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.

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

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.

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

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 :

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

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 .

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

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

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

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 операторды 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 операторды жұмыс істеу және жазу форматы.