 Шеңбердің ауданын есептейтін программаны жазыңыз. Радиустың мәні диалог режімінде енгізіледі және ол нөлден үлкен болуы тиіс.

1.5.9 Кесіндінің ұзындығы дюймде берілген. (1 дюйм = 2,54 см.), диалог режімінде енгізіледі. Ұзындықтың мәнін метрлік жүйеге, яғни метрге, сантиметрге және миллиметрге айналдырыңыз. Мысалы, 21 дюйм = 0м 53см 3,4мм.

1.5.10 α бұрышы диалог режімінде градуспен, минутпен және секундпен беріледі. Оның мәнін радиан шамасында табыңыз.

1.5.11 Екі нақты санның мәні диалог режимінде беріледі. Значения двух действительных чисел задается в режиме диалога. Найти среднее арифметическое этих чисел и среднее геометрическое их модулей.

1.5.12 t1 температурасымен v1 литр су t2 температурасымен v2 литр су араластырылған. Алынған қоспаның температурасын және көлемін тап. v1, t1, v2 және t2 диалог режимінде беріледі.

1.5.13 Шеңбердің ұзындығы диалог режимінде беріледі. Шеңбермен шектелген дөңгелектің ауданын тап.

1.5.14 Қабырғалары тең үшбұрыштың ұзындығы диалог режимінде енгізіледі. Бұл үшбұрыштың ауданын табу керек.

1.5.15 Үшбұрыш төбелерінің координаталары диалог режимінде берілген. Бұл үшбұрыштың периметрін есепте.

1.5.16 Куб қабырғасының ұзындығы диалог режимінде енгізіледі, Кубтың көлемін тап, және оның барлық беттерінің ауданың тап.

1.5.17 Екі кедергінің мәні диалог режимінде беріледі. Олардың тізбектей және параллель қосылу кезіндегі кедергіні тап.

1.5.18 X мәні диалог режимінде енгізіледі. Тек қана арифметикалық операцияларды жүргізе отырып, өрнекті есепте

$$Y = 1 - 2*X + 3*X2 - 4*X3.$$

Бір өрнекте операциялардың саны сегізден аспау керек.

1.5.19 X мәні диалог режимінде енгізіледі. Тек қана арифметикалық операцияларды жүргізе отырып, өрнекті есепте

$$Y = 1 + 2*X + 3*X2 + 4*X3.$$

Бір өрнекте операциялардың саны сегізден аспау керек.

1.5.20 Сақинаның ауданын тап, оның сыртқы радиусы 100 тең, ал ішкі радиусы диалог режимінде беріледі және 100-ден аспау керек.

1.6 СОӨЖ бойынша бақылау сұрақтары

1.6.1 NET платформа ұғымы дегеніміз не?
1.6.2 Тілдік орта ұғымы нені білдіреді? (CommonLanguageRuntime, CLR)?
1.6.3 «қосымша» термині нені білдіреді?
1.6.4 «жоба» термині нені білдіреді? Оның құрамы?
1.6.5 «Есім кеңістігі» нені қамтиды?

1.6.6 Visual Studio.NET ортасы нені қамтиды?
1.6.7 Алгоритм түсінігі? Алгоритмнің негізгі қасиеті.
1.6.8 Есепті шешу қадамдары?
1.6.9 C# тілінің қандай деректер типін білесіз?
1.6.10 C# тілінің қандай операциялар белгілерін білесіз?
1.6.11 C#? тілі операторының жазу пішімі? Оның жұмысын мысалда түсіндіріңіз.
1.6.12 «диалог режимінде» жазбаны қалай ұйымдастыруға болады? Мысал.
1.6.13 Экран мониторына деректер шығысын қалай ұйымдастыруға болады? Мысал.
1.6.14 Белгілі бір форматта экранға деректер шығысын қалай ұйымдастыруға болады? Мысал.
1.6.15 Берілген ауқымда кездейсоқ сандар қалай жасалады? Мысал.

ТАҚЫРЫП 2 C# БАҒДАРЛАМАЛАУ ТІЛІНІҢ КҮРДЕЛІ ОПЕРАТОРЛАРЫ

2.1 Екінші тақырыптың мақсаты

C# тілінің күрделі операторларын оқу. C# бағдарламалау тілінің күрделі операторларын қолданып тәжірибелік дағды алу.

2.2 Теориялық ақпарат

2.2.1 Шартты оператор if

Басқа бағдарламалау тілдері сияқты, C# тілінде де бірнеше параметрлердің біреуін таңдау үшін –if және switch операторлары қолданылады. Бірінші операторды шартты немесе шартты ауысу операторы деп атайды, екінші – бағдарламаны ауыстыру операторы. If немесе switch операторлары қолданылатын бағдарламаны «тармақталған» есептеу процесі деп аталады. Жазу пішімін және осы операторлардың әрқайсысының жұмысын қарастырайық. If операторының келесідей жазу пішімі бар:

```
if(выражение) { операторы_1; }
        else{ операторы_2; }
```

if логикалық өрнегі жақшаға алынып true немесе false мәнін қабылдай алады. Егер логикалық өрнек true болса, бағдарлама тек оператор_1 орындайды. Егер логикалық өрнек false болса, бағдарлама оператор_2 орындайды.

Егер оператор_1 немесе оператор_2 жекеше түрде берілген болса, олар үшін жақшаны қолданбауға болады.

Көп жағдайда else сөзінен кейін if операторы болмауы мүмкін. оператор_1 жағдайында if операторының қысқаша жазылуы альтернативті таңдау – істеу немесе істемеу – орындау немесе орындамау. Егер if

операторының ішінде if операторының басқа операторлары болса, онда else , егер ол бар болса, өзінің алдында ашық тұрған if операторына қатысты.

2.2. switch бағдарламаны ауыстыру операторы

Өрнек мәнін анықтау үшін бағдарлама ішінен қажетті нұсқаны таңдау үшін switch операторы қолданылады. switch операторын жазудің келесідей пішімі бар:

```
switch(выражение)
{
case константное_выражение_1: [операторы_1 оператор_перехода_1]
        …
caseконстантное_выражение_K: [операторы_Kоператор_перехода_K]
  [default: операторы_Nоператор_перехода_N]
}
```

Default бұтағы болмауы мүмкін. Жазу форматында сәйкес қос нүктеден кейін бос орын, басқа операторға ауысу қолайлы болып табылады. case –те тұрақты өрнек типі switch-өрнегінің типіндей болуы керек.

switch операторының жұмыс істеуі келесідей:

– switch-өрнегінің мәні есептеледі;

– белгілі бір ретпен case тұрақты өрнекпен салыстырылады:

– ұқсастық табылғаннан кейін, case-бұтағы операторының реті орындалады. Осы реттіліктің соңғы операторы ауысу операторы болғандықтан (көп жағдайда да бұл break операторы), switch операторының орындалуын аяқтайды;

– егер case-бұтағы бос реттілік операторы болса. Онда бұл бұтақтың тұрақты өрнегі switch-өрнегімен ұқсас болса, онда case-бұтағының бос емес реттілігі бірінші орындалады;

– егер switch-өрнегінің мәні ешқандай тұрақты өрнекпен сәйкес келмесе, default бұтағы операторының реттілігі орындалады;

– егер default бұтағы болмаса, онда switch операторы бос операторға тең.

Switch операторының әрбір case-бұтағы ауысу операторымен аяқталуы тиіс (бос реттілікті ескермей тұра тұрайық), әйтпесе компиляция периоды кезінде қателік туындайды.

Switch операторында case-өрнегі тек тұрақты өрнек бола алады. Тапсырма диапазонына case-өрнегін беру белгіленбеген.

Егер case - өрнек операторларының бір тобына сәйкес келетінін атап кетсе , кейс- өрнек операторларының осы топқа өткен жағдайда - білдіру алдында «бос» жазылады. Мысалы, екі аргумент арасына арифметикалық операция switch өрнегінде string типінде, операцияны таңдау үшін бірнеше жауап нұсқасы болуы мүмкін +, -, *, және /, келесідей бағдарламалық код түрінде көрсетуге болады:

```
// Өрнектер тізімін пайдалана отырып, жағдайларды талдау/
/arg1 – операцияның бірінші аргументі
//arg2 – операцияның екінші аргументі
```

```
// result – операция нәтижесі
switch (operation)
   {
case "+":
case "Plus":
case "Қосу":
result = arg1 + arg2;
break;
case "-":
case "Minus":
case "Азайту":
result = arg1 - arg2;
break;
case "*":
case "Mult":
case "Көбейту":
result = arg1 * arg2;
break;
case "/":
case "Divide":
case "Div":
case "Бөлу":
case "Бөлу":
result = arg1 / arg2;
break;
default:
result = 0;
break;
   }
```

Назар аударыңыз, операция белгісін аргументтерге әртүрлі тәсілдермен беруге болады, switch операторының бұтақтарына тұрақты өрнек тізімін беру мүмкіндігін көрсетеді.

2.2.3 for циклдік операторы

For циклдік операторы, if операторы сияқты, бағдарламалау тілдерінің барлығында бар. Оның келесідей жазылу пішімі бар:

for(инициализаторы; условие; списоквыражений)
{ оператор; }

Фигуралық жақша ішіндегі оператор цикл денесін анықтайды. Цикл денесінің қанша рет орындалатындығы жақша ішіндегі басқарушы үш элементе байланысты. Инициализаторлар бір немесе бірнеше айнымалы , жиі санауыш немесе айнымалы цикл деп аталатын бастапқы мәнді орнатады. Шарт true немесе false мәнін қабылдай алатын цикл аяқталғанын білдіреді.

Үтірлермен бөлінген өрнектердің тізімі, цикл санауыштарының әрбір қадамда қалай орындалатынын көрсетеді. Егер цикл шарты ақиқат болса, онда цикл денесі орындалады, одан кейін санауыштар мәні өзгереді және шарт қайта тексеріледі. Шарт жалған болған соң, цикл өз жұмысын аяқтайды. For циклінде цикл денесі орындалмауы мүмкін, егер инициализациядан кейін цикл шарты жалған болса , дұрыс істемеуі мүмкін, егер шарт ылғи ақиқат болып қалса. Қалыпты жағдайда цикл денесі соңғы рет санымен орындалады.

Цикл санауыштары жиі инициализаторда жарияланады және айнымалы болып табылады, цикл жерсіндірілген, цикл аяқталған соң олар қолданылмайды. Мерзімінен бұрын тоқтату мүмкіндігі оларға цикл шығуда мәні талдауға мүмкіндік береді операторлар көшу цикл санауыштар мәлімделген болады , бірыңғай цикл жабдықталған жағдайларда.

2.2.4 Шартты цикл

While циклі (өрнек) циклдің әмбебап түрі болып табылады,барлық бағдарламалау тілдерінде енгізілген. while өрнегі ақиқат болғанша цикл денесі орындала береді. C# тілінде бұл цикл түрінде екі модификация бар – цикл басында және аяғында шартты тексеру. Бірінші модификацияның келесідей синтаксисі бар:

While (өрнек) { оператор; }

Бұл модификацияның стретегиясы: "алдымен тексер, сосын орында". Тексеру нәтижесінде ештеме істемей қоюға болады жағдайлар болуы мүмкін. Бұндай цикл денесі бір ретте орындалмауы мүмкін. Әрине, кідірстер болуы мүмкін. Қалыпты жағдайда цикл денесінің әр орындалуы – циклды аяқтаудың қадамы.

Шартты соңында тексеретін циклдің стратегиясы: "алдымен орында, содан соң тексер". Мұндай цикл денесі кем дегенде бір рет орындалады. Модификации синтаксисі:

```
do
 { оператор; }
While (өрнек);
```

2.2.5 Ауысу операторы

Ауысу операторы, C# тіліндегі күрделі операторлардың орындау ретін тоқтатуға мүмкіндік береді. Олар басқа операторлармен фигуралық жақшада жазылады – күрделі оператор денесінде немесе блокта. Олар бірнешеу және бәрін кезекпен қарап шығамыз.

goto операторының қарапайым жазу пішімі бар:

goto [метка|caseконстантное_выражение|default];

C# тілінің барлық операторларында белгі болады - айрықша идентификатор. Бақылауды белгіленген бір операторға ұсыну – бұл классикалық goto операторын қолдану болып табылады.

break және continue операторлары. Құрылымдық бағдарламада пайдалысы «алдыға жылжу» болып саналады (тек артқа шегінуге емес), кейбір жағдайларда шарттарды орындау шағында циклден шығуға

мүмкіндік береді, блокты және операторды таңдауға. Break и continue операторлары арнайы осы мақсаттар үшін пайдаланады.

Break операторы цикл ішінде тұруы мүмкін немесе switch операторындағы case тармақшасын аяқтау мүмкіндігі бар. Оны switch операторында қолдану мысалына демонстрация жасалған. Операторды цикл ішінде орындау барысында, ішкі циклдің өзінің орындалуы аяқталады. Цикл ішінде оператор break көбінесе if операторының тармақшаларында орналасады, циклдің уақытынан бұрын аяқталу шартын тексереді.

Continue операторы цикл ішінде ғана қолданылады. Ішкі циклді аяқтайтын break операторына қарағанда, continue осы циклдің келесі итерацияға өтуін қадағалайды.

Тағыда бір келесі итерацияға өтетін топтардың ішіндегі операторладың бірі return операторы болып табылады, процедуралар мен функциялардың орындалуын аяқтаушы болып табылады. Оның формат жазуы келесі көрсеткіштерге ие:

return [выражение];

Функция үшін оның болуы шарт, неге десеңіз return операторы фукцияны қайтаратын белгілерді тапсырады.

2.3 Зертханалық жұмысты орындауға арналған мысал

Алдыңғы жылдары жеке тапсырмалар тізімінен орта қиындық проблемасын шешуді таңдалған кешенді операторлары C # тілі тақырыбында тағайындау мысалы ретінде .

Есеп 2.1 Сізде 1000 у.е. бар. Сіз 5 компаниялардың акцияларын сатып алуыңыз қажет. 5-тен 20 АҚШ долларына дейін диапазонында кездейсоқ қалыптасқан әрбір компания акцияларының құны (Бұл сауда-саттықтың ұзақтығы өзгеріссіз қалады). Сіздің міндет әрбір қоғам акцияларының шамамен бірдей санының бүкіл сомасын сатып алу болып табылады. Экран мониторына әрбір компанияның сатып алған акцияларының саны , сондай-ақ АҚШ долларынан қалған сомманы шығару керек. Біз мәселені шешу үшін алгоритм ауызша сипаттамасын әзірлеу :

– Бірінші қадам 5-тен 20 диапазонында 5 кездейсоқ сандарды қалыптастыру болып табылады– 5 компаниялардың акцияларының құны:

– Екінші қадам өз кезегінде әрбір компания акцияларын сатып алу сериясын ұйымдастыру болып табылады. Циклмен шарт қажет - біздің ақша кем дегенде бір үлесін бес компаниялардың кез келген сатып алу үшін жеткілікті болып табылады.

– цикл денсінде біздің ақша бөлек әр компанияның акцияларын сатып алуға жеткілікті қажетті жағдайды тестілеу. Егер шарт орындалатын болса, онда біз қарсы 1 компанияның акцияларын сатып алдымыз , және біздің ақша осы компанияның акцияларының құнын азайту үшін молаяды;

– соңғы кезеңінде (цикл аяқталғанда) сіз экранда әр компания үшін сатып алынған акциялардың саны және біздің ақшаның қалғанын көрсетуіңіз керек. Әрине санының қалған оң мен компаниялардың кез келген акцияларының құнынан кем болуы тиіс.

Бағдарлама коды:

```
usingSystem;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
classProgram
  {
staticvoid Main(string[] args)
    {
int a1,a2,a3,a4,a5,k1,k2,k3,k4,k5,c;
Random rnd = newRandom();
Console.WriteLine("Программа моделирования покупки акций у 5
предприятий");
// формируем случайным образом стоимость акций 5 предприятий
     a1 = rnd.Next()%16 + 5;
Console.WriteLine("Стоимость акции 1 предприятия = {0}", a1);
     a2 = rnd.Next() % 16 + 5;
Console.WriteLine("Стоимость акции 2 предприятия = {0}", a2);
     a3 = rnd.Next() % 16 + 5;
Console.WriteLine("Стоимость акции 3 предприятия = {0}", a3);
     a4 = rnd.Next() % 16 + 5;
Console.WriteLine("Стоимость акции 4 предприятия = {0}", a4);
     a5 = rnd.Next() % 16 + 5;
Console.WriteLine("Стоимость акции 5 предприятия = {0}", a5);
// Обнуляем счетчики купленных акций каждого предприятия
     k1=0; k2=0; k3=0; k4=0; k5=0;
// Наши деньги
     c = 1000;
// Запускаем цикл покупки акций
while ((c>=a1)||(c>=a2)||(c>=a3)||(c>=a4)||(c>=a5))
{
if (c>=a1) {k1++; c = c - a1;}
if (c>=a2) {k2++; c = c - a2;}
if (c>=a3) {k3++; c = c - a3;}
if (c>=a4) {k4++; c = c - a4;}
if (c>=a5) {k5++; c = c - a5;}
      }
Console.WriteLine("Купленоакций 1 предприятия = {0} ", k1);
Console.WriteLine("Куплено акций 2 предприятия = {0} ", k2);
Console.WriteLine("Куплено акций 3 предприятия = {0} ", k3);
Console.WriteLine("Куплено акций 4 предприятия = {0} ", k4);
Console.WriteLine("Куплено акций 5 предприятия = {0} ", k5);
Console.WriteLine("Остаток денег  = {0} ", c);
Console.WriteLine("Для продолжения нажмите клавишу Enter");
Console.ReadLine();
    }
```

```
    }
}
```

Бағдарлама жұмысы:
5 компаниялардың акцияларын сатып алуды моделдеу
1 Кәсіпорынның акцияларының құны = 15
2 Кәсіпорынның акцияларының құны = 16
3 Кәсіпорынның акцияларының құны = 11
4 Кәсіпорынның акцияларының құны = 8
5 Кәсіпорынның акцияларының құны = 15
1 Кәсіпорынның сатып алынған акциялар = 16
2 Кәсіпорынның сатып алынған акциялар = 15
3 Кәсіпорынның сатып алынған акциялар = 15
4 Кәсіпорынның сатып алынған акциялар = 16
5 Кәсіпорынның сатып алынған акциялар = 15
Қалдық ақша = 2
Жалғастыру үшін Enter батырмасын басыңыз

2.4 Зертханалық жұмысқа арналған үй тапсырмасы

Қатардың соммасын табыңыз:

$$s = 1 + \frac{1}{2} + \frac{1}{3} - \frac{1}{4} - \frac{1}{5} + \frac{1}{6} + \frac{1}{7} - \frac{1}{8} - \frac{1}{9} + \ldots + \frac{1}{N}$$

N – бүтін сан және диалог режімінде енгізіледі.

2.5 СРС-қа жеке тапсырма

2.5.1 Кітап жүз беттен тұрады Бір бетте 35-тен 45-ке дейін жолдар болуы мүмкін (кездейсоқ сан). Бір жолда «а» әрпі 2-5 аралығында кездесуі мүмкін. (кездейсоқ сан). Кітапта «а» әрпі неше рет басылған?

2.5.2 Сіз 200 беттен тұратын қызықты кітапты оқып жүрсіз. Бір бетті оқуға 50-80 секундты жұмсайсыз. Бетте суреттің болу ықтималдығы 5%. Суретті қарауға кететін уақыт – 5-10 секунд. Бүкіл кітапты оқуға барлығы қанша уақыт жұмсалады?

2.5.3 Кездейсоқ түрде 100 нүктенің X және Y координаттары құрылады. координаттардың диапазоны – минус 15-ден 150-ге дейін. Әрбір ширектікте орналасқан нүктелер санын есептеп, экранға шығарыңыз.

2.5.4 Қатардың қосындысын есептейтін программаны жазыңыз:

$$S = 1 + \frac{1}{2} - \frac{1}{3} + \frac{1}{4} + \ldots - \frac{1}{n}$$

n мәнін диалог режімінде енгізіледі.

2.5.5 Қатардың қосындысын есептеңіз, |x|<1 диалог режімінде енгізіледі.

$$y = \frac{2!}{x^2 * 3!} + \frac{3!}{x^4 * 4!} + \frac{4!}{x^6 * 5!} + \dots$$

Қатардың кезекті мүшесі 0.0001 мәнінен кіші болғанға дейін есептеуді жалғастыру керек.

2.5.6 Автобус бір ауысымда 10-нан 15-ке дейін рейстерді орындайды (кездейсоқ сан). Бір рейсте 130-230 жолаушыны тасымалдайды (кездейсоқ сан). Жолақы 30 теңге. Жолаушы куәлігінің, жолдық билетінің болу ықтималдығы немесе жолаушының төлемеу ықтималдығы - 30%. N ауысымда түсетін табысты анықтау керек. ауысым саны диалог режімінде енгізіледі.

2.5.7 Цехте 20 мың бірлік өнімді өндіретін жұмыс көлемін орындау керек. Күнделікті жұмысқа 10-нан 20-ға дейін жұмысшы шығады (кездейсоқ сан). Әрбір жұмысшының бір күндегі жұмыс өнімділігі 5-15 бірлік (кездейсоқ сан). Бүкіл жұмысты орындау үшін қанша күн керек?

2.5.8 Кітап сөресінде тарихтан, физикадан және химиядан 20 әдебиет орналасқан. Кітаптардың тарих пәнінен болу ықтималдығы – 40%, химиядан – 25%, физикадан – 35%. Тарихтан, физикадан және химиядан қанша кітаптар барын анықтап, монитор экранына шығарыңыз.

2.5.9 Бір студент әрбір сабаққа 3 минуттан 10 минутқа дейін кешігіп келеді (кездейсоқ сан). Бір аптада 20 сабақ бар. 20 сағат кешугуді студент нешенші аптада жинайды?

2.5.10 X натуралдық саны диалог режімінде беріледі. сандағы цифрлардың санын анықтау керек. Санның ең бірінші және ең соңғы цифрларын тауып, экранға шығарыңыз.

2.5.11 |x|<1 и m=-1 режиміндегі диалог үшін берілген қатардағы қосындыны табу. Келесі топтың мүшелері 0.0001 ден аз болған кезде, есеп аяқталады.

$$(1+x)^m = 1 + \frac{m}{1!}x + \frac{m(m-1)}{2!}x^2 + \frac{m(m-1)(m-2)}{3!}x^3 + \dots + \frac{m(m-1)\dots(m-n+1)}{n!}x^n + \dots$$

2.5.12 Кездейсоқ A (X, Y) үйлестіреді және D (X, Y) биіктерге қарама-қарсы жүз тіктөртбұрыш көрсетілген жинақталатын . -150 -Ден 150 дейін үйлестіру мәндер ауқымы баспа саны және координаттар жүйесі (шыңдары түрлі таймнан орналасқан болса, осы іске асуы қарау алынып тасталады) жоғарғы және төменгі тоқсандарда орналасқан тіктөртбұрыш саны.

2.5.13 Бір диалог режимі x (x > 0 және x < 1) берілген қатарының қосындысын есептеңіз. Есептеулер соңы сериясы келесі мерзімді аз болған кезде дәл енгізілген □ = 0,0001 :

$$\sqrt[3]{1+x} = 1 + \frac{1}{3}x - \frac{2}{2!*3^2}x^2 + \frac{2*5}{3!*3^3}x^3 - \frac{2*5*8}{4!*3^4}x^4 + \dots$$

2.5.14 Кездейсоқ 60 ұпай x және y координаттары жинақталаған. Міндер координатының диапазоны минус 50-ден 50-ге дейін. Қашықтық шығу тегі мүмкін , сондай-ақ нүктелер өздері түрлі тоқсандарда бар нүктелерінің тізімін көрсету.

2.5.15 Берілген x <0 қатарының қосындысын есептеңіз есептеу соңына модулінің бірқатар мүше басқа 0.0001 кем болған кезде:
$$arctg(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \ldots$$
2.5.16 Кездейсоқ 90 ұпай x және y координаттары жинақталаған. Міндер координатының диапазоны минус 150-ден 150-ге дейін. Қашықтық шығу тегі мүмкін , сондай-ақ нүктелер өздері түрлі тоқсандарда бар нүктелерінің тізімін көрсету.
2.5.17 X | көрсетілген санына сомасын есептеу | > 0 .
$$\ln\frac{x+1}{x-1} = 2\sum_{n=0}^{\infty}\frac{1}{(2n+1)x^{2n+1}} = 2\left(\frac{1}{x} + \frac{1}{3x^3} + \frac{1}{5x^5} + \ldots\right) \quad |x| > 1$$
Есептеулер соңы модуль бірқатар мүше басқа 0,0001 кем болған кезде .
2.5.18 Координаттар жүйесі X , Y « шығу тегіне орталығы 10 топтарының » нысанаға және 10 бірлік қадам радиусы боялған . 10 бірлік радиусы 10 ұпай салмақ мәніне сәйкес . Келесі « сақина » мақсатты Балл әрбір 1 9-дан төмендеді. Кездейсоқ 10 ұпай (он кадрлар) X және Y координаты жинақталатын . -150 -Ден 150 әр нүктесінде үйлестіру мәндер диапазоны . Ұпайдан атып және кадрлардың бүкіл сериясы жалпы балл жинап ретінде анықтау және басып шығару .
2.5.19 50 тізе радиусы - кездейсоқ орталығы және R X және Y координаты жинақталатын . 5-тен 15 радиусын мәндер шегінде , минус 150 150 үйлестіру мәндер ауқымы . Әр тоқсанда толығымен қанша шеңбер анықтау және басып шығару .
5.2.20 функциясын зерттеу
$$\operatorname{arctg} x = \frac{\pi}{2} + \sum_{n=0}^{\infty}\frac{(-1)^{n+1}}{(2n+1)x^{2n+1}} = \frac{\pi}{2} - \frac{1}{x} + \frac{1}{3x^3} - \frac{1}{5x^5}\ldots \quad x > 1$$
2-ден 10 аралықта . Есептеулер соңы модуль бірқатар мүше басқа 0,0001 кем болған кезде .

2.6 СРСП-да есеп-хатты қорғауға арналған бақылау сұрақтары

2.6.1 C# бағдарламалау тілінің қандай операторларын күрделі операторлар деп атайды?
2.6.2 C# тілінің күрделі операторының жұмыс істеу аймағы қалай белгіленеді?
2.6.3 Қандай есептеу процесін «тармақталған» деп атайды?
2.6.4 Қандай есептеу процесін циклдік деп атайды?
2.6.5 if шартты өту операторының жұмыс істеу және жазу форматы.
2.6.6 switch бағдарламаны ауыстыру операторының жұмыс істеу және жазу форматы.
2.6.7 for циклдік операторының жұмыс істеу және жазу форматы.
2.6.8 for циклдік операторын қолдану қашан орынды ?

2.6.9 for циклна жататын операторлар қандай жағдайда орындалмайды? Мысал келтіріп түсіндіріңіз.
2.6.10 while циклдік оператордың жұмыс істеу және жазу форматы
2.6.11 При каких условиях операторы, принадлежащие циклу while, не выполнятся ни одного раза? Пояснить на примере.
2.6.12 do – while циклдік оператордың жұмыс істеу және жазу форматы.
2.6.13 do – while циклдік операторы қандай жағдайда орындалмайды?
2.6.14 «циклдік» есептеу процесін цикл операторынсыз ұйымдастыруға бола ма? Мысал келтіріп түсіндіріңіз.
2.6.15 Қандай өту операторыларын білесіз?

3 ТАҚЫРЫП C# ТІЛІНІҢ БІРӨЛШЕМДІ МАССИВТЕРІ

3.1 Үшінші тақырыптың мақсаты

C# тілінің бірөлшемді массивтерін ұйымдастыру тәсілдерін оқу. Бірөлшемді массивтерді қолданып есептерді C# тілінді бағдарламалауға практикалық дағдылану.

3.2 Теориялық мәліметтер

3.2.1 Массив түсінігі

Массив біртипті айнымалыларды ұйымдастыруға жол анықтайды. Кейде бір типті айнымалы реттелген жиынтығын массив деп атайды. Әрбір айнымалы массив индексі – массивінде айнымалы саны. C# тілінде , басқа тілдердегі сияқты, индекс бүтін санды типпен беріледі. Бірінші массив айнымалысы 0 индекс, N айнымалы – N-1 индекс.

Кейбір бағдарламалау тілдерінде массивті жариялау кезінде массив айнымалылар саны көрсетіледі. Жиым элементтерінің саны оның жарияланған сәтте белгілі және ол бөлінген болуы мүмкін болса , жад аударма сатысында , мұндай массивтер статикалық деп аталады.

C# тілінде массивпен жұмыс істеу екі деңгейде орындалады. Бірінші деңгейде айнымалылар типі жарияланады . Екінші деңгейде – бағдарламаны орындау жұмысында массивті баптау орындалады, массив элементтерінің саны анықталады. Сондықтан C# тіліндегі барлық массивтер динамикалық болып табылады. Сіз массивті баптандырғанда ол әдетте бөл- жад үздіксіз ауданы болып табылады .

3.2.2 Массивті жариялау

Бірөлшемді массивтер жариялау келесідей:

<тип>[]<объявители>;

где <тип> – тип переменных, объединяемых в массив;
[]– признак одномерного массива;
<объявители> – айнымалы тізімі, айнымалы массив ретінде жариялады. Әрбір ұйымдастырушы атауы немесе баптандыру аты болуы мүмкін. Бірінші жағдайда біз жалқау баптандыру туралы айтып отырмыз . Сіз массивті

жалқау баптандыру кезде қалыптасады екенін түсіну керек , және анықталмаған құндылығы бар массив ғана сілтеме жасайды . Сондықтан массив шын мәнінде орнату емес, және оның элементтері , ол мүмкін емес бағдарламасының есептеу оны пайдалануға инициализацияланып жатқанда .Мысалы, үш массивті жариялау төмендегідей:

```
int[] a, b, c;
```

Әрбір массивті жеке жариялауға болады, мысалы:

```
int[] a;
int[] b;
int[] c;
double[] d;
```

Квадраттық жақшалар айнымалылар аттарына емес, типіне жазылған. Олар түрін анықтау ажырамас бөлігі болып табылады және түрі бүтін элементтерімен бір өлшемді массив түрі , деп түсінген жөн int[] жазыңыз. Айнымалы жарнама анықталған элементтердің оның саны бір өлшемді массив , олардың әрқайсысы көшірмелері , - шекаралары индекстер өзгерту өткендей, бұл тән түріне тиесілі емес , ол айнымалы осы түріне тән .

3.2.2 Массивті баптау

Егер массив баптаусыз жарияланса, онда void мәніне массивтен сілтеме жазылады.

Баптандыру орындалуы конструктор болса , массив элементтері тұрақты (арифметика үшін нөлдік , жол массивтерді бос жолдың) тиісті түрі , және осы массивке байланысты сілтеме бапталады (алапта орын бөлінген) , массив құрылды. Мысалы:

```
//объявление массивов с отложенной инициализацией
  int[] u;
  u = newint[3];
```

Жоғарыда келтірілген мысалда бастапқыда (жиым құрылысшы кезінде - жаңа) , содан кейін бүтін типті массив жариялайды , және қада кеңістікте жазылған нөлдер бүтін типті үш құндылықтарды массивке бөлінді.

Егер массив тұрақты массивпен бапталса, онда жадыда тұрақты массив құрылады, ол жерде сілтеме байланысады. Тұрақты массив элементтерін фигуралық жақшаға жазу керек, мысалы:

```
int[] x ={5,5,6,6,7,7};//размерность вычисляется, new подразумевается
int[] x = newint[]{5,5,6,6,7,7};   // размерность вычисляется
int[] x = newint[6] {5,5,6,6,7,7}; // считается избыточным описанием без ошибки
```

көп жағдайда бағдарламада алдымен массив жарияланады, бапталады. (элементтердің санын міндетті түрде көрсете отырып), одан кейін мәндер толтырылады (нөлдік мәндер өзінде тағайындалады). Мысалы:

```
int[] b = new int[6]; // элементыравны 0
   Random rnd = new Random();
```

```
// формируем случайным образом и выводим его на экран монитора
for (inti = 0; i< 6; i++)
        {
b[i] = rnd.Next()%101;
Console.Write(" {0}", b[i]);
};                              //Символ ; можно не ставить
       Console.WriteLine();
```

3.3 Зертханалық жұмысты орындауға арналған мысал

С # тілінің тақырыбында тағайындау мысалы ретінде қарапайым шешім алдыңғы жылдардың жеке тапсырмалар тізімінен таңдалады.
Есеп 3.1 минус 40-тан 40-қа дейінгі диапазонда 20 кездейсоқ саннан тұратын массив құру. Оны басып шығару. Сұраптауды оң сандардан кему ретімен орналастыру. Жаңа массивті басып шығару.

Есепті шешу алгоритмі есеп шартында анықталған – Біз мәлімдейміз және 20 бүтін айнымалы массив баптандырамыз ; кездейсоқ бүтін санлармен толтыру және бірден монитор экранына шығару; элементтерді сұрыптаймыз, оның мәні >0; массивті басып шығаруды қайталаймыз.
Бағдарлама коды:

```
usingSystem;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
classProgram
  {
staticvoid Main(string[] args)
    {
int i, j, k;
int[] a = newint[20];
Random rnd = newRandom();
// формируем случайным образом массив 20 чисел
// выводим их на экран монитора
Console.WriteLine("Исходный массив: ");
for (i = 0; i <= 19; i++)
      {
a[i] = rnd.Next() % 81 - 40;
Console.Write(" {0}", a[i]);
}
Console.WriteLine();
Console.WriteLine();
// Сортировка положительных чисел массива по убыванию
for (i = 0; i < 19; i++)
for (j = i+1; j <= 19; j++)
if (a[i] >= 0 && a[i] < a[j])
        {
         k = a[i]; a[i] = a[j]; a[j] = k;
```

```
}
// выводим их на экран монитора массива после сортировки
Console.WriteLine("Массив после сортировки: ");
for (i = 0; i <= 19; i++)
Console.Write(" {0}", a[i]);
Console.WriteLine();
Console.WriteLine();
Console.WriteLine("Для продолжения нажмите клавишу Enter");
Console.ReadLine();
    }
  }
}
```

Бағдарлама жұмысы:
Бастапқы массив:
-6 21 -7 -15 -36 17 36 1 15 -11 -31 23 31 -10 -24 16 -6 29 -35 24

Сұрыптаудан кейін массив:
-6 36 -7 -15 -36 31 29 24 23 -11 -31 21 17 -10 -24 16 -6 15 -35 1

Жалғастыру үшін Enter батырмасын басыңыз

3.4 Зертханалық жұмысқа арналған үй тапсырмасы

100 нүктенің X және Y координаттары кездейсоқ түрде құрылады. Координаттар мәндерінің диапазоны -150-ден +150-ге дейін. Әрбір ширектік үшін оның нүктелерін және араларындағы қашықтық минималды болатын екі нүктені басып шығару керек.

3.5 СРС-қа арналған жеке тапсырма

3.5.1 Диапазоны минус 40-тан 60-ке дейін болатын, кездейсоқ 25 бүтін сандардан тұратын массивті құру керек. Массивті басып шығару керек. 3 max чсанды тауып, оны массивтің басына орналастырыңыз (сандарға сұрыптауды жүргізуге болмайды). Жаңа массивті экранға шығарыңыз.
3.5.2 Диапазоны минус 50-ден 50-ге дейін болатын, кездейсоқ 30 бүтін сандардан тұратын массивті құру керек. Массивті басып шығару керек. Реті тақ болып келетін сандарды тауып, оларды жаңа массивке жазып, экранға шығару керек.
3.5.3 Диапазоны 0-ден 9-ға дейін болатын, кездейсоқ 100 бүтін сандардан тұратын массивті құру керек. Массивті басып шығару керек. Массивте ең жиі кездесетін санды тауып, экранға шығарыңыз.
3.5.4 Диапазоны минус 20-дан 20-ға дейін болатын, кездейсоқ 20 бүтін сандардан тұратын массивті құру керек. Массивті басып шығару керек. Массивтің алғашқы 10 санын өсу бойынша, ал екінші ондықты кему бойынша сұрыптау керек. Жаңа массивті экранға шығарыңыз.

3.5.5 Диапазоны минус 40-тан 40-қа дейін болатын, кездейсоқ 30 бүтін сандардан тұратын массивті құру керек. Массивті экранға басып шығару керек. Барлық теріс сандарды массивтің басына орналастыру керек (сандарға сұрыптауды жүргізуге болмайды). Жаңа массивті экранға шығарыңыз.
3.5.6 Диапазоны минус 50-ден 50-ге дейін болатын, кездейсоқ 20 бүтін сандардан тұратын массивті құру керек. Массивті экранға басып шығару керек. Барлық жұп сандарды массивтің сол жағына, ал тақ сандарды массивтің оң жағына орналастыру керек. Жаңа массивті экранға шығарыңыз.
3.5.7 Жүз тіктөрбұрыштың қарама-қарсы төбелері арқылы берілген A(X,Y) және B(X,Y) координаттары кездейсоқ түрде құрылады. Координаттардың диапазоны - минус 150-ден 150 дейін. Аудандары бірдей тіктөрбұрыштарды тауып, экранға шығыру керек.
3.5.8 50 шеңбердің центрлерінің X, Y координаттары және R – радиусы кездейсоқ түрде құрылады. Координаттардың диапазоны 5-тен 15-ке дейін. аралары ең алыс болып келетін шеңберлерді тауып, экранға шығыру керек.
3.5.9 Диапазоны минус 30-дан 30-ға дейін болатын, кездейсоқ 20 бүтін сандардан тұратын массивті құру керек. Массивті экранға шығару керек. Массивтің максимал және минимал элементтерінің орындарын ауыстару керек. Жаңа массивті экранға шығарыңыз.
3.5.10 Диапазоны минус 40-тан 40-қа дейін болатын, кездейсоқ 40 бүтін сандардан тұратын массивті құру керек. Массивті экранға шығару керек.
Массивтің максимал және минимал элементтерінің арасында орналасқан элементтердің көбейтіндісін экранға шығарыңыз.
3.5.11 0-ден 100 дейін кездейсоқ бүтін сандардың жиымын 100 құрайды. Оны басып шығару. Барлық нөмірлер 30 -ден көп , бірақ массивтің басында жазылған 70 кем . Жаңа массивті басып шығарыңыз.
3.5.12 -40 40 диапазонында 40 бүтін кездейсоқ сандардың массивін қалыптастыру үшін. Оны басып шығару. Барлық элементтері оны алып тастау арқылы массивті қысу модуль 20 бірліктен аспайд . массив соңында қалғандарды нөлмен толтыру. Жаңа массивті басып шығарыңыз.
3.5.13 0-ден 60 диапазонында 50 бүтін кездейсоқ сандардың массивін қалыптастыру үшін . Оны басып шығару. Массив тұрған элементтері - бірінші жартысында тақ позицияларда , ал екінші жартысында тұрған элементтерді шешіледі , сондықтан массивті түрлендіру . Жаңа массивті басып шығарыңыз.
3.5.14 0-ден 10 дейін кездейсоқ бүтін сандардың массивін 100 сан құрайды. Оны басып шығару. Басып шығару статистика - қанша рет массивте әрбір нөмірі.
3.5.15 -30 -Ден 30 диапазонында 20 бүтін кездейсоқ сандардың массивін қалыптастыру. Оны басып шығару. Квадраттарының барлық теріс элементтерін ауыстырыңыз және өсу тәртібімен массивті ұйымдастырыңыз. Жаңа массивті басып шығарыңыз.

3.5. -50 -ден 50 диапазонында 20 бүтін кездейсоқ сандардың массивін қалыптастырады. Оны басып шығару. Сұрыптау тақ жерлерде тіпті жер мен элементтері тұрған жеке элементтерін артуы бойынша . Жаңа массивті басып шығарыңыз.
3.5. Векторларын A және B , 7 элементтерінің әрбір өлшем қалыптастыру және басып шығару үшін . Векторларының элементтерінің мәндері минус 30 30 диапазонында тиесілі кездейсоқ сандарды генерациялау үшін . Табу және оның элементтері векторлар A және B тиісті элементтерін өнім айқындалады басып шығару вектор C (өлшемі 7) .
3.5.18 Пішін минус 50 -ден 50- диапазонында 20 кездейсоқ бүтін сандардың массиві. Оны басып шығару. Массвитің 1 элементін 20 элементпен, 2-ні 19-бен және 10-мен орынын ауыстыру. Жаңа массивті басып шығарыңыз.
3.5.19 Кездейсоқ A (X, Y) үйлестіреді және D (X, Y) биіктерге қарама-қарсы жүз тіктөртбұрыш көрсетілген жинақталатын . 0-ден минус 150 үйлестіру мәндер массиві . (- Квадраттар бөлек екі жақ жұп 4 тараптар және барлық тараптар қанағаттандырады онда тіктөртбұрыш санын басып шығару) анықтау және сол тараптардың осы тіктөртбұрыш кез келген жағдайда басып шығарыңыз.
3.5. 100 тізе радиусы - кездейсоқ орталығы және R X және Y координаты жинақталатын . 150 минус 150 мәндерді үйлестіру ауқымы , 5-тен 25 радиусы мәндер диапазоны. Максимум анықтау және « салынған » топтарының (шеңбер кейбір басқа бөлігінде , т.б. толық шеңбері) ретін басып шығарыңыз.

3.6 СРСП-да есеп-хатты қорғауға арналған бақылау сұрақтары

3.6.1 Сілтеме типі түсінігі. Мысал.
3.6.2 Массив түсінігі. Деректерді массив түрінде ұйымдастыру мысалдары.
3.6.Қандай массивтер бағдарламада жарияланады? Мысал.
3.6.4 Массивті баптандыру қалай орындалады? Мысал.
3.6.5 Кездейсоқ сандардың генераторын пайдаланып массивтерді қалай құруға болады? Мысал.
3.6.6 Массив элементтеріне ассоциатиті алгоритм бойынша қатынау. Мысал.
3.6.7 Массив элементтерін әдістік таңдау бойынша сұрыптау алгоритмі. Алгоритм және код фрагментін сұрыптаудың ауызша сипаттамасы .
3.6.8 «Көпіршік» алгоритмі бойынша массив элементтерін сұраптау. Алгоритм және код фрагментін сұрыптаудың ауызша сипаттамасы .
3.6.9 Орын ауыстыру алгоритмі (оңға немесе солға ауыстыру) деректер массивтерінде. Алгоритм және код фрагментін сұрыптаудың ауызша сипаттамасы .
3.6.10 Динамикалық массив түсінігі. Мысал.
3.6.11 Іздеу массивінің түсінігі және іздеу кілт. Мысал .
3.6.12 Массив элементтерін бірізділік іздеу алгоритмі. Бірізділік іздеу алгоритмінің артықшылығы және кемшілігі.

3.6.13 Массив элементінің блоктық іздеу алгоритмі. Блоктық іздеу алгоритмінің артықшылығы және кемшілігі.
3.6.14 Массив элементтерінің екілік іздеу алгоритмі. Екілік іздеу алгоритімінің артықшылығы және кемшілігі.
3.6.15 Адрес-хештеудің түрлену кілтінің іздеу алгоритмі.

ТАҚЫРЫП 4 C# ТІЛІНІҢ ӘДІС-ФУНКЦИЯСЫН ҚОЛДАНУ

4.1 Төртінші тақырып мақсаты

Класс әдістерін және мәліметтерін оқу, әдістерді пайдаланып консоль қосымшасын құруға практикалық дағдылану.

4.2 Теориялық мәліметтер

4.2.1 Әдіс түсінігі

C# тілі объектілі-бағдарланған бағдарламалау тілі болып табылады және класс оның негізі болып табылады. Класстар мәліметтер типі ретінде қарастырылады , оның компоненттері ретінде (басқалармен бірге) класс өрістерін қамтиды (оның деректері) және класс әдістері (оның функциясы). Класс әдістері "қызмет етеді" мәліметтерге, оларды өзгертумен айналысады.

C# тілінде функциялар кейбір класстардың әдістер ретінде қызмет етеді, олар класстан тыс жүреді.

C# тілінде арнайы кілттік сөздер жоқ – method, procedure, function, бірақ олар жайлы түсінік бар. Әдісті жариялау синтаксисі әдісті – процедура немес функция екенін анықтауға мүмкіндік береді.

4.2.2 Класс әдісінің жазу форматы

Класс әдісінің жазу форматы келесідей:

void немес тип_метода имя_метода(список_формальных_параметров)
{ тело метода}

Егер әдіс типінің орнына void мәні берілсе, онда әдіс процедура сияқты жұмыс істейді.

Егер void сөзі болмаса, орына әдіс типі көрсетілсе, онда әдіс функция ретінде жұмыс істейді.

Әдіс атауын жазу міндетті , ол формальды параметрлер тізімі емес жағдайда қажет болуы оның әдісінің түрін , әдісі аты мен жақшаларды , көрсету болып табылады. Әдіс атауы және формальді параметрлер тізімі әдістің қолын көрсетеді (бағдарламаны пайдаланған кезде қажет элементтер). Назар аударыңыз, қол қоюға формальді параметрлердің аты кірмейді, бұл жерде аргументтер аты маңызды. Қол қоюға әдіс типі де кірмейді.

Мысалы, әдіс процедурасын келесідей көрсетуге болады:

voidpoisk() {…};

оны бағдарламада қолдану келесідей көрсеткішпен шектеледі:

poisk();

Әдіс функциясының типін жариялағанда, мысалы:

intkol(){…};

бағдарламаны пайдалану түрі бүтін бір айнымалы берілуге тиіс (әдіс типіне сәйкес), мысалы:

d= kol();

4.2.3 Класс әдісінің формальді параметрлері

Әдіс параметрінің формальді тізімі бос болуы мүмкін және класс әдістері үшін қалыпты жағдай. Тізім параметрлері бекітілген сан болуы мүмкін, үтірмен бөлінген.

Бір формальді параметрдің жазу форматы:

[ref или out или params]тип_параметраимя_параметра

Жақшада берілген мәндер міндетті емес.

Тип пен параметр атауын көрсету міндетті болып табылады.

Ресми параметрлерге тіркелген санға қарамастан нақты параметрлерінің кез келген санына өтуге мүмкінді бар. Ресми параметрлер тізімінде осы мүмкіндікті іске асыру үшін params кілт сөзі болуы тиіс. Бұл өткен параметр тізімі түрі кез келген массив ретінде жарияланды ғана хабарландыруда пайда болуы мүмкін. Бұл ресми параметрді шақыру кезде нақты параметрлердің еркін санына сәйкес келеді.

Барлық әдіс параметрлері үш топқа бөлінеді: кіріс , шығыс және жаңартылған .

Бірінші топ параметрі әдіс ақпаратын береді, олардың мәні әдіс денесінде оқылады.

Екінші топ параметрі әдістің нәтижесін көрсетеді, әдіс жұмыс істеп тұрған кезде мән қабылдайды.

Үшінші топ параметрі екі функция атқарады. Бұл құндылықтар есептеу пайдаланылады және әдіс нәтижесінде жаңартылады. Шығыс параметрі out кілттік сөзімен бірге жүруі тиіс, жаңартылады – ref. Енгізу параметрлері не болсақ, олар кілт сөзсіз көрсетілген. Егер параметр out кілттік сөзімен шығыс ретінде жарияланса, онда әдіс денесінде міндетті түрде меншіктеу операторы болуы мүмкін, ол параметр мәнін көрсетеді. Әйтпесе, қате компиляция кезінде жүреді. Әдіс денесі операторлар тізбегін көрсетеді және айнымалыларды сипаттайды, фигуралық жақша ішінде көрсетілген. Егер сөз функция денесі жайлы болса, онда блокта кем дегенде бір оператор болуы керек, функция мәнін return<выражение> түрінде қайтарады (мән типі функция типімен бірдей болуы керек).

Әдіс денесінде көрсетілген айнымалылар сол әдіс ішінде локальді болып есептеледі.

Әдіс денесінің операторларын жазуда әдіс айнымалылардың локальді аты қолданылады, класстың өріс аттары (әдістер жаһандық айнымалылар класы ретінде қарастырылады) және әдіс параметрлері аттары.

4.3 Зертханалық жұмысты орындауға арналған мысал

Әдісті пайдаланып жұмыс істеу үшін таза білім беру бағдарламасын қарастырайық.

Есеп 4.1 минус 50-ден 50-ге дейінгі диапазонда кездейсоқ 20 саннан тұратын массив тұрғызу керек. Оны басып шығару. Массив мәнін 1 разрядқа солға жылжытуды орында. Тапсырманың әрбір тапсырмасын көрсететіндей бағдарламаны әдіс түрінде ұсыну. Әдіс ішінде кіріс, шығыс және жаңартылған параметрлерді қолдану. Мәзірді қарастыру.

Есептің шартына сәйкес 3 әдісті орындау қажет:

– массив құру (массив үшін шығыс параметрін қолданамыз);

– массивті басу (массив үшін кіріс параметрін қолданамыз);

– массивті жылжыту (массив үшін жаңартылған параметрді қолданамыз).

Бағдарламаның мәзірін ұйымдастыру үшін цикл while циклін және switch () операторын қолданамыз;

Бағдарлама коды:

```
usingSystem;
usingSystem.Collections.Generic;
usingSystem.Linq;
usingSystem.Text;
namespace ConsoleApplication1
{
classProgram
  {
publicstaticvoid sozd(outint[] ma)
    {
     ma = newint[20];
Random rnd = newRandom();
for (int i = 0; i < 20; i++)
       ma[i] = rnd.Next() % 101 - 50;
Console.WriteLine("Массивсоздан !!");
    }

publicstaticvoid zadvig(refint[] ma)
    {
int k;
for (int i = 0; i < 19; i++)
      {
       k = ma[i]; ma[i] = ma[i + 1]; ma[i + 1] = k;
}
Console.WriteLine("Сдвиг массива на 1 разряд выполнен !");
}

publicstaticvoid prinmas(int[] ma)
    {
for (int i = 0; i < 20; i++)
Console.Write(" {0}", ma[i]);
Console.WriteLine();
```

```
    }

staticvoid Main()
    {
int[] a = newint[20];
int k = 0;
string buf;
while (k < 4)
      {
Console.WriteLine("1 - Создатьмассив 20 чисел");
Console.WriteLine("2 - Переместить массив на 1 разряд влево");
Console.WriteLine("3 - Печать массива");
Console.WriteLine("4 - Выход из программы");
Console.WriteLine("Введите пункт меню программы");
buf = Console.ReadLine();
       k = Convert.ToInt32(buf);
switch (k)
        {
case 1: sozd(out a);  break;
case 2: zadvig(ref a); break;
case 3: prinmas(a); break;
default: break;
}
       }
}
  }
}
```

Бағдарлама жұмысы:
1 - 20 саннан тұратын массив құру
2 – массивті 1 рарядқа солға жылжыту
3 – массивті басып шығару
4 – бағдарламадан шығу
Бағдарламаның мәзірін енгізіңіз
1
Массив құрылды !!
1 - 20 саннан тұратын массив құру
2 - массивті 1 рарядқа солға жылжыту
3 - массивті басып шығару
4 - бағдарламадан шығу
Бағдарламаның мәзірін енгізіңіз
3
-24 -20 40 46 -26 -16 -45 -46 -39 32 38 -38 -18 -2 3 -26 -40 -17 -34 -39
1 - 20 саннан тұратын массив құру
2 - массивті 1 рарядқа солға жылжыту
3 - массивті басып шығару
4 - бағдарламадан шығу
Бағдарламаның мәзірін енгізіңіз
2

Массивті 1 разраядқа солға жылдыту орындалды!
1 - 20 саннан тұратын массив құру
2 - массивті 1 рарядқа солға жылжыту
3 - массивті басып шығару
4 - бағдарламадан шығу
Бағдарламаның мәзірін енгізіңіз
3
-20 40 46 -26 -16 -45 -46 -39 32 38 -38 -18 -2 3 -26 -40 -17 -34 -39 -24
1 - 20 саннан тұратын массив құру
2 - массивті 1 рарядқа солға жылжыту
3 - массивті басып шығару
4 - бағдарламадан шығу
Бағдарламаның мәзірін енгізіңіз

4.4 Зертханалық жұмысқа үй тапсырмасы

Диалог режимінде берілген x (x>0 және x<1) үшін әдіс-функцияны қолданып, қатардың соммасын табыңыз.Используя метод-функцию вычислить сумму ряда для заданного в режиме диалога x (x>0 и x<1). Қатардың кезекті мүшесі ε=0.0001 дәлдіктен кіші болғанша есептеуді жалғастыру керек.:

$$\sqrt[3]{1+x} = 1 + \frac{1}{3}x - \frac{2}{2!*3^2}x^2 + \frac{2*5}{3!*3^3}x^3 - \frac{2*5*8}{4!*3^4}x^4 + \ldots$$

4.5 СРС-да орындалатын жеке тапсырма

Назар аударыңыз! Орындалатын есептің әрбір әдіс-функциясында немесе әдіс-процедурасында кіріс және жаңартылатын формальді параметрлерболу керек. Егер осы параметрлер қолданбаса, жеке тапсырма қорғауға қабылданбайды.

4.5.1 20 кесіндінің (Y1,Y2) және (X1,X2) координаттары кездесоқ түрде құрылады. Координаттардың мәндері – 0-ден 100 дейінгі бүтін сандар. Максимал ұзындықты кесіндінің нөмерін тап. Әдіс-процедураны қолдан.
4.5.2 Қатарды есептейтін бағдарламаны жаз:

$$Y = \sum_{N=0}^{100} (x^{2*N+1}/(2*N+1)*\sin(2*N+1)*x/10),$$

мұнда x –диалог режимінде енгізіледі интервалы 0 ден 1 дейін; N – 1 қадаммен 0-ден 100 дейін өзгеріледі. Әдіс-функцияны қолдану керек.
4.5.3 $ax^2+bx+c = 0$ өрнектің түбірлерін табатын бағдарламаны жаз. a, b және c мәндері диалог режимінде енгізіледі. Өрнектің түблері бар болу шартын тексеріңіз және тиісті хабарламаны шығарыңыз. Бұл есепті шешу үшін әдіс-процедураны қолданыңыз.
4.5.4 Әдіс-функцияны қолданып C-нің мәнің есепте:

$$C_N^M = \frac{N!}{M!(N-M)!}$$

Егер N саны 0-ден үлкен болса, N! есепте, әйтпесе – экран мониторына тиісті хабарламаны шығар. N және M мәндері кездейсоқ түрде құрылады немесе экран мониторынан енгізіледі.

4.5.5Дөңгелек берілген, центрдің координаттары (50,50) және радиус = 30. 100 нүктенің (x,y) координаттары кездейсоқ түрде құрылады. Координаттар мәндерінің диапазоны 0-ден 100-ге дейін. Қанша нүкте дөнгелектің ішіне кіргенін анықтаңыз. Әдіс-процедураны қолданыңыз

4.5.6 Y = 1*2*3+2*3*4+3*4*5+…+(n-1)*n*(n+1) соммасын тап,n диалог режимінде беріледі. Әдіс-функцияны қолданыңыз.

4.5.7 Циклда диапазоны 0-ден 100-ге дейін болатын, кездейсоқ 20 бүтін сандар қалыптастырылады.Сандардың мәндерін басып шығарыңыз. max және min сандарды анықтау үшін әдіс-функциянықолданыңыз. Оларды басып шығарыңыз.

4.5.8 Диалог режимінде енгізілген және|x|>1 үшін қатардың соммасын табыңыз.

$$y = \frac{2!}{x^2 * 3!} + \frac{3!}{x^4 * 4!} + \frac{4!}{x^6 * 5!} + \dots$$

Қатардың кезекті мүшесі 0.0001 мәнінен кіші болғанға дейін есептеуді жалғастыру керек.Әдіс-функцияны қолданыңыз.

4.5.9 100 нүктенің X және Y координаттары кездейсоқ түрде құрылады. Координаттар мәндерінің диапазоны -150-ден +150-ге дейін. Әрбір ширекте орналасқан нүктелердің санын экранға шығарыңыз. Ось бойында орналасқан нүктелерді жеке қарастырыңыз. Әдіс-процедураны қолданыңыз.

4.5.10 |x| > 0 үшін әдіс-функцияны қолдана отырып, қатардың соммасын табыңыз.

$$\ln\frac{x+1}{x-1} = 2\sum_{n=0}^{\infty}\frac{1}{(2n+1)x^{2n+1}} = 2\left(\frac{1}{x} + \frac{1}{3x^3} + \frac{1}{5x^5} + \dots\right) \quad |x| > 1$$

Қатардың кезекті мүшесі 0.0001 мәнінен кіші болғанға дейін есептеуді жалғастыру керек.

4.5.11 Жүз тіктөртбұрыштардың қарама-қарсы төбелері A(X,Y) және B(X,Y) кездесоқ түрде құрылады. Координаттар мәндерінің диапазоны -150-ден +150-ге дейін. Координаттар жүйесінің жоғарғы және төменгі бөліктерінде орналасқан тіктөрбұрыштардың санын тауып, экран мониторына шығару керек (егер төбелері әртүрлі ширекте жатса, онда ол тіктөртбұрыш қарастырылмайды). Әдіс-процедураны қолдану керек.

4.5.12 x = 5 үшін әдіс-функцияны қолдана отырып, қатардың соммасын табыңыз.

$$\operatorname{arctg} x = \frac{\pi}{2} + \sum_{n=0}^{\infty}\frac{(-1)^{n+1}}{(2n+1)x^{2n+1}} = \frac{\pi}{2} - \frac{1}{x} + \frac{1}{3x^3} - \frac{1}{5x^5}\dots \quad x > 1$$

Қатардың кезекті мүшесі 0.0001мәнінен кіші болғанға дейін есептеуді жалғастыру керек.

4.5.13 Диапазоны минус 30-дан 30-ға дейін болатын, кездейсоқ 20 бүтін сандардан тұратын массивті құру керек. Массивті басып шығару керек. Массивтің максималды және минималды элементтердің орындарын ауыстыру керек. Жаңа массивті экранға шығарыңыз. Әрбір әрекетті әдіс-процедура түрінде көрсетіңіз.

4.5.14 60 нүктенің X және Y координаттары кездейсоқ түрде құрылады. Координаттар мәндерінің диапазоны -150-ден +150-ге дейін. Әр ширекте орналасқан аралары максимал болатын нүктелердің тізімін экран мониторына шығару керек. Әдіс-процедураны қолдану керек.

4.5.15 Диапазоны 0-ден9-ға дейін болатын, кездейсоқ 100бүтін сандардан тұратын массивті құру керек. Массивті басып шығару керек. Массивте ең жиі кездесетін санды тауып, экран мониторына шығару керек. Әрбір әрекетті әдіс-процедура түрінде көрсетіңіз.

4.5.16 Диалог режимінде берілген |x|<1 және m=-1 үшін әдіс-функцияны қолдана отырып, қатардың соммасын табыңыз. Қатардың кезекті мүшесі 0.0001мәнінен кіші болғанға дейін есептеуді жалғастыру керек:

$$(1+x)^m = 1 + \frac{m}{1!}x + \frac{m(m-1)}{2!}x^2 + \frac{m(m-1)(m-2)}{3!}x^3 + \ldots + \frac{m(m-1)\ldots(m-n+1)}{n!}x^n + \ldots$$

4.5.17 Диапазоны минус 40-тан 60-ке дейін болатын, кездейсоқ 25 бүтін сандардан тұратын массивті құру керек. Массивті басып шығару керек. 3 max санды тауып, оны массивтің басына орналастырыңыз (сандарға сұрыптауды жүргізуге болмайды). Жаңа массивті экранға шығарыңыз. Әрбір қадамды әдіс-процедура түрінде рәсімдеңіз.

4.5.18 60 нүктенің X және Y координаттары кездейсоқ түрде құрылады. Координаттар мәндерінің диапазоны -150-ден +150-ге дейін. Бір ширекте орналасқан аралары минималды болатын нүктелердің тізімін экран мониторына шығару керек. Әдіс-процедураны қолдану керек.

4.5.19 Берілген x үшін қатардың соммасын есепте. Қатардың кезекті мүшесі 0.001 мәнінен кіші болғанға дейін есептеуді жалғастыру керек:

$$arctg(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \ldots$$

Әдіс-процедураны қолдану керек.

4.5.20 Жүз тіктөртбұрыштардың қарама-қарсы төбелері A(X,Y) және B(X,Y) кездесоқ түрде құрылады. Координаттар мәндерінің диапазоны -150-ден +150-ге дейін. Әрбір ширекте орналасқан тіктөрбұрыштардың санын тауып, экран мониторына шығару керек (егер төбелері әртүрлі ширекте жатса, онда ол тіктөртбұрыш қарастырылмайды). Әдіс-процедураны қолдану керек.

4.6 СРСП-да есеп-хатты қорғауға арналған бақылау сұрақтары

4.6.1 C# тілінің класс әдісі түсінігі. Мысал.

4.6.2 C# тілінің класс әдісін жазу форматы.

4.6.3 Класс әдісіне қолжеткізудің қандай спецификаторын білесіз? Мысал.
4.6.4 C# тілінің класстар әдісінің `staticvoidMain()-мен` мәлімет алмасу механизмі?
4.6.5 Класс әдістерінің қандай кіріс параметрлерін білесіз? Мысал.
4.6.6 Класс әдістерінің қандай шығыс параметрлерін білесіз? Мысал.
4.6.7 Класс әдістерінің қандай жағартылған параметрлерін білесіз? Мысал.
4.6.8 C# тіліндегі функция түсінігі. Мысал.
4.6.9 C# тілінде қайда және қашан функция типі жазылады? Мысал.
4.6.10 C# тілінде процедура түсінігі. Мысал.
4.6.11 C# тілінің класс әдістері айнымалыларының локальді және глобальді түсінігі. Мысал.
4.6.12 C# тілінің класс әдістері денесінде қандай «аты» жазу операторы қолданылуы мүмкін?
4.6.13 Бір әдіс ішіне (мысалы, `voidMain()`) басқа әдісті жариялауға бола ма? Мысал.
4.6.14 Рекурсия түсінігі. Мысал.
4.6.15 Рекурсия жетістігі және кемшілігі.

5 C# ТІЛІНІҢ КӨПӨЛШЕМДІ МАССИВІ

5.1 Бесінші жұмыс мақсаты

Көпөлшемді массивтерді жариялау ережелерін оқу, көпөлшемді матрицаларды қолданып, консоль қосымша құруға практикалық дағдылану.

5.2 Теориялық мәліметтер

5.2.1 Көпөлшемді массив түсінігі

Массивтерді бірөлшемді және көпөлшемді бөлу тарихи сипаттама болып табылады. Олардың арасында принципті айырмашылық жоқ. Бір өлшемді массивтер - көпөлшемді массивтердің арнайы жағдайы.

Бірөлшемді массивтер математикалық құрылымды көрсетуге мүмкіндік береді, оларға векторлар, екіөлшемді матрицалар, үшөлшемді - трехмерные – деректер текшелері, жоғары өлшемді массивтер – көпөлшемді деректер текшелері.

Массив өлшемі типтің сипаттамасы болып табылады. Көпөлшемді массивті мәлімдемені жазу форматы бойынша жариялау төмендегідей:

<тип>[, … ,] <объявители>;

Мысалы: int[,] matri; int[,,] kubi;

Массив өлшемі жақша ішінде үтір арқылы көрсетіледі.

Үтірлер саны 1-ге артық болса массив өлшемін көрсетеді. Бірөлшемді массивтердегі сияқты көпөлшемді массивтерде де жариялау болады. Біз оны граф тарауында қолданатын боламыз. Әдетте , көп өлшемді массивтер баптандыру массивін құру үшін бағдарламалық қамтамасыз жұмыс істейді.

5.2.2 Массивтердің массиві

C# тілінде массивтердің тағы бір түрі массивтердің массиві болып табылады, оларды басқаша «сынған» массивтер деп атайды (jaggedarrays).

Массивтердің массивін бірөлшемді массив деп қарастыруға болады, олардың элементі массив болып табылады, олардың элементінің элементі де массив болып табылады, белгілі бір деңгейге дейін жалғаса береді.

Қандай жағдайларда бұл деректер құрылымдарында қажет болуы мүмкін? Бұл массивтер ағаштарды ұсыну үшін пайдаланылуы мүмкін, онда түйіндер балалардың кез келген санын беруі мүмкін. Бұл, мысалы, отбасы ағаш болады. Бірінші деңгей төбесі - Fathers, ікені білдіреді, бірөлшемді массив болып берілуі мүмкін, сондықтан Fathers[i] – бұл і-дің әкесі. Екінші деңгей төбесі массивтердің массиві - Children, сондықтан Children[i] – бұл массив і әкесінің баласы, ал Children[i][j] - і-ші әкенің j-ші баласы . Немерелерін көрсету үшін үшінші деңгей қажет, сондықтан GrandChildren [i][j][k] будет і-ші әкенің j-ші баласының к-ші немересін көрсетеді.

Массивтерді жариялаудың кейбір артықшылықтары бар. Егер көпөлшемді массивтердің өлшемдерін жариялағанда үтір қолданылса, «үзілген» массивтер үшін түсінірек белгі қолданылады – тік жақша жұптылығы; мысалы, int[][] массивті белгілейді, оның элементі - int типінің бірөлшемді массиві.

Ең қиыны массивтерді құру және баптау. Бұл жерде new int[3][5] конструкторын шақыруға болмайды, «үзілген» жиынтық массивін бермейді. Ең төменгі деңгейде әрбір массив үшін конструктор шақыру керек. Мұндай жариялау массивтердің күрделілігі болып табылады. Мысалы: masmasi бүтін типінің абстрактілі массивтің массивін жариялау төмендегідей:

```
int[][]masmasi = new int[3][]
  {
        newint[] {5,7,9,11},
        newint[] {2,8},
        newint[] {6,12,4}
};
```

Массив masmasi-дің тек екі деңгейі бар. Оның үш элементі бар деп санауға болады, оның әрбіреуі массив болып табылады. Ішкі массивті құру үшін әрбір массивке new конструкторын шақыру керек. Берілген мысалда ішкі массив элементтері мән иеленеді. Бұл массивті келесідей жариялауға және баптауға болады:

```
int[][]masmasi  = new int[3][]
{
        newint[4],
        newint[2],
        newint[3]
};
```

Бұл жағдайда массив элементтерін баптау кезінде нөлдік мән қабылдайды. Шынайы баптауды бағдарламалық жолмен орындау қажет.

Айта кетерлік жағдай, жоғарғы деңгей конструкторында 3 тұрақтысын ескермей және жай new int[][] деп жазуға болады. Ең қызығы, бұл конструкторды шақырған кезде оны мүлдем ескермей қою:

```
int[][]masmasi =
{
        newint[4],
        newint[2],
        newint[3]
};
```

Бірақ төменгі деңгей конструкторы қажет болып табылады.

5.2.3 Сызықтық алгебра алгоритмдері

Матрица дегеніміз m жолдан және n бағаннан тұратын сандар жиыны. Программист үшін матрица – екіөлшемді матрица. Егер m = n болса матрица квадраттық деп аталады, кері жағдайда тікбұрышты. m және n сандары матрица өлшемін анықтайды. Тікбұрышты матрицаны ауыстыру, қосу, көбейту операциялары анықтайды.

A - матрицасының өлшемі m*n болсын (m жолдан және n бағаннан) $a_{i,j}$ элементімен. Ауыстыру матрицасы деп B = AT өлшемі n*m, оның элементі $b_{i,j} = a_{j,i}$. Ауыстыру матрицасында берілген матрица жолы баған болып табылады.

$$A = \left\| \begin{array}{l} a_{1,1}, a_{1,2} \dots a_{1,n} \\ a_{2,1}, a_{2,2} \dots a_{2,n} \\ \dots \\ a_{m,1}, a_{m,2} \dots a_{m,n} \end{array} \right\| \quad B = A^T = \left\| \begin{array}{l} a_{1,1}, a_{2,1} \dots a_{m,1} \\ a_{1,2}, a_{2,2} \dots a_{m,2} \\ \dots \\ a_{1,n}, a_{2,n} \dots a_{m,n} \end{array} \right\|$$

Қосу операциясы бірдей өлшемді тікбұрышты матрицаны анықтайды. A, B, C - тікбұрышты матрицасы m*n өлшемді болсын. Онда матрицаның суммасы келесідей анықталады:

$$A = \left\| \begin{array}{l} a_{1,1}, a_{1,2} \dots a_{1,n} \\ a_{2,1}, a_{2,2} \dots a_{2,n} \\ \dots \\ a_{m,1}, a_{m,2} \dots a_{m,n} \end{array} \right\| \quad B = \left\| \begin{array}{l} b_{1,1}, b_{1,2} \dots b_{1,n} \\ b_{2,1}, b_{2,2} \dots b_{2,n} \\ \dots \\ b_{m,1}, b_{m,2} \dots b_{m,n} \end{array} \right\|$$

$$C = A + B = \left\| \begin{array}{l} a_{1,1} + b_{1,1}, a_{1,2} + b_{1,2} \dots a_{1,n} + b_{1,n} \\ a_{2,1} + b_{2,1}, a_{2,2} + b_{2,2} \dots a_{2,n} + b_{2,n} \\ \dots \\ a_{m,1} + b_{m,1}, a_{m,2} + b_{m,2} \dots a_{m,n} + b_{m,n} \end{array} \right\|$$

Көбейту матрицасы тікбұрышты матрицаларға анықталған, олардың бірінші көрсеткішінің бағандар саны екінші көрсеткіштің жолдар санына тең.

Матрица жұмысының жолдар саны бірінші көрсеткіштің жолдар санына, екінші көрсеткіштің бағандар санына тең.

A – матрицасының өлшемі m*p, B – өлшемі p*n, онда C матрицасы = A*B оның өлшемі m*n. Матрица жұмысының элементтері жүп элементтер ретінде анықталады, бірінші көрсеткіштің жолдар саны екінші көрсеткіштің бағандар санына сәйкес келеді.

$A = ||a_{i,j}|| \quad i = 1, ...m;\ j = 1, ...p; \quad B = ||b_{j,k}|| \quad j = 1, ...p;\ k = 1, ...n$
$C = A * B = ||c_{i,k}|| \quad i = 1, ...m;\ k = 1, ...n;$

$$c_{i,k} = \sum_{j=1}^{p} a_{i,j} * b_{j,k};$$

Көбейту ылғы тік және ауыстыру матрицасы үшін анықталған. Егер А – тікбұрышты матрицасының өлшемі m*n, онда ылғи В квадраттық матрица өлшемі m*m:

$$B = A * A^T = B^T = (A * A^T)^T = (A^T)^T * A^T = A * A^T$$

Мұндай жұмыстың нәтижесі симметриялық матрица. Квадраттық матрица симметрицлық деп аталады, егер $a_{i,j} = a_{j,i}$ барлық і және j, немесе егер $A = A^T$. Ауыстыру операциясы, қосу және көбейтудің келесідей қасиеттері бар:

$$(A^T)^T = A; \quad (A + B)^T = A^T + B^T; \quad (A * B)^T = B^T * A^T$$

5.2.4 Квадраттық матрицалар

Квадраттық матрица диагональндық деп аталады, егер барлық элементтер, диагональдық элементтерден басқа нөлге тең болса, яғни $a_{i,j} = 0$ егер $i/ = j$.

Квадраттық матрица бірлік деп аталады, егер барлық элементтер, диагональдықтан басқа, нөлге тең болса, ал диагональдық элементтер бірге тең. Яғни $a_{i,j} = 0$ егер $i/ = j$ және $a_{i,j} = 1$ егер $i = j$. Бірлік матрица Е әрпімен белгіленеді, және ол матрицаларды көбейткен кезде бірлік болып табылады:

$$A * E = E * A = A$$

Оның элементтерінің функциясын белгіленген квадраттық матрица үшін , айқындаушы болып табылады. Матрицаны анықтау сандар жиынтығы

$$D(A) = \begin{vmatrix} a_{1,1}, a_{1,2} \dots a_{1,n} \\ a_{2,1}, a_{2,2} \dots a_{2,n} \\ \dots \\ a_{n,1}, a_{n,2} \dots a_{n,n} \end{vmatrix}$$

айналасында бір желілері арқылы білдіреді :

Анықтауыш беретін функция маңызды қасиеттер атқарады.

Диагональдық матрица анықтауышы диагональдық элементтер мәніне тең. Бұдан, Е матрицасының анықтауышы 1 тең.

Матрицаның айқындаушы қарапайым өзгерістерді жүзеге асыру бойынша өзгерген жоқ. Қарапайым операциялар (трансформациялау) астында басқа жолдар сызықтық комбинациясы матрицаның кез келген жолдың қосымша жатады. Атап айтқанда, егер j нөмерлі жолға $k(k/ = j)$ нөмерлі жол қосса, кейбір санға көбейтілген, онда матрица анықтауышы өзгермейді.

Егер матрицаның барлық элементін кейбір q санына көбейтсе, онда матрица анықтауышы q ретке өзгереді (q көбейтіледі).

Егер j және k жолдардың орынын ауыстырса, онда анықтауыш модулы өзгермейді, бірақ белгі өзгереді, егер өзгешелік $|k-j|$ тақ сан болса.
Матрица анықтауышы анықтауыш мәніне тең:

$$D(A*B) = D(A)*D(B)$$

Ресми анықтама бермей, оның қасиеттері негізінде матрицаның анықтауышын есептеу алгоритмін қарастырайық.

Егер А квадраттық матрицасының анықтауышы нөлге тең болмаса, онда кері матрица бар, белгіленуі A^{-1}.Тікелей және кері матрицасы байланысты :

$$A*A^{-1} = A^{-1}*A = E$$

Төмендегідей матрицаны көшіру , көбейту және инверсия байланысты :

$$(A^T)^{-1} = (A^{-1})^T; \quad (A*B)^{-1} = B^{-1}*A^{-1}$$

Барлық анықтауышы бар бірөлшемді квадраттық матрицалар көбейту бойынша бір топ құрайды. Топта бірлік элемент бар, әрбір элемент үшін оған кері элемент болады.

Анықтауыштар n сызықтық теңдеулердегі n белгісі бар түбірді табу үшін кеңінен қолданылады.

5.3 Зертханалық жұмысты орындауға арналған мысал

0-ден 5-ке дейінгі диапазонда кездейсоқ толық санды А – 5*3 және В – 3*4 матрицасын тұрғызуды қарастырайық.

Бұл матрицаларды басып шығару керек, одан кейін А*В жұмысының нәтижесін ұсынатын С матрицасын тауып оны басу керек.

Әдістемелік нұсқаулықтың теориялық бөлімнің 5 модулында матрицаларды көбейтудің алгоритмі қарастырылған. С матрицасының мәнін тексеру үшін А және В матрицасының мәндер диапазоны аз болып таңдалды.
Бағдарлама коды:

```
usingSystem;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
classProgram
 {
publicstaticvoid sozd(int[,] ma, int[,] mb)
  {
Random rnd = newRandom();
Console.WriteLine("Матрицысозданы!!");
Console.WriteLine("МатрицаА 5*3:");
for (int i = 0; i < 5; i++)
   {
for (int j = 0; j < 3; j++)
    {
     ma[i, j] = rnd.Next() % 6;
Console.Write(ma[i, j] + "\t");
```

```
    }
Console.WriteLine();
    }
Console.WriteLine();
Console.WriteLine("МатрицаB 3*4:");
for (int i = 0; i < 3; i++)
  {
for (int j = 0; j < 4; j++)
    {
     mb[i, j] = rnd.Next() % 6;
Console.Write(mb[i, j] + "\t");
}
Console.WriteLine();
    }
Console.WriteLine();
  }

publicstaticvoid umnmatr(int[,] mc, int[,] ma, int[,] mb)
  {
int S;
for (int i = 0; i < 5; i++)
for (int j = 0; j < 4; j++)
     {
      S = 0;
for (int k = 0; k < 3; k++)
       S = S + ma[i, k] * mb[k, j];
      mc[i,j] = S;
}
Console.WriteLine("Матрица C 5*4:");
for (int i = 0; i < 5; i++)
   {
for (int j = 0; j < 4; j++)
    {
Console.Write(mc[i, j] + "\t");
    }
Console.WriteLine();
    }
Console.WriteLine();
  }

staticvoid Main()
  {
int[,] a = newint[5, 3];
int[,] b = newint[3, 4];
int[,] c = newint[5, 4];
int k = 0;
string buf;
while (k < 3)
    {
    Console.WriteLine("1 - Создать и напечатать матрицы A 5*3 и
B 3*4");
Console.WriteLine("2 - Найти и напечатать матрицу C = A * B");
```

```
Console.WriteLine("3 - Выход из программы");
Console.WriteLine("Введите пункт меню программы");
buf = Console.ReadLine();
     k = Convert.ToInt32(buf);
switch (k)
     {
case 1: sozd(a,b);  break;
case 2: umnmatr(c,a,b); break;
default: break;
     }
    }
   }
}
}
```

Бағдарлама жұмысы:
1 - A 5*3 және B 3*4 матрицасын құрып және басып шығару
2 - C = A * B матрицасын тауып оны басып шығару
3 – бағдарламадан шығу
Бағдарламаның мәзірін енгізіңіз
1
Матрицы құрылды!!
Матрица A 5*3:
2 1 3
5 5 3
2 0 2
4 4 0
4 2 3

Матрица B 3*4:
0 5 0 3
0 3 5 3
3 0 5 2

1 - A 5*3 және B 3*4 матрицасын құрып және басып шығару
2 - C = A * B матрицасын тауып оны басып шығару
3 - бағдарламадан шығу
Бағдарламаның мәзірін енгізіңіз
2
Матрица C 5*4:
9 13 20 15
9 40 40 36
6 10 10 10
0 32 20 24
9 26 25 24

1 - A 5*3 және B 3*4 матрицасын құрып және басып шығару

2 - C = A * B матрицасын тауып оны басып шығару
3 - бағдарламадан шығу
Бағдарламаның мәзірін енгізіңіз

5.4 Зертханалық жұмысқа арналған үй тапсырмасы

Минус 20-дан 60-қа дейінгі аралықтағы кездейсоқ бүтін сандардан тұратын 5×5 матрицасын құру керек. Матрицаны басып шығару керек. Максималды элементті тауып, ол орналасқан жол мен бағанды жою керек. Қалған мәндерді жаңа 4×4 матрицасына көшіріңіз. Жаңа матрицаны басып шығарыңыз. Бағдарламада мәзір қолданыңыз.

5.5 СРС-қа арналған жеке тапсырма

5.5.1 Минус 50-ден 50-ге дейінгі аралықтағы кездейсоқ бүтін сандардан тұратын 5×5 матрицасын құру керек. Матрицаны басып шығару керек. Минималды элементті табу керек, егер ол бірінші жолда орналаспаса, онда ол ораналасқан жолды бірінші жолмен орын ауыстыру керек. Жаңа матрицаны басып шығару керек.
5.5.2 Мәтін дилог режимінде енгізіледі және сөздер «пробел» символымен ажыратылатын қарапайым сөйлемді құрайды. Жолдың ұзындығы 60 тең етіп және жол бойында «пробел» символдары біртекті болатындай етіп қосымша «пробел» символдарды қосу керек.
5.5.3 0-ден 100-ге дейінгі аралықтағы кездейсоқ бүтін сандардан тұратын A6×6 матрицасын құру керек. Матрицаны басып шығару керек.Әр жолда орналасқан сандарды кему ретімен орналастыру керек. Жаңа матрицаны басып шығару керек.
5.5.4 Жолда сөздері «пробел» символымен ажыратылатын қарапайым сөйлем бар. Максималды және минималды ұзындықты сөздерді басып шығару керек.
5.5.5 Минус 20-ден 70-ге дейінгі аралықтағы кездейсоқ бүтін сандардан тұратын A5×5 матрицасын құру керек. Матрицаны басып шығару керек. Максималды элементті тауып, ол орналасқан бағанды жою керек. Қалған мәндерді B5×4 матрицасына көшіру керек. Жаңа матрицаны басып шығару керек.
5.5.6 Диалог режимінде енгізілген жолда екі жақшадан аспайтын C# тілінің шарты if операторы бар. Диалог режимінде енгізілген шарт дұрыс жазылғанын тексеру керек.
5.5.7 Минус 30-ден 50-ге дейінгі аралықтағы кездейсоқ бүтін сандардан тұратын A6×6 матрицасын құру керек. Матрицаны басып шығару керек. Матрицаның басынқы және бағыңынқы диагональда орналасқан сандарды өсу ретімен орналастыру керек. Жаңа матрицаны басып шығару керек.
5.5.8 Диалог режимінде енгізілген жол қарапайым сөйлемді құрайды және сөздер «пробел» символымен ажыратылады. Максималды және минималды

ұзындықты сөздерді тауып, олардың орындарын ауыстырыңыз, жаңа жолды экран мониторына шығарыңыз.

5.5.9 Минус 50-ден 50-ге дейінгі аралықтағы кездейсоқ бүтін сандардан тұратын A6×6 матрицасын құру керек. Матрицаны басып шығару керек. Оң және теріс сандырдың санын табыңыз. Егер модуль бойынша теріс сандардың соммасы оң сандардыңсоммасынан көп болса, онда оң сандардың соммасы артық болмағанша теріс сандардың минус таңбасын плюс таңбасына ауыстыру керек. Жаңа матрицаны басып шығару керек.

5.5.10 Диалог режимінде енгізілген жол қарапайым сөйлемді құрайды және сөздер «пробел» символымен ажыратылады. Жолдың сөздерін әріптердің өсу ретімен орналастырып, жаңа жолды қалыптастырыңыз.

5.5.11 0-ден 9-ға дейінгі аралықтағы кездейсоқ бүтін сандардан тұратын A10×10 матрицасын құру керек. Матрицаны басып шығару керек. Матрица қанша 0, 1, 2, …, 9 сандардың бар екенің есептеңіз.

5.5.12 Диалог режимінде енгізілген жол қарапайым сөйлемді құрайды және сөздер «пробел» символымен ажыратылады. A K×2 матрицаны құрыңыз, онда барлық K сөздерінің бастапқы және соңғы позициялардың нөмердерін енгізіңіз.

5.5.13 0-ден 20-ға дейінгі аралықтағы кездейсоқ бүтін сандардан тұратын A6×6 матрицасын құру керек. Матрицаны басып шығару керек. Матрицаның бағандарын элементтердің соммаларының өсу ретімен орналастыру керек. Жаңа матрицаны басып шығару керек.

5.5.14 0-ден 20-ға дейінгі аралықтағы кездейсоқ бүтін сандардан тұратын A6×6 матрицасын құру керек. Матрицаны басып шығару керек. Матрицада қайталанатын сандарынан тұратын векторды құрыңыз және басып шығарыңыз.

5.5.15 Диалог режимінде енгізілген жол қарапайым сөйлемді құрайды және сөздер «пробел» символымен ажыратылады. Мәтінде қайталанатын сөздерін анықтаңыз. Қайталанатын сөздерді және сөйлемде қайталау жиілігі бойынша экран мониторына шығарыңыз.

5.5.16 Минус 20-дан 60-қа дейінгі аралықтағы кездейсоқ бүтін сандардан тұратын A7×7 матрицасын құру керек. Матрицаны басып шығару керек. Матрицаның басты және бағаңынқы диагональдарынан «төмен» жатқан элементтердің соммасын табу керек.

5.5.17 0-ден 20-ға дейінгі аралықтағы кездейсоқ бүтін сандардан тұратын A5×5 матрицасын құру керек. Матрицаны басып шығару керек. Матрицада қайталанатын сандарды және олардың қайталану жиілігін табыңыз және басып шығарыңыз.

5.5.18 Диалог режимінде енгізілген жол қарапайым сөйлемді құрайды және сөздер «пробел» символымен ажыратылады. Сөйлемде ең жиі кездесетін символды тауып, экран мониторына шығарыңыз.

5.5.19 Минус 50-ден 50-ге дейінгі аралықтағы кездейсоқ бүтін сандардан тұратын A8×8 матрицасын құру керек. Матрицаны басып шығару керек. Басты диагональға паралель барлық диагональдарды басып шығарыңыз.

5.5.20 0-ден 20-ға дейінгі аралықтағы кездейсоқ бүтін сандардан тұратын A10×10 матрицасын құру керек. Матрицаны басып шығару керек. Ең көп бірдей сандар бар жолды тауып, оның нөмерін экран мониторына шығарыңыз.

5.6 СРС-да есеп-хатты қорғауға арналған бақылау сұрақтары

5.6.1 Көпөлшемді массив түсінігі. Мысалдар.
5.6.2 Көпөлшемді массив қалай жарияланады? Мысал.
5.6.3 0-ден 10-ға дейінгі диапазонда A5×5 матрицасын қалай тұрғызу керек мысал келтір.
5.6.4 Экран мониторына A5×5 матрицасын қалай енгізуге болады мысал келтір.
5.6.5 A5×5 матрицасындағы үлкен санды іздеу алгоритмін түсіндіріңіз.
5.6.6 Массивтердің массив түсінігі. Сондай массивтерді жариялау. Мысал.
5.6.7 Екі матрицаны қосу алгоритмні түсіндіріңіз. Мысал.
5.6.8 Екі матрицаны көбейту алгоритмні түсіндіріңіз. Мысал.
5.6.9 C# тілінің жол айнымалы түсінігі. Мысал.
5.6.10 C# тілінде escape-қолданбасы не үшін керек? Мысалдар.
5.6.11 C# тілінде жолдық айнымалылар қалай салыстырылады? Мысал.
5.6.12 Қалай кейбір белгілі фрагментті мәтіннен жоюға болады? Мысал.
5.6.13 Кейбір белгілі фрагменті мәтінге қалай қоюға болады? Мысал.
5.6.14 Қалай мәтіндегі таңбалар санын анықтауға болады? Мысал.
5.6.15 Split және Join әдістерін тағайындау. Мысалдар.

6 ГРАФТЫҢ ӨТУ АЛГОРИТМІ

6.1 Алтыншы тақырыптың мақсаты

Граф теориясының негізгі түсініктерін оқу және граф алгоритмінің орындалуын қарау, «транспорттық» есепті шешуге арналған консоль қосымшасында орындауға практикалық дағдылану.

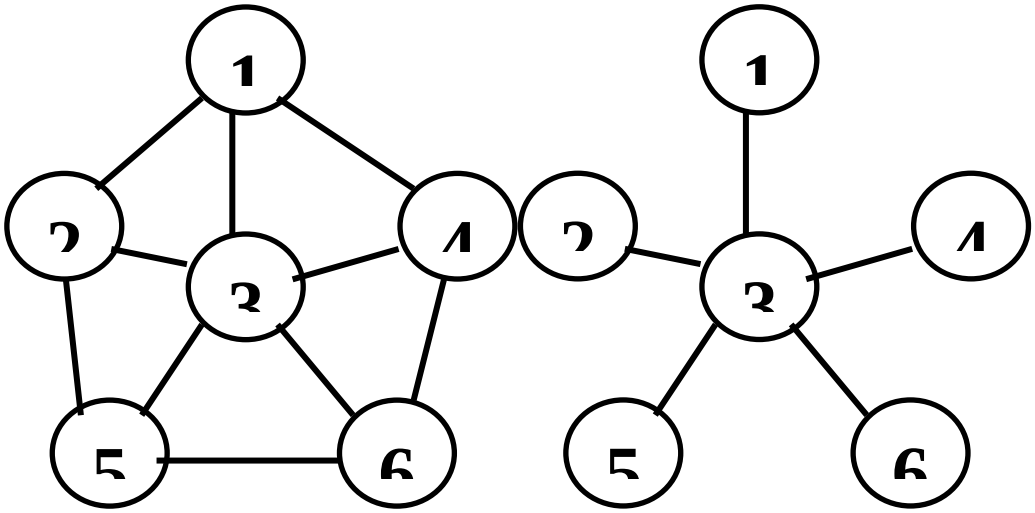
6.2 теориялық мәлімет

6.2.1 Қатаңдату ағаштары. Негізгі түсініктер

Графтар теориясында ағаш түсінігінің қарапайым екі анықтамасы бар. Теоремаларды қорытпай, дұрыстығын дәлелдеу үшін анықтамасын келтірейік. Бірінші анықтама, байланысқан граф ағаш болып табылады, онда доға саны төбе санынан бірге кіші.

Екінші анықтама анықтырақ – ағаш циклдар жоқ қосылған граф болып табылады. Ағаш , кейбір қабырғаны алып тастау арқылы граф алу, осы ағаштың графы немесе жақтауы деп аталады.

Мысал, 6.1 суретте «жұлдыз» граф көрсетілген бірнеше созылмалы ағашпен берілген. 6.1 суретте созылмалы ағаштың тағы бір түрі көрсетілген.



Сурет 6.1 – Байланысқан граф және оның созылмалы ағашы.

Граф теориясында қолданылмаған қабырға (созылған ағаш жойылған граф қабырғасы арқасында алынды) хорда деп аталады.

Егер созылған ағашқа еркін хорданы қосса, алынған граф бір циклды болады.

Көп циклді граф арқасында алгоритм құрылады, қолданбалы сипаттамасы бар, мысалы, Кирхгоф заңына сәйкес электронды сымдарға анализ жүргізу.

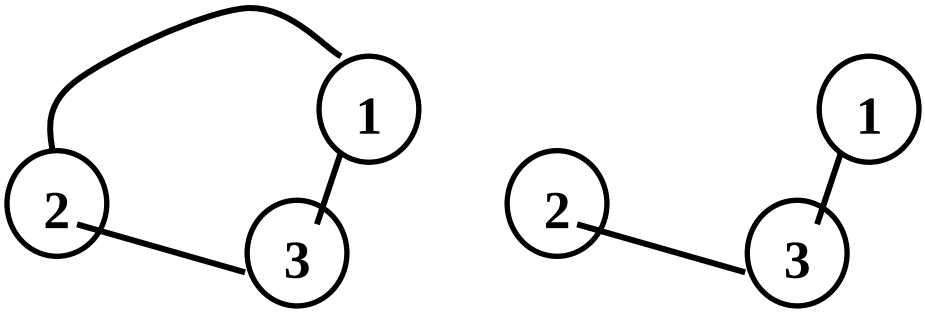
6.2.2 Созылған ағашты тұрғызу алгоритмі

Созылған ағаштың табудың әртүрлі алгоритмі бар.

Созылған ағашты табудың алгоритмін қарастырайық, онда бәріне белгілі Дейкстр алгоритмі қолданылады – графтың берілген төбесінен қалған төбелеріне дейінгі ең аз маршрут.

Бұл мүмкін, өйткені соны дәлелдейтін теория бар, берілген граф төбесінен қалған төбелеріне дейінгі ең аз қашықтық арқылы созылған ағаш тұрғызуға болады.

Граф төбелері арасындағы ең аз қашықтықты табудың алгоритмін қарастырайық. Мысалы, іргелес үш төбе бар 1,2 және 3. Төбелер арасындағы қашықтық 1 - 2 тең 5, төбелер 2 – 3 тең 3, төбелер 3 – 1 тең 1. Маршрутты таңдау 1 – 2 граф төбелері үшін 1 – 3 – 2 маршрут қолданыады . Қабырға 1 – 2 графтан өшіріледі.

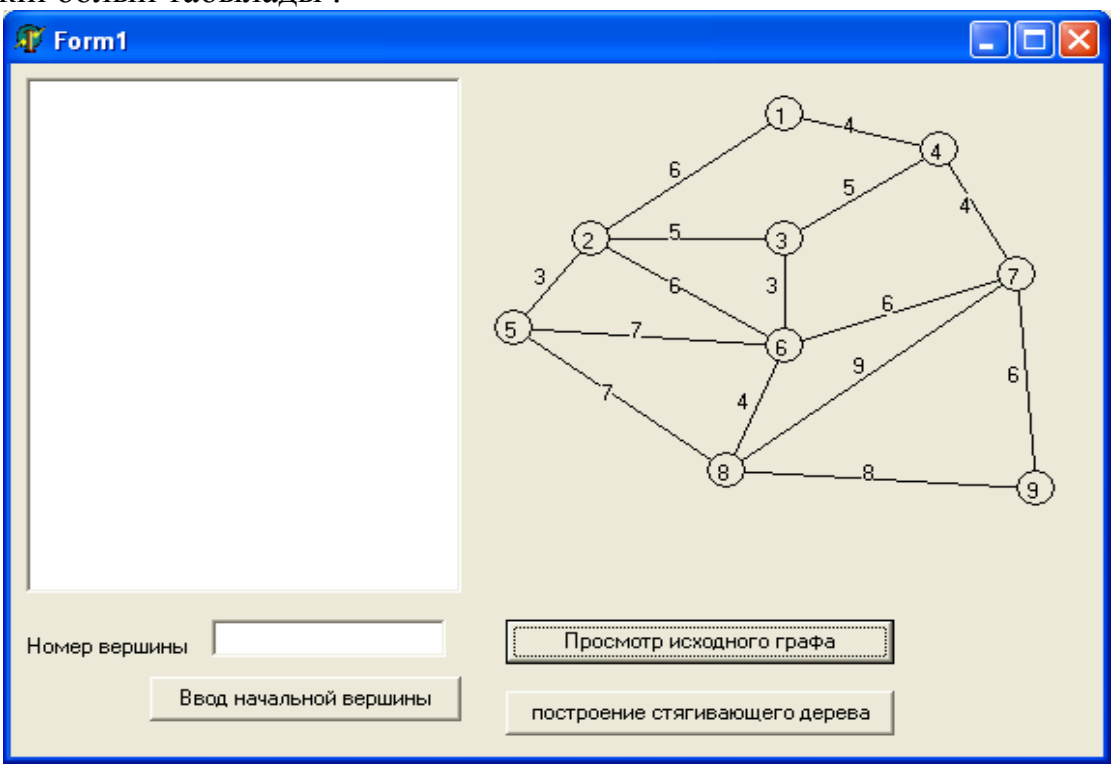


Сурет 6.2 – ең аз маршрутты табу алгоритмі

Көрсетілген алгоритмді қолдану созылған ағашты тұрғызуға көмек береді. «көпмөлшер» граф төбесі алгоритімде төбелердің өсу ретімен орындалады (for циклі қолданылған) , онда екі маршрут «тең» болғанда созылған ағаш болып бірінші нұсқа қалады.

Есепті бағдарламалау үшін бастапқы графты байланысқан граф ретінде созылған ағашты тұрғызу Delphi бағдарламалау ортасында бар (6.3 суретте граф көрсетілген).

Созылған ағашты құру алгоритмі Дейкстрдің ең аз қашықтықты табу алгоритмін қолданады. Осы алгоритмнің бұрын талқыланған бағдарламаны жүзеге асырудан айырмашылығы ол бастапқы төбелердің мәнін орнату үшін мүмкін болып табылады .



Сурет 6.3 – Бастапқы байланысқан граф

6.2.3 Дейкстр алгоритмі

Дейкстр алгоритмін үш массивті қолдануды талап етеді, олардың өлшемі граф төбелеріне сәйкес келеді.

Бірінші массив, «тұрақты» төбе массиві, берілген граф төбесінен қалған төбелерге дейінгі ең аз қашықтықты сақтайды. Алдымен бұл массивке бастапқы төбе нөмері жазылады (төбелер, басқа граф төбелеріне дейінгі ең аз қашықтықты табу үшін). Массив атауы Дейкстр алгоритіміндегі таңдап алынған төбелер атауымен сәйкес келеді, ең алдымен барлық граф төбелері «уақытша» болып жарияланады, ал таңдап алынған төбелер «тұрақты» болып табылады.

Екінші массив барлық басқа төбелерге ең аз арақашықтықты сақтау үшін пайдаланылады. Алдымен төбенің таңдап алған нөмеріне сәйкес іргелес матрица жолы жазылады.

Логикалық типті үшінші массивте таңдап алынған (тұрақты) граф төбелері көрсетіледі. Бастапқыда «тұрақты» граф төбесі алғашқы төбе болып табылады.

Әрі қарай «уақытша» төбе таңдап алынады, онда бастапқы (тұрақты) төбеге дейінгі қашықтық ең аз. Бұл төбе k төбесі болып табылады. Міндетті түрде «тұрақты» төбеден «уақытша» төбеге дейін барлық қашықтық тексеріледі, сол сияқты k төбесінен де тексеріледі. Ең аз қашықтық екінші массивте жазылады, ал бірінші массивке k жазылады, егер ең аз қашықтық k төбесі арқылы өтсе.

Одан кейін k төбесі «тұрақты» болып жарияланады (бұл үшінші массивте жазылады) және жаңа «тұрақты» төбеге сәйкес графтың келесі төбесін іздеу алгоритмі жалғасады.

6.3 Зертханалық жұмысты орындау мысалы

6.1 есеп. 6.3 суретте көрсетілген байланысқан графқа сәйкес созылған ағаш тұрғызу. Графтың алғашқы төбе нөмерін диалог режімінде беру. Созылған ағаш үшін бастапқы төбеден барлық қалған төбелеріне дейін маршруттың тізімін көрсету.

Бағдарлама коды:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication1
{
classProgram
 {
publicstaticint[,] a = newint[9, 9]
   {{   0,   6,1000,   4,1000,1000,1000,1000,1000},
    {   6,   0,   5,1000,   3,   6,1000,1000,1000},
    {1000,   5,   0,   5,1000,   3,1000,1000,1000},
    {   4,1000,   5,   0,1000,1000,   4,1000,1000},
    {1000,   3,1000,1000,   0,   7,1000,   7,1000},
```

```
    {1000,   6,   3,1000,   7,   0,   6,   4,1000},
    {1000,1000,1000,   4,1000,   6,   0,   9,   6},
    {1000,1000,1000,1000,   7,   4,   9,   0,   8},
    {1000,1000,1000,1000,1000,1000,   6,   8,   0}};

publicstaticint[] d = newint[10];
publicstaticint[] post = newint[10];
publicstaticbool[] t = newbool[10];
publicstaticint i, j, p, k, minras, begver;

publicstaticvoid poisk()
{
string buf;
//Ввод значения переменной a в режиме диалога
Console.WriteLine("Введите номер вершины графа целое значение 0
- 8");
buf = Console.ReadLine();
   begver = Convert.ToInt32(buf);
//начальные установк
for (i = 0; i < 9; i++)
{
     post[i] = begver;
     t[i] = true;
d[i] = a[begver,i];
    }
   t[begver] = false;
   post[begver] = 0;
   minras = 0;
for (i = 0; i < 8; i++)
    {
// поисквершины k
     minras = 1000;
for (j = 0; j < 9; j++)
if ((t[j] == true) && (minras > d[j]))
{
        minras = d[j]; k = j;
}
// поиск маршрутов и минимальных расстояний через вершину k
t[k] = false;
for (j = 0; j < 9; j++)
if ((t[j] == true) && (d[j] > d[k] + a[k,j]))
      {
       d[j] = d[k] + a[k,j];
       post[j] = k;
      }
    }
  }


publicstaticvoid print()
{
Console.WriteLine("Печатаемматрицусмежности : ");
```

```
for (i = 0; i < 9; i++)
    {
for (j = 0; j < 9; j++)
Console.Write("\t" + a[i, j]);
Console.WriteLine();
    }
Console.WriteLine();
// печатьномероввершинграфа
for (i = 0; i < 9; i++) Console.Write("\t" + i);
Console.WriteLine();
// печать массива минимальных расстояний
for (i = 0; i < 9; i++) Console.Write("\t" + d[i]);
Console.WriteLine();
// печатьмассивамаршрутов
for (i = 0; i < 9; i++) Console.Write("\t" + post[i]);
Console.WriteLine();
Console.WriteLine();
int dlin;
// печать маршрутов стягивающего дерева
for (i = 0; i < 9; i++)
    {
     dlin = d[i];
Console.Write("Путь № {0}-{1} длина пути = {2}\t\t{3}", i,
begver, dlin,i);
p = i;
if (i != begver)
      {
do
        {
         p = post[p];
if (p!=0) Console.Write("\t" + p);
}
while (p != 0);
Console.WriteLine();
      }
elseConsole.WriteLine();
    }
  }

staticvoid Main()
  {
   poisk();
   print();
Console.ReadLine();
  }
 }
}
```

Бағдарлама жұмысы:

0 - 8 бүтін мәннің граф төбелері санын енгізіңіз

4

Іргелес матрицаларды басып шығарамыз :

0 6 1000 4 1000 1000 1000 1000 1000

6	0	5	1000	3	6	1000	1000	1000
1000	5	0	5	1000	3	1000	1000	1000
4	1000	5	0	1000	1000	4	1000	1000
1000	3	1000	1000	0	7	1000	7	1000
1000	6	3	1000	7	0	6	4	1000
1000	1000	1000	4	1000	6	0	9	6
1000	1000	1000	1000	7	4	9	0	8
1000	1000	1000	1000	1000	1000	6	8	0

0	1	2	3	4	5	6	7	8
9	3	8	13	0	7	13	7	15
1	4	1	2	0	4	5	4	7

Жол № 0-4 жол ұзындығы = 9 0 1 4
Жол № 1-4 жол ұзындығы = 3 1 4
Жол № 2- жол ұзындығы = 8 2 1 4
Жол № 3-4 жол ұзындығы = 13 3 2 1 4
Жол № 4-4 жол ұзындығы = 0 4
Жол № 5-4 жол ұзындығы = 7 5 4
Жол № 6-4 жол ұзындығы = 13 6 5 4
Жол № 7-4 жол ұзындығы = 7 7 4
Жол № 8-4 жол ұзындығы = 15 8 7 4

6.4 Зертханалық жұмысқа үй тапсырмасы

Графты «тереңдігі» бойынша жүріп өту алгоритмін жүзеге асыратын алгоритмді бағдарлама құрыңыз. Өз графын қолданыңыз. Төбелер саны 10 кем болмау керек.

6.5 СРС арналған жеке тапсырма

6.5.1 Граф жасау - әуе тасымалы түйіндерді одан әрі түртіндінің өрісті қамтитын - облыс орталықтары атауы. Елдің барлық басқа орталықтарына берілген облыс орталығынан бастап, ең төменгі туристік маршрут іздеу қамтамасыз ету.
6.5.2 Егер граф жасау 10 шыңдары , белсенді элементтер кейбір электрондық тізбек сәйкес түйіндерінің кем емес . диалог режимінде орнатылған тізбектің тораптары арасындағы әрбір схеманың қарсылығын табыңыз.
6.5.3 граф түрі құрылымын енгізу схемасы қалалық автобус маршруттары . Түйіндері автобус маршруттарының құрылымына сәйкес келеді және одан әрі аялдама атын қамтиды . аялдама атына байланысты бағдарларды қамтамасыз ету .

6.5.4 Студенттер тобының атауларының графын жасау (атауы бірдей қолдануға рұқсат) . диалог режимінде көрсетілген аты бойынша студенттерге арналған іздеуді қамтамасыз ету.
6.5.5 Отбасылық ағашы 12 төбемен көрсетілген болып табылады. Әрбір төбесінде құрастыру одан әрі түрлі мүшесі кіреді. «тереңдігі» бойынша жүріп өтіу графы көмегімен барлық әйел затын табуды ұйымдастыру.
6.5.6 Қалалық төрт трамвай маршруты граф типінің құрылымы боцынша көрсетілген. Түйіндері трамвай маршруттарынан құрылымына сәйкес келеді және одан әрі аялдамасына атын қамтиды. Диалог режимінде енгізілген екі аялдамалар атауларының үшін , (аялдама саяхат қашықтыққа ең төменгі сомасы бойынша) екінші аялдамасына бірінші аз көші-қон маршрут табу үшін
6.5.7 Отбасылық ағашы 14 төбемен көрсетілген болып табылады. Әрбір төбесінде құрастыру одан әрі түрлі мүшесі кіреді. Ең жиі ерлер мен әйелдер атауларының іздеуді ұйымдастыру.
6.5.8 Қалалық төрт трамвай маршруты граф типінің құрылымы боцынша көрсетілген. Түйіндері трамвай маршруттарынан құрылымына сәйкес келеді және одан әрі аялдамасына атын қамтиды. Диалог режимінде енгізілген екі аялдамалар атауларының үшін , (аялдама саяхат қашықтыққа ең төменгі сомасы бойынша) екінші аялдамасына бірінші аз көші-қон маршрут табу үшін.
6.5.9 5 Отбасылық ағашы 15 төбемен көрсетілген болып табылады. Әрбір шыңында құрастыру одан әрі тектес және осы мамандық бойынша оның жұмысының уақыт негізгі қызметін қамтиды . Максималды жүгіру уақыт іздеу мамандық ұйымдастыру .
6.5.10 Қалалық төрт трамвай маршруты граф типінің құрылымы боцынша көрсетілген. Түйіндері трамвай маршруттарынан құрылымына сәйкес келеді және одан әрі аялдамасына атын қамтиды. Бұл тоқтау арқылы бағыттар санының кему тәртібімен қала аялдамалары атауын теріңіз.
6.5.11 Оның тораптары қалта каталогына сәйкес және қосымша қалта атын қамтиды граф ұсынылған компьютерлік диск - түрі құрамынан иерархиялық каталогтар құрылымы. Дискідегі бірдей қалта бар-жоғын анықтау үшін . Оларды басып шығару және оларға түбірлік каталогына жолын көрсетеді.
6.5.12 Егер граф жасау оның тораптары одан әрі түртіндінің өрісті қамтиды 10 төбедег кем емес. (Қосымша сипаты өрісі « дыбыстардың » символы бар) шыңдары ғана «мөлдір» « терең » бойынша графтар айналып алгоритмі әзірлеу .
6.5.13. Оның тораптары қалта каталогына сәйкес және қосымша қалта атын қамтиды граф ұсынылған компьютерлік диск - түрі құрамынан иерархиялық каталогтар құрылымы. Диск ойындары деп аталатын қалтаны тауып қанша рет анықтау үшін .
6.5.14 Егер граф жасау оның тораптары одан әрі түртіндінің өрісті қамтиды 10 төбеден кем емес . (Қосымша сипаты өрісі « дайын » символы бар)

шыңдары «сәйкес » тек « терең » бойынша графтар айналып алгоритмі әзірлеу .
6.5.15 Диаграмма түрін (10 шыңы кем емес) құрылымын ұсынуға автобус маршруттары аудандық ауданы схемасы . Түйіндері ауылы ауданы атауы құрылымына сәйкес келеді. Аялдама атына сәйкес маршруттар номерін көруді қарастыру. Ауылының атауы үшін , (аялдама саяхат қашықтыққа ең төменгі сомасы бойынша) осы ауылға аудан орталығынан аз көші-қон маршрут табу үшін, диалог енген .
6.5.16 Желінің өткізу қабілетін (секундына берілетін судың көлемі) бөлігі - қала , аудан Сумен жабдықтау желісі бағытталған кем дегенде 15 төбе графы, және доға ұсынылған. Су максималды сомасы түйін X. A , B және C тораптар жеткізілетін болуы мүмкін анықтаңыз. Барлық тораптар мәні диалог режимінде орнатылады. (Бұл қорғасын - доғаның қамтамасыз ете алады , егер) қабылдайтын тораптары арасындағы жем су біркелкі бөлу пайдаланыңыз.
6.5.17 Егер граф жасау 10 төбеден , белсенді элементтер кейбір электрондық тізбек сәйкес түйіндерінің кем емес. Диалог режимінде (параллель және сериялық қосылу схемасын ескеріңіз) белгіленген түйіндер схемасын арасындағы тізбектің кедергісі табыңыз.
6.5.18 Лабиринт 16 төбеден кем емес графпен көрсетілген, төбелері қмылыстарға сәйкес келеді. Лабиринттің кіру және шығу түйіндері белгілі. лабиринт жүруінің ең аз қашықтығын табу.
6.5.19 Қалалық төрт трамвай маршруты граф типінің құрылымы боцынша көрсетілген. Түйіндері трамвай маршруттарынан құрылымына сәйкес келеді және одан әрі аялдамасына атын қамтиды. Екі аялдамалар атауларының үшін , екінші аялдамасы (жасалған трансплантация саны , бірақ аялдамалары саны) бірінші аз көші-қон маршрут табу үшін , диалог режимінде енгізілген. Жинақтарын немесе өзгерту қажет аялдамалары және трамвай маршруттары бөлмелер атауларының жолын басып шығару, отыруға немесе өзгерту үшін сол қажеттілігі .
6.5.20 Қалалық төрт трамвай маршруты граф типінің құрылымы боцынша көрсетілген. Түйіндері трамвай маршруттарынан құрылымына сәйкес келеді және одан әрі аялдамасына атын қамтиды. Екі аялдамалар атауларының үшін , екінші аялдамасы (жасалған трансплантация саны , бірақ аялдамалары саны) бірінші аз көші-қон маршрут табу үшін , диалог режимінде енгізілген. Жинақтарын немесе өзгерту қажет аялдамалары және трамвай маршруттары бөлмелер атауларының жолын басып шығару, отыруға немесе өзгерту үшін сол қажеттілігі .

6.6 Контрольные вопросы для защиты отчета на СРСП

6.6.1 Граф түсінігі. Мысалдар.
6.6.2 Сырқаттанушылық түсінігі. Мысал.
6.6.3 Жалғастыру түсінігі. Мысал.

6.6.4 Граф жолы түсінігі. Графтың қандай жолы қарапайым деп аталады? Мысал.
6.6.5 Қандай граф байланысқан деп аталады? Мысал.
6.6.6 ЭВМ жадысында граф туралы мәліметті қалай сақтауға болады?
6.6.7 ЭВМ жадысында графты іргелес төбе тізімі ретінде қалай түсіндіруге болады?
6.6.8 Граф төбелері арасындағы ең аз қашықтықты іздеу алгоритмі.
6.6.9 Граф төбелері арасындағы ең аз қашықтықты іздеу алгоритмі.
6.6.10 «тереңдігі» бойынша графтың өту алгоритмі.
6.6.11 «ені» бойынша графтың өту алгоритмі.
6.6.12 «созылған» ағаш алгоритмін тұрғызу.
6.6.13 Екі төбе арасындағы барлық маршрутты табу алгоритмі.А
6.6.14 Неге «қайтару» жоғарғы жаңа меншік төбелері барлық циклдердің іздеу алгоритмі ?
6.6.15 Іздеу алгоритмі диаграммада екі берілген төбелер арасындағы барлық бағыттарды пайдаланылану құрылымын түсіндіріңіз .

7 Ұсынылған әдебиеттер тізімі

7.1. Негізгі әдебиет

7.1.1 Презентации лекций по дисциплине «Алгоритмизация и основы программирования» для студентов специальности 5В070400 – смотри портал кафедры ИС http : \\ www.do.ektu.kz
7.1.2 В.В. Фаронов Создание приложений с помощью C# Руководство программиста. - М.: “Эксмо”, 2008г.
7.1.3Т.А. Павловская C#, Программирование на языке высокого уровня. Учебник для вузов, СПб.: Питер, 2009г.
7.1.4Д. Кнут. Искусство программирования для ЭВМ. Т.3./ Сортировка и поиск / - М.:Мир,1976.

7.2 Қосымша әдебиет

7.2.1 Э. Йодан Структурное программирование и конструирование программ. М.: ”Мир”, 1989г.
7.2.2 Н. ВиртАлгоритмы и структуры данных. М. Изд-во «МИР», 1989г.
7.2.3 Э.ТроелсенC# и платформа .NET Библиотека программиста, СПб,: Питер, 2007г.