

Оқу жұмыс бағдарламасы
бекіту парағы (SYLLABUS)



Форма
ПМУ ҰС Н 7.18.4/19

Қазақстан Республикасының білім және ғылым министрлігі

С. Торайғыров атындағы Павлодар мемлекеттік университеті

Есептеу техникасы және бағдарламау кафедрасы

ЖҰМЫС ОҚУ БАҒДАРЛАМАСЫ

«Бағдарламау технологиясы» пәні бойынша

Павлодар, 2013 ж.

БЕКІТЕМІН

ФМЖАТ факультетінің деканы

_____ Испулов Н.А.

(қолы)

(аты-жөні)

«___» _____ 20__ ж.

Құрастырған: аға оқытушы _____ Ахмерова З.Р.
(лауазымы, ғылыми атағы, қолы) (аты-жөні)

ЖҰМЫС ОҚУ БАҒДАРЛАМАСЫ (Syllabus)

Бағдарламау технологиясы ТР 1205

(жұмыс оқу жоспары бойынша пәннің коды мен толық атауы)

күндізгі оқу формасындағы мамандық(тар) студенттері үшін

(оқу формасы)

Есептеу техникасы және бағдарламалық қамтамасыз ету 5В070400, Автоматтандыру және басқару 5В070200

(мамандық(тар)дың толық атауы мен шифрі)

Бағдарлама «___» _____ 20__ ж. бекітілген жұмыс оқу бағдарламасы негізінде құрастырылған

Кафедраның отырысында қарастырылған «___» _____ 20__ ж.

Хаттама № _____.

Кафедра меңгерушісі _____ Потапенко О.Г. «___» _____ 20__ ж.
(қолы) (аты-жөні)

ФМЖАТ факультетінің оқу әдістемелік кеңесінде құпталған

(факультет атауы)

«___» _____ 20__ ж. Хаттама № _____

ОӘК төрайымы _____
(қолы)

Исқакова А.Б. «___» _____ 20__ ж.
(аты-жөні)

1. Оқу бағдарламасының паспорты

Пән атауы Есептеу техникасы және бағдарламалық қамтамасыз ету 5В070400

Кредиттер саны мен оқыту ұзақтығы

Жалпы – 3 кредит

Курс: 2

Семестр: 2

Жалпы аудиториялық сабақтар– 45 сағат

Дәрстер - 15 сағат

Тәжірибелік /семинар сабақтары – 15 сағат

Зертханалық – 15 сағат

СӨЖ – 90 сағат, сонымен бірге СӨЖМ – 22,5сағат

Жалпы бейнетті- 135 сағат

Бақылау формасы

Курстық жұмыс (Есептеу техникасы және бағдарламалық қамтамасыз ету 5В070400) – 2 семестр (қорғау)

Қорытынды бақылау формасы

Емтихан – 2 семестр

Пререквизиттер

«Бағдарламау технологиясы» пәнінің алдында «Ақпараттану», «Алгоритмдік тілде бағдарламалау» сабақтары бойынша алынған білімдерге негізделеді

Постреквизиттер

Студенттер «Бағдарламау технологиясы» пәні бойынша алған білімі мен дағдыларын мамандандырылған пәндер мен дипломдық жобалауда қолданулары мүмкін.

2. Оқытушылар жайлы мәліметтер мен жеке ақпарат

Аты-жөні Ахмерова Зарема Равильевна

«ЕТЖБ» кафедрасы, аудитория А-403

E-mail: Ahmerova_Zarema@mail.ru

3. Пәні, мақсаты және мәселесі

Пән атауы Бағдарламау технологиясы

Пәнді оқыту мақсаты

«Бағдарламалау технологиясы» пәнінің мақсаты есептердің алгоритм негізін, программалаудың автоматтық негіздердің, программалау тілінің классификациясын, мәліметтер типтерін және Турбо Паскаль тілінің операторларының классификациясын оқыту: ішкі программаларды қолданып программа, стандартты модельдер, деректердің динамикалық құрылымын, программалық қамтамасыз етуді жобалау тәсілдерін, программалау стилін, программалаудың сапа көрсеткішін, программаны сынау мен қалыптастыру тәсілдерін, объектілі-бағытталған программалаудың негіздерін қолданып программа құру болып табылады.

Пәнді оқыту мәселесі

алгоритмдеу әдістер туралы және деректерды құрылымдау туралы түсініктерді алу және әртүрлі есептердің алгоритмдерін шешу әдістеріне үйрету.

4. Білімі, икемділігі, дағдысына және құзыретіне талаптары

Пәнді оқу барысында студенттер келесілерді білуі керек:

- негізгі алгоритмдік конструкцияларды;
- деректер, тип және құрылым деректер туралы түсініктері;

- әртүрлі есептерді шешу алгоритмдердің технологиясын жасау;
- Пәнді игеруде тәлімгерлер істей білуге тиісті:
- әртүрлі есептерге сәйкес еңгізу және шығару деректердің типіп анықтау;
- әртүрлі есептердің алгоритмдерің құру және оның дұрыстығын тексеру;
- дайын алгоритмдер бойынша программаларды құру.

5 Пәннің тақырыптық жоспары

Академиялық сағаттарды сабақтардың түрі бойынша бөлу

Оқу формасы	Пән жұмыс сыйымдылығы				Семестрлер бойынша бақылау формасы				Семестр	Семестрлер бойынша студенттердің жұмыс көлемі					
	кредиттер	академиялық сағаттар								кредиттер	аудиторных занятий (ак. часов)				СӨЖ (ак. сағат)
		жалпы	ауд	СӨЖ	емт.	сын.	КЖ	КЖ			жалпы	дәр.	тәж.	зерт	
ЖОБ 2013 негізінде күндізгі	3	135	45	90	2			2	2	3	45	15	15	15	90

6. Дәріс сабақтарының мазмұны

1 Тақырып. Кіріспе

«Бағдарламалау технологиясы» пәнінің мақсаты мен даму сатылары. Есептеу техникасы және ақпаратты өңдеу және басқарудың компьютерлік жүйелері. Ақпаратты өңдеу және басқарудың компьютерлік жүйелердің әдістері және есептері. Өндірістің салалары бойынша есептеу техникасының қолдауының мысалдары. Оқу барысында компьютерді пайдалану.

2 Тақырып. Компьютердің программалық құралдары

Программалауды автоматтандырудың әдістері. Алгоритмдік тілдер. Алгоритмдік тілдің қолдануы және оған қойылатын талаптар. Процедураға бағытталған тілдер және объектілерге бағытталған программалау туралы түсініктер. Компьютерді программалық қамтамасыз етудің жалпы түсінігі. Операциялық жүйелердің құрамы. Компьютер мен пайдаланушының диалогын ұйымдастыру. Программалаудың интеграцияланған жүйелері.

3 Тақырып. Есептерді алгоритмдеудің негіздері

Алгоритмнің анықтамасы. Алгоритмдерді баяндаудың тәсілдері. Алгоритм схемаларын жасау ережелері. Алгоритмнің құрылымдарының түрлері. Сызықты және тармақталған алгоритмдердің құрылымдарының баяндалуы. Циклдік құрылымдар. Алгоритмнің циклдік құрылымы. Ішкі циклдік құрылымының алгоритмдік баяндалуы.

Техникалық есептердің қойылымының классқа бөлінуі. Есептердің типтік компоненттерін талдау, синтез, шешім қабылдау.

Ғылыми – техникалық есептердің алгоритмдерінің схемаларының мысалдары.

4 Тақырып. Негізгі процедураларға бағытталған алгоритмдік тілдерде программалау

Оқып үйренетін алгоритмдік тілдің негізгі мінездемелері. Тілдің алфавиті. Тілдің объектілерінің жазулының ережелері. Мәліметтер типтері. Тұрақтылар. Айнымалылар. Ерекше белгілер. Өрнектер. Арифметикалық және логикалық өрнектер. Мәліметтер құрылымы: Массивтер. Жиындар. Жазбалар.

Алгоритмдік тілдің операторларын классқа бөлу. Меншіктеу операторы. Басқару операторы. Мәліметтерді енгіз – шығаруды ұйымдастыру.

Программаның құрылымы. Алгоритмнің схемасын программаның схемасына көшу. Алгоритмдердің сызықтық құрылымын программалау. Тармақталған құрылымдарды программалау. Алгоритмдер циклдық құрылымыдық программалау (сандық талдау есептері, сандық массивтерді өңдеу, массивтер компоненттерін есептеу және т.б. мысалдарында) массивтерді енгізу – шығарудың программалары.

Қатарлар. Символдық мәліметтерді өңдеудің есептерін программалау. Мәліметтер құрылымы қолданатын есептерді программалаудың ерекшеліктері.

Ішкі программалар және оларды классқа бөлу. Ішкі программаларды ұйымдастыру әдістері. Ішкі программаларды шақыру. Нақты параметрлерді тасымалдау. Жадының жалпы облысын пайдалану. Стандартты ішкі функциялардың қоры және процедуралары. Оларды шақырудың тәсілдері. Программаларды ұйымдастыру. Өртүрлі құрылымдық ұйымдасқан программалардың мысалдары.

5 Тақырып. Программаларды компьютерде орындауды ұйымдастыру

Компьютерде программаларды орындауға дайындық және орындау сатылары. Программаларды орындайтын, өндейтін, трансляция сатыларын орындайтын интеграцияланған турбожүйе құралдарымен жұмыс істеу.

Байланыс редакторы мен транслятордың жұмыс режимі. Алғашқы программаның трансляторы. Транслятордың қателер туралы хабарлауы. Программалардың орындалуы. Программаларды орындау барысында жүйенің қателері жайында хабарлауы.

6 Тақырып. Программалаудың әдістемесі.

Программа құрудың сатылары мен деңгейлері. Программа құруға техникалық тапсырма. Программаларды техникалық жобалаудың сатылары. Алгоритмдердің құрылымдық схемасын құру. Мәліметтерді ұымдастыру программалар мен программа ішіндегі интерфейсін құрылымын құрастыру.

Ақпаратты компьютердің сыртқы құрылғыларында көрсету. Файлдармен жұмыс істеу. Мәліметтердің динамикалық құрылымы. Сілтемелер. Кезектер және жұмыс істеу.

Графика. Графиктік бейнелеудің алгоритмдік негізделуі. Стандартты ішкі программалардың қоры. Стандартты ішкі программалардың қорын ұымдастырудың ережелері. Стандартты ішкі программалардың қорын есептерді шешу үшін қолдану.

Программаларды ұымдастыру. Өртүрлі құрылымдық ұйымдасқан программалардың мысалдары.

7 Тақырып. Программалау технологиясының негізі

Программалық жабдыктануды жобалаудың әдістері. Программалауды жобалаудың әдістері. Программаларды жобалаудың өрлеуі және құлдырауы, олардың сәйкестігі. Құрылымдық программалау. Модульдік программалау.

Программалық тілді таңдау. Программалау тәсілі. Программалаудың сапалық көрсеткіші. Программаның оқытылатындығы, комментарийлар. Қателерден қорғау арқылы программалау. Программаларды өңдеу сатылары. Программаларды құжаттау. МСТ бекіткен программалық құжаттау түрлері. Программалық жабдықтауды автоматтандыруды жобалау.

7. Тәжірибелік (семинар, зертханалық, студиялық өздік) сабақтар мазмұны, олардың сағаттық көлемі

Тәжірибелік жұмыс 1

Тақырып 1 Алгоритмдердің сызықты құрылымының программасын жазу. Тармақталған алгоритмдердің құрылымын программалау.

Бақылау сұрақтары:

1. Программа басы қалай жазылады?
2. Программаның құрылымында қандай негізгі бөлімдер болады? Олардың жазылу тәртібі?
3. Айнымалылар қандай қызметші сөзден кейін сипатталады? Мысал келтір?
4. Айнымалыларды және олардың типтерін сипаттау арқылы компьютерге нені хабарлаймыз?
5. Программа блогы қандай қызметші сөздер аралығында жазылады?
6. Оператор дегенді қалай түсінуге болады? Жазылуы қандай символмен аяқталуы керек?
7. Меншіктеу операторы қалай жазылады? Қалай орындалады? Жадыда не болады?
8. READ және READLN процедуралары не үшін қолданылады? Айырмасы қандай?
(мысалдар келтір).
9. WRITE және WRITELN процедуралары не үшін қолданылады? Айырмасы неде?
(мысалдар келтір).
10. Есепті алгоритмдеу процессін неден бастау керек?
11. Қандай кезеңдерден тұрады?
12. Сызықтық құрылымды алгоритмнің блок-схемасы қандай блоктардан тұрады?
13. Неге сызықтық құрылымды дейміз?

Тапсырмаларды орындау бойынша әдістемелік кеңестер:

Паскаль тілінің операторларын қарапайым және күрделі түрлерге бөлуге болады. Күрделі операторлар бірнеше қарапайым операторлардан тұрады.

Қарапайым операторлар қатарына меншіктеу, енгізу және шығару операторлары жатады. Кез келген тілдің ең негізгі операторының бірі, ол:

Меншіктеу операторы, жазылу форматы мынадай:

< айнымалы > := < өрнек >;

Мұндағы := символы меншіктеу операциясын орындайды. Оң жақтағы өрнектің мәні есептеліп, сол жақтағы айнымалыға (ұяшыққа) меншіктеледі.

Өрнектің типі айнымалы типімен сәйкес болуы керек.

Енгізу операторы:

READ(a1,a2,...,an),

READLN(a1,a2,...,an)-оқу; {LN-жаңа жолға өту}.

Мұндағы: a1,a2,...,an айнымалылар тізімі. Программа орындалу барысында READ операторы кезіксе компьютер тоқтап, осы айнымалыларға клавиатурадан мәндерді енгізуді күтеді.

Шығару операторы:

WRITE(<айнымалылар>,<символдық тұрақтылар>)

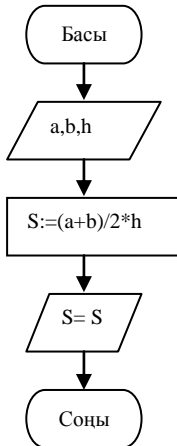
WRITELN(<айнымалылар>,<символдық тұрақтылар>)

экранға шығару, яғни тізімде тұрған айнымалылардың мәндерін, апострофқа алынған символдық тұрақтыны, санды экранға шығарады.

Мысалы:

WRITE (' значение Y=', Y); {значение Y= және Y -тің мәні шығады}

Есеп: Трапецияның ауданын есептейтін блок-схема және программа құрайық:



```
Program трапеция; {прогр-ң басы, аты}
  Vara,b,h,s: real; {айнымалылар, типтері}
Begin
  Writeln('вв. a,b,h'); {енгізуді хабарлау}
  Readln(a,b,h); {мәндерді енгізу};
  S := (a+b)/2*h; {ауданды есептеу}
  Writeln ('S=', S); {S -ң мәнін шығару}
  Readln; {ENTER-ы басуды тосу}
END.
```

Жаттығулар

1. Программаның мына берілімдер блогында қателер бар, соны түзетіп жаз:

a) Program ЕСЕП;

VAR: x,y real

Begin;

b) Program primer_b;

Var T,N;

b:= a*n;

Begin

w) Program

Var n,m,k : integer;

read(a);

Begin

g) Program tip;

Var i,j: integer;

Begin

k := n Div m;

2. Мына меншіктеу операторлар тізбегі орындалған соң айнымалылардың мәні неге тең болатынын анықта?

FOR < цикл параметрі > := n1 DOWNTO n2 DO
< цикл денесі > ;

Мұндағы: FOR (үшін), TO (дейін), DO (орындау) - қызметші сөздер;
< цикл параметрі > - циклді басқару айнымалысы, Integer типті (немесе кез келген реттелген тип).

n1, n2 - цикл параметрінің бастапқы және соңғы мәндері, (типі циклді басқару айнымалысымен сәйкес).

For операторының орындалу тәртібі

Ең алдымен бастапқы мән есептеліп цикл параметріне меншіктеледі. Содан кейін мына әрекеттер қайталанатын:

- 1.-< цикл параметрі > <= n2 - шарты тексеріледі, егер шарт орындалса, онда:
- 2.-Цикл денесіндегі операторлар орындалады;
- 3.-Цикл параметріне 1 қосылып, 1-ші пунктқа оралады;
- (Downto болса, < цикл параметрі > >= n2 тексеріліп, цикл параметрі -1-ге өзгеріп отырады.);
- 4.-Егер шарт орындалмаса, онда For операторы жұмысын аяқтайды;

For операторы нені анықтайды

а) басқарушы айнымалы мәндерінің өзгеру диапазонын және цикл ішіндегі оператордың қайталану санын;

б) айнымалы мәнінің өзгеру бағытын (өсу немесе кему);

в) әрбір қайталану сайын орындалатын іс-ірекеттерді.

Мысал:

```
For i := 1 To 5 Do
  Begin
    a := 2*i;
    b := 2*i + 1;
    writeln(a:3, b:3)
  end.
```

Циклдік бөлім бес рет қайталанатын, басқарушы айнымалы i, мәні 1,2, 5 өзгереді. Программа орындалу нәтижесінде айнымалылар мына мәндерді қабылдайды:

I	1	2	3	4	5
A	2	4	6	8	10
B	3	5	7	9	11

Осы программа үзіндісін кему бағытында ұймдастырса, онда, цикл параметрі 1-ге кеміп отырады:

```
For i := 5 Downto 1 Do
  Begin
    a := 2*i;
    b := 2*i + 1;
    writeln(a:3, b:3)
  end.
```

I	5	4	3	2	1
A	10	8	6	4	2
B	11	9	7	5	3

Басқарушы айнымалының мәндерінің өзгеру диапазоны өрнекпен берілуі мүмкін, ол цикл орындалмас бұрын бір рет есептеледі, циклдің қайталану саны анықталады.

Мысал: x := 3;

```
For j := x + 2 downto x - 2 do
  Begin
    a := 2*j;
    writeln(a:3);
  end;
```


бұл мысалда j басқару айнымалысының мәні біртіндеп $x+2$ мәнінен соңғы $x-2$ мәніне дейін азайды.

Жаттығулар:

Мына программалар үзінділерінде цикл неше рет қайталады?

- a) for k := -1 to 1 do ...
- b) for k := 10 to 20 do ...
- c) k := 5; r := 15;
for i := k + 1 to r - 1 do ...
- d) k := 5; r := 15;
for i := 0 to k * r do ...

Есеп

Мына тізбектің көбейтіндісін және қосындысын есептеу:

$$3*5*7* \dots 13;$$

Program S_and_P;

Var n,s,k : integer; {n-цикл параметрі,s-қосынды,k-қадам}

p : longint; {p-көбейтінді, ұзын бүтін сан болуы}

begin

k := 1; {қадамның бастапқы мәні }

s := 0; {қосындының бастапқы мәні }

p := 1; {көбейтіндінің бастапқы мәні }

For n := 1 To 13 div 2 Do {циклдің қайталану саны есептеледі}

Begin

inc(k,2); {қадамның мәні 2 өсіп отырады }

s := s + k; {қосындының есептелінуі}

p := p * k; {көбейтіндінің есептелінуі}

end;

Writeln(' s=', s, ' p=', p); {нәтижені шығару}

Readln

end.

(Жауабы: s=48 ,p= 135135).

Жаттығулар

1. 1-ден 20 дейін сандар квадраттарын экранға шығару;

2. Мына өрнектің мәнін есептеу программасын құру:

$$Y = ((\dots (20^2 - 19^2)^2 - 18^2)^2 - \dots - 1^2)^2$$

3. Y-тің мәндерін, X - тің 4,5, ... , 28, мәндері үшін есептеу:

$$Y := 2T^2 + 5,5T - 2, \text{ мұндағы } T = X + 2;$$

4. 7-ге көбейту кестесін құру;

5. Қазіргі доллар курсына сәйкес 1,2,...,20 АҚШ долларын теңгеге аудару кестесін құру. (курстың мәні клавиатурадан енгізіледі);

6. 10,11, ...,22 дюйм тізбегін сантиметрге аудару кестесін шығару (1 дюйм=25,4 мм).

Есептер

1. Берілген 10-нан 99 дейін сан аралығынан, цифрлар қосындысы N санына тең сандарды шығару.

($0 < N <= 18$).

2. Цифрларының қосындысы, берілген N санына тең, үш орынды сандардың санын есептеу. (Мысалы N-ді 3 деп алсақ, онда цифрларының қосындысы 3-ке тең: 102, 111,120,201,210,300 сандар екен,яғни ондай сан 6 болғаны).

3 25 тен 125 дейін сандардың кубтарының қосындысын есептеу.

4. Екі орынды сандар арасынан, цифрлар квадраттарының қосындысы 13- ке қалдықсыз бөлінетіндерін табу.

5. Екі орынды сандар арасынан мына қасиетке сай, іздеу программасын құру: егер, санның цифрлар қосындысына, осы қосындының квадратын қосқанда, санның өзі шығу керек.

6. Кейбір үш орынды сандардың, квадраттарының соңғы үш цифрі санның өзіне тең. Сондай сандарды іздеу программасын құру.

7. Төрт орынды сандар арасынан, 133 бөлгенде, қалдығы 125, ал 134-ке бөлгенде қалдығы 111 болатын сандарды іздеу программасын құру.

8. 100-ден кіші, оң тақ сандардың қосындысын есептеу.

9. А дан В-ға дейін аралығында, 4-ке еселі, бүтін оң сандар қосындысынын табу керек (А мен В-ң мәндері клавиатурадан енгізіледі).

10. 2,4 немесе 8 цифрларымен бітетін, 3-ке еселі, 20 артық, 100 кем, бүтін оң сандардың қосындысын табу.

11. Келесі заңдылықты пайдаланып, натурал сандарды квадрат дәрежеге шығару программасын құру:

$$\begin{aligned}1^2 &= 1 \\2^2 &= 1 + 3 \\3^2 &= 1 + 3 + 5 \\4^2 &= 1 + 3 + 5 + 7 \\&\dots\dots\dots \\n^2 &= 1 + 3 + 5 + 7 + 9 + \dots + 2n - 1\end{aligned}$$

Алдыңғы шартты While цикл операторы

Циклдің қайталану саны алдын ала белгісіз болса, онда алдыңғы шартты While операторы қолданылады.

Жазылу үлгісі

While < шарт > Do

Begin

< 1 оператор >;

< 2 оператор >;

...

< n оператор >;

End;

Мұндағы: While (әзірше) - қызметші сөзі;

< шарт >-логикалық типті өрнек.

While операторының орындалу тәртібі

Цикл қайталанар алдында, логикалық өрнектің мәні тексеріледі, егер True болса, онда цикл денесі қайталанып орындалып отырады. Керісінше False болса, циклдің орындалуы аяқталады.

Егер, ең басында, өрнектің мәні False болса, онда цикл бір ретте орындалмайды.

Есеп

Берілген N натурал санының цифрлар санын есептеу.

Шешімі Цифр санын есептеуді ақырғы цифрдан бастаймыз. Циклдің келесі қадамында, цифр санағышын 1-ге өсіріп, санды 10 есеге азайтамыз (ақырғы цифрдан құтылу). Осы әрекеттер N саны нөлге тең болғанға дейін қайталанатын.

```
Program san_esepteu;
```

```
Var m, n : Longint; {ұзын бүтін типті санның  
идентификаторы}
```

```
k : integer; {санағыш}
```

```
Begin
```

```
Writeln ('Введите натуральное число');
```

```
Readln (n);
```

```
m := n;
```

```
k := 0;
```

```
While m <> 0 Do {әзірше, m <> 0 True болса, орындау)}
```

```
Begin
```

```
Inc (k); {k := k + 1; }
```

```
m := m div 10; {ақырғы цифрасы жойылады }
```

```
end;
```

```
Writeln (' ', n, ' - ', k, ' '); {цифрлар санын шығару }
```

```
Readln;  
End.
```

Жаттығулар

1. Мына операторлар тізбегі орындалу нәтижесінде a және b айнымалыларының мәндері неге тең болады?

```
a := 1; b := 1;  
While a <= 3 Do a := a + 1; b := b + 1;
```

2. Мына операторлар орындалғанда S айнымалысы қандай мән қабылдайды?

```
a) s := 0; i := 0;          b) s := 0; i := 0;  
   While i < 5 Do Inc(i);   While i > 1 Do  
   s := s + 100 div i;      Begin  
                             s := s + 100 div i;  
                             dec(i);  
                             End;
```

Есептер

1. Санның цифрларының қосындысын табу.
2. Санның жоғарғы разрядында тұрған цифрді табу.
3. Жазылған N санының басына және аяғына 1 тіркестіріп жазу керек. Мысалы, егер N=3456 болса, шығарғанда 134561 болуы керек.
4. Санның бірінші және ақырғы цифрларының орнын ауыстыру.
5. Натурал санның жұп цифрлар санын есептеу.
6. Бүтін санның ең үлкен цифрін табу.
7. Бүтін санның, 5 артық цифрлар қосындысын табу.
8. Берілген цифр бүтін санда неше рет кезігеді?
9. Клавиатурадан енгізілген 10 бүтін сандар тізбегі, осу ретімен орналасқанба?
10. Берілген натурал сан палиндром екенін тексеру программасын құру.

Кейінгі шартты Repeat цикл операторы

Қайталану саны алдын ала белгісіз алгоритмдерді ұйымдастыруға, кейінгі шартты Repeat операторы қолданылады.

Жазылу үлгісі

```
Repeat  
  <оператор 1>;  
  <оператор 2>;  
  <оператор 3 >;  
  ...  
  <оператор n >;  
Until <шарт >;
```

Мұндағы: Repeat (қайталау), Until (дейін) - қызметші сөздер;

< шарт > - логикалық типті өрнек;

Бұл цикл алдыңғы циклге ұқсас, бірақ шарт, цикл денесі орындалғаннан кейін тексеріледі. Сондықтан, цикл ең жоқ дегенде бір рет орындалады.

Орындалу тәртібі

1. Алдымен Repeat сөзінен кейінгі операторлар тізбегі орындалады;
 2. Until сөзінен кейінгі шарт тексеріледі;
 3. Егер, өрнектің мәні False болса, онда цикл қайталанады, True болса цикл аяқталады.
- Begin . . . End операторлық жақшаларды қолданбағанда болады, себебі мұнда операторлар Repeat . . . Until аралығында жазылып сол жақшалардың функциясын орындап тұр.
- Мысал:

```
x := 8;  
Repeat
```

```

Y := x * x;
Writeln( x :3 , y : 5 );
x := x + 2
Until x < 0

```

Мына операторлар орындалған соң s неге тең болады?

```

s := 0; i := 1;
Repeat
s := s + 5 div i;
i:=1-1;
Until i <= 1;

```

Ұсынылатын әдебиет тізімі (1-5)

Тәжірибелік жұмыс 3

Тақырып 3 Екі өлшемді массив. Символдық мәліметтерді өңдейтін программалау. Жазбалар.

Тапсырмаларды орындау бойынша әдістемелік кеңестер:

Массив - деп, бір типті, саны шектелген, барлық мүшелері нөмірмен реттеліп және бір идентификатормен бірлестіріп аталған, берілімдер жиынын айтады.

Мысалы: нақ
типті кездейсоқ сандар тізбегін:

1.2, 14.6, -5.2, 8.3, 0.15, 3.9

нақ типті A массиві деуге болады.

Массивтерді өңдеу процессінде, оның әр мүшелеріне тура қатынас жасау үшін, оны нөмірлеу қажеттілігі туындайды:

Мүшелердің нөмірі (индексі)	1	2	3	4	5	6
Массивтің идентификаторы: A	1.2	14.6	-5.2	8.3	0.15	3.9

Массивтің мүшелері индекстің нөмірлерімен реттеледі, сондықтан оларды индекстелген айнымалылар дейді. Паскаль тілінде индекс тік жақшаға алынады.

Қарастырылған мысалда A массивінің мүшелері:

$A[1] = 1.2$, $A[2] = 14.6$, $A[3] = -5.2$, $A[4] = 8.3$, $A[5] = 0.15$, $A[6] = 3.9$.

Мүшелері бір индекспен нөмірленген массивті, бір өлшемді дейді.

Массивті сипаттау

Егер программада массив қолданылса, онда ол, айнымалы Var бөлімінде немесе Type типтер бөлімінде сипатталуға тиіс. Алдымен Var бөлімінде сипатталу тәртібін қарастырайық.

а) Var айнымалылар бөлімінде сипаттау

Жазылу үлгісі:

Var

<массивтің идентификаторы> : Array [T1] Of T2;

мұндағы:

Аггау (массив), Of (одан)-қызметші сөздер;

T1 -массив индексінің типі, тек реттелген тип қолданылады.

Индекстер тұрақты диапазонымен беріледі. Яғни, массив мүшелерінің саны тұрақты болу керек.

T2-массив мүшелерінің типін анықтайды.(Паскаль тілінің кез келген типі).

Жоғарыдағы мысалдың сипаттамасы мынадай:

Var A : Array [1..6] Of Real;

Массивті дұрыс сипаттау мысалдары:

```

Var
A:      Array [1..10] Of Integer;
B:      Array [0..40] Of Char;
C:      Array [-2..2] Of Boolean;

```

б) Type бөлімінде сипаттау. Массивтің типі

Паскаль тілінде, массивтерді сипаттаудың тәртібі екі кезеңнен тұруы мүмкін. Алдымен Type бөлімінде массивтің типі көрсетіледі. Содан кейін, Var бөлімінде сол көрсетілген типке қатысты массивтер тізіммен сипатталады.

Жазылу үлгісі

```

Type <типтің идентификаторы > = Array [T1] Of T2;
Var < массивтің идентификаторы > : <типтің идентификаторы >;

```

Мысалы:

```

Type
MAS = Array [1..10] Of Integer;
Var M1: MAS;

```

Егер, программада бір неше, мысалы R,A,B,C массивтері, көрсетілген MAS типті болса, онда, тек айнымалылар бөлімінде өзгеріс енгізіледі:

```

Var
R,A,B,C : MAS;

```

Екі өлшемді массивтер

Егер, кез келген берілімдер тізімі жолмен және бағаналармен берілсе, онда оны екі өлшемді массив дейді. Математикада матрица деп аталатынын білеміз. Екі өлшемді массивтің мүшелері екі индексмен нөмірленеді.

```

Мысалы:  5   4   3   6
          2   8   1   7
          4   3   9   5

```

матрицасы, бүтін сандар типті, өлшемі -3 жолдан, 4-бағанадан тұрады.

Бүкіл матрицаны -A, жолды -I, бағананы -J десек, әр мүшесі мына түрде жазылады: A[I,J].

Сонда: A[1,1]=5, A[2,3]=1, A[3,2]=2, т.с.с.

Сипаттау түрлері

```

1). Var
Mas Array[1..3] Of Array[1..4] Of Integer;

```

Бұл сипаттаманы 1-шіден, бірнеше массивтерден тұратын массив; 2-шіден екі өлшемді массив сипаттамасы деп ұғуға болады. Бұл мысалдың эквиваленті төмендегідей жазылады:

```

2). Var
Mas : Array[1..3,1..4] Of Integer;

```

Массив мүшелерінің қолдану түрлері:

```

Vector[1]      V2[3,7]      V2[K]
Vector[(i+1)*2]  V2[i,j]  V2[K,5]

```

Массив мүшелері айнымалы болуы мүмкін:

```

MAS[i,j]:=MAS[i,j-1]+5;

```

Мысалдар

1-ші есеп. Нақты сандар қосындысын есептеу.

```

Program    summa;
Const N=7;    {сандар саны}
Var
A : Array[1..n] Of Real;    {N-санды массив}
S : Real;    {қосынды}
I : Integer;    {цикл параметрі}
Begin
Writeln('вв.числа через пробелы'); {мүшелерін енгізу}
For I := 1 To N Do
Read( A[ I]);
S:=0;
For I := 1 To N Do    {қосындыны есептеу}
S:=S+A[ I];
Writeln('сумма чисел =', S:6:2);
Readln;
End.

```

2-ші есеп. В[3..5]- өлшемді нақ типті матрица берілген. Матрицаның барлық мүшелерінің көбейтіндісін есептеу керек.

```

Program    Matric;
Const N=3;    {жол саны}
      M=5;    {бағана саны}
Type Mas=Array[1..N,1..M] Of Real;    {матр-ң типін сипаттау}
Var
B : Mas;    {матрица}
I : 1..N; {жол индексі}
J : 1..M;    {бағана индексі}
P : Real;    {көбейтінді}
Begin
Writeln('вв.значения матрицы');    {мәндерін енгізу блогы}
  For I := 1 To N Do
    For J := 1 To M Do
      Read(B[ I,J]);
P:=1;    {көбейтіндіні есептеу блогы}
  For I := 1 To N Do
    For J := 1 To M Do
      P := P*B[ I,J];
Writeln('произведение=',P);
Readln;
End.

```

Қате беретін жағдайларды көрсетейік

```

1).  VAR
      Vector: array[0..10] of real;
      Vector[11]:=0.5;

```

Бұл жағдайда қате индекске байланысты: сипаттама бөлігінде индекстің жоғарғы шекарасы 10 деп алынған, ал программа денесінде 11-ші, яғни индекстер диапазонына кірмейтін индексті мүшені қолданбақ болғанымыз дұрыс емес.

```
2. Vector[i]:= Vector[i-1]+i;
```

Бұл жағдайда қате i , $i-1$ мәндеріне байланысты болады.

8.4. Массивтерді өңдеудің негізгі алгоритмдер кестесі

а) бір өлшемді массивтерді өңдеу

Алгоритмдер	Түсініктеме
FOR I := 1 TO N DO READLN(M[I]);	Клавиатурадан бір бірлеп енгізу
FOR I := 1 TO N DO M[I]=Random(a+b)-a	[a;b]-аралығынан кездейсоқ сандармен толтыру
FOR I := 1 TO N DO WRITE(M[I], ' ');	Жолмен шығару
FOR I := 1 TO N DO WRITELN(M[I]);	Бағанамен шығару
Min:=m[i]; FOR I:=2 TO N DO If m[i]<Min then Min=m[i];	Ең кіші мүшені іздеу
FOR I := 1 TO N Div 2 DO Begin P:=m[2*i-1]; M[2*i-1]=M[2*i]; M[2*i]=P; End;	Жұп және тақ орындарда тұрған мүшелердің орнын ауыстыру
FOR I := 1 TO N DO Begin M[2*i-1]=A[i]; M[2*i]=B[i]; End;	Екі массивті біріктіру Таңдау кезекпен

б) екі өлшемді массивтерді өңдеу

Алгоритмдер	Түсініктеме
FOR I := 1 TO N DO Begin FOR J := 1 TO M DO A[i,j]:=Random(a+b)-a; End;	Матрицаны кездейсоқ сандармен толтыру
FOR I := 1 TO N DO Begin FOR J := 1 TO M DO Write(A[i,j]:3, ' '); Writeln; End;	Матрицаны кесте түрінде шығару
FOR I := 1 TO N DO Begin FOR J := 1 TO M DO S[i,j]:=A[i,j]+B[i,j]; End;	Матрицаларды қосу

<pre> FOR I:= 1 TO N DO Begin FOR J := 1 TO M DO Begin P:=0; FOR h:= 1 TO k DO P:=P+A[i,h]*b[h,j]; S[i,j]:=P; End; End; </pre>	Матрицаларды көбейту
<pre> FOR I:= 1 TO N DO Begin FOR J := 1 TO M DO Begin P:=A[i,j]; A[i,j]:=B[i,j]; B[i,j]:=P; End; End; </pre>	Матрицаны транспонирлеу (басқа жерге ауыстыру)

Есептер

1. Массивтің оң мүшелерінің қосындысын табу.
 2. Массивтің жұп мүшелерінің қосындысын табу (немесе берілген санға есе мүшелер қосындысын).
 3. Массивтің жұп индексті, жұп мүшелерінің қосындысын табу
 4. Массивтің бірінші бес мүшелерінің қосындысын табу.
 5. K1-ден k2-ге дейін мүшелердің қосындысын табу,мұнда K1 және k2 клавиатурадан енгізіледі.Олардың енгізу дұрыстығын тексеру керек.
 6. Берілген А санынан үлкен мүшелердің қосындысын табу.(А-саны клавиатурадан енгізіледі).
 7. А-дан В-ға дейін аралыққа жататын мүшелердің қосындысын табу.(А мен В клавиатурадан енгізіледі).
- Ескерту. 1-7 есептерде көбейтіндіні де табуға болады.
8. Барлық мүшелері әр түрлі болған жағдайда,массивтің ең үлкен мүшесін және оның нөмірін табу.
 9. Барлық теріс мүшелерінің нөмірлерін тауып,экранға шығару,егер ондай жоқ болса,сондай хабарлама шығару.
 10. Ең үлкен мәнді мүшелердің нөмірлерін табу.
 11. Ең кіші мүшені табу.
 12. Тақ мүшелердің санын есептеу.
 13. Теріс мүшелердің санын есептеу.
 14. Берілген А санынан массивтің неше мүшесі, модулі бойынша үлкен?
 15. 3-ке және 5-ке еселі барлық мүшелерді табу. Олар нешеу?
 16. Берілген массивте көршілес екі оң мүшелер бар ма?
Бірінші (соңғы) жұп мүшелердің нөмірлерін табу.
 17. Берілген санға тең мүше бар ма? Егер бар болса,онда біреуінің нөмірін шығару.
 18. Бес бүтін сандар арасынан жұп сандардың санын табу.(массивті пайдаланбай).
 19. Модулі бойынша ең үлкен мүшенің таңбасын ауыстыру.
 20. Барлық жұп мүшелерді өзінің квадраттарымен ауыстыру,ал тақ мүшелерді екі еселеу.
 21. Оң мүшелерден, k1 нөмірлі мүшені алу,ал теріс мүшелерге, k2 нөмірлі мүшені қосу,нөлдік мүшелерді өзгеріссіз қалтыру.
 22. Екі массив берілген. Әр массивтің мүшелерінің арифметикалық орташасын табу және оларды салыстыру.
 23. Матрицаның әр жолындағы теріс мүшелерінің санын есептеу.

24. Матрицаның ең үлкен мүшесін және оның нөмірін табу.
25. Екі А және В квадрат матрицаларының көбейтіндісін есептеу.

Символдық массивтер

Мүшелері символдық (Char) типтен тұратын бір өлшемді массивтерді символдық массив немесе символдық жол дейді. Жазылу үлгісіне мысалдар:

```
VAR  
S: array[1..13] of char;  
S2:= 'жол мысалы';
```

Паскаль тілінде символдық массивтермен жұмыс жасайтын қосымша мүмкіндіктер бар.

Конкатенация амалы

Символдық жолдарды бір-біріне тіркестіру, басқаша айтқанда қосу амалы -деп аталады. Қосу таңбасы (+) амалдың белгісі болады.

Мысалы, төмендегідей сипаттама бөлімі бар болсын:

```
Var  
S1: array[1..5] of char;  
S2: array[1..6] of char;
```

онда келесі жазулар орынды болады:

```
S1:= 'жолды';  
S2:= 'массив';  
Writeln(S1+ ' '+S2);
```

Сонда, осы программа үзіндісі орындалу нәтижесінде экранға мынадай хабарлама шығады:
жолды массив

Ескерту

1. Жолды меншіктеу кезінде, жол ұзындығы мен массив өлшемі бірдей болу керек.

2. Символдық массивтер үшін, меншіктеу операторының оң жағында конкатенация амалын орындауға болмайды.

Егер S:=S1+ ' '+S2 -деп жазылған болса, S символдық массивтің өлшеміне қарамай, программаны компиляциялау кезінде қате шығады. Мұндай жағдайда ұзындығы айнымалы болатын жолдарды қолдану керек.

Ұсынылатын әдебиет тізімі (1-5)

Тәжірибелік жұмыс 4

Тақырып 4 Ішкі программаларды құру және дайындау. Сыртқы құрылғыларда орналасқан құрылымдық мәліметтерді өңдеу есебін программалау

Тапсырмаларды орындау бойынша әдістемелік кеңестер:

Біреу кезде, қайталанғыштықта формаға бағдарламаларда кең таралған және сол последовательность әсерлердің мәліметте әр түрлі кезеңдерде орындалуға тиісті. Бағдарламаларда әр түрлі орындарда бөліктер кездесіп жатыр, бірдей атқархатын әсерлер бойынша және мәндерде тек қана айыратын бастапқы осы. Тапсырма беру дәл келу бағдарламаларда құрастыруда бір және сол әрбір қайталанатын бөліктерден лайықты операторлардың топты. Тілде ұқсас қайталаулар көп тиімді программалау үшін ішкі программалар ұғымы енгізген. Операторлардың қайталанатын тобы дербес бірліктерге түрде пісіп жетіп жатыр – ішкі программалар, бір рет жазылады, ал бағдарламаларға лайықты орындарда қамтамасыз етіп жатыр тек қана үндеу ода. Ішкі программалар аппараттың қолдануысы көлем қысқарту мүмкіндік беріп жатыр және көрнекіліктен және оқыландықтан көзқарастан бағдарламалар құрылымды жақсарту керек. Қарастырылған мүмкін болу ішкі программалардан дербес бағдарламадан сияқты өз кіретін және демалыс осы).

Өз сипаттамадан көмекпен бағдарламаға ендірген процедураларға және функцияларға түрде іске асыруға ішкі программаларға Паскальға тілде.

Процедуралар. Сипаттама.

Процедуралар айнымалы бөлімнің артында артына бағдарламаларға сипаттама бөлікте арнайы бөлімде суреттеліп жатыр.

Любая процедура түр, ұқсас бағдарламаға, процедурадан және блоктан бастан.

ПАРАМЕТРЛЕРДІҢ <ТІЗІМІ PROCEDURE <АТЫ>,</p></div>
<div data-bbox="138 80 931 127" data-label="Text">
<p>PROCEDURE қайда – қызметтік сөз, АТЫ – ПАРАМЕТРЛЕРДІҢ процедуралар, ТІЗІМІ аты - белгі үшін атылардың бұрышқа-бұрыш бастапқы осы және олардың түрлерінен нұсқаумен процедуралар жұмыстары нәтижелердің. Тізімде аталған параметрлер, үстірт деп аталып жатыр.</p></div>
<div data-bbox="197 125 386 141" data-label="Text">
<p>PROCEDURE <АТЫ>;</p></div>
<div data-bbox="138 141 931 202" data-label="Text">
<p>Процедуралар маңызды бөлігі өзі блок ұсынып жатыр және тұрады, демек, сипаттамалардан бөлімнен (мерген, тұрақтыларды, түрлерді, айнымалы, процедураларды, және өзі құрама BEGIN оператор ұсынатын операто-ровамицин функциялардың) және бөлімнің – END. Үтірден нүктемен процедуралар блогы бітіп жатыр.</p></div>
<div data-bbox="138 201 931 291" data-label="Text">
<p>Функция – орындаулар бұл ішкі программа, нәтиже атыға бұл функциялар иемденхатын жалғыз скалярлық мән бар. Демек, функциялар дербес жағдаймен процедураларды келіп жатыр және оларға маңызды айырмашылығы болып жатыр сол, не, - бірінші, функциялар орындаулары нәтижесі – бір мән, ал процедуралар – бір немесе бірнешесі ; -екінші, бұл функциялар атыға мәні сияқты, функциялар орындаулары нәтижесі негізгі бағдарламаға жұғысып жатыр, ал процедуралар орындаулары нәтижелері – оның параметрлердің мәндің сияқты.</p></div>
<div data-bbox="197 290 921 307" data-label="Text">
<p>Процедуралар ұқсас сипаттамаға функциялар сипаттамасы және бас және блоктан тұрады.</p></div>
<div data-bbox="197 318 686 335" data-label="Text">
<p>ПАРАМЕТРЛЕРДІҢ <ТІЗІМІ FUNCTION <АТЫ> : <ТҮР></p></div>
<div data-bbox="138 334 931 382" data-label="Text">
<p>FUNCTION қайда – қызметтік сөз, АТЫ – ПАРАМЕТРЛЕРДІҢ функциялар, ТІЗІМІ аты – үстірт параметрлердің тізім (бастапқы осы) олардың түрлерінен, ТҮРІНЕН нұсқаумен – нәтиженің түрі : функциялар атысы алуға тиісті значе-ние.</p></div>
<div data-bbox="197 381 704 397" data-label="Text">
<p>Параметрлерсіз функциялар сипаттамасы рұқсат етіліп жатыр :</p></div>
<div data-bbox="197 396 439 412" data-label="Text">
<p>FUNCTION <АТЫ> : <ТҮР>;</p></div>
<div data-bbox="197 411 693 427" data-label="Text">
<p>Атыда бағдарлама-функцияда маңызды бөлікте тұрмыс тиісті</p></div>
<div data-bbox="138 427 931 457" data-label="Text">
<p>Атыда бағдарлама-функцияда маңызды бөлікте жауаптың) (мән кейбір мән берған болуға тиісті т.е., Аты иемденулерге кейбір операторға сол бөлікте қатысуға біреу рет тиісті.</p></div>
<div data-bbox="197 470 686 486" data-label="Text">
<p>ПАРАМЕТРЛЕРДІҢ <ТІЗІМІ FUNCTION <АТЫ> : <ТҮР></p></div>
<div data-bbox="138 485 931 532" data-label="Text">
<p>FUNCTION қайда – қызметтік сөз, АТЫ – ПАРАМЕТРЛЕРДІҢ функциялар, ТІЗІМІ аты – үстірт параметрлердің тізім (бастапқы осы) олардың түрлерінен, ТҮРІНЕН нұсқаумен – нәтиженің түрі : функциялар атысы алуға тиісті значе-ние.</p></div>
<div data-bbox="197 531 704 547" data-label="Text">
<p>Параметрлерсіз функциялар сипаттамасы рұқсат етіліп жатыр :</p></div>
<div data-bbox="197 547 439 562" data-label="Text">
<p>FUNCTION <АТЫ> : <ТҮР>;</p></div>
<div data-bbox="197 562 693 577" data-label="Text">
<p>Атыда бағдарлама-функцияда маңызды бөлікте тұрмыс тиісті</p></div>
<div data-bbox="138 577 931 607" data-label="Text">
<p>Атыда бағдарлама-функцияда маңызды бөлікте жауаптың) (мән кейбір мән берған болуға тиісті т.е., Аты иемденулерге кейбір операторға сол бөлікте қатысуға біреу рет тиісті.</p></div>
<div data-bbox="138 606 931 696" data-label="Text">
<p>Мәліметтерді сақтау үшін оған екі массмвке орын бөлуімізге болатын еді. Бірінші массив (string типті) оқушылардың аты жөнін сақтауға, екіншісі сақтауға арналған. Егер бізге қосымша мәліметтер (мысалы: сыныбы, ұлты, т.б.) енгізу керек болса, онда енгізуге тура келеді. Бір мезгілде бірнеше массивпен жұмыс істеу және оларды смпаттамалары бойынша сұрыптауға, әрине болады, бірақ үлкен жұмысты талап етеді. Сонымен қатар бұл мәліметтерді файлда сақтау керек болса не істейміз?</p></div>
<div data-bbox="138 695 931 742" data-label="Text">
<p>Файлдар туралы сұрақтарды талдағанда айтылғандай, файл элементтері тек бір типті элементтер болып келеді. Олай болса әр түрлі типтегі элементтерді әрбір типке – жолдық типке айналдырып, мәтіндік файлдармен жұмыс істеуіміз керек.</p></div>
<div data-bbox="138 741 931 771" data-label="Text">
<p>Мәліметтерді сақтай аламыз, бірақ болашақта бұл мәліметтермен жұмыс істеу жеткілікті қиындық туғызады.</p></div>
<div data-bbox="138 770 931 832" data-label="Text">
<p>Бұл қиындықтан шығу үшін мәліметтердің құрама типі қолданады. Мәліметтердің бұл типі **жазба** деп аталады. Жазбалар әр түрлі типтегі элементтер жиынтығынан тұрады. Жазба элементтерін құраушыларды **өріс** деп атайды да, әрбір өрістің өзіндік атауы болады. Әрбір жазбаның бірегей атауы болады.</p></div>
<div data-bbox="173 844 886 893" data-label="Table">
<table border="1">
<thead>
<tr>
<th rowspan="2">Жазба атауы</th>
<th colspan="3">Тізім</th>
</tr>
<tr>
<th>Өрістің атауы</th>
<th>Фамилия</th>
<th>Аты</th>
<th>Туған жылы</th>
</tr>
</thead>
<tbody>
<tr>
<td>Өрістер мәні</td>
<td>Ахметов</td>
<td>Алдияр</td>
<td>1980</td>
</tr>
</tbody>
</table></div>
</div>

Жазбаны өңдеу элементар объектте түрінде шығады. Мұндай объект типі – record (жазба) деп аталады. Олай болса, мәліметтердің типтеріне байланысты шектеулер алынып тасталады.

Жазбаларды сипаттау үшін, оның атауы, берілген өрісте сақталатын мәліметтердің атауы және типі көрсетілуі керек. Жазбалардың жалпы түрде сипатталуы келесідегідей:

```
Түре<жазба атауы>=Record
  <1 өріс>:<1 тип>;
  <2 өріс>:<2 тип>;
  .....
  <n өріс>:<n тип>;
End.
```

Ұсынылатын әдебиет тізімі (1-5)

Зертханалық жұмыс 1. Тақырып 1. Өрнекті жазу, меншіктеу операторы. Паскаль тілінде алгоритмдеудің сызықты құрылымының программаларын жазу. Алгоритмдердің тармақталатын құрылымдарын программалау.

Зертханалық жұмыс 2. Тақырып 2. Алгоритмдердің циклдік құрылымдарын программалау.

Зертханалық жұмыс 3. Тақырып 3. Бір өлшемді массивтер және екі өлшемді массивтерді қолданып программа құру.

Зертханалық жұмыс 4. Тақырып 4. Қатарларды өңдеу есебін программалау. Жазбалар типін қолданып программа құру.

Зертханалық жұмыс 5. Тақырып 5. Процедура мен функцияны қолданып программа құру.

Зертханалық жұмыс 6. Тақырып 6. Графиканы қолданып программа құру.

Зертханалық жұмыс 7. Тақырып 7. Сыртқы тасымалдаушыларда орналасқан мәліметтер құрылым есебін программалау.

8. Өздік жұмысы тапсырмалары

1 Тақырып Қатарлар. Қатарлармен жұмыс істеу үшін қолданылатын процедуралар мен функциялар.

2 Тақырып Стандартты функциялар қорын пайдаланып программа құру.

3 Тақырып Модульдік программалауды қолдану мысалдары .

4 Тақырып Графикалық құрастыруларды қолданып программа құру.

5 Тақырып Динамикалық жадыны қолданып программа құру.

9. Курстық жұмыстардың (жобалардың) және т.б. ұсынылатын тақырыптары

1 Өртүрлі сандық әдістері қолданып, сызықты емес теңдеу жүйесін шешу.

2 Таңдап алу әдістерін пайдалану («ат туры» есебін шешу үшін программа құрастыру, шахмат тақтасындағы берілген екі алаңды жалғайтын өртүрлі шахмат тураларының қозғалысының ең қысқа жолын табудың программасын құрастыру).

3 Берілген сөз тізбегін қолданып, кроссворд құрастыратын программа құру.

4 Файлдарды өңдеу.

5 Берілген сөз тізбегін қолданып, кроссворд құрастырылатын программа құру.

6 Өртүрлі іздеу және сұрыптау әдістерін қолданып программаларды жасау.

10. Кеңес алу графигі СРОП (СРО –дан СРОП 25% құрайды)

Барлық сұрақтар бойынша кеңестер СРОП-тың ағымдағы семестр графигіне сәйкес жүзеге асады.

11. Тәлімгерлердің білімін тексеру кестесі

Тәжірибелік, зертханалық сабақтары мен СӨЖМ 0-100 баллмен бағаланады

Пән бойынша тапсырмаларды орындау мен тапсыру графигі

№	Жұмыс түрлері	Тақырып, мақсат және тапсырма мазмұны	Ұсынылатын әдебиет	Орындау ұзақтығы	Бақылау формасы	Тапсыру мерзімі
1	2	3	4	5	6	7
1	Тәжірибелік жұмыс 1	Желі топологиясының конфигурациясын оңтайлы таңдау.	1,2	4 жұма		5-ші жұма
2	Тәжірибелік жұмыс 2	Желілік топологияның динамикалық моделін құру	1,2,5	4 жұма		8-ші жұма
	Зертханалық жұмыс 1	Желідегі жұмыс статистика анализі	1,2,3	7 жұма		8-ші жұма
3	Межелік бақылау	2 тәжірибелік жұмыспен 1 зертханалық жұмысты қорғау			сынақ	8-ші жұма
4	Тәжірибелік жұмыс 3	Желіні жоспарлау	1,2,3	4 жұма		13-ші жұма
	Зертханалық жұмыс 2	Желілік клиент-серверлік жүйені құрастыру	2, 3, 4	8 жұма		14-ші жұма
5	Курстық жұмыс	Бекітілген тақырып бойынша		8 жұма	сынақ	14-ші жұма
6	Межелік бақылау	3 тәжірибелік жұмыспен 2 зертханалық жұмысты қорғау			сынақ	15-ші жұма

12. Тәлімгерлердің білімін бағалау критерийлері

Пәнді оқыту барлық өткен мәліметтерден тұратын тест түріндегі емтиханмен аяқталады. Емтиханға жіберілудің міндетті шарты бағдарламада берілген барлық тапсырмаларды орындау болып табылады.

Әрбір тапсырма 0-100 баллмен бағаланады.

Жіберілу рейтингі ағымдағы сабақтардағы (дәрісте болу, үй жұмыстары, СРО бойынша тапсырмалар, тәжірибе бойынша тапсырмалар және басқалары, межелік бақылау) орындалған тапсырмалардың орташа арифметикалық ортасын табу арқылы анықталады.

Пән бойынша қорытынды бақылауға (ҚБ) жұмыс оқу бағдарламасының барлық талаптарын (барлық тәжірибелік, зертханалық жұмыстары мен СРО бойынша тапсырмаларды орындау және тапсыру) орындаған, курстық жобаны (жұмысты) қорғауда жақсы баға алған және жіберілу рейтингін (50 баллдан кем емес) алған студенттер ғана жіберіледі.

Әрбір пән бойынша студенттердің оқу жетістіктерін ЖР мен ҚБ (емтихан, дифференциалды сынақ немесе курстық жұмыс/жоба) бағаларын салмақтық үлестерін ескере отырып қосындыдан анықталатын қорытынды (К) баға бойынша анықтайды.

$$К = ЖР * 0,6 + ҚБ * 0,4$$

Салмақтық үлестер В жыл айын университет ғылым кеңесінде бекітіледі және ЖР үшін 0,6-дан көп емес, ал ҚБ үшін 0,3-тен кем емес.

ҚЖ/КЖ комиссия алдында қорғалады. Баға көрсетілген білім мен оқытушы пікірі негізінде қойылады.

Пән бойынша қорытынды баға тәлімгердің жіберу рейтингі мен қорытынды бақылау бойынша жақсы бағалары болған жағдайда саналады. Қорытынды бақылауға себепсіз келмеу «қанағаттанбайтын» бағаға тең болады. Пән бойынша емтихан мен межелік аттестация нәтижелері сол күні не егер жазбаша емтихан күннің екінші жартысында өткізілсе келесі күні студенттерге айтылады.

Білімнің қорытынды бағасының орындылығы үшін тәлімгер межелік бақылауда (рейтинг) және қорытынды емтиханда 0-ден 100%-ге дейінгі пайызбен бағаланады.

Межелік бақылау бағасы ағымдағы баға мен межелік бақылау бағасы қосындысынан құралады.

Оқу жетітіктері, яғни білімдер, дағдылар, икемдіктері мен құзыреттері көпбалдық әріптік жүйе бойынша, сонымен бірге оның цифрлық эквиваленті мен дәстүрлі баға шкаласына сәйкес бағаланады:

Әріптік жүйе бойынша баға	Баллдардың цифрлік эквиваленті	Проценттік мазмұны	Дәстүрлі жүйе бойынша бағасы
A	4,0	95-100	Өте жақсы
A-	3,67	90-94	
B+	3,33	85-89	Жақсы
B	3,0	80-84	
B-	2,67	75-79	
C+	2,33	70-74	Қанағаттанатын
C	2,0	65-69	
C-	1,67	60-64	
D+	1,33	55-59	
D	1,0	50-54	
F	0	0-49	Қанағаттанбайтын

13. Оқытушы талаптары, саясат пен тәртіптер

Тәлімгерлердің аудиториялық сабақтарына кешігусіз қатынасу міндетті болып табылады. Сабаққа келмей қалған кезде деканатта бекітілген тәртіп бойынша өтеледі. Максимум екі рет сабақтан қалу ғана жіберіледі. Сабаққа екі рет кешігу бір рет сабаққа келмеуге теңеледі. Екі реттен көп рет сабақтан қалған кезде оқытушының студентті мәселенің әкімшілік шешіміне дейін сабаққа кіргізбеуге құқығы бар. Дәрісте бөтен студент емес адамдардың болуы рұқсат етілмейді.

Жұмыстарды белгіленген мерзімде тапсыру қажет. Барлық тапсырмаларды тапсырудың ақырғы мерзімі – емтиханға дейінгі 3 күн.

Барлық тапсырмаларды тапсырмаған, курстық жұмысты қорағмаған студенттер емтиханға жіберілмейді.

Әрбір оқу сабағының өтілген материалдары бойынша тақырыпты қайталау мен өтелу міндетті болып табылады. Оқу материалдарын игеру деңгейі тестер мен жазбаша жұмыстармен тексеріледі. студенттерді тестілеу ескертусіз өтілуі мүмкін.

Оқытушы жетекшілігімен өтетін өздік жұмысты орындауда (СӨЖМ) келесі төрт негізгі функцияны ескеру қажет.

Біріншісі – пәннің бекітілген оқу бағдарлама бойынша өткізілетін сабақтар кезінде оқытушының беретін ақпаратын студенттер активті қабылдауы керек.

Екінші функция бойынша студенттер оқытушы кеңесі бойынша оқу*әдістемелік құралдар мен әдеби оқыту көздерін қолдана отырып үй жұмысын, бақылау жұмыстарын және курсытқ жұмыстарды орындайды. Бұл кезеңде студенттен жұмыс әдістерін білу, өз-өзін ұйымдастыру, өз-өзін ұстау талап етіледі.

Үшінші функция студенттер қиын жағдайда анализ бен жүйелеу, қиыншылық себептерін түсіну мен басқа оқу әрекеттерін орындауы тиіс. Студенттер қиын сұрақтарды оқытушыдан сұрап, өз жауаптарын тексереді.

Студенттің төртінші функция оқытушыдан кеңес алудан тұрады.

14. Әдебиет тізімі

Негізгі

1) Попов И.И., Голицын О.Л. Основы алгоритмизации и программирования. Учебное пособие. – «Форум», 2010. – 432с., ил.

2) Канцедал С.А. Алгоритмизация и программирование. Учебное пособие. – «Форум», 2010. – 352с., ил.

3) Дональд Э. Кнут. Искусство программирования. Том 1. Основные алгоритмы. Учебное пособие. – Вильямс, 2012. – 720с., ил.

Қосымша

4) Гамма Э., Джонсон Р., Хелм Р. Приемы объектно-ориентированного проектирования. Паттерны проектирования. – Питер, 2012. – 368с., ил.

5) Сеницын С.В., Михайлов А.С., Хлытчев О.И. Программирование на языке высокого уровня. Учебник для ВУЗов. – Academia, 2010. – 400с., ил.

15. Мультимедиялық сүйемелдеу тізімі

1) Borland Pascal 7.0