

ЖА. 945, 26 - 018
А 303



ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ
ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Абай атындағы Қазақ ұлттық
педагогикалық университеті

АЛГОРИТМДЕУ ЖӘНЕ ПРОГРАММАЛАУ ТІЛДЕРІ



http://www.kaznu.kz

у-әдістемелік кешені

nur-print
Алматы 2012.



ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ
МИНИСТРЛІГІ

Абай атындағы Қазақ ұлттық педагогикалық университеті

АЛГОРИТМДЕУ ЖӘНЕ ПРОГРАММАЛАУ ТІЛДЕРІ

*Білім алушыларға арналған пәнінің
оқу-әдістемелік кешені*

5В070300 – Ақпараттық жүйелер

Алматы, 2012



ӘОЖ 004.4 (075)
КБЖ 32.973.202 я73
А 39

Пікір жазғандар:

М.Тынышбаев атындағы ҚазККА, Есеттеу техникасы және ақпараттық жүйелер кафедрасының меңгерушісі, т.ғ.к., доцент, Д.Р.Қуандықова
ф.-м.ғ.к., доцент К.А. Искакова

Құрастырушылар:

т.ғ.к., доцент Б.Қ. Тульбасова
п.ғ.к., доцент О.С.Ахметова
аға оқытушы А.Б. Дүйсебаева

Алгоритмдесу және программалау тілдері: Оқу-әдістемелік кешен.
– Алматы: Нур-Принт, 2012-117 б.

ISBN 9965-894-95-7

Абай атындағы Қазақ ұлттық педагогикалық университеті физика-математика факультетінің Ғылыми Кеңесінің шешімімен баспаға ұсынылған.

Хаттама № 1, « 31 » 08 2012 жыл.

ISBN 9965-894-95-7

ӘОЖ 004.4 (075)
КБЖ 32.973.202 я73
А 39

© Нур-Принт, 2012 ж.

Мазмұны

№	Құжаттаманың атауы	Беті
1	Силлабус	4
2	Дәрістер тезистері	12
3	Семинар және практика сабақтарының қысқаша сипаттамасы (жоспарлар, семинар және практика сабақтарын жүргізуге арналған тапсырмалар, СОӨЖ, СӨЖ);	72
4	Зертханалық және студиялық сабақтардың тақырыптары мен қысқаша сипаттамалары	72
5	Емтиханға дайындалуға және өзін-өзі тексеруге арналған тапсырмалар, соның ішінде тесттер	111
6	Негізгі және қосымша әдебиеттер тізімі, соның ішінде электрондық тасуыштардағы әдебиеттер	114
7	Интернет-ресурстардың тізімі	115
8	Глоссарий	115

Студенттерге арналған пән сипаттамасы

1. Пән туралы ақпарат

Пән атауы	Пән коды	Кредит саны	Курс, семестр
<i>Алгоритмдеу және программалау тілдері</i>		4	1-к, 1-семестр
Мамандық аты	Мамандық шифры	Кафедра	факультет
<i>Ақпараттық жүйелер</i>	5B070300	<i>Оқытудың ақпараттық жүйелері</i>	<i>Физика-математика</i>
Оқыту формасы - күндізгі, сырттай		Оқыту тілі – қазақша	
Пәнді өткізу уақыты және орны дүйсенбі 8.00-9.50, 408 аудитория			
<i>Лекция</i>	сабақ кестесіне сәйкес		
<i>Зертханалық сабақ</i>	сабақ кестесіне сәйкес		
<i>СӨЖ</i>	сабақ кестесіне сәйкес		
Консультация уақыты: сабақ кестесіне сәйкес			
Аралық бақылау кестесі: 7-апта, 14-апта			
Оқытушының аты-жөні, лауазымы, дәрежесі, атағы	Контактілік ақпарат (телефон, e-mail)		
<i>Тудьбасова Ботакөз Қарабекқызы, п.ғ.к., доцент</i>	жұмыс телефоны: 272-63-49		
	Кафедра меңгерушісі Сыдықов Б.Д.		

2. Пәннің қысқаша сипаттамасы

Мақсаты: Есептердің алгоритмдеу негізін, программалаудың автоматтық негіздерін, программалау тілінің классификациясын, мәліметтер типтерін және Турбо Паскаль тілінің операторларының классификациясын оқыту: ішкі программаларды қолданып программа стандартты модульдер, деректердің динамикалық құрылымын, программалық қамтамасыз етуді жобалау тәсілдерін, программалау стилін, программалаудың сапа көрсеткішін, программаны сынау мен қалыптастыру тәсілін, объектілі – бағытталған программалаудың негіздерін қолданып программа құру болып табылады.

Студенттер білуге тиіс:

- әртүрлі алгоритмдердің құрылымдық схемасын;

- талап ететін есептерге сәйкес мәліметтер құрылымын ұйымдастыруды;
- тіл құрылысын қолданып берілген программалау тілінде программа құру;
- программаны сынау және қалыптастыру, сапалы программалық құжаттарды құра білуді меңгеру.

3. Пән пререквизиттері

«Программалау технологиясы», «Информатика» т.б.

4. Пән постреквизиті

«Жоғары деңгейдегі программалау тәсілдері», «Деректер қоры жүйесі», «Алгоритмдер теориясы», «Программалау тілдері» т.б.

5. Күнтізбелік-тақырыптық жоспар.

№	Пән тақырыптары	Апта	Аудиториялық Сабақтар		СӨЖ	СӨЖ	Барлығы
			Дәріс (с.)	Пр/сем./ лаб./студ д саб (с.)			
Кіріспе							
1	«Алгоритмдеу және программалау тілдері» пәнінің мақсаты мен даму сатылары. Ақпаратты жүйелерде қолданылатын есептеу техникасының ролі. Оқу барысында компьютердің пайдалануы.	1	1	1	4	4	12
Компьютердің программалық құралдары							
2	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.	2	3	3	4	4	12

№	Пән тақырыптары	Апта	Аудиториялық Сабақтар		СӨӨ Ж	СӨ Ж	Бар лы ғы
			Дәріс (с.)	Пр/сем./ лаб./сту д саб (с.)			
	Амалдық жүйелердің құрамы. Компьютермен пайдаланушының диалогын ұйымдастыру. Программалаудың интеграцияланған жүйелері.						
Алгоритмдеу негіздері							
3	Алгоритмнің анықтамасы. Алгоритмдерді баяндаудың тәсілдері. Алгоритм схемасын жасау ережелері. Алгоритмдердің құрылымдарының түрлері. Сызықтық және тармақталған алгоритмдердің құрылымдарының баяндалуы.	3	2	2	4	4	12
	Циклдік құрылымдар. Алгоритмнің циклдік құрылымы. Ішкі циклдік құрылымның алгоритмдік баяндалуы. Техникалық есептердің қойылымының класқа бөлінуі. Есептердің типтік компоненттері: талдау, синтездеу, шешім қабылдау. Ғылыми техникалық есептердің алгоритмдерінің схемаларының мысалдары.	4	2	2	4	4	12
Негізгі процедураларға бағытталған алгоритмдік тілдерді программалау							
	Турбо Паскаль тілінің негізгі сипаттамалары. Тілдің алфавиті. Тілдің объектілерінің жазылуының ережелері. Мәліметтер типтері. Тұрақтылар. Айнымалылар. Ерекше белгілер. Өрнектер. Арифметикалық және	5	2	2	4	4	12

№	Пән тақырыптары	Апта	Аудиториялық Сабақтар		СӨӨ Ж	СӨ Ж	Бар лы ғы
			Дәріс (с.)	Пр/сем./ лаб./сту д саб (с.)			
	логикалық өрнектер. Мәліметтер құрылымы: массивтер, жындыр, жазбалар.						
	Алгоритмдік тілдік операторларын класқа бөлу. Меншіктеу операторлары. Басқару операторлары. Мәліметтерді енгізу шығаруды ұйымдастыру. Программаның құрылымы. Алгоритмнің схемасынан программаның схемасына көшу. Алгоритмдердің сызықтық құрылымын программалау.	6	2	2	4	4	12
	Тармақталған құрылымдарды программалау. Алгоритмдердің циклдік құрылымдарын программалау (сандық талдау есептері, сандық массивтерді өңдеу, массивтер компоненттерін реттеу есептері және т.б. мысалдарында) массивтерді енгізу шығаруды программалау.	7	2	2	4	4	12
	Қатарлар. Символдық мәліметтерді өңдеудің есептерін программалау. Мәліметтер құрылымында қолданатын есептерді программалаудың ерекшеліктері.	8	2	2	4	4	12
	Ішкі программалар және оларды класқа бөлу. Ішкі программаларды ұйымдастыру әдістері. Ішкі программаларды шақыру.	9	2	2	4	4	12

№	Пән тақырыптары	Апта	Аудиториялық Сабақтар		СОӨ Ж	СӨ Ж	Бар лы ғы
			Дәріс (с.)	Пр/сем./ лаб./сту д саб (с.)			
	Нақты параметрлерді тасымалдау. Жалпының жалпы облысын пайдалану. Стандартты ішкі функциялардың қоры және процедиуралары. Оларды шақырудың тәсілдері.	10	2	2	4	4	12
	Программаларды ұйымдастыру. Өртүрлі құрылымдық ұйымдасқан программалардың мысалдары. Ақпаратты компьютердің сыртқы құрылғыларында көрсету. Файлдармен жұмыс істеу.	11	2	2	4	4	12
	Мәліметтердің динамикалық құрылымы. Сілтемелер. Кезектер және ағымдармен жұмыс істеу.	12	2	2	4	4	12
	Графика. Графиктік бейнелеудің алгоритмдік негізделуі. Стандартты ішкі программалардың қоры. Стандартты ішкі программалардың қорын ұйымдастырудың ережелері. Стандартты ішкі программалардың қорын есептерін шешу үшін қолдану.	13	2	2	4	4	12
	Программаларды ұйымдастыру. Өртүрлі құрылымдық ұйымдасқан программалардың мысалдары. Құрылымдық программалау.	14	2	2	4	4	12
	Модулдік программадау. Программалық тілдерді талдау. Программалау тәсілі. Программалаудың сапалық көрсеткіші.	15	2	2	4	4	12

№	Пән тақырыптары	Апта	Аудиториялық Сабақтар		СӨЖ	СӨЖ	Бар лы ғы
			Дәріс (с.)	Пр/сем./ лаб./сту д саб (с.)			
	Барлығы:		30	30	60	60	180

6. Әдебиеттер

Негізгі әдебиеттер:

1. Борібаев Б. Программалау тілдеріне кіріспе: Жоғары оқу орындарына арналған оқулық. –Алматы: АЭСА, 2008. -376 б.
2. Құралбаев З.Қ. Алгоритмдеу және программалау тілдері. –Алматы: «ТST-сompану» баспасы, 2008. -354 б.
3. Фаронов В.В. Турбо-Паскаль 7.0. Начальный курс. Учебное пособие. –М.: Нолидж, 2004
4. Марченко А.И., Марченко Л.А. Программирование в среде Turbo Pascal 7.0. Киев: ВЕУ+, 2000
5. Федоренко Ю. Алгоритмы и программы на Turbo Pascal. Учебный курс. –СПб: Питер, 2001
6. Культин Н.Б. Turbo Pascal в задачах и примерах. –СПб: БХВ-Петербург, 2003
7. Зуев Е.А. Turbo Pascal Практическое программирование. –М.: ПРИОР, 1999
8. Гусева А.И. Учимся программировать: Pascal 7.0. Задачи и методы их решения. –М., 1999
9. Епанешников А., Епанешников В. Программирование в среде Turbo Pascal 7.0. –М.: ДИАЛОГ-МИФИ, 2000

Қосымша әдебиеттер:

10. Абрамов С.А., Гнездилова Г.Г., Капустина Е.И., Селюнов М.И. Задачи по программированию. –М.:Наука, 1988
11. Юркин А.Г. Задачник по программированию. –СПб.:Питер, 2002
12. Немнюгин С.А. Turbo Pascal: практикум. –СПб.: Питер, 2000
13. Гуденко Д., Петроченко Д. Сборник задач по программированию –СПб.:Питер, 2003
14. Климова Л.М. PASCAL 7.0 Практическое программирование. Решения типовых задач. –М.: КУДИЦ-ОБРАЗ, 2000
15. Борібаев Б. Компьютердің арифметикалық негіздері: Оқу құралы. –Алматы: Қазак университеті, 2009. -80 б.
16. Анисимов А.Е., Пупышев В.В. Сборник заданий по основам программирования. 2007. // www.intuit.ru
17. Сыдыков Б.Д., Досмайлов Т.Қ., Ануарбекова Г.Ж. Бағдарламалау. Оқу құралы. –Алматы, 2012. -202б.

7. Баға қою саясаты

Жұмыс түрі	Баға (мақ балл)	Саны	Қосындысы
Қатысуы	10 баллов	2	20
Теория	30 баллов	2	60
Практикалық жұмыс	30 баллов	2	60
Өзіндік жұмыс	30 баллов	2	60
Емтихан	100 баллов	1	100
Барлығы			300

Баға	Баллдың эквиваленттік саны	Пайыздық мазмұндама	Дәстүрлі формадағы баға
A	4,0	95-100	Өте жақсы
A-	3,67	90-94	
B+	3,33	85-89	
B	3,0	80-84	Жақсы
B-	2,67	75-79	
C+	2,33	70-74	
C	2,0	65-69	Қанағаттанарлық
C-	1,67	60-64	
D+	1,33	55-59	
D	1,0	50-54	
F	0	0-49	Қанағаттанарлықсыз

8. Оқытушының талабы.

Курстың саясаты – бұл оқытушының курсты оқыту барысында студентке қоятын талаптар жүйесі.

Төменде ұсынылатын талаптар тізімі көрсетілген:

- сабақтан кешігуге, оны жіберуге болмайды;
- сабақ үстінде ұялы телефондарды өшіру керек;
- барлық тапсырмаларды өз уақытында орындау керек;
- оқытушы ұсынған сабақ жоспарын мұқият тыңдап, оған белсенді түрде қатысу керек;
- себепті жағдаймен қатыспаған сабағын (деканаттың рұқсатымен) қайта тапсыру керек;
- өз бетінше кітапханада және үйде дайындық жасау керек;

Академиялық этика ережесі:

- тәртіп сақтау;
- тәрбиелік;
- адамшергілік;
- шыншылдық;
- жауапкершілігінің бар болуы;

Оқу топтарындағы келіспеушілік жағдайларды оқытушылар немесе тьюторлардың қатысуымен ашық түрде талқыға салу керек, шешілмеген жағдайда деканат қызметкерлері көмегімен қарастырылады.

Дәрістер тезистері

1-Дәріс. «Алгоритмдеу және программалау тілдері» пәнінің мақсаты мен даму сатылары. Аппаратты жүйелерде қолданылатын есептеу техникасының рөлі. Оқу барысында компьютердің пайдалануы.

«Алгоритмдеу және программалау» пәнінің мақсаты- есептердің алгоритмдеу негізін, программалаудың автоматтық негіздерін, программалау тілінің классификациясын, мәліметтер типтерін және Турбо Паскаль тілінің операторларының классификациясын оқыту: ішкі программаларды қолданып программа, стандартты модульдер, деректердің динамикалық құрылымын, программалық қамтамасыз студия жобалау тәсілдерін, программалау стилін, программалаудың сапа көрсеткішін, программаны сынау мен қалыптастыру тәсілін, объектілі – бағытталған программалаудың негіздерін қолданып программа құру болып табылады.

Әдеттегі сөйлесу тілі төрт негізгі элементтерден: символ, сөз, сөз тіркестері мен сөйлемдерден тұрады. Алгоритмдік тіл соған ұқсас элементтерден, тек сөзді элементар конструкция, ал сөз тіркестерін – өрнек (выражения), сөйлемдерді – операторлар дейді. Элементар конструкциялар- символдар тізбегінен, өрнек- элементар конструкциялар мен символдардан, ал оператор-өрнектен, элементар конструкция мен символдардан түзілетіндіктен, символдар, элементар конструкциялар, өрнек және операторлар иерархиялық құрылымдарды құрайды. Тілдің баяндалуы аталған төрт элементтерден баяндалады. Символдардың жазылуы тілдің пайдаланылатын символдарынан жүзеге асады. Элементар конструкциялардың жазылуы деп олардың түзілуінің заңдылығын түсінеміз. Өрнектердің жазылуы деп тілдегі мағынасы болатын кез келген өрнектің түзілуінің заңдылығын түсінеміз. Операторлардың жазылуы тілде пайдаланылатын барлық оператор типтерінің қарастырылуынан тұрады. Тілдің аталған төрт элементтерінің баяндалуы оның СИНТАКСИСІ мен СЕМАНТИКАСЫМЕН беріледі. Синтаксистік анықтама тілдің элементтерінің құрылуының заңдылығын анықтайды. Семантика синтаксистік анықтама берілген тілдің элементтерінің мағынасын және қолдану заңдылығын анықтайды. Тілдің символы бұл-тексті құру үшін пайдаланылатын белгілер. Элементар конструкциялар бұл - тілдің минималь бірлігі. Олар тілдің негізгі символдарынан құралады.

Өрнек алгоритмдік тілде мәнді есептеу заңдылығын беретін элементар конструкция мен символдардан тұрады. ОПЕРАТОР орындау қажет әрекеттерді бейнелеуді көрсетеді. Күрделі әрекеттерді бейнелеуге операторлар тобы қолданылады. Мұндай жағдайда операторлар құрама операторлар иен блоктарға біріктіріледі. Операторлармен берілген әрекеттер берінгілерден орындалады. Берілгендердің типі жөнінде мәлімет беретін алгоритмдік тілдің сөйлемі бейнеленуі (жазылуы) немесе орындалмайтын операторлар делінеді. Бейнеленуі мен операторлар алгоритмі бірігін алгоритмдік тілдегі программаны құрайды. Меңгерілетін алгоритмдік тілді терминдермен бейнелейтін тіл-метатіл делінеді.

Синтаксистік анықтама формальды немесе формальды емес тәсілдермен беріледі. Формальды тәсілдердің үш түрі бар:

- Бэкус-Науер формуласы делінетін металингвистикалық символика;

-синтаксистік Вирт диаграммалары;

-жақшалы конструкциялар.

2-Дәріс. Программалауды автоматтандырудың әдістері.

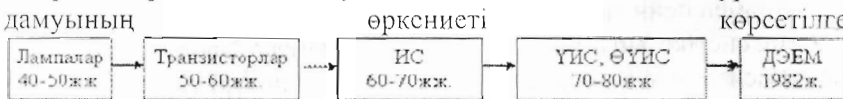
Алгоритмдік тілдер. Алгоритмдік тілдің қолдануы және оған қойылатын талаптар. Процедураға бағытталған тілдер және объектілерге бағытталған программалау туралы түсініктер. Компьютерді программалық қамтамасыз етудің жалпы түсінігі. Амалдық жүйелердің құрамы. Компьютермен пайдаланушының диалогын ұйымдастыру. Программалаудың интеграцияланған жүйелері.

Программалаудағы фраза программа-транслятор арқылы автоматты түрде машинада бегіленген кодқа айналады, сондықтан кез келген бірмәнді емес тұспал (ишара) осы фразаны аударылмайтын етеді немесе категе келтіреді.

Бэкус-Науер формуласы тілі математикалық формулаға ұқсас ықшамды формула түріндегі формада ұсынылады. Тілді түсіну үшін жалғыз метаформула бар болады. Ол оң және сол бөліктен тұрады. Сол бөлігінде анықталылатын ұғымдар, оң бөлігінде осы ұғымға біріктірілетін тілдің қолданылатын конструкциялар жиыны беріледі. Формуладағы анықталатын ұғымдарға бұрынығық жақша түріндегі арнайы метасимволдар (сол бөлікте) қолданылады немесе алдың-ала анықталған ұғым (оң бөлікте), ал сол және оң бөліктің бөлінуі.

мағынасы "анықтама бойынша бар" сөзіне эквивалентті, ":= " метасимволымен көрсетіледі.

Бағдарламалау технологиясының дамуы компьютерлік техниканың дамуымен тығыз байланысты. Алғашқы ENIAC ЭЕМ-ы 1946 жылы құрылған. ЭЕМ ауданы 100 шаршы метр бөлмеде орналасып, өнімділігі – 500 оп/сек болған. Ол АҚШ ӨТФ снарядтарының ұшу траекториясын есептеуге арналған болатын. Әрі қарай компьютерлік техниканың дамуы ЭЕМ қуатын арттыру және көлемін азайту жолымен дамыды. ЭЕМ дамуының өркениеті микроэлектрониканың дамуымен тығыз байланысты. 1-суретте ЭЕМ дамуының өркениеті көрсетілген.



1.1-сурет-ЭЕМ дамуының өркениеті ЭЕМ-н бір буынынан басқасына көшуі элементтік базаның жақсаруымен ғана сипатталмайды, сонымен қатар ЭЕМ-н құрылымының өзгеруімен, олардың функционалды-техникалық мүмкіндіктерінің және пайдаланушылық сипаттамалардың кеңеюімен сипатталады. Бірінші буын ЭЕМ электрондық лампалар негізінде құрылды. Сыртқы ЕСК ретінде магниттік барабандар қолданылды. Өнімділігі мен сенімділігі төмен болды. Алғашқы ЭЕМ-лар әр ЭЕМ-на арнайы машиналық командаларды ғана түсінді (тілдердің I буыны). Барлық командалар цифр түрінде келтірілді және оларды оқу мүлдем мүмкін емес. ЭЕМ командаларының жүйелері және форматтарын білу талап болды. Мысалы, екіадресті команданың форматы:

Операция коды	Адрес 1	Адрес 2
---------------	---------	---------

ЭЕМ «МИНСК – 22» үшін бағдарлама жолы мынандай түрде болады

код I адрес II адрес 10 2000 2001 10-ға тең операция коды 2000 ұяшығының мәнін 2001 ұяшығының мәнімен қосып, нәтижесін бірінші адрес бойынша орналастыру керектігін көрсетеді. Мұндай технологиядағы есептің алгоритмі элементарлы іс-әрекеттерге бөлінді, бағдарламалаушы қай ұяшықта не барын есте сақтауы керек болды. Мұндай технология өте ауыр болды және бағдарламалаушының жоғары маман болғанын қажет етті. Екінші буын ЭЕМ жартылай өткізгіш құрылғылар (диодтар, транзисторлар) негізінде құрылды. Құрылымдық негіз ретінде баспалық платалар қолданылды.

Сыртқы ЕСК магниттік таспаларда орындалды. Командалардың сандық коды әріптік кодқа ауысатын, ал ұяшық адрестері әріптік немесе шартты болатын ASSEMBLER (тілдердің 2-буыны) тілі пайда болды. Алдыңғы мысалдың жазбасы мынадай болды:

ADD a b

Ассемблердің басты артықшылығы—программалаушы ЭЕМ командаларының жүйесін білу және ЭЕМ жадысын ұяшық нөмірін көрсетіп үлестіру талап емес. Ассемблер тілі адресстер мен амалдардың мнемоникалық кодтарының символдық белгіленуін қолданады. Мысалы—`ADD`; `MOV` — қайта жіберу. ЭЕМ машиналық кодты ғана түсінетіндіктен, ассемблерден транслятор пайда болды. ЭЕМ-ға бағдарлама енгізгеннен кейін транслятор Ассемблер тілін ЭЕМ-ға түсінікті машиналық кодтарға аударды. Ассемблер пайда болғаннан кейін-ақ жұмыстың ауырлығы төмендеген жоқ, есептеулерде қажеттіліктердің керекті көлемін қамти алмады. Үшінші буын ЭЕМ-да құрылымдық негіз ретінде интегралдық схемалар (ИС) қолданылды. Құрылысында модульдік принцип қолданылған. Сыртқы ЕСК ретінде магниттік таспалардағы және қатты дисктердегі жинағыншалар болды. Көлемі мен энергияны пайдалану азайғаннан кейін өнімділік пен сенімділік деңгейі артты. Үшінші буын ЭЕМ-нің сипаттамалық өкілдері болып ЕС ЭЕМ және СМ ЭЕМ табылады. ЭЕМ базалық бағдарламалық қамтамасыздандырудың болуымен және бірнеше бағдарламалардың бір уақытта орындалу мүмкіндігімен сипатталады. ЕС ЭЕМ ғылыми-техникалық, экономикалық және ақпараттық-басқарушы мәселелерді шешуге арналған әмбебап ЭЕМ тобының өкілі болып табылады. СМ ЭЕМ — технологиялық процестерді автоматты басқару жүйелерінде (ТПАБЖ) қолдануға арналған проблемалық-бағытталған мини-ЭЕМ тобына жатады. Бағдарламалаудағы қиыншылықтарды азайтуға бағытталған келесі қадам — жоғары деңгей тілдерінің пайда болуы. 1955 ж. алғашқы жоғары деңгей тілі **FORTRAN** (FORMULA TRANSLATOR) — ғылыми есептеулер тілі пайда болды. 1960 ж. **Алгол** (Algorithmic Language) пайда болды.

1965 ж. **Basic** (Beginner's All purpose Symbolic Instructions Code) пайда болды. Авторлары Курт және Кемени. 1970 ж. Никлаус Вирт студенттерге бағдарламалауды үйретуге арналған Паскаль тілін шығарды.

1972 ж. **Си** тілі пайда болды (авторы Денис Ричи). 1980 ж. 3-ші буындағы ең күшті Ада тілі пайда болды.

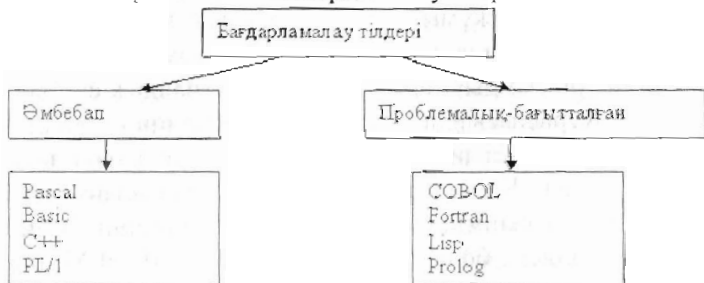
3-буын бағдарламалау жүйелері қолданушыға бағдарламалар жасаудағы қуатты және қолайлы құралдарды ұсынады:

- компилятор және интерпретатор
- редактор
- кітапханалар
- қосымша бағдарламалар (жөндеушілер, қателерді, айқастырылған сілтемелерді табатын бағдарламалар).

3-ші буын жоғары деңгей тілдері екі үлкен класка бөлінеді (2-сурет):

1. Әмбебап;
2. Проблемалық-бағытталған.

Біріншілері мәселелердің кез-келгенін шеше алаа, екіншілері – арнайы мәселелерді шешуге арналған.



1.2-сурет – Жоғары деңгей тілдерінің классификациясы

Жоғары деңгей тілдерінің пайда болуымен бағдарламаларды құру жылдамдығы жоғарылады, бағдарлама құрудағы уақыттың 80%-н алатын түзету процесін автоматтандыру мүмкіндігі пайда болды. 3-ші буын жоғары деңгей тілдерінде қосу командасының жазылу мысалдарын қарастыратын болсақ. Мысалы: Basic тілінде қосу командасы былай жазылады: $Y = a + b$. Ал Pascal тілінде қосу командасы былай жазылады: $Y := a + b$;

3-Дәріс. Алгоритмнің анықтамасы. Алгоритмдерді баяндаудың тәсілдері. Алгоритм схемасын жасау ережелері. Алгоритмдердің құрылымдарының түрлері. Сызықтық және тармақталған алгоритмдердің құрылымдарының баяндалуы.

Алгоритмдеу – есепті шығару алгоритмін құрастыру процесі, мұның нәтижесінде мәліметтерді өңдеу процесінің кезеңдері айқындалады да, кезеңдер мазмұны формальды (жасанды) түрде

жазылып, солардың орындалу реттілігі анықталады. *Алгоритм* – бастапқы берілген мәліметтерден қажетті нәтижеге қол жеткізу жолында атқарылатын есептеу процесін анықтайтын дәлме-дәл нұсқаулар жиыны.

Алгоритм қасиеттері:

1. детерминділік (анықтылық, бір мәнділік) – басқаша түсінуге жол бермей, тек қана көрсетілген әрекеттерді айқын түрде орындауға арналған нұсқаулар дәлдігі;

2. дискреттілік – есептеу процесін жекеленген қарапайым операцияларға бөлу қасиетінің болуы, яғни күрделі есепті атқарылуына күдік келтіруге болмайтын шағын бөліктерге жіктеу мүмкіндігінің болуы;

3. нәтижелілік – белгілі бір әрекеттер саны атқарылған соң, процестің қажетті нәтижесін алып, оны аяқтау мүмкіндігінің болуы немесе есептеу процесін ары қарай жалғастыруға болмайтындығы жайлы мәлімет алу;

4. жалпылық – алгоритмнің осы сияқты көптеген басқа да есептерге қолданылу мүмкіндігінің болуы

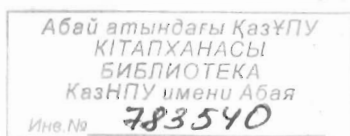
Алгоритмдік тіл – алгоритмдерді жазуға арналған символдар мен сол символдардан тұратын конструкцияларды құрастыру және түсіндіру ережелерінің жиыны. **Программалау тілі** компьютерлерде программаларды орындау ісін атқарады. **Программа** – машинаға түсінікті түрде жазылған алгоритм. Программала берілген мәліметтердің сипаттамаларымен бірге оларды өңдейтін командалар болады. Командалар қандай мәліметтер қандай операцияларға қатынасатынын, олар қандай реттілікпен орындалатынын және нәтиженің қандай түрде шығарылатынын көрсетеді. Бұлар операторлар арқылы жүзеге асырылады. Мәліметтер– белгілі бір процесс көмегімен тасымалдап, өндеуге болатын, формальды түрде бейнеленген фактілер мен идеялар. **Оператор** – операциялар мен мәндерді көрсететін немесе солардың элементтерінің қай жерде орналасқанын білдіретін символдар жиыны. Мысалы:

$a = b+c$; // a, b, c – айнымалылар;

$k= 2$;

if ($t<0$) ...

Айнымалы – программа орындалуы барысында өз мәнін өзгерте алатын объект.



Айнымалы қасиеттері:

1. Айнымалы белгілі бір мәнге ие болмағанша, анықталмаған болып саналады. Оған мәні беру мынадай тәсілдермен орындалады:

- сырттан енгізу арқылы;
- константаны меншіктеу арқылы;
- бұрын анықталған айнымалының мәнін беру арқылы;

2. Кезкелген сәтте айнымалының белгілі бір мәні болады немесе ол анықталмаған болып есептеледі:

3. Айнымалыға соңғы берілген мән оның алдыңғы мәнін жойып (өшіріп) жібереді. Айнымалыны таңдау (оқу) және оны пайдалану айнымалының мәнін өзгертпейді. Бұл пәннің заттық негізі болып (компьютерде шығару мақсатында) алгоритмдер мен программаларды құрастыру тәсілдері мен құралдары саналады. Программалар құру үшін программалау жүйелері пайдаланылады.

Программалау жүйесі – программалауды автоматтандыру құралдары. Олар программалау тілінен, осы тілдің трансляторынан, құжаттамаларынан және де программаларды дайындау, әрі орындау құралдарынан тұрады.

Транслятор – бір тілді екінші тілге аудару программасы. Ол интерпретатор және компилятор сияқты екі топқа бөлінеді.

Интерпретатор – бұл командаларды аударып, оларды бірден орындауға арналған трансляторлық программа.

Компилятор – бұл алгоритмдік тілдің конструкцияларын толығымен машиналық кодқа түрлендіретін программа. Есептің нәтижесін алу үшін машиналық кодты орындау керек.

Алгоритмдерді бейнелеу тәсілдері

Алгоритмдерді бейнелеудің негізгі тәсілдеріне оларды жазудың келесідей түрлері жатады:

- табиғи тіл сөздері арқылы;
- формулалық-сөздік тәсіл арқылы;
- графикалық түрде бейнелейтін блок-схемалар арқылы;
- псевдокодтар арқылы;
- программалау тілі арқылы.

Алгоритмдерді **табиғи тіл сөздері арқылы** бейнелеуде – есептеу кезеңдері мазмұны кез келген түрде табиғи тілде жазылады. Бірақ бұл тәсілде алгоритм көрнекілігі жоқ және дәлдік, яғни детерминділік қасиет, толық формальдау мүмкіндігі сақталмайды, сол себепті ол сирек қолданылады.

Алгоритмдерді формулалық-сөздік тәсіл арқылы бейнеленуі-тапсырманың математикалық символдар мен өрнектердің және сөздердің араласуымен берілуі болып табылады. Мысалы, үшбұрыш ауданын оның үш қабырғасының ұзындығы арқылы есептеу алгоритмін құру керек болсын делік.

1. үшбұрыштың жарты периметрін есептеу

$$p = (a + b + c) / 2$$

2. үшбұрыштың ауданын есептеу

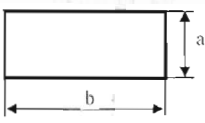
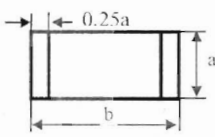
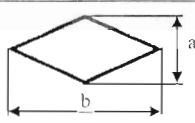
$$S = \sqrt{p(p-a)(p-b)(p-c)}$$

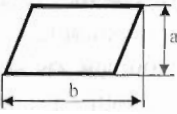
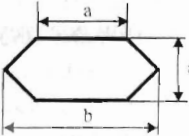
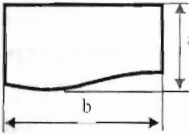
3. нәтиже ретінде S мәнін шығарып, алгоритм жұмысын аяқтау.

Бұл тәсілді пайдаланғанда, алгоритмді кез келген деңгейде айқындап көрсетуге болады, бірақ формальды түрде анық бейнелеу қиын. Алгоритмді **графикалық түрде блок-схемалар арқылы** көрсету – оның логикалық құрылымын графикалық түрде бейнелеу болып саналады. Мұнда мәліметтерді өндеудің әрбір кезеңі атқарылатын операцияға сәйкес әр түрлі геометриялық фигуралар (блоктар) түрінде көрсетіледі (1.1-кесте).

1.1 кесте

Алгоритмдерді бейнелеу блоктары

Іс-әрекет аты	Блоктың пішімі	Атқаратын жұмысы
Процесс		Математикалық өрнектерді есептеу
Басы – соны		Алгоритмдерді бастау, аяқтау
Алдын ала анықталған процесс (подпрограмма)		Қосалқы программаларға кіру және шығу
Шешім		Есеп шығару жолын таңдау

Енгізу-шығару		Мәліметтерді енгізу және шығару
Модификация		Цикл басы
Құжат		Нәтижені баспаға (қағазға) шығару

Жоғарыдағы мысалдардың алдыңғысы алгоритмін блоктар арқылы графикалық түрде бейнелеу 1.1-суретте көрсетілген. Ол нақты тәсіл, сол себепті кең қолданылады.

Псевдокодтар – нақты программалау тілінің синтаксистік ерекшеліктерін есепке алмай, тек формальды түрде программа логикасын бейнелеуге мүмкіндік береді. Бұл тәсіл программалау тілінің операторлары мен табиғи тіл сөздерінің араласуы арқылы құрастырылып, блок-схема орнына пайдалануға болатын программа логикасын бейнелеу құралы болып табылады.

Алдыңғы алгоритмді псевдокод арқылы былай жазуға болады:

```

Бірінші элементті таңдаймыз ( $i := 1$ )
IF  $A > x_i$  немесе  $x_i > B$  THEN
    хабарлама шығару және соңына көшу
ELSE
    келесі элементке көшу ( $i := i + 1$ )

IF жным біткен жок ( $i \leq N$ ) THEN
    интервалды тексеруге көшу
ELSE
    интервалға элементтер толық кіретіні жайлы
    хабарлама шығару
Соңы

```

Осы алгоритм блок-схема түрінде төмендегі 1.1-суретте көрсетілген.

Алгоритмдердің негізгі канондық құрылымдары

Программалаудың ең негізгі әрі күрделі кезеңі алгоритм құру болып табылады. Программалау теориясында күрделі есептің шығару жолын, яғни программаны үш түрлі құрылымды пайдалану арқылы жазып шығуға болатындығы дәлелденген:

- сызықтық құрылым немесе операторлар тізбегі;
- тармақты құрылым немесе шартты оператор;
- қайталау немесе циклдік оператор.

1) сызықтық құрылым

A; B;

A және B әрекеттері мыналардың бірі болуы тиіс:

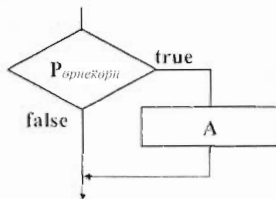
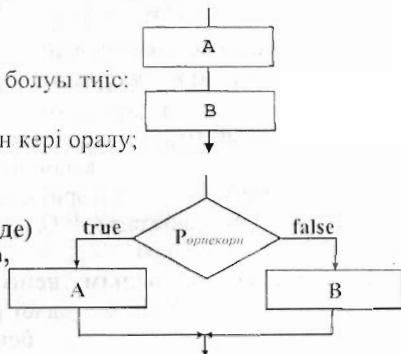
- кез келген жеке оператор;
- белгілі бір функцияны шақырып, одан кері оралу;
- басқа бір басқару құрылымы.

2) тармақты құрылым

if P then A else B; (Паскаль тілінде)

if P A else B; (C/C++, C#, Java, PHP тілдерінде)

P-ны тексеру предикат болып табылады, яғни ол мәні ақиқат (true) немесе жалған (false) бола алатын логикалық өрнектен немесе шарттан тұратын функция түрінде болады. Бұл құрылым толымсыз (қысқаша) түрде болуы мүмкін, онда логикалық өрнектің мәні жалған болғанда ешқандай әрекет орындалмайды. Мұндай құрылым түрі төменде көрсетілген:



if P then A; (Бейсик, Паскаль тілдерінде)

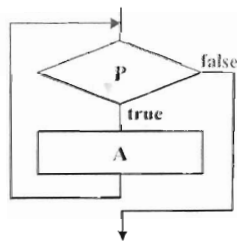
if P A; (C/C++, C#, Java, PHP тілдерінде)

3) циклдік құрылым (қайталау)

цикл – әзірше

while P do A;

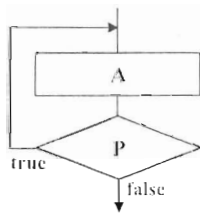
Мұнда A әрекеті P предикат мәні ақиқат болып тұрса, қайталана береді. Сондықтан A әрекеті орындалуы кезінде P-ға әсер ететін айнымалылар мәні өзгеруі тиіс. Бұлай болмаған жағдайда шексіз цикл орын алады. Предикат мәні A әрекетіне дейін анықталады, сол себепті кейде A әрекеті бір де бір рет орындалмауы да мүмкін.



цикл – дейін

repeat A until P; (Бейсик, Паскаль тілдерінде)

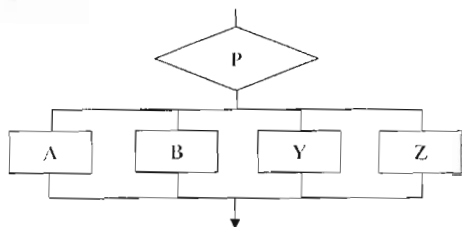
do A while P; (C/C++, C#, Java, PHP тілдерінде)



Мұндағы қайталау кем дегенде бір рет орындалады, өйткені шарт A әрекетінен кейін тексеріледі. A әрекеті предикат мәні жалған болған кезде орындалмайтын болады.

4) таңдау – case (switch) ауыстырғыш (қол тармақты) құрылымы программалауды жеңілдететін мүмкіндік болып табылады. Таңдау құрылымы бірнеше мүмкіндіктердің біреуін ғана орындау кезінде қолайлы болып табылады.

P мәніне қарай A, B, \dots, Z әрекеттерінің бірі орындалады, сонан соң кейінгі орналасқан құрылымдар атқарылады.



Күрделі алгоритмдерді құру үшін қарапайым канондық (бірыңғайланған) алгоритмдік құрылымдар қолданылады. Олар сызықтық, тармақталу және цикл құрылымдарынан тұрады.

1. **Сызықтық құрылым** немесе қарапайым сызықтық алгоритм іс-әрекеттердің орындалу ретіне қарай тізбектеле орналасқан блоктардан тұрады. Мысалы, $y = a + b$ формуласы бойынша есептеу тіктөртбұрыш арқылы кескінделетін есептеу блогы (3-блок) арқылы өрнектеліп, нәтижені қағазға басу үшін көпбұрышты құжат алу блогын (4-блок) пайдаланып, оның ішіне нәтиженің атауларын жазамыз. Осы көрсетілген $y = a + b$ формуласын есептеу үшін a және b -ның сандық мәндерін программаға енгізіп (2-блок), содан кейін қосу амалын орындап, ақырында y -ті экранға (қағазға) басын шығарып, жұмысты тоқтатамыз. Осы алгоритмнің схемасы 1.2-суретте

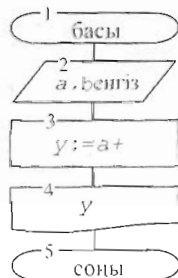
Паскальдағы программа:

```
Program P(input,output);
var a,b,y: integer;
begin
  write('a, b=');
  readln(a,b);
  y:= a+b;

  writeln('y=',y:6:3);
  readln;
end.
```

Си тіліндегі программа:

```
#include <stdio.h>
#include <conio.h>
main()
{
  int a,b,y;
  printf("a, b =");
  scanf("%i%i", &a, &b);
  y = a + b;
  printf("y = %i", y);
  getch();
}
```



1.2-сурет.
Алгоритм

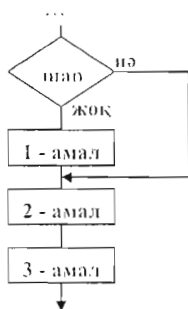
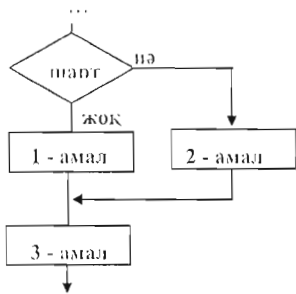
көрсетілген, ал оның жанында Паскаль және Си тіліндегі программасы жазылған.

2. **Тармақталу алгоритмдері** құрылымы қарапайым болып келеді. Мұнда арифметикалық теңсіздік (теңдік) түрінде берілген логикалық шарт тексеріледі.

Егер ол орындалса, онда алгоритм бір жолмен, ал орындалмаса екінші жолмен жүзеге асырылады, яғни есепті шығару жолы тармақталып екіге бөлініп кетеді. Тармақталу алгоритмдеріне шартты тексеру блогы міндетті түрде кіреді. Ол ромб түрінде кескінделіп, басқа блоктармен 1 кіру және 2 шығу сызықтары арқылы байланысады. Көбінесе тармақталу алгоритмі екі түрде кездеседі, олар "таңдау" және "аттап өту" мүмкіндіктерін іске асыруға көмектеседі.

"Таңдау" жолымен тармақталуда берілген шарт тексеріледі (1.3-сурет), егер ол шарт орындалса (ақиқат болса), онда 2-амал жүзеге асырылып, содан кейін келесі 3-амалға көшеміз. Ал, егерде шарт орындалмаса, яғни оның орындалу мүмкіндігі жалған болса, онда 1-амал атқарылып, содан кейін 3-амал атқарылады. Сонымен, шарттың ақиқат немесе жалған болуына байланысты 1-амал немесе 2-амал орындалады.

"Аттап өту" (1.4-сурет) алгоритмінде шарт орындалса, 1-амалды аттап өтіп, бірден 2-амалды, содан кейін 3-амалды орындаймыз. Ал шарт жалған болса, онда 1-амал міндетті түрде орындалып, одан кейін 2- және 3-амалдар жүзеге асырылады.

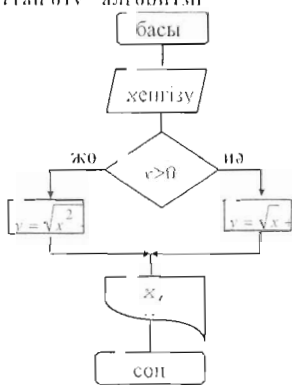


1.3-сурет. "Таңдау" алгоритмі. 1.4-сурет. "Аттап өту" алгоритмі

1-мысал. y функциясын төмендегі формула бойынша есептеп шығарайық.

$$y = \begin{cases} \sqrt{x} + x, & x \geq 0 \\ \sqrt{x^2 + 1}, & x < 0 \end{cases}$$

Мұнда x айнымалысының таңбасына (оң, теріс) байланысты не жоғарғы, не төменгі формуланы таңдап алып, сол арқылы y функциясының мәнін табамыз (1.5 сурет). 2-блоктың орындалу барысында x айнымалы-



1.5-сурет. Тармақталу алгоритмі

сына белгілі бір мән беріледі де, ол мән енгізу операторлары арқылы программаға енгізілуі тиіс. Бұдан кейін енгізілген мәннің оң немесе теріс екендігі үшінші шартты тексеру блогы арқылы айқындалады. Шарттың "ақиқат" (иә) немесе "жалған" (жоқ) болуына қарай 4- не 5-блоктардың бірі ғана орындалып, "таңдау" орындалады. 6-блок x айнымалысының және y функциясының сандық мәндерін экранға немесе қағазға басып шығарады.

Бақылау сұрақтары

1. Алгоритм және программа дегеніміз не, олардың қандай ұқсастықтары мен айырмашылықтары бар?
2. Компьютерде орындалатын алгоритмдердің қандай қасиеттері болады?
3. Алгоритмдерді өрнектеу жолдары.
4. Компьютерде есеп шығару кезеңдері.
5. Алгоритм схемаларының әр түрлі блоктары, олардың бейнеленуі, байланыстары.
6. Программалау тілдері деген не?

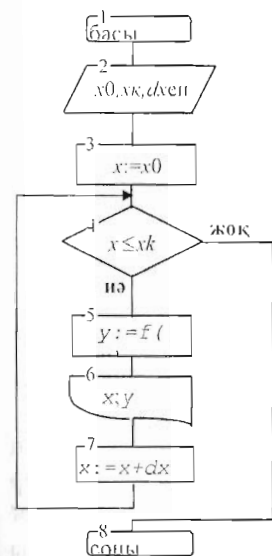
4-дәріс. Циклдік құрылымдар. Алгоритмнің циклдік құрылымы. Ішкі циклдік құрылымның алгоритмдік баяндалуы. Техникалық есептердің қойылымының класка бөлінуі. Есептердің типтік компоненттері: талдау, синтездеу, шешім қабылдау. Ғылыми техникалық есептердің алгоритмдерінің схемаларының мысалдары.

Математикада, экономикада көптеген есептерді шығару кезеңінде бір тездеуді пайдаланып, ондағы айнымалының өзгеруіне байланысты оны бірнеше рет қайталап есептеуге тура келетін сәттер де жиі кездеседі. Осындай қайталап орындалатын есептеу процессінің белгілі бір бөліктерін цикл деп атайды. Осы бірнеше рет қайталанатын бөлігі бар алгоритмдер тобы циклдік алгоритмдерге жатады. Циклдік алгоритмдерді пайдалану оларды кейіннен программаларда цикл операторы түрінде қысқартып жазу мүмкіндігін береді. Циклдер қайталану санының алдын ала белгілі және белгісіз болуына байланысты екі топқа бөлінеді. Қайталану сандары алдын ала белгілі болып келген циклдер тобы *арифметикалық цикл* болып есептеледі, ал орындалу саны белгісіз циклдер – *қадамдық (итерация) цикл* болып аталады.

Практикада белгілі бір айнымалының сандық мәніне байланысты орындалатын арифметикалық циклдер жиі кездеседі. Мұнда арифметикалық прогрессияға ұқсас болып келетін циклдер ең қарапайым арифметикалық цикл болып табылады.

Оны басқару қайталану кезеңінде прогрессияның заңына сәйкес тұрақты шамаға өзгеріп отыратын цикл параметрінің сандық мәнімен байланысты болуы тиіс.

Цикл орындалуы алдында оның айнымалы аргументі – параметрі алғашқы мәнге не болуы керек, сонан кейін қайталану кезеңінде цикл параметрі белгілі бір шамаға (қадамға) өзгере отырып, ол алдын ала берілген ең соңғы мәнге дейін жетуі қажет.



2.1-сурет.

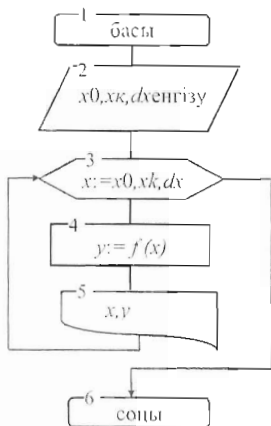
Қарапайым

мұнда $x_0 + dx, x_0 + 2dx, \dots, x_0 + (n-1)dx, x_k$, мұндағы n – циклдің қайталану саны, ол былай анықталады:

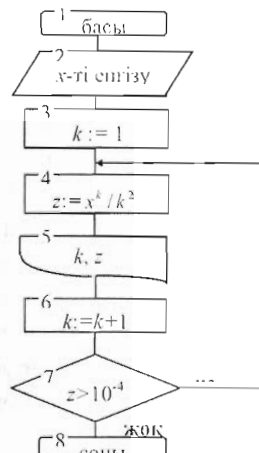
$$n = \left[\frac{x_k - x_0}{dx} \right] + 1,$$

мұнда [...] – өрнектің бүтін бөлігі алынатынын көрсетеді, n – циклдің қайталану саны әрқашанда бүтін натурал сан болуы тиіс, егер ол аралас сан болса, онда оның бөлшегі алынып тасталады. Арифметикалық цикл үшін $y=f(x)$ функциясының есептелу жолы алгоритм ретінде 2.1-суретте көрсетілген. Мұндағы 3-ші, 4-ші, 7-блоктар циклді ұйымдастыру үшін қажет. Олар цикл параметрінің алғашқы мәнін, өзгеру қаламын белгілеп және оның ең соңғы мәніне жеткен-жетпегенін тексереді. Ал 5- және 6-блоктар бірнеше рет қайталанып циклдің өзін құрайды. 4-блок шартты тексеріп қайталану процесін ұйымдастырады.

Алгоритм схемасын салуды және программаны жазуды жеңілдету үшін цикл алгоритмдері ықшамдалған түрде "модификатор" немесе "цикл басы" блогын пайдалану арқылы жазылады. Онда 1.6-суретте көрсетілген 3-ші, 4-ші, 7-блоктардың орнына "цикл басы" блогы орналасады.



2.2-сурет. Модификаторлы циклік алгоритм



2.3-сурет. Қадамдық цикл

4. Қадамдық циклдер. Циклді орындаудың алдында, оның қайталану саны белгісіз болған жағдайда қадамдық циклдер пайдаланылады. Мұнда циклді жазу үшін тек қана "шартты тексеру" блогын қолдану қажет, ол циклді аяқтау үшін белгілі бір шартты тексереді. Қадамдық циклдердің схемасын сызғанда модификаторды (алтыбұрышты) қолдана алмаймыз, себебі алдын ала циклдің неше рет қайталанатыны бізге белгісіз. Енді осындай циклдер жұмысына мысал келтірейік.

3-мысал. $z = \frac{x^k}{k^2}$ функциясының мәндерін $k = 1, 2, 3, \dots$ және $Z > 0.0001$ -ден артық болған жағдайда есептейік, мұндағы $0 \leq x \leq 1$. Бұл мысалда алдын ала цикл неше рет қайталанатынын айта алмаймыз, өйткені бізде тек k параметрінің алғашқы мәні мен қадамы ғана белгілі. Сонымен қатар Z функциясының 0.0001 -ден артық болуы циклді қайталау шарты болып есептеледі ($Z > 0.001$). 2.3-суретте осы есептің алгоритм схемасы көрсетілген.

Бақылау сұрақтары

1. Алгоритм және программа дегеніміз не, олардың қандай ұқсастықтары мен айырмашылықтары бар?
2. Компьютерде орындалатын алгоритмдердің қандай қасиеттері болады?
3. Алгоритмдерді өрнектеу жолдары.
4. Компьютерде есеп шығару кезеңдері.

5. Алгоритм схемаларының әр түрлі блоктары, олардың бейнеленуі, байланыстары.
6. Сызықтық, тармақталу және циклдік алгоритмдер.
7. Қадамдық циклдер және олардың ерекшеліктері.

5-Дәріс. Турбо Паскаль тілінің негізгі сипаттамалары. Тілдің алфавиті. Тілдің объектілерінің жазылуының ережелері.

Мәліметтер типтері. Тұрақтылар. Айнымалылар. Ерекше белгілер. Өрнектер. Арифметикалық және логикалық өрнектер.

Мәліметтер құрылымы: массивтер, жиындар, жазбалар.

1. Кез-келген тексті құру үшін пайдаланылатын элементарлық белгілерді *тілдің символы* дейді. Сондай символдардың тобын *тілдің алфавиті* дейді. Паскаль тілінің алфавиті төмендегі символдар тобын қамтиды:

- а) 26 латын әрпі: A, B, C, D, ... Z;
- ә) араб цифрлары: 0, 1, 2, ... 9;
- б) 32 орыстың баспа әріптері: A, Б, В, ... Я;
- в) арнайы символдар: *, /, +, -, ...;

2. Кейбір түсініктер.

Тілдің алфавиттерін көп жағдайларда - *тілде пайдаланылмайтын символдар* деп те атайды. Тілдің жеке символдары арқылы белгілі мағынаны білдіретін тілдің сөздері және сөйлемдері құрастырылады. Тектегі сөздерді бір-бірінен ажыратып тұру үшін бос орын немесе әр түрлі бөлгіш белгілері (жақша, қос нүкте, т.с.с.) пайдаланылады. Бір сөз бен екінші сөз арасына бір немесе бірнеше бос орын қалдыруға болады.

Орындайтын міндеттеріне байланысты сөздерді: *қызмет сөздері, атаулар және тұрақтылар* деп бөледі.

Қызмет сөздері. Мағынасы алдын ала анықталып қойылған Паскаль тілінде пайдаланылатын сөздер тобын *Паскаль тілінің қызмет сөздері* дейді. Мысалы, PROGRAM, VAR, BEGIN, END және т.б.

Атау (идентификатор). Атау бір өрнекті белгілеуге арналады. Паскаль тілінде атауды *стандартты және пайдаланушылар* атаулары деп бөледі.

Стандартты атаулар – стандартты объектілерді (стандартты функциялар, стандартты бағдарламалар т.с.с.) белгілеуге арналады. Мысалы, енгізу - READ, шығару - WRITE, (INPUT, OUTPUT) т.с.с.

Пайдаланушылар атауы ретінде әріптен басталатын кез-келген әріптер мен цифрлар тізбегі, сөздер, тіпті сөйлемдер алынады. Мысалы, жылдамдық, SUMMA, A, K, A5, AB т.с.с. Бірақ пайдаланушылардың атауы үшін стандартты атауларды пайдалануға болмайды.

Берілгендердің элементі. Паскаль тілінде берілгендердің элементіне *тұрақтылар және айнымалылар* жатады.

Паскаль тілінде тұрақтылар деп алдын-ала мәні берілетін, бағдарлама орындалу барысында мәндерін өзгертпейтін шамаларды айтады. Ал бағдарлама орындалу барысында мәндері өзгеріп отыратын шамаларды айнымалы шамалар дейді.

Түсініктеме. Бағдарламаға, оның бөліктеріне немесе жеке операторларға түсінік беру үшін түсініктеме пайдаланылады. Паскаль тілінде түсініктеме {} ішіне алынып жазылады. Мысалы, {квадрат теңдеуді жазу}.

3. Паскаль тілінде төмендегі арифметикалық амал белгілері пайдаланылады:

+ - қосу амалы;

- - алу /азайту/ амалы;

* - көбейту амалы;

/ - бөлу амалы;

DIR – бүтін бөлу;

MOD – бүтін бөлгендегі қалатын қалдық.

DIR, MOD амалдары тек бүтін сандар үшін ғана пайдаланылады.

Мысалы, A=25., B=2

A DIR B (нәтиже 12)

A MOD B (нәтиже 1)

@ - адресті алу (анықтау)

shl - солға жылжыту

shr- оңға жылжыту

in - элементтің жиынға жататындығын немесе жатпайтындығын анықтау операциясы.

Басқа программалау тілдерінен ерекшелігі Паскальда логикалық амалдар жақпаға алынып жазылады.

A=b and c<d - дұрыс емес.

Синтаксистік қате пайда болмас үшін бұл қатынастарды былай жазған дұрыс.

(a=b) and (c<d).

Логикалық типке келесі стандартты логикалық функциялар қолданылады:

ODD(x)

EOLN(x)

EOF(x).

Егер x бүтін сан болса $ODD(x)$ логикалық функциясының мәні $=1$ (TRUE), қарама-қарсы жағдайда $=0$ (FALSE).

Егер x мәтіні файл жолының соңын білдіретін символ болса, онда $EOLN(x)=1$ (TRUE) ақиқат мәнін береді. Егер x кез келген файлдың соңын белгілейтін символ болмаған жағдайда $EOF(x)$ логикалық функциясының мәні (FALSE) жалған болмақ.

Арифметикалық амал белгілері және жақшалар арқылы жалғастырылған сандарды және функцияларды *өрнектер* дейді. Паскаль тілінде арифметикалық өрнектердің мәндері есептелгенде кәдімгі математикадағыдай амалдар өз орындалу тәртібі бойынша орындалады. Егер өрнекте жақша болса, онда алдымен жақша ішіндегі өрнектің мәні табылады.

4. Паскаль тілінде тұрақтылар, айнымалылар, функциялар, өрнектер пайдаланылады.

Берілгендердің типтері: *қарапайым* және *күрделі* болып екіге бөлінеді.

Қарапайымға *стандартты* және *айнымалы* типтер жатады.

Стандартты типтер: INTEGER (бүтін), REAL (нақты), BOOLEAN (логикалық), CHAR (символдық).

Айнымалы типтерді ЭЕМ-ді пайдаланушылар өздері анықтайды. Оларға *шектелген* және *көрсетілетін* типтер жатады.

Әр түрлі қарапайым типтердің комбинацияларынан құралған типтер *күрделі* типтерді береді (мысалы, массивтер, жиымдар, жазулар және файлдар типтері).

Бүтін типтегі тұрақтылар – кез-келген нүктесіз жазылған бүтін сан. Олар теріс немесе оң таңбалы болуы мүмкін. Диапазоны: **-32768** ден **32767** дейін. Типі бүтін болатын айнымалыны көрсету үшін INTEGER қызмет сөзі пайдаланылады. Айнымалыларды белгілеу үшін оларға *атау* береді. Паскаль тілінде атау ретінде әріптерден басталатын, әріптер мен цифрлардан құрылған символдар тізбегі пайдаланылады, мысалы, A, B, ..., X2, CU, ... Бүтін сандарға мына амалдар қолданылады: +, -, *, /, DIV, MOD.

Нақты тұрақты. Паскаль тілінде нақты тұрақтылар екі түрде жазылады:

- сандардың үтірлері өзгермейтін түрі – $\pm m$; Мысалы: 25.3, 5.0 т.б.
- сандардың үтірлері өзгертін түрде жазылуы – $\pm mE+D$

Мысалдар:

математикалық берілуі

Паскаль тілінде

жазылуы

$4 \cdot 10^{-3}$

4E - 3

$0,62 \cdot 10^4$

0.62E+4 (0.62E4)

$-10,52 \cdot 10^{12}$

-10.52E12 (-

10.52E+12)

Нақты айнымалы. Нақты мән қабылдайтын айнымалыны бейнелеу үшін REAL қызмет сөзі пайдаланылады. Бүтін және нақты типтерді арифметикалық тип дейді. Нақты типтегі шамаларға базалық арифметикалық амалдарды орындауға болады.

Логикалық тип.

Логикалық айнымалы: тип BOOLEAN, TRUE (ақиқат) немесе FALSE (жалған) тұрақтыларының біріне қабылдайды. Логикалық берілгендер шарт тексеруге жиі пайдаланылады, логикалық шарттарды жазу үшін <, <=, =, >, >=, > белгілері пайдаланылады. Мысалы, $5 > 3$ нәтиже TRUE; $5 = 3$ нәтиже FALSE.

Логикалық берілгендерге мына амалды қолдануға болады:

OR – логикалық қосу (немесе);

AND – логикалық көбейту (және);

NOT – жоққа шығару (жоқ).

Символдық типтер. Символдық тұрақтылар екі апостроф ішіне алынып жазылады. Мысалы, 'A', '+', ...

Символдық айнымалылар. Символдық айнымалылардың типін көрсету үшін Паскаль тілінде CHAR қызмет сөзі пайдаланылады.

5. Паскаль тілінде стандартты функциялардың мәндері автоматты түрде есептеледі. Олардың мәнін есептеу үшін функцияны дұрыс жазса болғаны.

Стандартты функцияларды жазу ережелері төмендегідей:

- а) функция аты латын әрпімен жазылады;
- ә) функция атауы 6 әріптен аспауы қажет;
- б) функцияның аргументі функция атауынан соң дөңгелек жақша ішіне алынып жазылады;
- в) функция аргументі тұрақты, айнымалы типті аргументтің өрнек түрінде терілуі де мүмкін;

г) тригонометриялық функцияларда аргумент мәні радиан өлшемімен берілуі қажет.

Егер α бұрыштың мәні градус өлшемімен берілсе, онда оны алдын-ала мына формуланы $\beta = \alpha \cdot \pi / 180^\circ$ пайдаланып радиан өлшеміне аударып содан соң тригонометриялық функциялардың мәндерін есептейді.

Функция	Орындайтын міндеті	Аргумент типі	Функция типі
ABS(x)	x мәнін табу	нақты бүтін	нақты бүтін
SQR(x)	x^2 мәнін табу	нақты бүтін	нақты бүтін
Sin(x)	sinx есептеу	нақты бүтін	нақты
Cos(x)	cosx есептеу	нақты бүтін	нақты
ARCTAN(x)	arctgx есептеу	нақты бүтін	нақты
EXP(x)	e^x есептеу	нақты бүтін	нақты
SQRT(x)	\sqrt{x} табу	нақты бүтін	нақты
LN(x)	ln x табу	нақты бүтін	нақты
ADIBB	A-ны B-ға бүтін бөлу	бүтін	бүтін
AMDB	A-ны B-ға бөлгендегі қалдық	бүтін	бүтін
TRUNC(x) INT(x)	x-тің бүтін бөлігін табу	нақты бүтін	бүтін
ROUND(x)	x санын жақын бүтінге дейін дөңгелектеу	нақты бүтін	бүтін
ODD(x)	TRUE, егер x-так FALSE, егер x-жұп	бүтін	логикалық
ORD(x)	x элементінің реттік нөмірін табу ORD(x)=x x символының реттік нөмірін табу	логикалық бүтін символдық	бүтін бүтін бүтін

CHR(x)	реттік нөмірі (коды) бойынша символды анықтау	бүтін	символдық
SUCC(x)	табылған элементтен кейінгі элементті анықтау	бүтін логикалық символдық	бүтін логикалық символдық
PRED(x)	табылған элементтің алдындағы элементті табу	есептелетін	есептелетін

Паскаль тілінде дөңгелектеуге арналған екі функция бар, олар TRUNC(x) (trunk - деген ағылшынның *truncate* - кию, қысқарту деген сөзінен қысқартылып алынған) және ROUND(x) (*round* – дөңгелек). TRUNC(x) - функциясы x-тің “бөлшексіз” цифрын береді, мысалы, trunk (3.14)=3, trunc (-3.14) = -3 т.с.с.

ROUND(x) – функциясы x-ті дөңгелектеу ережесі бойынша дөңгелектейді, яғни: ROUND(x)={ trunc(x+0.5), егер $x \geq 0$,

trunc(x-0.5), егер $x < 0$,

мысалы, ROUND (3.14)=3, ROUND(3.7)=4

a^x , EXP (x*LN(A));

Сонымен Паскаль тілінде бағдарламаның жалпы құрылымын былай көрсетуге болады:

PROGRAM аты

бейнелеу
бөлігі

BEGIN

операторлар бөлігі

END.

Сонымен бағдарлама түрі былай болады:

PROGRAM аты(INPUT, OUTPUT);

LABEL – белгі бөлігі;

CONST – тұрақтылар бөлігі;

TYPE – тип бөлігі;

VAR – айнымалы бөлігі;

PROCEDURE, FUNCTION – процедура және функция бөлігі,

BEGIN

l - оператор;

n-оператор;

END.

Бейнелеудің жеке бөліктері

Белгі. Белгі ретінде кез-келген бүтін оң сан пайдаланылады. Белгі операторды немесе бағдарламаның бөлігін табуға пайдаланылады. Белгі оператор алдында орналасады да “;” арқылы ажыратылады. Қажетті операторға (бағдарламаның бөлігіне) оралу белгі арқылы жүзеге асырылады. Белгіні пайдалану үшін ол алдын-ала белгі бөлігінде LABEL (белгі) қызмет сөз арқылы хабарлануы (бейнеленуі) қажет.

Белгіні хабарлаудың жазылу үлгісі:

LABEL сан, сан, ... , сан;	Мысалы: LABEL 5,13,45;
----------------------------------	------------------------

Стандартты типтегі тұрақтылар мен айнымалыларды бейнелеу.

1) Тұрақтылар.

Бағдарламада тұрақтылар өз түрімен (ашық) немесе атауы арқылы берілуі мүмкін. Егер тұрақтылар атауы арқылы берілсе, онда оны бағдарламаның тұрақтылар бөлігінде міндетті түрде бейнелеу қажет.

Жазылу үлгісі:

CONST тұрақты атауы = мәні;

Мысалы: CONST N=28;

Бір бөлімде бірнеше тұрақты жазуға болады, бірақ олар “;” ажыратылып жазылуы тиіс. Мысалы: CONST

K = 23; {бүтін тип}

V = 1.8E-3; {нақты тип }

P = 3.14; {нақты тип }

S = 'R'; {символдық тип }

L = TRUE; {логикалық тип }

Жолдық тұрақты - программаның бір жолына жазылған және тырнақшаға (апостроф) алынған символдар тізбегі болып табылады. Жолдық тұрақтылардың мысалдары: 'Text', 'A', # 61, #S A2.^ G.

Логикалық тұрақтылар «Ақиқат» және «Жалған» деген түсініктерге сәйкес TRUE және FALSE идентификаторларымен өрнектеледі.

2) Айнымалылар

Бағдарламаларда пайдаланылатын кез-келген айнымалы бағдарламаның айнымалылар бөлігінде бейнеленуі қажет.

Айнымалыларда бейнелеу қызмет сөзі VAR (айнымалы) сөзінен басталады.

Жазылу үлгісі:

VAR айнымалы атауы : тип;

Мысалы, VAR K: integer; {K – айнымалысының типі бүтін}.

Бір мезгілде бірнеше айнымалыны бейнелеуге болады. Мысылы:

VAR

A: INTEGER; {A – бүтін типтегі айнымалы}

S: REAL; {S – нақты типтегі айнымалы}

S2: CHAR; {S2 – символдық типтегі айнымалы}

L: BOOLEAN; {L – логикалық типтегі айнымалы}.

Егер бірнеше айнымалының типтері бірдей болса, онда оларды бір жолға үтір арқылы бөліп жазып типін бір-ақ рет көрсетсе де болады.

VAR

M, N: integer; {M, N – типі бүтін}

SUM, K, L: real; {SUM, K, L – типі нақты}

R, RI: CHAR; {R, RI – типі символдық}

Сонымен, *айнымалы деген* – атау берілген әр түрлі мән қабылдайтын объект.

Типтерді [TYPE], функцияларды [FUNCTION] және процедураларды [PROCEDURE] бейнелеуге төменде арнайы тоқталамыз.

3) Операторлар бөлігінің жазылуы.

Операторлар бөлігі операторлық жақша BEGIN (бастау) және END (бітіру) ішіне алынып жазылады және END қызмет сөзінен соң “;” қойылады.

Операторлық бөлікте біртіндеп орындалатын операторлар тізбегі жазылады. Әр операторлар не орындалатынын көрсетеді. Орындалатын операторлар бір-бірімен “;” арқылы ажыратылады, ал соңғы оператордың соңына еш нәрсе қойылмайды. Мысалы:

BEGIN

1-оператор;

2-оператор;

n-1-оператор;

n-оператор

END.

Бір жолға бір немесе бірнеше оператор жазуға болады, бірақ олар “;” арқылы бір-бірінен ажыратылады. Бір жолға сыймаған операторды екінші жолға жалғастырып жазуға болады.

Бір-ақ бағдарламаны оқуға қолайлы етіп жазған дұрыс, бұл жағдайда оны түзету, редакциялау ыңғайлы болады.

Паскаль тіліндегі программаның жалпы құрылымын келесі түрде көрсетуге болады:

```
Program программаның аты
Uses модульдер
Label белгілер;
Const тұрақтыларды хабарлау;
Type деректер типін сипаттау;
Var айнымалыларды хабарлау;
Процедураларды (Procedure) және функцияларды
(Function) сипаттау.
Begin
    Программаның денесі
End.
```

Типтердің үлкен үш класы бар: жай типтер, құрылымды типтер (массивтер, жазбалар, жиындар, файлдар); көрсеткіш (указатель) тип. Типті хабарлау **TYPE** кілттік сөзінен, сипатталып отырған типтің идентификаторынан, = (тең) белгісінен, типтің анықтамасынан және ; (нүктелі үтір) белгісінен тұрады.

Мысалы:

```
Type
Number = Integer;
Color = (Red, Green, Blue);
CharVal = Ord (' A ' ) ... Ord (' Z ' );
TestIndex = 1 ... 100;
TestValue = -99 ... 99;
TestList = Array [TestIndex] Of TestValue;
TestListPtr = ^ Testlist;
Date = Record
    Year: Integer;
    Month : 1 ... 12;
    Day : 1 ... 31;
End;
```

Түгенделетін тип. Паскаль тілінде тек бір ғана стандартты түгенделетін тип бар. Ол – логикалық тип.

Программаға программист өзіне қолайлы түгенделетін типті енгізуіне болады. Түгенделетін тип реттелген идентификаторлар жиынында анықталады.

Мәндердің кез келген стандартты емес типі программада типті баяндау бөлімінде анықталуы қажет. Бұл тип (түгенделетін) типтер бөлімінде былай баяндалады.

TYPE NT = (w1,w2...wn).

Мұндағы NT - анықталатын типтің идентификаторы.

w1,w2...wn - тұрақты идентификаторлар, NT - типті айнымалы қабылдайтын нақтылы мәндер. Бұл мәндер реттелген, реттелу тәртібі типті баяндайтын идентификаторлардың мынадай орналасуымен анықталады:

$w1 < w2 < \dots < wn$

Мысалы,

Түре Апта= (дүйсенбі, сейсенбі, сәрсенбі, бейсенбі, жұма, сенбі, жексенбі);

Бұл Мысалда Апта типі

(дүйсенбі, сейсенбі, сәрсенбі, бейсенбі, жұма, сенбі, жексенбі) тұрақты идентификаторлар жиыны екені анықталған.

Егер салыстыру операндаларының айнымалылар типтері бірдей болса, онда барлық түгенделетін типтерге келесі салыстыру амалдарын қолдануға болады:

$=, <, >, <=, >=$

Барлық скалярлы типті айнымалыларға (Real типінен басқа) келесі стандартты функцияларды қолдануға болады:

SUCC(x) - реттелген тізбекте x-ке жалғасатын келесі элементті анықтайды.

Pred(x) - реттелген тізбекте x-тің алдындағы элементті анықтайды.

ORD(x) - реттелген тізбекте x-элементінің нөмірін анықтайды.

Мысалы,

SUCC(B)=C;

SUCC(M)=N;

SUCC (дүйсенбі) = сейсенбі

PRED(M) = I.;

PRED(жұма) = бейсенбі;

Uprise('n')='N.' **Тек** латын әріптеріне қолданылады, кіші әріпті үлкен әріпке түрлендіру функциясы. **Шектелген тип.** Паскальда типтерді баяндаған кезде программа талабына орай белгілі бір деректердің белгілі бір аралығын қамтып отыру мүмкіндігі қарастырылған. Осындай аралықты белгілейтін айнымалыны анықтағанда шектелген немесе түгенделетін типтер қолданылады.

Мысалы, **VAR A: MIN..MAX**; Мұндағы, А - айнымалысы MIN - төменгі және MAX - жоғарғы шектермен шектелген ішкі жиында анықталған, жиынның шекаралары екі нүктемен (..) бөлінеді. Мысалы, **i - айнымалысы 1900-1995** жиынның мәндерін қабылдау үшін баяндау бөлімінде бұл талап былайша көрсетіледі:

Var i: 1900..1995;

i - айнымалысы integer, яғни жиынның шегі бүтін сандар болмақ. Шектелген типтерді символдық және түгенделетін типтерде де анықтауға болады, Мысалы:

Түре Апта = дүйсенбі, сейсенбі, сәрсенбі, бейсенбі, жұма, сенбі, жексенбі;

Var

Жұмыс_күні: дүйсенбі.. жұма;

Мұндағы, Жұмыс_күні айнымалысы Апта типіне сәйкес дүйсенбіден Жұмаға дейінгі күндердің мәндерін қабылдай алады.

6-Дәріс Алгоритмдік тілдік операторларын класқа бөлу.

Меншіктеу операторлары. Басқару операторлары. Мәліметтерді енгізу шығаруды ұйымдастыру. Программаның құрылымы.

Алгоритмінің схемасынан программаның схемасына көшу.

Алгоритмдердің сызықтық құрылымын программалау.

Бағдарламаға енетін нұсқауларды (бұйрықтарды) Паскаль тілінде *операторлар* дейді. Паскаль тілінің операторларын *қарапайым* және *күрделі* (құрама) операторлар деп екі топқа бөледі. Қарапайым операторлардың құрамына басқа оператор енбейді. Ал күрделі операторлардың ішінде бінеше қарапайым операторлар еніп тұруы мүмкін.

Паскаль тілінде **қарапайым операторларға**: *меншіктеу, көшіру, бос, енгізу және шығару* операторлары жатады.

Күрделі операторға: *шартты көшіру, қайталау, таңдау, жалғастыру* т.б. операторлар жатады.

Меншіктеу операторлары

“Мен меншікте ” деген бұйрықты орындайтын операторды меншіктеу операторы дейді. Жалпы жазылу үлгісі:

W:=E;

Мұнда: W-айнымалы атауы, “ :=”- меншіктеу белгісі; E-өрнек.

Жалпы жағдайда арифметикалық өрнек

E= {тұрақты, айнымалы атауы, функция, өрнек болуы мүмкін.

Меншіктеу операторы тек арифметикалық өрнектер үшін ғана емес, логикалық және символдық берілгендер үшін де пайдаланылады. Мысалы: $K:=A \text{ AND } B$, $B:=\text{'T'}$; $B5:=\text{'9'}$;

Меншіктеу операторын пайдаланғанда айнымалы атауының типі мен қабылдайтын мәнінің типі бірдей болуы қажет. Мысалы, егер $\text{VAR } A, B, C, D:\text{REAL}$;

Мысалы: $A:=15.7$ т.с.с.

Бос оператор

Бос оператор ешкандай амал орындамайды. Бос оператор қалдырылып кеткен оператордың орнын белгілеу үшін пайдаланылады. Әдетте бос оператор орнына “;” қойылады. Мысалы, $S:=A$:

$R:=5$;

;

$M:=19.36$;

Сандық берілгендерді енгізу

Берілген мәндерді айнымалыға жазу үшін меншіктеу операторын пайдалануға болады. Мысалы, $A:=15$ т.б. Бірақ бұл жағдайда бағдарлама әмбебап болмайды. Бағдарламаны әмбебап етіп жазу үшін айнымалылардың мәндерін өзгеретін түрде жасау қажет, бұл жағдайда бағдарлама әр түрлі мәндер үшін дұрыс болады. Ол үшін енгізу операторы READ – (ағылшынша оқу) пайдаланылады.

Жазылу үлгісі:

$\text{READ}(d_1, d_2, \dots, d_n)$;

Мұнда d_1, d_2, \dots, d_n – айнымалы атаулары, оларды енгізу операторларының параметрлері деп те атайды. Параметрлердің сандық мәндері бір-бірінен бос орын арқылы ажыратылып жазылады. Мысалы, $\text{READ}(A, B, C)$; берілсе, параметрлерге мән мына түрде беріледі:

5 6.2 4.5 (ЕП) Мәндерді енгізгенде мән енгізілген жолдан келесі жаңа жолға көшіру үшін параметрсіз немесе параметрімен сәйкес READLN , $\text{READLN}(d_1, d_2, \dots, d_n)$; операторын пайдалану қажет.

Шығару операторы.

Жазылу үлгісі: $\text{WRITE}(d_1, d_2, \dots, d_n)$;

Мұнда, d_1, d_2, \dots, d_n жай айнымалылар немесе апострофтар ішінде алынған символдар тобы болуы мүмкін. Мысалы, егер $B=17,15$ болып мына команда орындалғанда: $\text{WRITE}(\text{'B='}, B)$ экранда $B 17.15$ көрінеді.

Бүгін және нақты сандарды шығару үшін сандардың форматын беру қажет. Формат айнымалы атынан соң “;” арқылы жазылады. Нақты сан үшін формат екі саннан тұрады:

I – санға берілетін барлық орын,

II – үтірден кейін бөлшек бөлік саны,

Мысалы, WRITE (Y:5:2);

Мұнда, 5 нәтижеге берілген барлық орын, 2 үтірден соң алынатын бөлшек бөліктің саны. Жалпы түрде:

Бүгін сан үшін: WRITE (N:S); немесе WRITE ('K=', K:S); мұнда S формат, яғни бүгін сана берілген орын.

Нақты сан үшін. WRITE ('Y=', Y:8:3); егер $y=1,76$ болса, онда $y=...1.760$. Егер операторда формат көрсетілмесе, онда нәтижеде қанша орынды сан алынса, сол сан экранға шығады.

Паскаль тілінде параметрсіз шығару операторы WRITELN – жаңа жоға көшіру үшін пайдаланылады.

Егер WRITELN (d_1, d_2, \dots, d_n) операторы пайдаланылмаса, онда d_1, d_2, \dots, d_n мәндерін экранға шығарған соң курсор келесі жолға көшеді.

WRITE операторын түсініктеме беруге пайдалануға болады. Мысалы WRITE ('A, B, C енгіз:'); READ (A, B, C);

Мысалы, Радиусы R-ге тең шардың көлемін есептеуге бағдарлама құралық.

{Шар көлемін есептеу}

Program B1 (input, output);

const PI=3.14;

VAR

R: real;

V: real;

begin

writeln ('R- мәнін енгіз :');

read (R);

V:=4*PI*R*R*R;

WRITELN;

writeln ('нәтиже:');

writeln ('шар көлемі V=', V:8:3);

end.

7-Дәріс. Тармақталған құрылымдарды программалау.
Алгоритмдердің циклдік құрылымдарын программалау (сандық талдау есептері, сандық массивтерді өңдеу, массивтер компоненттерін реттеу есептері және т.б. мысалдарында) массивтерді енгізу шығаруды программалау.

Шартсыз көшу операторы.

Жазылу үлгісі:

GOTO белгі;

Белгі ретінде таңбасыз 1-9999 дейінгі бүтін сан пайдаланылады. Көшіру операторы қарапайым оператор қатарына жатады. Белгі LABEL – белгі бөлігінде өрнектеледі. Оның жазылу үлгісі: LABEL белгі;

Белгіні былай жазуға да болады:

LABEL 1–белгі , 2–белгі,..., N–белгі ;

Мысалы, LABEL 10,99;

...

GOTO 99;

...

99: WRITELN ('АУЫСУ ОПЕРАТОРЫ ОРЫНДАЛАДЫ');

Құрама операторлар

Бірнеше операторларды бір операторға біріктіргенде шыққан операторды *құрама оператор* дейді. Ол операторлардың жазылу үлгісі:

BEGIN

1-оператор;

2-оператор;

n-1-оператор;

n-оператор

END.

Бұл жағдайда BEGIN – END қызмет сөздері операторлық жақшалар деп аталады. Құрама оператор бүтін бір оператор қызметін атқарады, сондықтан оны бағдарламаның кез-келген бөлігіне қоюға болады. Құрама оператордың құрамында өзінің құрама операторы болуы мүмкін.

Құрама операторларға шартты көшіру және қайталау операторлары жатады.

Шартты оператор

Жазылу үлгісі:

```
IF шарт THEN 1-серия ELSE 2-серия;
```

Шартты операторда құрама команданы пайдалануға болады:

```
IF логикалық өрнек THEN
```

```
BEGIN
```

```
1-оператор;
```

```
2-оператор;
```

```
.....
```

```
n-оператор;
```

```
END
```

```
ELSE
```

```
BEGIN
```

```
1-оператор
```

```
.....
```

```
n-оператор;
```

```
END;
```

Қысқартылған жазылу үлгісі:

```
IF шарт THEN оператор;
```

Таңдау операторы

Жазылу үлгісі:

```
CASE өрнек OF
```

```
1-тұрақты: 1-оператор;
```

```
2-тұрақты: 2-оператор;
```

```
...
```

```
n-тұрақты: n-оператор;
```

```
END.
```

CASE-жағдай, OF- солар, төмендегілер

Мысал:

```
CASE k+1 of
```

```
5: y:= SQR(x);
```

```
11: y:= SQRT(x);
```

```
4: z:=y*(A-B);
```

```
7: WRITE (A, B);
```

```
END.
```

Өрнек символдық та болуы мүмкін, онда тұрақтыларда символдық типте болуы керек. Мысалы:

```
CASE s of
```

```
  '+', '-' , '*', '/' : P:=1;
```

```
  'A', 'B' : P:=2
```

```
  '.' : P:=3
```

```
END.
```

Циклдік құрылымдарды программалау

Паскаль тілінде қайталау процесін үш түрлі жолмен ұйымдастыруға болады:

- алдын ала берілген шарт бойынша;
- келесі шарт бойынша;
- параметрдің мәні бойынша;

1. Алдын ала берілген шарт бойынша қайталау.

Алдын ала берілген шарт бойынша қайталау қайталаудың саны белгісіз болған жағдайда пайдаланылады.

Жазылу үлгісі:

```
WHILE шарт DO
BEGIN
    серия (шарт денесі)
END.
```

WHILE – әзір, DO – орындалу.

Егер қайталаудың денесі бірнеше оператордан құралса, онда олар BEGIN – END қызмет сөздерінің арасына алынып жазылады. Ал егер қайталаудың денесі бір ғана оператордан тұрса, онда оны операторлық жақшасыз жазуға болады:

```
WHILE шарт DO оператор;
```

2. Келесі шарт бойынша орындалатын қайталау операторы.

Бұл оператор көп жағдайда қайталаудың саны белгісіз болған жағдайда пайдаланылады.

Оның жазылу үлгісі:

```
REPEAT
    қайталау денесі
UNTIL логикалық өрнек
```

Мұнда, REPEAT – қайтала, UNTIL - соған дейін - қызмет сөздері.

Орындалуы: қашан логикалық өрнектің мәні ақиқат болғанша циклдің денесі қайталана береді.

Оператордың соңы UNTIL сөзінен бітетін болғандықтан , мұнда операторлық ашылатын және жабылатын жақшалар орнына жазылатын BEGIN – END қызмет сөздері жазылмайды.

Мысал: $x=8, 6, 4, 2$ болғанда $y=x^2$ мәндерін табуға бағдарлама құру керек.

```
program B10;
var   x, y: integer;
begin
    writeln ('x, y: ');
    x:=8;
REPEAT
    y:=x*x;
    writeln ('X', X:3, 'Y', Y:5);
x:=x-2;
UNTIL x=0;
end.
```

Параметрлік қайталау

Қайталау саны белгілі болған жағдайда параметрлік қайталау операторын пайдалануға болады. Жазылу үлгісі:

```
FOR I:=m1 TO {DOWNT0} m2
DO
BEGIN
    операторлар
END.
```

Мұнда, FOR- үшін, TO - дейін, DO - орындалу деген мағынаны білдіреді. I - цикл параметрі, m1, m2 – сәйкес параметрдің бастапқы және соңғы мәндері, олар бүтін сандар немесе бүтін мән беретін арифметикалық өрнектер болуы мүмкін. Паскаль тілінде параметрдің қадамы тек екі мән., 1 немесе -1 –ді ғана қабылдай алады. Қадамның мәніне тәуелді операторда TO немесе DOWNT0 қызмет сөздері пайдаланылады. Егер қадам 1 – ге тең болса, онда операторда TO, ал қадам -1 болса, онда DOWNT0 пайдаланылады.

Мысал: $A=2i, B=2i+1$ мәндері i -дің 1, 2, 3, 4, 5 болғандағы мәндерін табуға бағдарлама құру қажет.

1 - вариант.

```
program B14;
var   a, i, b: integer;
begin
```



```

    for i:=1 to 5
begin
    A:= 2*I; B:=2*I+1;
    writeln ('A=', A:3, 'B=', B:3);
end
end.

```

2 - вариант.

```

program B15;
    var A, I, B: integer;
begin
    for i:=5 downto 1 Do
begin
    A:=2*I; B:= 2*I+1;
    writeln ('A=', A:3, 'B=', B:3);
end
end.

```

Егер бағдарламаның денесі тек жалғыз ғана оператордан тұратын болса, онда begin-end жазбауға да болады.

<pre> FOR I:=m1 TO m2 DO оператор; </pre>

немесе

<pre> FOR I:=m1 TO m2 DO оператор </pre>
--

Күрделі циклдер

Көп есепті шешу үшін бірінің ішінде бірі жатқан күрделі деп аталатын циклді пайдалануға тура келеді. Бұндай бағдарламаларды құрғанда ішкі цикл толығымен сыртқы циклдің ішінде орналасуы қажет. Ішкі циклдің өзі де басқа ішкі циклдерді қамтуы мүмкін.

Мысал: $N=1, 2, 3$ және $k=2, 4, 6, 8$ болғандағы $y=2k+N$ мәндерін есептеуге бағдарлама құру қажет.

```

program B18;
    var N, k, y: integer;
begin
    for N:=1 to 3 do
begin k:=2;
    while k<=800

```

```

begin y:=2*k+N;
      writeln (N:4, k:4, 9:4);
      k:=k-2;
end;
end;

```

end.

Массив дегеніміз—бір типтегі берілгендер жиыны.баскаша айтқанда массив—бір атауға біріктірілген айнымалылардың реттелген тізбегі. Айнымалылардың массив элементтерінің типтері бірдей болады. Массив бір ғана атпен белгіленеді.мысалы: нақты сандардан құралған тізбекті К атаулы массив деуге болады. Массивтің әр элементі массивтың атымен белгіленеді де,оның индексі қойылады, яғни массив элементтері индекстерібойынша реттеліп жазылады. Мысалы: A_1, A_2, \dots, A_n немесе жалпы түрде $A_i, i=1, 2, \dots, n$. Паскаль тілінде массив элементтерінің индексі алгаритмдік тілдегі тәрзіді тік жақшаға алынып жазылады.

Мысалы: $A[i]$

Паскаль тіліндегі массив ұғымы алгаритмдік тілдегі массив ұғымына сәйкес келеді. Алгаритмдік тілдегі ТИП Атау өлшем (мысалы: нақ таб $A[1:N]$) орнына Паскаль тілінде атау:массив (индекстердің алғашқы .. соңғы мәні элемент типі — түрінде жазылады.

Егер бағдарламада массив пайдаланылатын болса, онда ол немесе айнымалы VAR бөлігінде немесе тип TYPE бөлігінде бейнеленуі қажет.

Массив айнымалы бөлігінде былай белгіленеді:

VAR массив аты:ARRAY [t1] of t2:

Мұндағы ARRAY(массив), of (одан) қызмет сөздері, t1—real, integer базалық тілінен өзге кез келген стандартты типі, t2—Паскаль тілінде пайдалануға болатын массив элементтерінің типі. Индекс типі стандартты бүтін немесе нақты бола алмайды.

Массивтерді дұрыс жазуға мысалдар келтірсейік:

Var

Mas:ARRAY [1..N] of real;

жыл :Array [қаңтар .. желтоқсан]

of integer

L:Array [жол] of Bool AN;

M:Array [char] of кругж

Егер бірнеше массив бірдей бағалық және индекстік типте болса, онда оларды бейнелгенде бір массив ретінде үтір арқылы бөліп жазуға болады. Мысалы: `Var A,B,C: Array [1..10] of real`; мұнда әр бір A,B,C- массивтері, элемент типтері нақтылы болатын 10 элементтер тұрады.

Бағдарламада «индекс» және «индекстар типі» ұғымдарын шатастыруға болмайды. Индекс типі массивті бейнелгенде пайдаланылса, индекс массив элементтерін белгілеу үшін операторлық бөлікте пайдаланылады. Индексті белгілегенде қандай тип берілсе индекс сол типте ғана болуы қажет.

Индекс, өрнек, айнымалы немесе тұрақты болып берілуі мүмкін. Сондықтан да көп жағдайларда массивтерді *индексті айнымалылар* деп те атайды. Ал индексіз айнымалыларды массивтерден ажырату үшін *қарапайым айнымалылар* дейді. Базалық типтегі айнымалыларға қандай амалдар қолданылса, массив элементтеріне де сондай амалдарды қолдануға болады. Мысалы: бағдарламаның операторлық бөлігінде массив элементтеріне мынадай амалдар: `A[3]:=C[4]+5`; `S:=S+X[K]`; `P:=X[3*I+1]`; және т.с.с. қолданылады. Массив элементтеріндегі енгізу және шығару үшін бағдарламаның мына үзінділерін пайдалануға болады:

а) енгізу:

```
FOR I:=1 TO N DO
```

```
  READ ( A[I]),
```

```
  .....
```

б) шығару:

```
FOR I:=1 TO N DO
```

```
  WRITE ( A[I]);
```

```
  .....
```

Массивтің тип бөлігіндегі жазылуы

Массив тип бөлігінде былай бейнеленеді:

```
TYPE тип аты = array [t1] of t2
```

```
VAR массив аты: тип аты;
```

Яғни мұнда тип (TYPE) бөлігінде массивтің типі көрсетіледі де, одан соң айнымалы (VAR) бөлігінде осы типке жататын массивтер көрсетіледі. Массивті TYPE бөлігінде бейнелеу, бейнелеулер бөлігінің көлемін үлкейтеді, бірақ бағдарламаны тез жөндеуге бағдарламада көп қате жібермеуге көмектеседі. Мысалы, бағдарламада нақты 40 элементтер тұратын A массиві пайдаланылатын болса, онда оны тип бөлігінде былай йөрнектеледі:

TYPE N= array [17740] of real;

Var O:N;

Бұның артықшылығы, егер бағдарламада типі N болатын бірнеше массив A,B,C,D пайдаланылатын болса, онда олар үшін:

Var A,B,C,D: N;

түрінде жазса жеткілікті.

Көп өлшемді массивтер

Біз жоғарыда бір өлшемді массивті, яғни әр элементтің бір ғана индексі бар массивтерді қарастырдық. Математикада массивтің ішіндегі массивті екі өлшемді (екі өлшемді тік бұрышты кесте) немесе матрица деп атайтыны белгілі.

Мысалы:

$A = \begin{pmatrix} 54 & 36 \\ 28 & 17 \\ 43 & 95 \end{pmatrix}$, бұның элементтерін қысқаша a_{ij} ($i=1,2,3, j=1,2,3,4$) арқылы белгілеп $A=(a_{ij})$ түрде жазуға болады.

Бұл матрицаның өлшемі 3x4. Матрица тек бір атпен белгіленеді де, индекстерді үтір арқылы ажыратылып жазылады. Мысалы: A [I,J] немесе A [I] [J] түрде жазуға болады. Паскаль тілінде мұндай матрицаны тип бөлігінде төмендегідей өрнектеуге болады:

1. TYPE array=ARRAY [1..N, 1..M] of mun;

Var A: array;

Мысалы:

TYPE T=ARRAY [1..3, 1..4] OF INTEGER;

VAR A: T;

немесе

2. TYPE array= ARRAY [1..N] OF ARRAY [1..M] OF mun; VAR A:

array;

мысалы,

TYPE T= ARRAY [1..3] OF ARRAY [1..4] OF тип;

VAR A:T;

1-ші жағдайда индекстің әрбір типі белгіленеді де, одан соң массив элементтерінің базалық типі көрсетіледі. 2-ші жағдайда алдымен [1..N] типі көрсетіледі де, одан соң күрделі базалық тип [1..M] OF тип көрсетіледі. Егер бағдарламада матрицаның тік және жатық жолдарын былай жазу қажет болса, онда оны былай бейнелесу қолайлы болады:

TYPE

T1= ARRAY [1..M] OF INTEGER;

```
T= ARRAY [1..N] OF T1;
VAR
A: T;
B:T1;
```

Мұнда алдымен бір жолдың типі T1 көрсетіледі де, одан соң ол жолдың типі арқылы матрицаның типі (T) көрсетіледі. Айнымалы бөлігінде екі өлшемді массив A және бір өлшемді Массив B көрсетілген.

Массивтерге мысалдар:

1-мысал $S = \sum_{i=1}^N a_i$ бағдарлама құру керек. Бағдарламаны үш вариантта құралық:

а) массив элементтерін сақтамай есептеу:

```
PROGRAM MA 1;
VAR
  A, S: REAL;
  N, I: INTEGER;
BEGIN
  S:= 0;
  WRITE IN (N санын енгіз.);
  READ (N);
  WRITELN (□ сандарды бос орын арқылы енгіз. );
  FOR I:=1 TO N DO
  BEGIN
    READ (A);
    S:= S+A
  END;
  WRITE LN (□ □);
  WRITELN (□ қосынды S=□, S:6:2)
  END.
```

1) Мұнда массив үшін бір ғана ұяшық беріледі.

2) Массив жадыда сақталмайды.

б) Массивті айнымалы бөлігінде бейнелеу және массив элементтерін жадыда сақтау:

```
PROGRAM MA 2;
const N=7;
VAR
A: ARRAY [1..N] OF REAL;
S: REAL;
I: INTEGER;
```

```

BEGIN
WRITELN (□санды бос орын арқылы енгіз:□);
FOR I:=1 TO N DO READ ( A[I] );
S:= 0;
FOR I:=1 TO N DO S:= S+A [I];
WRITE LN (□_□_□_□_□_□_□);
WRITELN (□ косынды S=□, S:6:2)
END.

```

1) Мұнда массив үшін 7 ұяшық (N-ұяшық) беріледі;

2) массив жадыда сақталады.

в) массивті тип бөлігінде бейнелеу және массив элементтерін жадыда сақтау.

```

PROGRAM M 3;
const N=7;
TYPE T= ARRAY [1..N] OF REAL;
VAR
T:T;
S: REAL;
I: INTEGER;
BEGIN
WRITELN ( массив элементтерін бос орын арқылы енгіз:□);
FOR I:=1 TO N DO READ (A[ I ]);
S:= 0;
FOR I:=1 TO N DO
S:= S+A[ I ];
WRITE LN (□_□_□_□_□_□_□);
WRITELN (□ косынды S=□, S:6:2)
END.

```

8-Дәріс. Жолдар. Символдық мәліметтерді өндеудің есептерін программалау. Мәліметтер құрылымында қолданатын есептерді программалаудың ерекшеліктері.

Мәтін деп жолдар тізбегін айтады. Жол символдар тізбегінен тұрады. Жолдың тұрақты, енгізілген символдарының саны 256-дан артпайтын апострофтар ішіне алынған символдар тізбегін айтады. Жолдың айнымалысының типі бағдарламаның тип бөлігінде немесе тікелей айнымалы бөлігінде бейнеленеді.

Айнымалы бөлігінде белгіленуі:

VAR

<айнымалы аты>: string [жолдың максимал ұзындығы];

Мысалы:

VAR

tex: string [29];

AT: string [40];

Жолға мынадай амалдар қолданылады:

1. **Жалғастыру амалы**, ол қарапайым “+” белгісімен белгіленеді

‘ауыл’ + ‘адамдары’

‘ауыладамдары’

2. **Салыстыру амалы**, олар: =, <, >, >=, <=, <

Мысалы:

‘SAM’= ‘SAM’ TRUE

‘SAM’> ‘SAM’ TRUE

‘BBB’< ‘BB’ FALSE

3. **Меншіктеу амалы**, ол “:=” арқылы белгіленеді.

а) A: string [6]; A:= ‘қалам қас;’

A:= ‘ қалам ’

Жолдың символдарын, олардың орналасу орындары (индексі) бойынша табуға болады. Индекстің типі бүтін болады, ол айнымалы атауынан соң квадрат жақша ішіне алынып жазылады.

Мысалы: S= ‘Алматы’ болса, онда S[2]= ‘л’, S[4]= ‘А’ береді.

Жолдық функциялар

Паскаль тілінде төмендегі стандарт функциялар пайдаланады.

1. CHR(I) - I санына сәйкес символды береді. Мысалы:

CHR(1)= ‘Г’, CHR(250)= ‘z’, CHR(255)= ‘r’

2. ORD(S)-S символының кодын береді. Мысалы:

ORD(z)=250, ORD(r)=255.

3. COPY(S,N,M)-S жолының N-1 орындағы символынан бастап M символ боліп алады. S-жолдың айнымалы, N,M- бүтін сан.

Мысалы:

Program BOLY;

VAR N,M:integer;

S:string [12]; R:string[8];

begin

writeln(‘N,M,S- енгіз.’);

readln(N,M,S);

R:=copy(S,N,M);

Write ('R=', R)

end.

егер N=4, M=6, S='Алматылық' болса, онда R='мағылы' болады.

4. CONCAT (S1, S2, ..., SN) – жолдардың реті бойынша жалғастырады. Мысалы: concat ('a', 'b', 'c')='abc'

5. LENGTH (S) – Жолдың айнымалысының мәніндегі символдар санын береді. Мысалы: length ('ABCD')=4

6. POS (S1, S2) - S1 кіші жолы S2- жолының бөлігі болатынын немесе болмайтынын анықтайды. Бұл функциялардан шыққан нәтиже бүтін сан болады да, S1-мәні, S2-мәніне сәйкес келетін орынды көрсететін санға тең болады. Егер нәтиже нөлге тең болса, онда S1 жолы S2 жолының бөлігі бола алмайды. Мысалы:

POS ('ab', 'abcd')=1;

POS ('ab', 'array')=0;

Мысал: Текстегі кіші әріптерді үлкен әріптерге алмастыру.

program NAK;

VAR t:string [15];

str,prop:string[33];

I,l,p;integer;

begin

writeln ('ұзындығы U -дан кем текст енгіз;');

readln (t);l:=length (t);

str:='a,б,в,г,д...s';PROP:='АВВ...S';

for I:=1 TO l DO

begin

P:=PAS (t[I],str);

write (PROP[p])

end

end.

UPCASE (CH)-латынның кіші әріптерін оған сәйкес үлкен әріпке аударады. Мысалы: UPCASE ('a')='A';UPCASE ('q')='Q'.

Жолдық процедуралар

Процедура деп атау берілген, арнайы амалдарды орындайтын, өзінше тәуелсіз бағдарламаны айтады. Процедура тақырыбынан және оның денесінен тұрады. Процедураны оның атауы арқылы шақырады.

1. DELETE (S,M,N)-S-тің мәнінің B-ші орнынан бастап N символды алып тастап нәтижені S-ке жазады.

Мысал .

Нәтиже

...

S:= 'абав';

DELETE (S,2,2);

writeln('S=',S);

S= 'ав'

2. INSERT (S1,S2,N)-S2-нің Nпозициясынан бастап S1 жолын S2 жолының арасына орналастырады да нәтижені S2-ге жазады.

Мысалы:

Берілуі

Нәтиже

Var S1,S2:string [10];

...

S1:= ' физ.фак-Т';

S2:= '- мат. ';

INSERT (S2,S1,);

write (S1);

'физ-мат.фак-т'.

3. str (I,S)-I санын жолға айналдырады да S-айнымалысына орналастырады.

Мысалы:

Нәтиже

...

var I:real;

s:string [20];

I:=15;

str(I,S);

writeln (s);

'15'

str(I:4,s);

writeln(s);

' 15'

4. Var (s,x,cod)- сандық S жолын бүтін немесе нақты санға аударады да нәтижесі I-ге жазады. cod- бүтін типтегі сан. Егер cod≠0 онда аударылғаны дұрыс, cod=0 қате,оның мәні I-ші қате кеткен орынды көрсететін санға тең.

Мысалы:

Берілуі:

Нәтиже:

Var

L:tcuar;

st:string [6];

cod,I:integer;

begin

```
lit:= 'F';
st:= '1992';
val (st,I,cod);
writeln ('I=',I, 'Acod=',cod);      I=1992 cod=0
val (lit,I,cod);
writeln('I=',I, ' cod',cod);      I- кез келген сан cod=I
```

9-Дәріс. Ішкі программалар және оларды класқа бөлу. Ішкі программаларды ұйымдастыру әдістері. Ішкі программаларды шақыру.

Практикалық ессиптерге бағдарлама құрғанда, ал үлкен бағдарламаның белгілі бір бөлігін әр түрлі мәндер бойынша бірнеше (тіпті одан да көп) рет қайталап пайдалануға тура келеді. Мұндай бір типтес бағдарлама бөлігін бағдарламаның әр бөлігінде қайталап жаза беруіне мақсатында оны жеке көмекші бағдарлама ретінде бөліп тастау қолайлы.

Жеке бағдарлама түрінде бөлек жазылған, қажет кезінде оған оралып, оны пайдаланып отыруға болады болатын негізгі бағдарламаның арнайы бөлігін **көмекші** бағдарлама (ішкі бағдарлама) дейді. Көмекші бағдарламаға автоматты түрде енгізілетін және одан шығуға болады.

Көмекші бағдарламаны пайдалану төмендегідей мүмкіндіктер береді:

- негізгі бағдарламаның көлемі кішірейеді;
- негізгі бағдарламада пайдаланылған айнымалыларды көмекші бағдарламада пайдалануға болады;
- көмекші бағдарламаға берілген жады ұяшықтарын, ол орындалмай тұрғанда бос ұяшық ретінде (бос айнымалы ретінде) пайдалануға болады;
- көмекші бағдарламаны пайдалану структуралық бағдарлама-лауға мүмкіндік береді.

Паскаль тілінде алгоритмдік тіл тәрізді көмекші бағдарламаның екі түрі пайдаланады. Олар процедура (PROCEDURE) және функция (FUNCTION).

Бір бағдарламада бір немесе бірнеше процедура және функция пайдалануы мүмкін. Паскаль тілінде процедуралар мен функциялар бағдарламаның басында айнымалылар бейнеленіп болғаннан кейін бейнеленеді, яғни жазылады.

Жоғарыда ескерілгендей негізгі және көмекші бағдарламада бір айнымалы, мысалы К қатар пайдалануы мүмкін, бірақ әр бағдарламада пайдаланғанда К-ның функциялық мағынасы бірдей болуы шарт емес. Атау берілген операторлар тізбегін Паскаль тілінде процедура дейді. Яғни: әр көмекші бағдарламаның (процедураның немесе функцияның) атауы (аты болуы керек, көмекші бағдарламаға сан атау негізінде оралады).

Процедуралар

Кез келген процедура бағдарламаға ұқсас түрінде жазылады. Олардың басы, бейнелеу және операторлар бөліктері болады.

PROCEDURE аты (формалды параметрлер тізімі);

Бейнелеу бөлігі

BEGIN

Оператор бөлігі

END

Процедураны параметрсіз пайдалануға болады. Екі процедуралық бағдарлама схемасы төмендегідей болады: PROGRAM P (INPUT, OUTPUT);

Ортақ берілгендерді белгілеу бөлігі

PROCEDURE P1;

P1-дің процедураларын бейнелеу бөлігі

BEGIN

P1-дің операторлар бөлігі

END;

PROCEDURE P2;

P2-нің процедураларын бейнелеу бөлігі

BEGIN

P2 операторлар бөлігі

END.

BEGIN

Негізгі бағдалманың операторлар бөлігі

END.

Бағдарламаның орындалуы негізгі бағдарламалық операторының орындалуынан басталады. Негізгі бағдарламада қашан процедураның орындалуы қажет болғанда ғана процедураға көшеді. Бұл жағдай да берілгендер негізгі бағдарламадан процедураға (енетін параметрлер) беріледі.

Процедура орындалып болған соң оның нәтижесі (шығатын параметрлер), процедураға негізгі бағдарламадан қай жерден көшті, сол жерге береді. Одан соң негізгі бағдарлама орындала бастайды.

Жеке бағдарламаларды бағдарлама кітапханасына жинап, қажет кезінде процедура ретінде пайдалануға болады. Кітап ханадан қажетті процедураны «ала тұру» үшін EXTERN (немесе EXTERNAL;) операторын пайдаланады.

Кытапханадан қажетті процедураны «алу» үшін процедураның басынан (тақырыбынан) соң (PROCEDURE атай) EXTERN; операторы жазылады;

Функция

Паскаль тілінде функцияны арнайы бағдарлама ретінде жеке жазуға болады, сондықтан оны көмекші бағдарлама ретінде қарауға болады. Функция- бағдарламаның, процедура бағдарламадан ерекшелігі:

- функция бір ғана белгілі мән қабылдайды, яғни функция - бағдарламадан алатын нәтиже біреу-ақ;

- нәтиже функция аты арқылы алынады да, негізгі бағдарламада пайдаланылады.

Функция- бағдарламаның жазылу үлгісі:

FUNCTION аты (формальді параметрлер):тип;

бейнелеу бөлігі;

BEGIN

параметрлер бөлігі

END;

Функция фактілі параметрлердің мәндері берілгеннен кейін өз аты арқылы шақырылады. Сонымен қатар функцияны тікелей өрнектің ішіне шақыруға болады.

Процедура мен функция айырмашылықтарын түсіну үшін екі санды қосу бағдарламасын процедура және функция түрінде жазалық.

Мысал: Екі санның қосындысын табу керек.

а) PROGRAM KPR;

VAR A: integer;

PROCEDURE KOS (B,C: integer; VAR k:integer);

begin

k:= B+C;

end;

begin

```
kos (15,4,A);  
WRITELN ('kos=',A: 2);  
END.
```

- процедура мен функцияның ұқсастығы олардың екеуінде де айнымалыларды, белгілерді, тұрақтыларды бейнелеу бөліктері боладыж

- олардың бір-бірінен бірінші өзгешелігі- 1-сінде PROCEDURE, ал 2-сінде FUNCTION қызмет сөздері пайдаланылады;

- процедурада, нәтиже (айнымалы) параметрінің типі процедура атауынан соң тұратын кіші жақша ішінде бейнеленсе, ал функцияда нәтиже үшін арнайы параметр алынбайды, нәтиже (функция мәні) функция атына меншіктеледі , сондықтан функция типі, аргументтер типтері бейнеленген жақша сыртында, ":" арқылы көрсетіледі.

Мысалы:

```
function f (a:real,var b:real):real;  
function f (var a,b: integer.):integer;  
function f (a:integer): char;
```

- негізгі бағдарламада функцияның мәнін тек функция атауы арқылы пайдаланады.

Мысал: $F = m! \cdot k!$ есептеу керек.

```
PROGRAM FAK;  
VAR F: real; M,K: integer;  
FUNCTION FACT ( N: integer): real;  
VAR I: integer;  
    P: real;  
begin  
    P:=1;  
    FOR I:=1 TO N DO  
        P:= P*I; FACT:= P  
END;  
BEGIN  
READLN( M,K);  
F:= FACT (M)- FACT (K);  
WRITELN ('F=',F)  
END.
```

10-Дәріс. Нақты параметрлерді тасымалдау. Жадының жалпы облысын пайдалану. Стандартты ішкі функциялардың қоры және процедуралары. Оларды шақырудың тәсілдері.

Турбо Паскальда сегіз стандартты модульдер бар. Бұл модульдерде көптеген әртүрлі типтер, тұрақтылар, процедуралар мен функциялар қолданылады. Оларға SYSTEM, DOS, CRT, PRINTER, GRAPH, OVERLAY, TURBO3 және GRAPH3 жатады.

GRAPH, TURBO3 және GRAPH3 модульдері жеке TPU-файлдарға белгіленген, ал қалғандары TURBO.TPL кітапханалық файлдың құрамына кіреді. Тек қана бір SYSTEM модулі кез келген программаға автоматты түрде қосылады. Басқа модульдерге USES сөзінен кейін тізімде аттары көрсетілгеннен соң қатынас құруға болады.

Стандартты модульдер

Негізгілері: -system, -crt, -graph, -dos, -printer.

Graph модулі және tpu-файлда, ал қалғандары turbo.tpl (tpl – Turbo Pascal Library) кітапханалық файл құрамына кіреді. System модулі ғана кез келген программаға автоматты түрде қосылады, қалған модульдерді uses сөзінен кейін көрсету қажет.

System модулі. Турбо Паскаль стандартты ішкі программаларының негізгі кітапханасы. Онсыз бірде – бір программа орындалмайды. Бұл модульге барлық негізгі стандартты функциялар мен процедуралар (енгізу /шығару, қатармен жұмыс процедуралары, математикалық амалдар және функцияға тән процедуралар мен функциялар) жатады.

crt модулі. Экран жұмысының мәтіндік режимін басқару, кілтжиынды, дыбыстық жұмыстарға сәйкес процедуралар мен функциялар қамтылады.

Graph модулі. Экранның графиктік жұмыс режимін басқаруға қажетті типтер, тұрақтылар, процедуралар, функциялар жиынын қамтиды. Түрлі графиктік көріністер құруға болатын ішкі программалармен қамтылған.

dos модулі. MS-DOS амалдық жүйесінің барлық жабдықтарына қатынас ұйымдастыратын процедуралар мен функциялар қамтылған.

printer модулі. Ақпаратты баспаға шығарудың қарапайым тәсілін ұсынады.

Экранды басқарудың процедуралары мен функциялары
Textmode (mode); мәтіндік режимді орнату. Мұндағы, mode - word бүтін типті тұрақты.

Терезелермен жұмыс. Терезені құру үшін қолданылатын процедура: window (X1, Y1, X2, Y2); X1, Y1 - терезенің жоғарғы сол жақ бұрышының координаталары;

X2, Y2 - терезенің төменгі оң жақ бұрышының координаталары.

Мысалы: **window(1,1,80,25)** – толық экранды алып тұратын терезе құрылады.

Негізгі процедуралар мен функциялар:

-clrScr - активті терезені тазарту;

- clrEol активті терезедегі қатарды курсор тұрған жерден бастап тазарту;

- GoTo (X, Y) - активті терезенің шеңберінде X, Y координаталы позицияға курсорды орнату;

- where X - курсордың ағымдық позициясындағы X - координатасын қайтару функциясы;

- where Y - курсордың ағымдық позициясындағы Y координатасын қайтару функциясы;

-delay (time) - компьютер жұмысын ақырындату процедурасы; - time - миллисекундтағы кідіріс уақыты. **Дыбысты басқару процедуралары:**

-sound (Hz) ішкі динамиканы қосу процедурасы. Hz - сигнал жиілігі (герц-пен өлшенеді). Дыбыстық сигналды өшіру үшін nosound процедурасы қолданылады.

11-Дәріс. Программаларды ұйымдастыру. Әртүрлі құрылымдық ұйымдасқан программалардың мысалдары. Ақпаратты компьютердің сыртқы құрылғыларында көрсету.

Файлдармен жұмыс істеу.

Файлдық тип. Турбо Паскальда файлдың үш түрі бар:

-мәтіндік файл (text типімен анықталады);

-типтелген файл (file of тип сөйлемімен беріледі);

-типтелмеген файл (file типімен анықталады).

Программада файлмен жұмыс үшін баяндау бөлімінде файлдық айнымалыны (файлдық типті) анықтау қажет.

Ол былай беріледі:

type

***Tun* *аты1* =text; *Tun* *аты2* =file of *Tun*; *Tun* *аты3* =file;**

Var;

Айнымалы Аты1: Тип аты1; Айнымалы Аты2: Тип аты2;

Айнымалы Аты3: Тип аты3;

Немесе

Var; Айнымалы Аты1=text; Айнымалы Аты2=file of Tun;

Айнымалы Аты3= file;

Мұндағы, тип –Паскальдағы кез келген тип; (файлдардан басқа).

Мысалы:

Type file type= text {файлдық тип}

var

**ftmp, f: filetype; {файлдық айнымалы} немесе var f1, lst: text;
f2: file;**

Программада баяндалатын файлдық айнымалылар логикалық файлдар деп аталады.

Деректерді енгізу/шығаруды қамтамасыз ететін барлық негізгі процедуралар мен функциялар тек логикалық файлдармен жұмыс жасайды. Сондықтан да, физикалық файлды қолданар алдында оны логикалық файлмен (файлдың айнымалыларымен) байланыстыру қажет. Файлды ішкі және сыртқы деп бөлуге болады. Программа тақырыбында қамтылған файлдар сыртқы деп аталады (sbu f1-f4). Ішкі файлдар сол программа ішінде құрылады, мұндай файлдар программаның орындалуы кезінде ғана жұмыс істейтін болады. Тізбектей қатынас құруға арналған файлдар үшін файлға деректерді бір мезгілде жазу және оқуға тыйым салынған.

Файлға тікелей қатынас құру тәсілі. Мұнда файл элементтеріне олардың адресі (нөмірі) арқылы қатынас құрылады. Қажетті элементті іздеген кезде позициялық нөмірін көрсетсе жеткілікті. Бұл тәсілде деректерді бір мезгілде оқуға және жазуға мүмкіндік бар.

Файлмен жұмыстың жалпы схемасы

Файлмен жұмыс істеуге арналған кез келген программаға төп қасиеттер:

файлдық типті (логикалық файлды) айнымалыны анықтау.

assign (белгілеу) процедурасының көмегімен программалаушы қолданылатын физикалық файлдардың әрқайсысына файлдық типті айнымалыны көрсетеді.

файлды иницирлеу (иницировать), яғни деректерді беру бағытын көрсету.

Файлмен жұмыс файл құрылған жағдайда ғана мүмкін болмақ. Файл жоқ болған жағдайда, оны құру қажет. Бұрыннан бар болса –

reset қайта құру процедурасы қолданылады (переустановка). Екінші жағдайда –rewrite қайта жазу (перезапись) процедурасы қолданылады.

Файлды ашу және құру кезінде оның деректерін уақытша сақтау үшін компьютердің жедел жадысынан автоматты түрде аумақ бөлінеді, оны файл буфері деп атайды. Деректерді оқу, енгізу/шығару инструкциялары: read, readln, write, writeln. Файлдық айнымалыны жабу және деректерді жұмыс аяғында сақтау үшін close процедурасы қолданылады.

Файлмен жұмысқа арналған процедуралар және функциялар

-assign (файлдық Айнымалы, файл Аты)-логикалық файлды файлдық айнымалымен байланыстыру процедурасы. Ашық файлға қолданылмайды. Файл Аты – қатарлық типті өрнек.

-reset (файлдық Айнымалы) процедурасы –бұрыннан бар файлды оқу үшін ашады. Файлды оқу кезінде нұсқағышты оның басына қайтару керек болса, онда reset процедурасын осы файлға қолданса жетіп жатыр.

-iorresult функциясы файлдың дискіде бар-жоғын тексереді.

-rewrite (файлдық Айнымалы) процедурасы –деректер жазу үшін жаңа файл құрады және ашады. Бұл процедура сәтті орындалған жағдайда файлға бірінші элементті жазуға болады. Егер бұрыннан бар файлға rewrite процедурасын қолдансақ, онда ол файл жоғалып кетеді (ішіндегі деректер өшіп қалады) да, сол аттас бос жаңа файл құрылады.

-close (файлдық Айнымалы) процедурасы –программа өңделгеннен кейін программалаушы файлды жабу керек. Қарсы жағдайда деректер жойылып кетуі мүмкін.

-rename (файлдық Айнымалы, файл Аты) процедурасы - кез келген типті ашық емес файлдың атын өзгерту үшін қолданылады. Жаңа ат файл Аты қатарында беріледі.

-erase (файлдық Айнымалы) процедурасы кез келген типті ашық емес сыртқы файлды жояды, файлдық Айнымалы параметрімен беріледі. Rename және erase процедураларын ашық файлдар үшін қолдануға болмайды. Файлды алдымен жазып алу қажет.

-eof (файлдық Айнымалылар) логикалық функциясы- (End of File)-файлдың соңы екенін тексереді (одан деректерді оқығанда).

~ функция true мәтін қайтарады, егер файл соңы болса;

~ қарама- қарсы жағдайда false болмақ.

Read (оқу файлдан) **reset** (f) процедурасының операторы орындалғаннан кейін ғана қосылады.

12-Дәріс. Мәліметтердің динамикалық құрылымы.

Сілтемелер. Кезектер және ағымдармен жұмыс істеу.

Сілтемелер – деректердің ерекше типі. Сілтемелік типтер деп те аталады.

Осы уақытқа дейін біз тек статикалық программалық объектілерді қарастырдық. Бұл объектілер программа орындалмастан бұрын туындап, программаның орындалуы барысында мәндер өлшемі өзгерместен қалып отырады (машиналық жады көлемі өзгермейді).

Статикалық объектілерді баяндаумен туындайды.

Мысалы:

Var

X: array [1..10] of real;

Программаға реттік типті статикалық айнымалылар енгізеді. Статикалық объектілер программаның орындалуына дейін туындайтындықтан және олардың мәндерінің өлшемі алдын-ала белгілі болғандықтан да, осы мәндерді сақтау үшін қажетті машина жадысындағы орын программаның бастапқы мәтінін машиналық тілге трансляциялау кезінде бөлінеді.

Алгоритмдік тілде жазылған программада программалық объектілерге сілтеме олардың аттары арқылы жүзеге асырылады, яғни объектіге аты бойынша қатынас құрылады.

Қазіргі кездегі компьютерлерде бұл орын объекті адресі арқылы, яғни жады ұяшығының көмегімен беріледі. Осы әрекеттердің барлығы (статикалық айнымалыларға тән) трансляция кезінде орындалмақ. Дегенмен, программалауда тек статикалық объектілерді қолданудың өзіндік қиындықтары (тиімді машиналық программа алу көзқарасы тұрғысынан қарағанда) да жоқ емес.

/* Кейде біз программаны құру кезінде алдын-ала белгілі бір программалық объектінің мән өлшемін, мән өлшемі тұрмақ сол объектінің өзінің программа жұмысы кезінде болатынын немесе болмайтынын біле алмаймыз*/.

Программаның орындалу процесінде туындайтын немесе программаның орындалу барысында мәндерінің өлшемі анықталатын объектілер динамикалық объектілер деп аталады.

Мысалы, берілген мәтіннен (кілең сөзден тұратын) бірінші сөзді табу керек болсын.

Мәні ізделінетін сөз болып табылатын айнымалы, программаның орындалу барысында мүлдем пайда болмауы мүмкін. Егер ол пайда болса, онда осы айнымалының мәні болып табылатын сөз ұзындығы алдын-ала белгісіз болуы мүмкін. Мұндай айнымалылар – динамикалық программалық объектілер болып табылады.

Динамикалық объектілерді қолданудың да өзіндік күрделіліктері бар:

Соның бірі, программаның орындалу процесінде динамикалық объектілер пайда болады, ал олармен орындалатын әрекет программаны жазу барысында берілуі қажет. Осыған байланысты, әлі туындамаған объектіге программада қалай сілтеме жасауға болады деген сұрақ туындайды. Ол үшін Паскальда динамикалық объектілермен жұмыста қолданылатын –сілтемелік тип қарастырылады. Бұл типтің мәні қандай да бір болмасын программалық объектіге сілтеме болмақ. Машиналық тілдегі мұндай сілтеме сәйкес объектінің жадыдағы орнын (адресін) көрсетеді.

Динамикалық объектілер және сілтемелер

Сілтемелік типті беру синтаксисі:

<сілтемелік типті беру>:= ↑ <тип Аты>,

↑ -сілтемелік тип белгісі, <тип Аты> -стандартты ат. Сілтемелік типті айнымалылар айнымалыларды баяндау бөлімінде беріледі.

Мысалы:

```
type  
массив = array[1..100] of integer;
```

```
динмас =↑ массив;
```

```
var
```

```
p: ↑ integer
```

```
g: ↑ char
```

```
рабмас: динмас;
```

мұндағы, p–бүтін типті динамикалық объектіге сілтеме;

g –логерлік типті динамикалық объектіге сілтеме.

Рабмас -100 саннан тұратын массивті құрайтын динамикалық объектіге сілтеме. Барлық статикалық айнымалылардың ортақ қасиеті -олардың мәндері сәйкес динамикалық объектінің жадыдағы орындарын **көрсету**, сондықтан сілтемелік типті айнымалыларды **сілтемелер** деп жиі атайды.

Кейбір жағдайларда нұсқағыш мәні ретінде ешқандай объектімен байланыспайтын «бос» сілтемелер құру қажеттілігі туындайды. Мұндай мән Паскалда nil қызметтік сөзімен беріледі де, кез келген

сілтемелік типке жатады. /* Енді динамикалық объектілермен қалай жұмыс жасауға болады, оған мәндерді қалай меншіктейміз, соны қарастырайық*/.

Динамикалық объектіге сілтеме жасау үшін нұсқағышты айнымалы деген түсінік қолданылады. <нұсқағышты айнымалы>:= <сілтемелік айнымалы>↑

<сілтемелік айнымалы> - программадағы берілген динамикалық объектіге сәйкес сілтемелік типті статикалық айнымалының аты. Сілтемелік типтен кейін қойылған ↑ белгісі – сол сілтемелік айнымалының мәнін емес, осы сілтемелік тип көрсететін программалық объектінің мәнін көрсетеді.

Мысалы: программада *p* айнымалысы бар болсын:

var

p: ↑ integer;

new(p) процедурасының орындалуы нәтижесінде, *integer* типті динамикалық объект туындайды. Одан соң: *p↑:=58;* меншіктеу операторын жазатын болсақ, онда жоғарыдағы динамикалық айнымалыға 58-ге тең мән меншіктеледі. Егер *r – integer* типті айнымалы болса, онда мынадай меншіктеу операторын жазуға болады:

R: -r+p↑+2 ;

r -дің бастапқы мәні берілсе, онда $58+2 = 60$ болады.

Сілтемелік айнымалы ретінде индексті айнымалыларды (массивке байланысты) қолдануға болады;

$A[2]$ немесе $A[K+5] \rightarrow A[2]↑$ және $A[k+5]$.

13-Дәріс. Графика. Графиктік бейнелеудің алгоритмдік негізделуі. Стандартты ішкі программалардың қоры.

Стандартты ішкі программалардың қорын ұйымдастырудың ережелері. Стандартты ішкі программалардың қорын есептерін шешу үшін қолдану.

GRAPH модулі

GRAPH модулі экранның графиктік режимде жұмысын басқаратын процедуралар мен функциялардан, тұрақтылардан және келген типтер жиынынан тұрады. GRAPH модуліне жататын ішкі программалар көмегімен шрифтары әртүрлі немесе стандартты мәтіндерді және кез келген графиктік бейнелерді экранға шығаруға болады. GRAPH модулінің ішкі программалары сәйкестеніп орнатылса, аппаратты графиктік жабдықтардың әртүрлі типін қолдай алады.

Графликтік драйверді жүктеу және графиканы анықтау `initgraph` процедурасының көмегімен жүзеге асырылады.

Форматы: **initgraph (Драйвер, Режим; Драйверге жол).**

-драйвер ~ `integer` типті айнымалы, графликтік драйвер типін анықтайды;

-режим ~ `integer` типті айнымалы, адаптер басқару режимі:

-драйверге жол ~ `string` типті айнымалы, драйвер жолын көрсетеді.

Негізгі процедуралары мен функциялары

`Init Graph (gd, gm, path)` - графликтік режимді орнату;

`Graph Result` - графикаға көшу кезінде қате кодын қайтаратын бүтін сан;

`Get Graph Mode` - ағымдық графликтік режимнің кодын қайтарады;

`Get Graph Mode (Mode)` - жаңа графликтік режимді орнату;

`Restore Grt Mode` - мәтіндік режимге уақытша көну;

`Close Graph` - графликтік режимнен шығу, оған дейінгі режим қайта калыпка келеді;

`Clear Device` - графликтік режимде экранды тазарту.

Координаталармен жұмысқа арналған негізгі функциялар:

-`Get Max X ~ X` -тің максималды координатасын қайтару;

-`Get Max Y ~ Y` -тің максималды координатасын қайтару;

-`Get X ~ X` ағымдық координатасын қайтару;

-`Get Y ~ Y` ағымдық координатасын қайтару;

-`Get Pixel (X, Y) ~ (X, Y)` координаталы нүктелер түсін қайтару.

Графликтік көріністер салудың кейбір процедуралары:

-`Put Pixel (X, Y Color) ~` экранға (X, Y) координаталы, `Color` түсті нүкте шығады;

-`Line (X1, Y1, X2, Y2) ~ (X1, Y1)` координаталы нүктелерден $(X2, Y2)$ координаталы нүктелерге түзу сызық жүргізу;

-`Move To (X, Y) ~` ағымдық нұсқағышты (X, Y) нүктесіне орнату;

-`Line To (X, Y) ~` ағымдық нұсқағыш тұрған жерден (X, Y) координаталы нүктеге дейін түзу сызық жүргізу. Ағымдық нұсқағыш (X, Y) нүктесіне орналасады.

-`Rectangle (X1, Y1, X2, Y2) – (X1, Y1)` - жоғарғы сол жақ бұрыш және $(X2, Y2)$ төменгі оң жақ бұрыш. Төртбұрыш сызу.

-`Bar (X1, Y1, X2, Y2) –` шрихталған тік төртбұрыш. $(X1, Y1)$ – жоғарғы сол жақ бұрыш. $(X2, Y2)$ төменгі оң жақ бұрыш. Стандартты түстер мен күйе стилі қолданылады.

-`Circle (X, Y, Radius) – (X, Y)` нүктесімен ортасы берілген, `Radius` арқылы радиусы берілген шеңбер.

-Ellipse (X, Y, St Angle, End Angle, Xradius, Yradius) – (X,Y) эллипс ортасы, St Angle =0 және End Angle =36 ~ толық эллипс сызылады.

Мысалы, үйшік салу программасы

Uses graph;

Var gd, dm:integer;

Begin

Gd:=detect;

Initgraph (gd, gm, ‘’);

If graphresult <> grok then begin

Writeln (‘графиканы жүктеу кезіндегі қате’);

Readln; halt; (программадан шығу);

End;

Restangle (280, 180, 360, 300);

Bar (305, 210, 335, 270);

Line (280, 180, 320, 150);

Line (320, 150, 360, 180);

Circle (320, 165, 8);

Readln;

CloseGraph;

End.

Түсі орнату және фигура стилін орнату

Set Color (Color) – көріністің түсін орнату;

Set Bk Color (Color) –фон түсін орнату;

Set Line Style (Style, Pattern, Thickness) – сызық параметрлерін орнату;

Set Fill Style (Style, Color) – штриховка түсін, түсін орнату.

14-Дәріс. Программаларды ұйымдастыру. Әртүрлі құрылымдық ұйымдасқан программалардың мысалдары.

Құрылымдық программалау.

Файлдық тип-компоненттері файлдық тип немесе компоненті файлдық тип болатын құрылымдық типтен басқа кез келген тип болатын сызықты тізбектен тұрады. Компоненттер саны файлдық типті хабарлауда бекітілмейді. Файлдың әрбір элементі 0 – ден бастап нөмірленеді. Паскальда файлдардың үш түрі бар: типтелген, текстік және типтелмеген. Паскальдағы кез келген файлдық айнымалы ол файлдық типтегі кез келген айнымалы.

Файлдық тип типті сипаттаудың келесі көрсетілген операторларының көмегімен хабарланады:

Type TF= File of

Файлдық айнымалыны қолданудың алдында ол **ASSIGN** процедурасының көмегі арқылы сыртқы файлмен байланыстырылуы керек. Әдетте сыртқы файл сыртқы есте сақтау құрылғысында аты аталған деректер жиыны, сондай – ақ ол пернелік немесе дисплей сияқты құрылғы да болуы мүмкін. Сыртқы файлдарда файлдарға жазылған ақпарат сақталады немесе олар файлдан оқылатын ақпарат көзі ретінде қызмет атқарады.

Сыртқы файлмен байланыс орнатылғаннан кейін файлдық айнымалыны енгізу немесе шығару операциясына дайындау үшін ол «ашылған» болуы керек. Бар файлды **RESET** процедурасының көмегімен құрып, ашуға болады, ал жаңа файлды **REWRITE** процедурасының көмегімен құрып, ашуға болады.

Файлдарға қатынау әдетте жүйелік түрде ұйымдастырылады, яғни элемент стандартты **READ** процедурасымен оқылған кезде немесе стандартты **WRITE** процедурасының көмегімен жазылған кезде, файлдың ағымдағы позициясы реті бойынша келесі элементке көшеді.

Аталған **READ** және **WRITE** процедураларынан басқа да, **END-OF-FILE** жағдайы туған кезде (басқаша айтқанда файлдан тыс жерден оқуға әрекет етілген кезде) **TRUE** мәнін қабылдайтын **EOF** логикалық функциясы да жиі қолданылады.

EOF функциясының форматы:

EOF (Name)

Мұндағы **NAME** ретінде **EOF** жағдайына тексерілетін файлдың аты көрсетіледі. Программа файлмен жұмысын аяқтағаннан кейін ол **CLOSE** процедурасының көмегімен жабылуы керек.

Файл жабылғаннан кейін онымен байланысқан сыртқы файл толығымен жаңартылады. Кейіннен файлдық айнымалы басқа сыртқы файлмен байланыстырыла алады. Енгізу-шығарудың стандартты функциялары мен процедураларына қатынаған кезде қателерді тексеру өздігінен автоматты түрде орындалады. Қате табылған жағдайда, экранға қате туралы хабар беріледі және программа өзінің орындалуын тоқтатады. Қателерді автоматты түрде тексеруді компилятордың директивасының көмегімен қосуға **{S1+}** немесе алып тастауға **{S1-}** болады. Тексеруді алып тастаған кезде енгізу-шығарудың қателері программаның тоқталуына әкелмейді.

Бұл жағдайда енгізу-шығару амалдарының орындалу нәтижесін тескеру үшін бүтін сан қайтаратын **IORESULT** стандартты функциясы қолданылады. Егер бұл функцияның нәтижесі нөлге тең болса, онда алдыңғы енгізу-шығару операциясы ойдағыдай аяқталған. Кері жағдайда бұл сан қатенің кодын көрсетеді (олардың кейбіреуі төменгі кестеде келтірілген).

Орнату және аяқтау амалдары

Бұл топқа келесі стандартты процедуралардың көмегімен жүзеге асырылған амалдар кіреді: **ASSIGN**, **APPEND**, **RESET**, **REWRITE**, **CLOSE**.

1. ASSIGN процедурасы файлдық айнымалымен сыртқы файлды байланыстыру үшін арналған және форматы келесідегідей:

Assign(F, Str);

мұндағы F — файлдық айнымалының аты, STR сыртқы файлдың немесе псевдофайлдың (нақты құрылғының: CON -енгізу үшін пернелік және шығару үшін экран; LPT1, LPT2, PKN — параллель порттарға жалғанған құрылғылар; COM1, COM2, AUX — тізбектелінген порттарға жалғанған құрылғылар; NUL — жалған құрылғы) аты болатын жолдық өрнек.

Мысалдар:

a) Assign(F, 'D:\stud\spisok.dat');

F файлдық айнымалысын D:\ дискісіндегі STUD каталогында орналасқан SPISOK.DAT файлымен байланыстырады.

5) Assign(F, 'Con:');

F файлдық айнымалысын консольмен (енгізген кезде пернелікпен және шығарған кезде экранмен) байланыстырады.

2. RESET процедурасы файлды ашады, файлдың көрсеткішін бастапқы орнына (бірінші элементтің алдына) қояды және форматы келесідегідей:

Reset(F);

мұндағы F — файлдық айнымалының аты.

RESET процедурасының орындалуы алдында файлдық айнымалы сыртқы файлмен немесе құрылғымен байланыстырылуы керек.

3. REWRITE процедурасы файлды ашады, файлдың көрсеткішін бастапқы орнына (бірінші элементтің алдына) қояды және форматы келесідегідей:

Rewrite (F);

мұндағы F — файлдық айнымалының аты.

Егер REWRITE процедурасының орындалуы алдында файлдық айнымалымен байланысқан сыртқы файл жоқ болса, онда ол құрылады. Егер бар болса, онда ол файл жойылады және деректердің бос жпыны болып қайта құрылады.

4. **APPEND** процедурасы бар файлды қосымша жасауға ашады және форматы келесідегідей:

Append (F);

мұндағы F — файлдық айнымалының аты.

5. **CLOSE** процедурасы файлды жабады және форматы келесідегідей:

Close (F);

мұндағы F - файлдық айнымалының аты.

Программаның жұмысы аяқталғаннан кейін программада ашылған барлық файлдар автоматты түрде жабылады.

Арнайы амалдар. Бұл топтың амалдары файлдық жүйенің элементтерімен (каталогтар және файлдардың аттары) әрекеттер жасауға арналған.

1. **ERASE** процедурасы сыртқы файлды жою үшін арналған және форматы келесідегідей:

Erase (F);

мұндағы F - жойылатын файлмен алдын ала байланысқан файлдық айнымалының аты.

2. **RENAME** процедурасы сыртқы файлдың атын өзгертуге арналған және форматы келесідегідей:

ReName(F,STR);

мұндағы F — аты өзгертілетін файлмен байланысқан файлдық айнымалының аты, STR - сыртқы файлдың жаңа атын құратын жолдық өрнек.

3. **CHDIR** процедурасы ағымдағы каталогты орнатуға арналған және форматы келесідегідей:

ChDir(STR):

мұндағы STR - C:\CҚ-ғы каталогтың атын құратын жолдық өрнек.

4. **MKDIR** процедурасы жаңа каталогты құру үшін арналған және форматы келесідегідей:

MkDir(STR);

мұндағы **STR** - СЕСҚ-ғы каталогтың атын құратын жолдық өрнек.

5. **RMDIR** процедурасы каталогты жою үшін арналған және форматы келесідегідей:

Rmdir (STR);

мұндағы **STR** — СЕСҚ-ғы каталогтың атын құратын жолдық өрнек.

Текетік файлдар

Бұл бөлімде стандартты **TEXT** типіндегі файлдық айнмалысын пайдаланатын енгізу/шығару амалдары сипатталады. Бір айта кететін жай, Паскальда **TEXT** типі **FILE OF CHAR** типінен өзгеше.

15-Дәріс. Модулдік программалау. Программалық тілдерді талдау. Программалау тәсілі. Программалаудың сапалық көрсеткіші.

Кітапханалық модуль баяндау бөлімінің түрлі элементтерін кейбір амалдарды қамтитын және компиляцияланатын программалық бірлік. Модуль құрамына тұрақтыларды, айнымалыларды, типтерді, процедуралар мен функцияларды баяндау кіреді. Жұмыс барысында программалаушы өзі үшін пайдалы модульдердің тұтас коллекциясын жинақтайды. Ол өзінің жеке кітапханасы болып табылады.

Модульдік программалаудың пайдасы:

-процедура мен функциядан айырмашылығы программаның бастапқы кодына қосылған модульдерді дискіде компиляцияланған түрде сақтауға болады. Бұл жағдайда программаны дайындау процесі аз уақыт алады, себебі негізгі программа ғана компиляцияланады да, модуль коды компоновкалау кезінде қосылады.

- Турбо Паскальда программаны орындау кезінде модульдің орындамалық коды жадының жекеленген сегменттерінде орналасады, бір мезгілде қолданылатын модульдер саны жады көлемімен шектеледі. Бұл үлкен көлемді программа құруға мүмкіндік береді.

-жеке-жеке модульдер түрлі программаларда құрылуы мүмкін.

Модуль құрылымы. Кез келген модульдің бастапқы мәтіндерін бірнеше бөлімдерге бөлуге болады:

- тақырыбы;
- интерфейстік бөлімі;
- орындалатын бөлім;
- анықталатын (иницирующая) бөлім.

Жалпы құрылымы

Unit <аты>; {модуль тақырыбы}
{SR+} {компилятордың глобалды директивалары}
interface {интерфейстік бөлімнің басы}
uses ... {модульдер тізімі}
label ... {қатынас құру Тамғаларын баяндау}
const ... {қатынас құру тұрақтыларын баяндау}
type ... {қатынас құру типтерін баяндау}
var ... {қатынас құру айнымалыларын баяндау}
procedure ... {қатынас құру процедураларының тақырыбы}
function ... {функциялар тақырыбы}
implementation {орындамалық бөлімнің басы}
uses ... {модульдерді жүзеге асыруда қолданылатындар}
label ... {жасырын глобалды тамғаларды баяндау}
const ... {жасырын глобалды тұрақтыларды баяндау}
type ... {жасырын глобалды типтерді баяндау}
var ... {жасырын глобалды айнымалыларды баяндау}
procedure . . . {жалпы және жасырын процедуралардың тақырыбы мен денелері}
function . . . {жалпы және жасырын функциялардың тақырыбы мен денелері}
begin ... {кез келген операторлар}
end. {модуль соңы}

Білу керек:

-Модуль аты басқа модульдермен және негізгі программамен байланыс үшін керек (ол міндетті);

- Модуль аты модульдердің бастапқы мәтіні орналасатын файл атымен сәйкес болу керек; мысалы unit triangle, онда сәйкес модульдің бастапқы мәтіні triangle.pas файлында орналасуы маңызды;

- Интерфейстік бөлімінде кез келген программаға немесе модульге қолдануға ыңғайлы хабарлар жазылады;

- Орындалатын бөлімде интерфейстік бөлімде баяндалған ішкі программаның толық суреттелуі қамтылады;

- Орындалатын бөлімде баяндалған барлық қосымша программалық элементтер жасырын деп аталады, себебі олар сол модульде ғана қолданылады, ал сол модульді қолданатын программада көрінбейді;

- Аяқтау бөлім модульді аяқтайды.

Ол (begin сөзінен басталады) бос болуы мүмкін. Ондай жағдайда begin сөзінен кейін бірден end қойылады.

Модульді программаға қосу үшін немесе басқа модульге қосу үшін uses модульдер тізімі қолданылады. Программада бұл басқа баяндаулардан бұрын орналасады.

Мысалы: **uses crt, graph, triangle;**

Модуль тақырыпшасы және модульдердің бір-бірімен байланысы

Модуль тақырыпшасы:

UNIT <модуль аты>;

Модуль аты басқа модульдермен және негізгі программамен байланысу үшін қажет:

USES <модульдер тізімі >

Мұндағы USES – қордағы сөз (қолданылады);

<модульдер тізімі> - байланыс орнатылатын модульдер тізімі; тізім элементтері бір-бірінен үтір арқылы бөлінген, модуль аттары болып табылады, мысалы:

Uses CRT, GRAPH, METHOD;

Егер USES... баяндауы қолданылса, ол негізгі программаны сипаттау бөлімін ашу керек. Модульдер басқа модульдерді қолдана алады.

Интерфейстік бөлім

Интерфейстік бөлім INTERFACE қордағы сөзімен ашылады. Бұл бөлімде модульдің барлық глобальды объектілерін баяндау (типтер, тұрақтылар, айнымалылар және ішкіпрограммалар) қамтылады, өз кезегінде негізгі программаға және басқа модульдерге қол жеткізу мүмкіндігін береді.

Орындамалық бөлім

Орындамалық бөлім IMPLEMENTATION сөзінен басталады және интерфейстік бөлімде баяндалған ішкіпрограммалар сипаттамасын қамтиды.

Модульдің интерфейстік бөлімінде баяндалған ішкіпрограмманы сипаттаған кезде ішкіпрограмма тақырыпшасын жазбаса да болады, себебі ол интерфейстік бөлімде баяндалып кетеді. Егер ішкіпрограмма толық қамтылса, онда ол интерфейстік бөлімде баяндалған тақырыпшамен сәйкес болу керек.

Анықтаушы бөлім

Бұл бөлімде программаның кейбір фрагменттерін қамтитын орындалатын операторлар орналасады. Бұл операторлар негізгі программаға басқару берілгенге дейін орындалады.

Модульдерді компиляциялау

Компиляцияның үш режимі анықталған: COMPILE, MAKE және BUILD. Режимдер бір-бірінен негізгі программаның немесе компиляцияланатын модульдің USES сөйлемінде баяндалған басқа модульдермен байланысу тәсілдері арқылы ерекшеленеді.

COMPILE режимінде USES сөйлемінде келтірілген барлық модульдер алдын ала компиляцияланған болу керек, және компиляция нәтижелері TPU кеңейтілулі аттас файлга орналасу керек.

MAKE режимінде компилятор TPU – файлдардың бар болуын тексереді (әрбір баяндалған модуль үшін). Егер қандай да бір TPU-файл баямаса, немесе модульдің бастапқы мәтінінде қандай да бір өзгеріс болса, онда жүйе PAS кеңейтілулі аттас файлды компиляциялайды.

BUILD режимінде жүйе USES сөйлемінде баяндалған әрбір модуль үшін сәйкес PAS-файлдарды компиляциялайды.

Семинар және практика сабақтарының қысқаша сипаттамасы (жоспарлар, семинар және практика сабақтарын жүргізуге арналған тапсырмалар, СӨЖ, СӨЖ);

1. Қатарлар. Қатарлармен жұмыс істеу үшін қолданылатын процедуралар мен функциялар.
2. Стандартты функциялар қорын пайдаланып программа құру.
3. Модульдік программалау, қолдану мысалдары.
4. Графикалық құрастыруларды қолданып программа құру.
5. Динамикалық жадыны қолданып программа жасау.

Зертханалық және студиялық сабақтардың тақырыптары мен қысқаша сипаттамалары

№1-Зертханалық жұмыс.

Өрнекті жазу, меншіктеу операторы. Паскаль тілінде алгоритмдердің сызықты құрылымының программаларын жазу.

Жұмыстың мақсаты: Сызықтық құрылымдық есептеу процесінде алгоритм құрудың практикалық дағдысын, оны программалау (Турбо Паскаль жүйесіне) программаны аударып,

енгізіп, оны жүргізе білу іскерлігін, стандартты типтер мен стандартты функцияларды пайдалана білу дағдысын, енгізу, шығару және меншіктеу операторларын пайдалана білу іскерлігін қалыптастыру.

Бақылау мысалы

$$y = \ln \frac{1 + \sqrt{x^2 + 1}}{x} - \sqrt{x^2 + 1}$$

Program primer1;

var x:integer;

y real;

begin

write('x=');

readln(x);

y:=ln(1+sqrt(sqrt(x)+1)/x)-sqrt(sqrt(x)+1);

writeln ('y=',y:6:3);

end.

Төменде берілген есептерді әр түрлі типтермен сипаттау арқылы есептің блок-схемасын және программасын жазыңыз.

Жұмысты орындауға арналған тапсырма нұсқалары

$$1) P = \frac{x \sin x}{(2x - y)^2} - \frac{\sqrt{2x - y}}{3y \cos y}$$

$$2) P = \frac{\sqrt{a^2 + b^2 + c^2}}{2a + b \sin c} - \frac{\ln(a + 2bc)}{a^2 + b^2 + c^2}$$

$$3) Z = \frac{x^3 + y^2}{y \sin x} - \frac{\ln(y + 5e^{y+x})}{\sqrt[4]{x^3 + y^2}}$$

$$4) P = \frac{e^{2a} + 2,5 \sin 3b}{\sqrt{a^2 + b^2}} - 2 \operatorname{tg}(a^2 + b^2)$$

$$5) Y = \sqrt{a^2 + x^2} + \frac{\operatorname{tg}(a+x) - \ln(a+x)}{\sin x + \cos^2 a}$$

$$6) Z = \frac{\sin^2(x+y)}{(x+y)^2} - \frac{\sqrt{|3x-4y|}}{\cos x + e^{5x}}$$

$$7) Y = \cos x - \frac{\ln x + \sin^2(5x+2)}{5,3x + \frac{7}{\sqrt{2x^2+20}}}$$

$$8) P = \frac{\ln a + \ln b}{x + \frac{ax+b}{\sqrt{x^2+2b^4}}} - \operatorname{tg}(ax+b)$$

$$9) Y = \frac{\sqrt{|5x-45|}}{\operatorname{tg} x^2 + \operatorname{tg}^2 x} + \frac{\sqrt{2x^2+6}}{\ln|5x-45|}$$

$$10) Y = \cos^3 a + \frac{2ab - \sqrt{a^2 + b^2}}{e^{2ab} + \ln a}$$

$$11) Z = \frac{\operatorname{tg}(x+y) - |x-5y|}{\sin 5,3x + \frac{2x}{\sqrt{|x-5y|}}} - \frac{\cos^3 y}{x+y}$$

$$12) P = \frac{x \sin^y x}{\cos x + e^x} - \sqrt{|\sin^y x + \cos^2 x|}$$

$$13) \quad z = \frac{y \sin x - \cos x^2 y^3}{\sqrt{|x^y - \cos x + 1|}} + \ln x$$

$$14) \quad z = \frac{\cos^2(x+y) - 2x}{\sin^3(x+y) + 3y} - \sqrt{x^4 + y^2}$$

$$15) \quad p = 2 \operatorname{tg}(a + 3b) + \frac{e^{2a} + 2,5 \operatorname{Sin} 3b}{\sqrt{|a + 3b|} - 5}$$

$$16) \quad z = \frac{\sqrt{x^2 + y^4}}{\cos^2 x + e^{x+y}} + \frac{\sin^2(x+y)}{2x+y}$$

$$17) \quad p = \frac{y \sin^2 x}{y^{x-2} + e^{xy}} - \frac{\operatorname{tg} x - 3y}{x-2}$$

$$18) \quad y = \frac{7 \cos a}{\sqrt{|5a + 2b|}} - \frac{\sin^2(5a + 2b)}{e^{\cos a}}$$

$$19) \quad y = \frac{\sin^2(5x+2)}{5,3x+9} + \frac{7x - e^{2x}}{\sqrt{|5x+2|}}$$

$$20) \quad p = \frac{\sin a + \cos b}{x + \operatorname{tg}(ax+b)} - \frac{a^3 + \sin b}{\sqrt{x^2 + 7b^2}}$$

$$21) \quad y = \frac{\operatorname{tg} x^2 - \sqrt{2x^2}}{\ln|5x-45|} - \frac{\sqrt{|5x-45|}}{x + \operatorname{tg}^2 x}$$

$$22) \quad y = \frac{\sqrt{a^2 + 5b^2}}{e^{2ab} + \ln a} + \cos^3(a^2 + 5b^2)$$

$$23) \quad z = \frac{\cos y}{\operatorname{lg} x - |y-x|} + \frac{x+y - \sqrt{|xy|}}{5x + \frac{2xy}{\sqrt{|y-x|}}}$$

$$24) \quad z = \frac{x \sin x - \cos x}{x^y + \ln x} + \sqrt{|x^y - x \sin x|}$$

$$25) \quad z = \frac{\sin^2(x^4 + y^2) - 2}{\sin^3(xy) + e^{2y-x}} - \sqrt{x^4 + y^2}$$

$$26) \quad y = \frac{2 \sin a}{\sqrt{|a+2b|}} - \frac{\cos^2(a+2b)}{e^{\sin a}}$$

$$27) \quad p = \frac{\sin(ax+b)}{x + \operatorname{tg}(ax+b)} - \frac{a^3 + \sin b}{\sqrt{x^2 + 3b^2}}$$

$$28) \quad p = \frac{x + \cos x}{(3y-x)^2} - \frac{\sqrt{|3y-x|}}{3y \cos x}$$

$$29) \quad p = \frac{\sqrt{a^2 + b^2}}{a + 2bc} - \frac{\ln(a + 2bc)}{a^2 + b^2 + c^2}$$

$$30) \quad p = \frac{\sin^2 x}{y^{x-2} + e^{xy}} - \frac{x-2}{\cos y^2}$$

31. Кубтың қыры берілген. Кубтың көлемі мен бүйір бетінің ауданын анықтаңдар.

32. Екі санның арифметикалық және геометриялық ортасын есептеп, нәтижесін шығаратын операторлар тізбегін жазыңыздар (он сандар берілген деп есептейміз). x, y сандарының геометриялық

ортасы: $\sqrt{x \cdot y}$, ал $a_1, a_2, a_3, \dots, a_n$ сандарының геометриялық ортасы: $\sqrt[n]{a_1 \cdot a_2 \cdot a_3 \cdot \dots \cdot a_n}$

33. Екі санның қосындысын, айырмасын және көбейтіндісін есептейтін программа жазыңыз.

34. Берілген екі катеті бойынша тікбұрышты үшбұрыштың гипотенузасы мен ауданын есептейтін программа жазыңыз.

35. Координаттары x_1, y_1 және x_2, y_2 болып келген екі нүкте ара қашықтығын табындар.

36. x саны берілген. $2x^4 - 3x^3 + 4x^2 - 5x + 6$, $1 - 2x + 3x^2 - 4x^3$ және $1 + 2x + 3x^2 + 4x^3$ өрнектерін есептейтін программа құрыңыздар. Мұнда көбейту, қосу және алу амалдарын ғана қолдануға болады. Орындалатын операциялар саны мүмкіндігінше аз болатын болсын.

37. Тең қабырғалы үшбұрыштың қабырғасы берілген. Осы үшбұрыштың ауданын табатын программа құрыңыздар.

38. x, y, z берілген. a, b -ны есептеу керек:

$$a = \frac{\sqrt{|x-1|} - \sqrt{|y|}}{1 + \frac{x^2}{2} + \frac{y^2}{4}} \quad b = x * (\text{arctg}(z) + e^{-(x+z)})$$

39.

40. x, y, z берілген. a, b -ны есептеу керек:

$$a = \frac{3 + e^{y-1}}{1 + x^2 * |y - \text{tg}(z)|}, \quad b = 1 + |y - x| + \frac{(y-x)^2}{2} + \frac{|y-x|^3}{3}$$

№2-Зертханалық жұмыс.

Алгоритмдердің тармақталған құрылымдарын программалау
Жұмыстың мақсаты: тармақталуы бар процесстерге алгоритмдер құра білу дағдысын қалыптастыру, есеп шығаруда белгілерді, тұрақтыларды, жай және құрама шарттарды пайдалана білу іскерлігін қалыптастыру.

Бақылау мысалы. IF шартты операторды колдана отырып INTEGER форматындағы a, b мәліметтер үшін берілген бүтін сандық өрнектердің әдісін табындар. Алгоритмның блок –схемасын құру және программасын жазу.

$$1. \quad X = \begin{cases} b/a - 2, & \text{есер } a > b, \\ -7, & \text{есер } a = b, \\ (a^3 - 3)/a, & \text{есер } a \leq b; \end{cases}$$

Program primer2;

Var a, b: integer;


```

X: real;
Begin
Write('a='); readln(a);
Write ('b=');readln(b);
If (a=b) then x:=-7 else
If(a>b)then x:=a/b-2 else x:=(sqr(a)*a-3)/a;
Writeln('X=',x:6:2);
End.

```

$$2. \quad y = \begin{cases} e^x, & x < 0 \\ 0, & 0 \leq x \leq 1 \\ \sin x, & x > 1 \end{cases}$$

```

Program primer2_1;
Label m1;
Var i, n:integer; x, y:real;
Begin
Write('N ді енгіз');
Read(n);
i:=1;
m1: Write('x ті енгіз);
Read(x);
If x<0
then y:=exp(x)
else
if(x>=0)and(x<=1)
then y:=0
else y:=sin (x);
write ('Y=',y:10:3);
i:=i+1;
if i<=n then goto m1
end.

```

Төмендегі тапсырмаларды орындауда If шартты операторды қолдана отырып Integer форматындағы a, b мәліметтер үшін берілген бүтін сандық өрнектердің есептеу әдісін табыңыз. Алгоритмнің блок-схемасын және программасын жазыңыз.

Жұмысты орындауға арналған тапсырма нұсқалары

$$1. \quad y = \begin{cases} |(2x-7)|, & x \leq -3 \\ 5(x+1)^2 + 5(x+2), & -3 < x \leq 0 \\ e^{-4x} - 7x, & x > 0 \end{cases}$$

$$2. \quad y = \begin{cases} \sin x + 2, & x < 1 \\ \cos x - 2, & 1 \leq x \leq 5 \\ \operatorname{tg} x, & x > 5 \end{cases}$$

$$3 \ y = \begin{cases} \sqrt{x^2 + 5}, & x < 1 \\ 3x + \cos(x + 5.7), & x \geq 1 \end{cases}$$

$$4 \ c = \begin{cases} a + b, & a \leq b \\ b^2 + 2ab, & a < b \end{cases}$$

$$5 \ y = \begin{cases} x^2 + 2x, & x > 0 \\ \operatorname{ctg} 2x, & x \leq 0 \end{cases}$$

$$6 \ y = \begin{cases} \lg(4x^2 + 5), & x \geq 0 \\ \sqrt{4x^2 + 5}, & x < 0 \end{cases}$$

$$7 \ y = \begin{cases} \sin x^2 + 4, & x \geq 2 \\ \operatorname{ctg}(\sqrt{5}x), & x < 2 \end{cases}$$

$$8 \ y = \begin{cases} \cos x + 5x - 2, & x \geq 2 \\ \operatorname{tg} x^2, & x < 2 \end{cases}$$

$$9 \ y = \begin{cases} x^2 + 5, & x > 5 \\ x^2 + 5x + 2, & x \leq 5 \end{cases}$$

$$10 \ y = \begin{cases} 2x^4 + \sqrt{x-4}, & x \geq 4 \\ x + 12 - 5x^2, & x < 4 \end{cases}$$

$$11 \ y = \begin{cases} \cos(x\sqrt{|x+2|}), & x \leq 0 \\ \operatorname{tg} x - \sin^2 x, & 0 < x < 1 \\ 3x\sqrt{x}, & x \geq 1 \end{cases}$$

$$12 \ y = \begin{cases} (x + \cos 2x)^{-2}, & x < 5 \\ (x + x^3) - \sin 7x, & x > 5 \\ 5 \operatorname{ctg}^2 x, & x = 5 \end{cases}$$

$$13 \ y = \begin{cases} \ln 2x - \cos^2 x, & x < 0 \\ \operatorname{tg} 3x, & x > 0 \\ 5, & x = 0 \end{cases}$$

$$14 \ z = \begin{cases} y^2 - 0.3, & y < 0 \\ e^{y+4} - \sin(y+4), & 0 \leq y \leq 1 \\ \operatorname{ctg}(y^2 + y), & y > 1 \end{cases}$$

$$15 \ Y = \begin{cases} \sin x, & x = 0 \\ \cos x, & x > 0 \\ \sqrt{x^2 + 3}, & x < 0 \end{cases}$$

$$16 \ Y = \begin{cases} (2x+5)^2 + \sqrt[3]{x}, & x > 5 \\ \frac{5x^2 - 7}{x^3 + 2x^2 + 1}, & x \leq 5 \end{cases}$$

$$17 \ y = \begin{cases} \sin^2(x-a), & x > 0 \\ 2a + \sqrt{7x}, & x \leq 0 \end{cases}$$

$$18 \ y = \begin{cases} \sin x + 2, & x > 0 \\ \lg(x+2), & x \leq 0 \end{cases}$$

$$19 \ y = \begin{cases} \lg x, & x > 7 \\ \sqrt[3]{\sin x}, & x \leq 7 \end{cases}$$

$$20 \ y = \begin{cases} |1-x|, & x > 7 \\ \operatorname{ctg} x - \operatorname{tg} x, & x \leq 7 \end{cases}$$

$$21 \ y = \begin{cases} (\sqrt{x^2 + 6x - 1})^2, & x < 3 \\ \sqrt[3]{7x^2 + 6x - 1}, & x \geq 3 \end{cases}$$

$$22 \ y = \begin{cases} a - b, & a > b \\ a + b, & a < b \\ a(b-1), & a = b \end{cases}$$

$$23 \ y = \begin{cases} \frac{|a-b|}{a+b}, & a < b \\ \sqrt{a-b}, & a \geq b \end{cases}$$

$$24 \ y = \begin{cases} 3 \cdot 1x^4 + 5, & x < 2 \\ \sqrt[3]{x^2 + x + 7}, & x \geq 2 \end{cases}$$

$$25 \quad y = \begin{cases} e^{2x} + 2,5 \sin x, x < 3 \\ \sqrt{x+1}, x > 3 \\ 45, x = 3 \end{cases}$$

$$26 \quad y = \begin{cases} \frac{a}{a+b}, a > b \\ \frac{b}{a+b}, a \leq b \end{cases}$$

$$27 \quad y = \begin{cases} x^2 + 2x, x > 0 \\ \sqrt{2}, x = 0 \\ 2x - x^4, x < 0 \end{cases}$$

$$28 \quad y = \begin{cases} x^3 + 5x + 3, x \geq 1 \\ |12x - 2x^2| + \sqrt[4]{4x}, 0 \leq x < 1 \\ \sqrt{4x^2 + 5}, x < 0 \end{cases}$$

$$29 \quad y = \begin{cases} \sin x^2, x = 0 \\ \cos^2 x, x > 0 \end{cases}$$

$$30 \quad y = \begin{cases} a^3 - b^2, a > b \\ \sqrt{|a+b|}, a < b \end{cases}$$

31 Нақты x, y ($x \neq y$) берілген. Кішісін олардың жарты қосындысымен, ал үлкенін - екі еселенген көбейтіндісімен алмастырыңыз.

32 Үш нақты сан берілген. Теріс емес сандарды квадраттаңыздар.

33 Егер берілген нақты x, y, z сандарының қосындысы 1-ден кем болса, онда бұл үш санның ең кішісін қалған екі санның жарты қосындысымен алмастырыңыз, кері жағдайда x және y -тің кішісін қалған екеуінің жарты қосындысымен алмастырыңыз.

34 Нақты a, b, c, d сандары берілген. Егер $a \leq b \leq c \leq d$ болса, онда әр санды ең үлкен санмен алмастырыңыз, егер $a > b > c > d$ болса, сандарды өзгеріссіз қалдырыңыз, кері жағдайда барлық сандарды олардың квадратымен алмастырыңыз.

35 Нақты x, y, z сандары берілген. Егер x -ке қалдықсыз бөлінсе және yz -ке қалдықсыз бөлінсе, онда барлық сандарға бірді қосыңыз, кері жағдайда барлық сандарды нөлге теңестіріңіз.

36 Нақты a, b, c сандары берілген ($a \neq 0$). $ax^2 + bx + c = 0$ теңдеуінің нақты түбірлері бар екенін тексеріңіз. Егер бар болса оларды есептеңіз, кері жағдайда экранда "нақты түбірі жоқ" деген мәлімдемесі шығу керек.

37 Нақты x, y сандары берілген. Егер x және y оң сан болса, онда оларды түбірден шығарыңыз; егер тек біреуі оң болса, онда оларды квадраттаңыз. Егер x және y теріс сан болса, онда олардың абсолют шамасын табыңыз.

38 Нақты x, y сандары берілген. Егер x және y теріс сан болса, онда олардың әрқайсысын квадраттаңыз; егер тек біреуі теріс болса, онда оларды 10-ға кемітіңіз. Егер x және y оң сан болса және біреуі

[1;3] аралығында жатса, онда оларды түбірден шығарыңыз. Қалған жағдайда “Шарттың ешқайсысы орындалмады” деген мәлімдеме шығарыңыз.

39 Нақты a, b, c, d сандары берілген. Осы сандардың терістерін квадраттап, ал оң сандарын түбірден шығарыңыз.

40 Нақты a, b, c, d сандары берілген. Егер кем дегенде бір сан нөлге тең болса, ол жайлы экранға мәлімет шығарыңыз, кері жағдайда a -ның b -ға және c -ның d -ға қалдықсыз бөлінетіндігін тексеріңіз.

41 Бүтін a, b, c сандары берілген. Егер $a \leq b \leq c$ болса, онда барлық сандарды олардың квадратымен алмастырыңыз; егер $a \cdot b > c$ болса, онда әр санды ең үлкен санмен алмастырыңыз, кері жағдайда барлық сандардың таңбасын кері таңбаға ауыстырыңыз.

42 Нақты x, y, z сандары берілген. $\max(x + y + z, x * y * z) + 10$ өрнегін есептейтін программа жазыңыз.

43 Нақты x, y, z сандары берілген. $\max(x^2 + y^2, y^2 + z^2) - 1$ өрнегін есептейтін программа жазыңыз.

44 Бүтін k, l, m сандары берілген. Оң сандардың квадратының қосындысын есептеңіз. Егер бір де бір оң сан жоқ болса, ол жайлы экранға мәлімет шығарыңыз.

№3-4. Зертханалық жұмыстар.

Алгоритмнің циклдік құрылымдарын программалау.

Жұмыстың мақсаты: қайталанатын алгоритмдерді программалай білу дағдысын қалыптастыру, есеп шығаруда қайталану саны белгілі және белгісіз болып келген қайталану процесстерін ұйымдастыра білу іскерлігін; Паскаль тіліндегі қайталану операторларын дұрыс пайдалана білу және олардың ерекшеліктерін ажырата білу, операторларды алмастыра пайдалана білу іскерлігін қалыптастыру.

Бақылау мысалы. $Y = A * K^2$, егер K 1 мен 10 арасында өзгерсе, бұл есепті бірнеше жолмен шығарайық

```

Program Primer3_1;
var k: integer; a, y: real;
begin
  read (a);
  for k := 1 to 10 do
    begin
      y := a * sqr(k);
      writeln('K=',k:2,'Y=',y:10:3)
    end
  end.

```

```

Program Primer3_2;
var k: integer; a, y: real;
begin
  read (a);
  for k := 10 downto 1 do
    begin
      y := a * sqr(k);
      writeln('K=',k:2,'Y=',y:10:3)
    end
  end.

```

```

Program primer3_3;
var k:integer; a,y: real;
begin
  read (a);
  k:=1;
  while k<=10 do
    begin
      y := a * sqr(k);
      writeln('K=',k:2,'Y=',y:9:3);
      k := k + 1
    end
  end.

```

```

Program primer3_4;
var k:integer; a,y: real;
begin
  read (a);
  k:=1;
  repeat
    y := a * sqr(k);
    writeln('K=',k:2,'Y=',y:9:3);
    k := k + 1
  until k > 10
end.

```

Берілген n - бүтін және x - нақты сандары үшін берілген өрнекті есептеу алгоритмінің блок-схемасын және программасын құрыңыз. Өрнектердің мәні циклдік операторлар көмегімен есептеледі. №11 тапсырмадан бастап өрнектерді есептеу берілген дәлдікпен жүргізіледі. Программа құру барысында әртүрлі циклдік операторларды қолдану қажет.

Есептің нұсқалары (№1ден №10 нұсқалар үшін берілген n мен x үшін өрнектерді есептеу):

$$\begin{aligned}
 & 1. \sum_{i=1}^n \frac{x^i}{i!} \quad 2. \sum_{i=1}^n \frac{x + \cos(ix)}{2^i} \quad 3. \prod_{k=1}^n \left(1 + \frac{\sin(kx)}{k!} \right) \quad 4. \prod_{k=1}^n \left(\frac{k}{k+1} - \cos^k|x| \right) \\
 & 5. \prod_{k=1}^n \frac{(1-x)^{k+1} + 1}{((k-1)!+1)^2} \quad 6. \sum_{i=1}^n \left(\frac{1}{i!} + \sqrt{|x|} \right)^i \quad 7. \sum_{i=1}^n \frac{x^i + \sin(ix)}{3^i} \quad 8. \sum_{i=1}^n \frac{1 + \sqrt{|x|} + x^i}{i!} \\
 & 9. \prod_{k=1}^n \left(\frac{2k}{3k+1} - \cos^2|x| \right) \quad 10. \prod_{i=1}^n \frac{1 + (x+1)^i}{((k-1)!+3)^2}
 \end{aligned}$$

№11ден №20 варианттар үшін берілген n мен ε ($\varepsilon=10^{-6}$, берілген дәлдік) үшін қосындыларды есептеу

$$\begin{aligned}
 & 11. \sum_{k=1}^n \frac{1}{x^k k^2} \quad 12. \sum_{k=1}^n \frac{1}{x^2 + k^3} \quad 13. \sum_{k=1}^n \frac{x^2}{k^2} \quad 14. \sum_{k=1}^n \frac{1}{\sqrt{|x|} + k^2} \\
 & 15. \sum_{i=1}^n \frac{\sqrt{|x^3|}}{k^3} \quad 16. \sum_{k=1}^n \frac{x}{k^3 + k\sqrt{|x|} + 1} \quad 17. \sum_{k=1}^n \frac{(-1)^k x^k}{k} \quad 18.
 \end{aligned}$$

$$\sum_{k=1}^n \frac{(-1)^{k+1}}{(2k+1)!} * \left(\frac{x}{3} \right)^{4k+3}$$

$$19. \sum_{k=1}^n \frac{(-1)^k x^k}{(k+1)^2} \quad 20. \sum_{k=1}^n \frac{(-1)^k (k+1)kx^k}{3^k}$$

21. Кестедегі функциялар мәндерін x x_0 -ден x_k -ға дейін dx қадамымен өзгерген кезде анықтау керек.

Нұсқа	Функция	Берілгендері
1	$Y = \begin{cases} \sqrt{ a-x } \cdot \sin^2 x, & \text{егер } a < x; \\ \left(\frac{x}{ a+x }\right) \cdot \sqrt[3]{ \sin x }, & \text{егер } a = x; \\ e^{\sqrt{ x }}, & \text{егер } a > x; \end{cases}$	$a = 0.265 \cdot 10^2$ $x_0 = 10$ $x_k = 30$ $dx = 1.5$
2	$Y = \begin{cases} ax + 0.23x^2 \log_2 a, & \text{егер } a < x; \\ \left(\frac{xe^a}{ a+x }\right) \cdot \sqrt{ \cos x }, & \text{егер } a = x; \\ x \cdot \operatorname{tg} a, & \text{егер } a > x; \end{cases}$	$a = 1.5$ $x_0 = -2$ $x_k = 5$ $dx = 0.5$
3	$Y = \begin{cases} (a^2 + x^2) \cdot e^x, & \text{егер } a < x; \\ \sqrt{ a } \cdot \sin^4 x, & \text{егер } a = x; \\ e^{\frac{ a-x \cos x^4}{a}}, & \text{егер } a > x; \end{cases}$	$a = 2.5$ $x_0 = 0$ $x_k = 3$ $dx = 0.25$
4	$Y = \begin{cases} \ln a+x \cdot \cos x^4 , & \text{егер } a < x; \\ e^{x^2} - \sqrt{ a+x }, & \text{егер } a = x; \\ \frac{\sqrt[3]{ a+x }}{(a-x)}, & \text{егер } a > x; \end{cases}$	$a = 2.5$ $x_0 = 0$ $x_k = 3$ $dx = 0.25$
5	$Y = \begin{cases} a^2 + \sqrt{a^2 + x \cdot \sin x}, & \text{егер } a < x; \\ 2x^2 + a^3 \cdot \operatorname{tg} x, & \text{егер } a = x; \\ \frac{x^2}{\sqrt{ a }}, & \text{егер } a > x; \end{cases}$	$a = 0.637$ $x_0 = -3$ $x_k = 3$ $dx = 0.5$

22. Ең соңында 9999 саны тұрған бүтін сандар тізбегі берілген. Солардың:

- жалпы санын;
- арифметикалық ортасын;
- максималы мәнін;
- теріс және оң элементтері санын;
- минималы элементтің нөмірін;

- тізбек ішіндегі жұп сандар санын;
- жұп сандары ішіндегі ең кіші элементін анықтау керек.

№5-6.Зертханалық жұмыстар.

Тілдің басқарушы операторы: цикл операторы –for циклы, while, repeat цикл операторлары

Жұмыстың мақсаты: қайталанатын алгоритмдерді программалай білу дағдысын қалыптастыру, есеп шығаруда қайталану саны белгілі және белгісіз болып келген қайталану процесстерін ұйымдастыра білу іскерлігін; Паскаль тіліндегі қайталану операторларын дұрыс пайдалана білу және олардың ерекшеліктерін ажырата білу, операторларды алмастыра пайдалана білу іскерлігін қалыптастыру.

Бақылау мысалы. Берілген n натурал санынан аспайтын так сандардың қосындысын табу керек.

```
Var n, i, S:word;
```

```
Begin
```

```
  write ('n=');
```

```
  readln(n);
```

```
S:=0;
```

```
i:=2;
```

```
while i<=n do
```

```
begin
```

```
  S:=S+i;
```

```
  i:=i+2;
```

```
end;
```

```
writeln ('S=',S);
```

```
end.
```

Осы берілген есепті for ... to.. do қолданып шығаруымызға да болады.

```
Var n, i, S:word;
```

```
Begin
```

```
  write ('n=');
```

```
  readln(n);
```

```
S:=0;
```

```
For i:=1 to n do
```

```
  If i mod 2=0 then S:=S+i;
```

```
Writeln ('S=',S);
```

```
End.
```


Төмендегі тапсырмаларды FOR, WHILE және REPEAT циклдық операторларды қолдана отырып вариант бойынша тапсырманы орындаңыз. Алгоритімнің блок – схемасын және программасын жазыңыз.

Жұмысты орындауға арналған тапсырма нұсқалары

1. A мен B ($A < B$) екі бүтін сан берілген. Осу тәртібі бойынша осы екі санның арасындағы (бұл сандарды қоса отырып) барлық бүтін сандарды шығару, сонымен бірге сол сандардың N жалпы санын шығару.

2. A мен B ($A < B$) екі бүтін сан берілген. Кему тәртібі бойынша осы екі санның арасындағы (бұл сандарды қоспай) барлық бүтін сандарды шығару, сонымен бірге сол сандардың N жалпы санын шығару.

3. A нақты сан мен N ($N > 0$) бүтін сан берілген. A санның N дәрежесін табу керек: $A^N = A * A * \dots * A$ (A сандары N рет көбейтіледі)

4. A нақты сан мен N ($N > 0$) бүтін сан берілген. A санның 1 ден N дәрежесін табу.

5. A нақты сан мен N ($N > 0$) бүтін сан берілген. $1 + A + A^2 + \dots + A^N$ шығару.

6. A нақты сан мен N ($N > 0$) бүтін сан берілген. $1 - A + A^2 - A^3 + \dots + (-1)^N A^N$ шығару.

7. N ($N > 1$) бүтін сан берілген. Осы сандар тізбегіне ең үлкен санды табыңыздар.

8. N ($N > 1$) бүтін сан берілген. Осы сандар тізбегіне ең кіші санды табыңыздар.

9. A ($A > 1$) нақты сан берілген. $1 + 1/2 + \dots + 1/N$ қосындыны шығару.

10. N ($N > 0$) бүтін сан берілген. $1 * 2 * \dots * N$ көбейтіндісін шығару.

11. N ($N > 0$) бүтін сан берілген. Егер N – тақ сан болса $1 * 3 * \dots * N$ көбейтіндісін шығару; егер N – жұп сан болса – $2 * 4 * \dots * N$ көбейтіндісін шығару.

12. N ($N > 0$) бүтін сан берілген. $1 + \frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{N!}$ қосындысын табу ($N! -$ “N – факториал” – 1 ден N бүтін сандардың көбейтіндісін белгілейді: $N! = 1 * 2 * \dots * N$). Алынған сан $e = \exp(1)$ ($= 2.71828183\dots$) тұрақтыға жуықталған сан болып табылады.

13. X нақты сан мен N ($N > 0$) бүтін сан берілген. $1 + X + \frac{X^2}{2!} + \frac{X^3}{3!} + \dots + \frac{X^N}{N!}$ шығару. Алынған сан X нүктесіндегі exp функциясына жуықталған сан болып табылады.

14. X нақты сан мен N ($N > 0$) бүтін сан берілген. $1 - X^2/2! + X^2/4! - \dots + (-1)^N X^{2N}/(2N)!$ ($N! = 1 * 2 * \dots * N$) шығару. Алынған сан X нүктесіндегі \cos функциясына жуықталған сан болып табылады.

15. X ($|X| < 1$) нақты сан мен N ($N > 0$) бүтін сан берілген. $X - X^2/2! + X^3/3! - \dots + (-1)^N X^N/N!$ шығару. Алынған сан $X+1$ нүктесіндегі \ln функциясына жуықталған сан болып табылады.

16. X ($|X| < 1$) нақты сан мен N ($N > 0$) бүтін сан берілген. $X - X^3/3! + X^5/5! - \dots + (-1)^N X^{2N+1}/(2N+1)!$ шығару. Алынған сан X нүктесіндегі \arctg функциясына жуықталған сан болып табылады.

17. Жалпы мүшесі $A_k = 1/(k * k + 3)$ болатын тізбектің алғашқы n мүшесі мен олардың қосындысын табу керек ($k = 1, 2, \dots, n$).

18. Жалпы мүшесі $A_n = A_1 * q^{(n-1)}$ формуласы арқылы берілетін шексіз кеміменгі геометриялық прогрессияның ϵ -тен кем емес мүшелерінің қосындысын табу керек ($0 < q < 1$; $\epsilon = 0,001$)

19. $D (> 0)$ бүтін сан берілген. A^N сандардың тізбегі келесі түрде анықталады:

20. $A_1 = 1, A_2 = 2, A^N = (A^N - 2 + A^N - 1)/2, N = 3, 4, \dots, 11$ $< D$ шарты орындалатын кезіндегі K нөмірлердің біріншісін іздеп табу, оны шығару, сонымен бірге $A_K - 1$ мен A_K сандарын шығару.

21. Банктік салымының көлемін есептейтін және сол көлемді әр жыл үшін салымның жылдық пайызын (3%) есептей отырып (ол салым екі есе көбейгенше) есептеп шығаратын программаны жазу.

22. Кез келген n және m үшін биноминалдық коэффициентті есептейтін программа жазу:

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

23. 0° ден 90° берілген қадаммен \sin, \cos функциялардың мәндер кестесін шығаратын программаны жазу.

24. $1 + \frac{1}{2} + \frac{1}{3} + \dots$ қатардың бірінші алты элементтерінің қосындысын есептеу (i элементінің мәні оның нөмірімен келесі формуламен байланысты $1/i$). Кезекті элементін есептегеннен кейін ол элементтің нөмірімен мәнін шығару.

25. Клавиатурадан енгізілетін оң сандардың қосындысын есептейтін программаны жазу.

26. Клавиатурадан енгізілетін жүйе сандардың қосындысын есептейтін программаны жазу.

27. Клавиатурадан енгізілетін сандардың жай сандар екендігін анықтайтын программаны жазу.

28. Келесі шексіз қосындының $\sum_{k=1}^{\infty} \frac{(-2)^k}{(k!)^2}$ (қосындының есептеу дәлдігі берілетін болсын) жуықталған мәнін есептейтін алгоритмді құру және программасын жазу.

Келесі тапсырмаларда берілген n мен x мәнін есептеңіз:

$$28. \sum_{l=1}^n \frac{x^l}{l!} \qquad 30. \sum_{l=1}^n \left(\frac{1}{l!} + \sqrt{|x|} \right)$$

$$29. \sum_{l=1}^n \frac{x + \cos(ix)}{2^l} \qquad 31. \sum_{l=1}^n \frac{x^l + \sin(ix)}{3^l}$$

Келесі тапсырмаларда x және ε (ε=10⁻⁶ дәлдігімен берілген) берілген мәндері арқылы қосындыны табыңдар:

$$32. \sum_{k=1}^{\infty} \frac{1}{x^3 k^2} \qquad 34. \sum_{k=1}^{\infty} \frac{x}{k^3 + k\sqrt{|x|} + 1}$$

$$33. \sum_{k=1}^{\infty} \frac{1}{x^2 + k^3} \qquad 35. \sum_{k=1}^{\infty} \frac{(-1)^k x^3}{k}$$

36. Есептендер: $P=(N+1)!$

37. Есептендер: $P = \frac{1}{1} \cdot \frac{3}{2} \cdot \frac{5}{3} \cdot \frac{7}{4} \cdot \dots$

38. Есептендер: $S = \sin x + \sin x^2 + \sin x^3 + \dots + \sin x^n$

40. Есептендер: $P = \sqrt{3 + \sqrt{6 + \sqrt{9 + \dots + \sqrt{3(n-1) + \sqrt{3}}}}}$

№7-9. Зертханалық жұмыстар.

Бір өлшемді және екі өлшемді массивтер

Жұмыстың мақсаты: массивтерді есептер барысында пайдалана білу, бір өлшемді массивтерді типтер және айнymалылар бөлімінде сипаттай білу, массив элементтерін енгізу жолдарын білу, массив элементтерімен ртүрлі амалдар: іздеу, сұрыптау, алмастыру, элементтің алып тастау және т.б. амалдарды қолданып есептер шығарып үйрену.

Бақылау мысалы. A(10) массиві берілген. Массивтің барлық элементтерінің әртүрлі екенін анықтау.

Program primer4:

Var a:array[1..10] of integer;

i,j:integer;

begin

writeln('массивтің 10 элементтерін енгізіңіз');

for i:=1 to 10 do readln (a[i]);

for i:=1 to 9 do begin

```

for j:=i+1 to 10 do begin
if a[i]=a[j] then break;
end;
if j<10 then break;
end;
if i<9 then writeln ('массивте бірдей элементтері бар') else writeln
('массивтегі барлық элементтер әртүрлі'); end.

```

2. A(10) массивінің арифметикалық ортасын анықтау.

```

Program primer4_2;
type T = array [1..5] of real;
var A:T; S, SA:real;
i, K: integer;
begin
for i := 1 to 10 do
read (a[i]);
S := 0; K := 0;
for i := 1 to 10 do
if a[i] > 0 then
begin
S := S + a[i]; K := K + 1
end;
SA := S/K;
writeln ('SA=', SA:10:2)
end.

```

Тапсырма нұсқаларында көрсетілгендей бірөлшемді массивті өңдеуді жүзеге асырыңыз. Алгоритмнің блок-схемасын және программасын жазыңыз.

Жұмысты орындауға арналған тапсырма нұсқалары

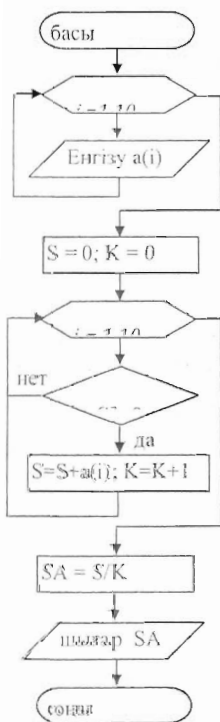
1. N өлшемді массив берілген. Массив индекстерін оларға сәйкес элементтер өсу реті бойынша түзілетіндей тәртіпте шығару.

2. N өлшемді массив берілген. Алдымен оның жұп индексті элементтерін шығару, сосын-тақ индекстісін шығару.

3. Өлшемі 10-ға тең A массиві берілген. $A[1] < A[i] < A[10]$ теңсіздігін қанағаттандыратын A[i] элементтерінің ішінен біріншісінің нөмірін шығару. Егер мұндай элементтер болмаса, онда 0 мәнін шығару.

4. А саны және $X(10)$ массиві берілген. Тізбектің А санына тең $X(i)$ элементі рет бойынша нешінші орында орналасқанын анықтаңдар. Егер мұндай сан болмаса, онда жауабы 0 болуы керек.

5. Массивте кем дегенде бір теріс элементін және оның реттік нөмірін жауапқа шығарыңдар.



6. У массивінің оң элементтерін Z массивіне рет бойынша көшіріңдер.

7. $A(10)$ сандар тізбегін екі массивке бөліңдер, олардың біріншісіне жұп нөмірлі элементтерді жазыңдар.

8. $Z(10)$ массивіндегі оң элементтердің қосындысын есептеңдер.

9. $X(20)$ массивіндегі жұп индексті элементтердің көбейтіндісі тақ сан екендігін анықтаңдар және “иә” немесе “жоқ” жауабын баспана шығарыңдар.

10. $Y(10)$ массивінің тақ индексті оң элементтерінің арифметикалық ортасын есептеңдер. Егер ондай элемент жоқ болса, “0” белгісін жауапқа шығарыңдар.

11.10 элементтен тұратын тізбектегі оң элементтер санын есептендер.

12.X(10) массивінің элементтерінің қосындысы жұп болатындығын анықтаңдар және «иә» немесе «жоқ» жауабын баспаға шығарыңдар.

13.X(12) массивіндегі оң элементтер орнына «1», ал теріс элементтер орнына «0»-ді жазыңдар.

14.X(10) өлшемді массивтің оң сандық элементтерін минимальды мәнге алмастыру.

15.N өлшемді массив берілген. Массивтің минимальды мен максимальды элементтерінің арасындағы тұрған элементтерді кері тәртіппен орналастыру.

16.N өлшемді массив берілген. Массивтің элементтерін бір позицияға солға қарай циклдык жылжытуды жүзеге асыру.

17.N өлшемді нөлдік емес бүтін санды массив берілген. Оның ішінде тақ пен жұп сандары кезектескенін тексеру. Егер кезектессе 0 шығару, егер кезектеспесе, онда бұл заңдылықты бұзатын бірінші элементтің нөмірін шығару. N өлшемді массив берілген. Оның жергілікті минимумын табу.

18.N өлшемді массив пен R нақты сан берілген. Берілген санға ең жақын болатын массивтің элементін табу.

19.N өлшемді массив берілген. Бұл массивтегі екі ең жақын сандардың нөмірлерін табу.

20.N өлшемді массив берілген. Оның бірдей элементтерінің максимальды санын табу.

21.N өлшемді массив берілген. Екіден кем рет кездескен элементтерін алып тастау.

22.N өлшемді массив берілген. Егер ол құрамында 1 ден N дейінгі сандарды қамтыса, онда нөлді шығару, әйтпесе бірінші кездескен ретсіз элементінің нөмірін шығару.

23.N өлшемді массив берілген. Барлық оң элементтің алдына нөлдік элементті қосып өзгерту.

24.N өлшемді массив берілген. Әр серияға бір элементті қосып өзгерту.

25.N өлшемді массив берілген. Бірінші кездескен ең ұзын серияға бір элементті қосып өзгерту.

26.N өлшемді массив берілген. Әр сериядан кейін нөлдік элементін қойып шығу.

27. N өлшемді массив пен k саны берілген. Массивтің соңғы және k-ші серияларының орындарын ауыстыру. Егер массивте сериялардың саны k-дан төмен болса, онда массивті өзгертпей шығару.

28. N өлшемді массив пен k саны берілген. Сериялардың ұзындығы k-ға тең барлық серияларды массивтен алып тастау.

29. N өлшемді массив пен k саны берілген. Серияның ұзындығы k санынан кіші әрбір серияны бір нөлдік элементке ауыстырып отыру.

30. Пернетақтадан енгізілген сан массивте қанша рет кездесетінін тексеру.

Көпөлшемді массивтерді қолдану

Екі өлшемді нақты сандардан тұратын массив берілген (A матрицасы), массивті төмендегі варианттарды қолданып, өңдесу. Бастапқы деректерді тапсырма ерекшеліктерін назарға ұстай отырып қолдан енгізу. Алгоритм блок-схемасын құрып, программасын жазу.

Бақылау мысалы

$N \times N$ екі өлшемді массив берілген. Басты және қосымша диагональдардағы элементтердің қосындысын табу.

```
program primer5;
uses crt;
const n=5;
var a:array[1..n,1..n] of integer;
i,j,s,s1,s2:integer;
begin
clrscr;
for i:=1 to n do
for j:=1 to n do
read(a[i,j]);
s:=0; s1:=0; s2:=0;
for i:=1 to n do
for j:=1 to n do
begin
if i=j then s1:=s1+a[i,j] else
if i+j=n+1 then s2:=s2+a[i,j];
end;
s:=s+s1+s2;
write('s=',s);
end.
```

2. A(4,5) матрицасының басты диагоналінің жоғары жағындағы он элементтерінің санын анықтау программасын қарастырайық.

```

program primer4_3;
type T = array [1..4,1..5]of
real;
var A:T; i, j, K: integer;
begin
  for i:=1 to 4 do
    for j:=1 to 5 do
      read (a[i,j]);
    K:=0;
    for i:=1 to 4 do
      for j:=1 to 5 do
        if (a[i,j]>0) and (i < j)
          then
            K:=K+1;
      writeln ('K=', K)
    end.

```

Жұмысты орындауға арналған тапсырма нұсқалары

1. k ($0 < k < 11$) саны мен 4×10 матрицасы берілген. K бағанының элементтерінің қосындысы мен көбейтіндісін табу.

2. 5×9 матрицасы берілген. Оның барлық жұп бағаналарының элементтерінің қосындысын табу.

3. 5×10 матрицасы берілген. Оның барлық тақ бағанасындағы минимальды мәнді табу.

4. 5×10 матрицасы берілген. Оның барлық жұп бағаналарының элементтерінің қосындысын табу.

5. Берілген 5×10 матрицасында максимальды элементі орналасқан жлды минимальды элемент орналасқан жолмен ауыстыру.

6. 5×10 матрицасы берілген. Барлық бағаналардың элементтерінің қосындысының арасындағы минимальды мәнді табу және сол минимальды мән орналасқан бағананың нөмірін табу.

7. 5×10 матрицасы берілген. Әр жолдың максимальды элементтерінің арасындағы минимальдысын табу.

8. 5×10 матрицасы берілген. Оң және теріс элементтерінің саны тең болған бірінші жолдың нөмірін шығару (нөлдік элементтер есептелмейді). Егер ондай жолдар болмаса, 0 шығару.

9. 5×10 матрицасы берілген. Тек оң элементтерінен құралған соңғы жолдың нөмірін шығару. Егер ондай жолдар болмаса, 0 шығару.

10. $M \times N$ матрицасы берілген. Егер жолдардр (бағаналарда) кездескен сандардың көптігі сәйкес келсе, онда ол жолдарды (бағаналарды) ұқсас деп атаймыз. Соңғы бағанаға ұқсайтын бағаналардың санын табу.

11. $M \times N$ матрицасы берілген. Барлық элементтері әр түрлі жолдардың санын табу.

12. $M \times N$ матрицасы берілген. Бірдей элементтердің саны максималды болған бірінші кездескен жолдың нөмірін шығару.

13. M квадратты матрицасы берілген. $A[1, M]$ бірэлементтік диагональдан бастап басты диагональға параллельді диагональдардың элементтерінің қосындысын табу.

14. M квадратты матрицасы берілген. $A[1, 1]$ бірэлементтік диагональдан бастап қосымша диагональға параллельді диагональдардың элементтерінің қосындысын табу.

15. M квадратты матрицасы берілген. $A[1, 1]$ бірэлементтік диагональдан бастап қосымша диагональға параллельді диагональдардың элементтерінің арасындағы максималдысын шығару.

16. M квадратты матрицасы берілген. Басты диагональдан төмен жатқан элементтерді нөлге ауыстыру.

17. M квадратты матрицасы берілген. Басты диагональмен қоса оң жағы жатқан және қосымша диагональмен қоса оңдан төмен жатқан элементтерін нөлге ауыстыру.

18. M квадратты матрицасы берілген. Матрица симметриясына горизонтальды оське байланысты оның элементтерін айналық түрде көрсету.

19. M квадратты матрицасы берілген. Матрица симметриясына вертикальды оське байланысты оның элементтерін айналық түрде көрсету.

20. M квадратты матрицасы берілген. Матрицаның басты диагоналына байланысты оның элементтерін айналық түрде көрсету.

21. M квадратты матрицасы берілген. Матрицаның қосымша диагоналына байланысты оның элементтерін айналық түрде көрсету.

22. M квадратты матрицасы берілген. Оны 90 градуске оң жаққа қарай бұру.

23. М квадратты матрицасы берілген. Оны 180 градуске оң жаққа қарай бұру.

24. М квадратты матрицасы берілген. Оны 270 градуске оң жаққа қарай бұру.

25. К саны мен 4×10 өлшемді матрица берілген. К нөмірлі жолды матрицадан алып тастау.

26. 5×10 өлшемді матрица берілген. Минималды элементті қамтитын бағананы матрицадан алып тастау.

27. 5×10 өлшемді матрица берілген. Тек оң элементтерін қамтитын барлық бағаналарды матрицадан алып тастау.

28. 4×9 матрица берілген. К нөмірлі жолдың алдына нөлдерден құрылған жолды қою.

29. 4×9 өлшемді матрица берілген. Максимальды элементті қамтитын бағананың көшірмесін құру.

30. 5×9 өлшемді матрица берілген. Тек оң элементтерден құрылған соңғы бағанадан кейін тек 1-ден құралатын бағананы қосу.

№ 10-Зертханалық жұмыс

Қатарларды өңдеу есебін программалау.

Жұмыстың мақсаты: жолдық және символдық жұмыс істеу алгоритмдерін құрып, үйрену, жолдық және символдық жолдық мәліметтерге әртүрлі амалдар қолдана білу.

Кілтжиынынан символдар тізбегін қолдан енгізіп, вариантта көрсетілген тапсырмалар бойынша өңдеңіз. Бастапқы деректерді қолдана отырып тапсырмалар нақтылығы ескерілуі тиіс.

Жұмысты орындауға арналған тапсырма нұсқалары

1. С1-ден бастап С1 мен С2 кезектесіп символдардан тұратын $N(N\text{-жұп})$ жолдың ұзындығын шығару.

2. Жол берілген. Жол элементтерін керу тәртіпте шығару.

3. Жол берілген. Ішіндегі бас әріптердің санын есептеу.

4. Жол берілген. Ішіндегі бас әріптердің санын есептеу.

5. Жол берілген. Барлық кіші әріптерді бас әріптерге түрлендіру.

6. Жол берілген. Егер ол бүтін сандарды қамтыса-1 шығару, егер нақты санды жазба болса – 2 шығару; егер жолды сапға ауыстыруға болмаса - 0.

7. Бүтін сан берілген. Кері ретіндегі сандардан қамтитын символдар жиынын шығару.

8. $N(>0)$, бүтін сан мен жылжымалы үтірлі нүкте форматындағы нақты сандарды қамтитын S жолы берілген. Сол санның бөлшек

бөліктеріндегі алғашқы N сандарын символдар жиыны түрінде көрсету.

9. Бүтін санды екілік жазба түрінде көрсететін жол берілген. Сол санның ондық жазбасын көрсететін жолды шығару.

10. Бүтін санды көрсететін жол берілген. Сол сандардың қосындысын шығару.

11. S жолы мен N сан берілген. S жолын N ұзындықты жолға келесі түрде өзгерту қажет: егер S жолының ұзындығы N-нан үлкен болса, онда бірінші символдарды лактырамыз, егер S қатарыN-нан кіші болса, онда оның бас жағына «.» символын қосу керек.

12. N1 және N2 екі саны, S1 және S2 екі жол берілген. N1 санын S1 қатарының бірінші символына қосып, N2 санын S2 жолының соңғы символдарына қосып, жаңа қатар алу.

13. S1 және S2 екі жол берілген. S2 жолы S1 жолының құрамында барын тексеру. Егер болса, онда S2 жолы S1 жолының құрамына бастап кірген позицияның нөмірін шығару; егер кірмесе -0 шығару.

14. S1 және S2 екі жол берілген. S2 жолы S1 жолына қанша рет кіретін санын анықтау.

15. S жолы және C символы берілген. S жолына C символының әр кіруін еселеу.

16. S1, S2 жолдары және C символы берілген. S1 жолына C символдың әр кіруінің алдында S2 жолын енгізу.

17. S1 және S2 екі жол берілген. S1 жолынан S2 жолымен сәйкес келген ішкі жолды алып тастау. Егер ондай подстрока болмаса, онда S1 жолын өзгертпей шығару.

18. S1, S2 және S3 үш жол берілген. S1 жолында S2 жолының соңын кіруін S3 жолына ауыстыру.

19. Жол берілген. Сол жолдың бірінші мен екінші нүктелерінің арасындағы ішкі жолды шығару. Егер жолда нүктелердің саны екіден кем болса, онда берілген жолды шығару.

20. Бір немесе бірнеше пробелмен бөлінген орыс сөздерінен құралған жол берілген. Жолдағы сөздердің санын анықтау.

21. Бір немесе бірнеше пробелмен бөлінген орыс сөздерінен құралған жол берілген. Бір әріптен басталып сол әріпке аяқталған сөздердің санын анықтау.

22. Бір немесе бірнеше пробелмен бөлінген орыс сөздерінен құралған жол берілген. Үш «А» әріп кездесетін сөздердің санын анықтау.

23. Бір немесе бірнеше пробелмен бөлінген орыс сөздерінен құралған жол берілген. Ең қысқа және ең ұзын сөздердің ұзындығын анықтау.

24. Бір немесе бірнеше пробелмен бөлінген орыс сөздерінен құралған жол берілген. Сол сөздерден құралған, бірақ «.» (нүктемен) бөлінген жолын шығару. Ең соңында нүкте қойылмайды.

25. Бір немесе бірнеше пробелмен бөлінген орыс сөздерінен құралған жол берілген. Сол сөздерден құралған, бірақ аралары бір пробелмен бөлінген, сөздер кері ретінде жазылған жолын шығару.

26. Бір немесе бірнеше пробелмен бөлінген орыс сөздерінен құралған жол берілген. Жолдағы әр сөздерден оның бірінші әріпінің келесі кірулерін алып отырып өзгерту.

27. Бір немесе бірнеше пробелмен бөлінген орыс сөздерінен құралған жол берілген. Сол сөздерден құралған, бірақ аралары бір пробелмен бөлінген, сөздер алфавит бойынша жазылған жолын шығару.

28. Сөйлем жол берілген. Әр сөз бас әріппен басталатындай жолды өзгерту.

29. Сөйлем жол берілген. Жолдағы тыныс белгілері мен дауысты әріптердің санын есептеу.

30. Сөйлем жол берілген. Сөйлемдегі ең қысқа мен ең ұзын сөзді шығару (егер ондай сөз бірнеше болса, онда соңғысын шығару).

№ 11-Зертханалық жұмыс

Жазбалар мен файлдар типін қолдана отырып программалар құру.

Берілген құрылымға сәйкес жазбалар массивін құрыңыз. Жазбалар файлын қолдана отырып жазбалардың енгізуін және өңдеуін ескеріңіз. Програмада кіріс деректерді, дерек қордың файлдарын енгізу-шығаруын өндейтін сәйкесті процедуралар мен функцияды алдын ала қарастырылуы қажет. Қолданушының әрекеттері бақылану және нақты хабарлармен қамтылу керек. Нәтижесі (дерек қоры) файлда сақталып келесі (соңғы) өңдеуіне жеңіл алынатын болуы керек.

Жұмысты орындауға арналған тапсырма нұсқалары

1-6 варианттары:

Келесі кілттер арқылы: заңгерлік (юрисетік) кеңесінің адресі, заңгерлік кеңесінің аты, адвокат немесе нотариустің аты-жөні

бойынша, адвокат немесе нотариустың телефоны бойынша жазба массивін элементтерінің өрісіне арналған міндерін шығаруын ескерту



Сұраныстар нұсқалары:

- Белгілі ауданда неше заңгерлік кеңесі бар екенін анықтау
- Ең көп адвокаттар қай заңгерлік кеңесінде істейтінін анықтау
- Ең аз нотариустар қай заңгерлік кеңесінде істейтінін анықтау
- Белгілі заңгерлік кеңесінде неше адвокат пен нотариустар жұмыс істейтінін анықтау.
- Бір белгі көшеде неше заңгерлік кеңес орналасқанын анықтау
- Бір белгі ауданда неше заңгерлік кеңес орналасқанын анықтау

Келесі қаттер арқылы: мерзімі (уақыты), соттық тексеріс затына, фирма-дугер аты бойынша, фирма-жауапшы бойынша жазба массивін элементтерінің өрісіне арналған міндерін шығаруын анықтау.

Сұраныстар:

- Қай жылы залымдыққа байланысты істердің саны ең жоғары болғанын анықтау.
- Қай жылы банкротқа ұшырауына байланысты істердің саны ең жоғары болғанын анықтау
- Қай жылы ажырасуға байланысты істердің саны ең жоғары болғанын анықтау
- Қай жылы дискриминацияға байланысты істердің саны ең жоғары болғанын анықтау

• Белгілі фирма қарсы қай жылы ең ккп қылмыстық іс қазғалғанның анықтау

- Екі реттен астам қай фирма бойынша қылмыстық іс қозғалған
- Қай фирмф өткен жылдар бойынша даугер рөлінде ең жиі шыққанын анықтау

14-24 варианттары

Келесі кілттер арқылы: мердігер фирмасының аты (подрядчик) бойынша, құрылыстың бағасы (құны) бойынша, құрылыстың түрі бойынша, банктегі салымның сомасы (көлемі) бойынша жазба массивтің элементерінің өрісіне арналған мәндерен шығартуын ескерту

Сұраныстар:

- Банкідегі салымның сомасы құрылыстың бағасынан кем болған клиенттерді анықтау
- Құрылыстың бағасын толық төленген клиентті анықтау
- Клиенттер екіден кем емес рет тапсырған құрлыстың типін анықтау
- Банідегі ең жоғары соманың есепшотына ие болған клиентті анықтау
- Максималды бағасы бар құрылғысын анытау
- Минималды бағасы бар құрылғысын анытау
- Несисні ең жоғары пайызға алынған клиентті анықтау
- Несисні ең төмен пайызға алынған клиентті анықтау
- Барлық клиенттердің төленбеген сомасын
- Ең аз төленбеген сомасына ие болған клиентті анықтау.
- Ең көп төленбеген сомасына ие болған клиентті анықтау.

25-34 варианттары

Келесі кілттер арқылы: тауарлардың атына байланысты, келіп түскен жылы бойынша, тауарды әкелген мемлекет бойынша, тауар әкелетін фирма бойынша жазба массивтің элементерінің өрісіне арқылы арналған мәндерін шығаруын ескеру

Сұраныстар

- Қай фирманың тауары ең жиі / аз түскенін анықтау .
- Барлық жылдар бойынша түскен тауардың көлемін анықтау.
- Ең қымбат / арзан тауардың атын анықтау
- Барлық жылдар бойынша тауарлар алу үшін кеткен шығынды есептеу.

- Бәрінен жиі / аз тауарға заказ берілген жабдықтаушы-өнді анықтау .
- Қай айда ең жиі /аз тауарлар түскенін анықтау .

Бақылау мысалы

Студенттер жайлы мәліметтер берілген : аты жөні , туған жылы , туған жері , тұратын адресі .

Файлды құру керек.Құрылған файл Алматылық студенттер жайлы мәлімет оқып және 18 – жастан кіші студенттер жайлы мәліметтерді шығару керек.

Program primer 9;

Uses crt ;

Type

Mydata = record

Fio :INTEGER ;

Year :integer ;

Place : string;

Addr : string;

End;

Myfile = file of mydata ;

Var

Fin , fout :myfile ;

Trec : mydata ;

N , I , k :integer;

Procedure inputdata ;

Begin

Write('Kolichestvo=');

Readln(n);

№ 12-Зертханалық жұмыс.

Процедуралар мен функциялар

Жұмыстың мақсаты: процедуралар мен функциялар процедураларымен жұмыс істеу, программасын құру.

№1-Тапсырма. Бұл топтың ішіндегі тапсырмаларда integer және real типті сандық параметрлері бар процедуралар мен функцияларды жүзеге асыру қажет. Бұл типтердің кіріс параметрлері әдетте мәнді –параметрлері сипатталады.

Жұмысты орындауға арналған тапсырма нұсқалары

1. А мен В нақты сандардың біреуін минимальді болып табатын $\text{Min } 2(A, B)$ нақты типті функцияны сипаттау. Сол функцияның көмегімен А мен В, А мен С, А мен D арасынан минимальді санды табу, егер А, В, С және D берілген сандар болса.

2. А мен В мәндердің арасындағы минимальді мәнін А айнымалысына, максимальді мәнін В айнымалысына жазатын $\text{Minmax}(A, B)$ процедурасын сипаттау (А мен В-нақты параметрлер, олар бір мезгілде кіріс және шығыс параметрлері болып табылады). Осы процедураның төрт шақыруын пайдалана отырып А, В, С, D сандардың арасындағы минимальді және максимальді сандарын табу.

3. 2 тапсырмадағы $\text{Minmax}(A, B)$ процедурасын қолдана отырып А, В, С нақты сандардың арасынан максимальді санды табатын $\text{Max3}(A, B, C)$ нақты типті функциясын сипаттау. Осы функция арқылы А, В, С, D сандардың арасындағы келесі жиындардың арасынан (А, В, С), (А, В, D), (А, С, D) максимальдісін табу.

4. 1 тапсырмадағы процедурасын қолдана отырып А, В, С, D нақты сандардың арасынан минимальді санды табатын $\text{Min } 3(A, B, C, D)$ нақты типті функциясын сипаттау. Осы функция арқылы А, В, С, D, E сандардың арасындағы келесі жиындардың арасынан (А, В, С, D), (А, В, С, E), (А, С, D) минимальдісін табу.

5. $N! = 1 \cdot 2 \cdot \dots \cdot N (N > 0)$ (бүтін санның параметрі) факториальдің мәнін есептейтін $\text{Fact}(N)$ бүтін типті функциясын сипаттау. Сол функция арқылы берілген 5 санның факториалдарын есептеу.

6. $N! = 1 \cdot 2 \cdot \dots \cdot N (N > 0)$ (бүтін сандар үшін) факториалдің жуықталған мәнін есептейтін бүтін типті $\text{FactR}(N)$ функциясын сипаттау. Сол функция арқылы берілген 5 санның факториалын есептеу.

7. Егер N—тақ болып, $N! = 2 \cdot 4 \cdot 6 \cdot \dots \cdot N$; N—жұп болса ($N > 0$) (бүтін санның параметрі) $N! = 1 \cdot 3 \cdot 5 \cdot \dots \cdot N$ екі еселенген факториалдің* мәнін есептейтін бүтін типті $\text{Fact2}(n)$ функциясын сипаттау. Сол функция арқылы берілген 10 санның екі еселенген факториалын есептеу.

8. И-бүтін санның цифрлерінің S-қосындысын табатын $\text{SumDigit}(N, S)$ процедурасын сипаттау. (И-кіріс параметрі, S-шығыс параметрі) сол процедура арқылы 5 санның факториалын есептеу.

9. А мен В натурал санның ең үлкен бөлгішін табатын бүтін типті рекурсивті емес функциясын сипаттау (Евклид алгоритмі бойынша $\text{НОД}(A, B) = \text{НОД}(B \bmod A, A)$ егер $A > 0$;

$\text{НОД}(0, B) = B$. Сол функция көмегімен А мен В, А мен С, А мен D сандардың ең үлкен бөлгішін табу, егер А, В, С, D ,сандары берілген болса.

10. Power A(x,a,eps) нақты типті функцияны сипаттау (x,a,eps параметрлері –нақты, $[x] < 1, a > 0, \text{eps} > 0$), ол $(1+x):(1+x) = 1 + a * x + a * (a-1)x^2 + \dots + a * (a-1) * \dots * (a-n+1) * x^n + \dots$

функцияның жуықталған мәнін табады. Қосындыда eps модулінен үлкен барлық қосылғыштарды ескеру керек. PowerA көмегімен x пен a деректердің eps-н алты әртүрлі мәндер үшін (1+x) –н жуықталған мәнін табу.

№2- Тапсырма. Бұл топтың ішіндегі тапсырмаларындағы процедуралар мен функциялардың сипаттамасында массивтерді параметр арқылы беруінің әрекеттерін ескеру қажет. Бір өлшемді массивтер үшін Borland Pascal 7.0 енгізілген ашық массивтерінің механизмін пайдалану керек. Екі өлшемді массивтерінде алдын ала сәйкесті қолданушылық типін анықтау қажет. Оны кейде параметр - матрицаны сипаттағанда қолданатын боламыз. Кірістік параметрлер-массивтарін әдетте параметрлар-мәндер арқылы сипаттамайды. Егер массив процедуралар /функциялар орындалу мерзімінде өзгермесе , оны параметр – айнымалы ретінде алу керек.

Жұмысты орындауға арналған тапсырма нұсқалары

1. Белгілі X нақты нүктесіндегі P полиномның мәнін табатын $\text{Polinom}(A, N, X)$ нақты типті функцияны сипаттау. P полиномы келесі параметрлер арқылы берілген; N(полиномның дәрежесі $0 < N < 8$) және A(полиномның(коэффициенты N+1

өлшемді нақты массив) $P(X) = A[1] * X^{n-1} + \dots + A [N+1]$ Осы функцияны қолдана отырып белгілі бес нүктедегі полиномның мәнін табу.

2. N нақты санның құралатын A массивтегі максимальді элементін табатын $\text{Max}(A, N)$ нақты типті функциясын сипаттау. Сол функция көмегімен NA, NB, NC өлшемге сәйкесті A, B, C массивтегі минималді элементтерін табу.

3. N нақты санның құралатын A массивтегі минималді элементін табатын $\text{NMin}(A, N)$

нақты типті функциясын сипаттау. Сол функция көмегімен NA , NB , NC өлшемге сәйкесті ABC массивтегі минимальді элементтерін табу.

4. N нақты саннан куралатын A массивтегі минимальді мен максимальді элементтерінің нөмірлерін табатын $Nminmax(A, N, Nmin, Nmax)$ процедурасын сипаттау. Шығыс параметрлері; $Nmin$ (минимальді элементтің нөмірі) және $Nmax$ (максимальді элементтің нөмірі) Сол функция көмегімен NA , NB , NC өлшемге сәйкесті A , B , C массивтегі минимальді мен максимальді элементтерінің нөмірлерін табу.

5. A натурал санды жай көбейткіштерге ажыратуын табатын $Factors(A, N, F)$

процедурасын сипаттау. Сол процедура арқылы берілген 5 санның жай көбейткіштерге ажыратуын көрсету.

6. N нақты саннан инверторлы ("инвертерілген массивке") A массивтегі элементтерінің еру тәртібін өзгертетін $Invert(A, N)$ процедурасын сипаттау. A массиві – кіріс және шығыс параметрі N кіріс параметрі. Сол процедура көмегімен NA , NB , NC өлшемге сәйкесті A , B , C массивтерін инверторлау.

7. N өлшемді A нақты массивті K позициясына ($0 < k < 5, k < N$) солға оңға қарай циклдік жылжытуды жүзеге асыратын $MoveLeft(A, N, k)$ $MoveRight(A, N, k)$ процедурасын сипаттау

A массиві – кіріс және шығыс параметрі, N мен k кіріс параметрі. Сол процедура көмегімен N өлшемді массивті $k1$ – позициясына жылжыту содан кейін – алынған массивтегі элементтерін $k2$ позициясына жылжыту ($k1$ мен $k2$ берілген) Процедураның әр орындалғаннан кейін экранға жаңа массивті шығару.

8. N өлшемді A нақты массивтің әр элементін онымен көршілес элементтерінің орта арифметикалығына ауыстыратын $Smooth(A, N)$ процедурасын сипаттау. A массиві – кіріс және шығыс параметрі, N кіріс параметрі. Сол процедура көмегімен N өлшемді массивті 5 рет орындау. Процедураның әр орындалғаннан кейін экранға жаңа массивті шығару.

9. N бүтін саннан құралатын A массивтен X бүтін санды алып тастайтын $RemoveX(A, N, X)$ процедурасын сипаттау. A массиві мен N саны кіріс және шығыс параметрі болып табылады. Сол процедура көмегімен NA , NB , NC өлшемді A , B , C массивтерден сәйкесті XA , XB , XC сандарын алып тастау.

10. N бүтін саннан құралатын A массивтегі X санға тең элементерін қайталайтын

DoubleX(A,N,X) процедурасын сипаттау. A массиві мен N саны кіріс және шығыс параметрі болып табылады. Сол процедура көмегімен NA, NB, NC өлшемді A, B, C массивтерге XA, XB, XC сандарының әр кірісін сәйкесті түрінде қайталауы.

```
For i:=1 to n do
  Begin
    With trec do
    Begin
      Write('surname:') read(fio);
      Write('year:'); readN(year);
      Write('please:'); readln(please);
      Write('address:'); readln(addr);
      End;
    Write(ln,trec);
      End;
    End;

  Procedure select
  Begin
    Reset(fin);
    For i:=1 to n do
    Begin
      Read(fin,trec);
      K:=k+1;
      End;
    End;

  Procedure outdara;
  Begin
    Reset(fout);
    If k:=0 then writeln('<<<<<svedeni net>>>>')
    Else for i:=1 to k do
    Begin
      Writeln('*****');
      Read(fout,trec);
      With trec do
      Begin
        Writeln(fio);
```

```

        Writeln(year);
        Writeln(pleace);
        Writeln(addr);
            Readln;
                End;
                    End;
                        End;
Begin
    Clrscr;
    Assign(fin,'data.txt');
    Rewrite(fin);
    Rewrite(fout,'outdata.txt');
    Rewrite(fin);
    Rewrite(fout);
        K:=0
        Inputdata;
        Select;
            Outdfata;
                Close(fin);
                    Close(fout)
                        End.

```

№ 13-Зертханалық жұмыс.

Функция және процедура құру

Жұмыстың мақсаты: жолдық шамалармен жұмыс істеу алгоритмдерін құрып үйрену, жолдық стандартты процедуралар мен функцияларды пайдалана білу іскерлігі мен дағдысын қалыптастыру; жолдық мәліметтерге әртүрлі амалдар қолдана білу дағдысын қалыптастыру.

Бақылау мысалы

Программада 5 пен 7 екі бүтән сан берілген, ол сандар оларды еселейтін Inc2 процедурасына беріледі. Параметрлердің брейі параметр-айнымалы ретінде, ал екінші – параметр-мәні ретінде берілені. Процедураның шығару алдындағы және солан кейінгі алынған еселену нәтижесі экранға шығарады

```

Program primer 8;
Var n ,m :integer ; f :real ;

```

```

Function fact (n :integer ) :integer ;
Var i,p:integer ;
Begin p:=1;
For i:=2 to n do p :=p*i
Fact :=p
End;
Begin
  Read (m,n) ;
  F:= fact (n) * fact (m)/ fact (n+m);
  Writeln (f);
End.

```

Төмендегі тапсырмаларда жолдық деректерді процедуралар/функциялар параметрі ретінде беру керек. Кірістік жолдық параметрлерін, егер олар процедуралар/функциялардың ішінде өзгермесе, онда оларды параметрі константта арқылы сипаттау қажет.

Жұмысты орындауға арналған тапсырма нұсқалары

1. S жолы паскаль тілінің рұқсат етілетін идентификаторы болатынын тексеретін IsIdent(S) бүтін сандық функциясын сипаттау. Егер дұрыс болса 0 қайтарылады. Егер S болса жол болса -1 қайтарылады, егер жол саннан басталса -2 қайтарылады. Егер S рұқсат берілмеген символдардан тұрса, бірінші рұқсат берілмеген символдың номері қайтарылады. Сол функция көмегімен 5 берілген санын тексеру.

2. S шаблон-жолының қайталанатын копияларымен толтырылған Len ұзындығы бар жолды қайтарылатын FiLLStr(S,Len) жолдық типті функциясын сипаттау. Сол функция көмегімен Len берілген санмен және 5 берілген шаблон-жолының 5 жаңа Len ұзындығы бар жолы құру.

3. S жолының барлық бес әріптерін кіші әріптерге ауыстыратын (басқа символдар өзгермейді) LowCase процедурасын сипаттау. Сол процедура көмегімен берілген 5 жолын өзгерту.

4. S жолынан бірінші және соңғы пробелдерін алып Trim(S), тастайтын процедурасын сипаттау. Сол процедура көмегімен берілген 5 жолын өзгеру.

5. S жолында subs (k>0) ішкі жолының соңғы кірісінен тұратын PosK(subs,S,k) позиция номерін қайтарып тұратын функциясын сипаттау. Егер S жолында subS ішкі жолы болмаса, ол функция 0

қайтарылады. Берілген subS және S мәндеріне бұл функциясының міндерін шығару,

6. S жолында ($k > 0$) ішкі жолының k-кірісі кездескен позиция нөмірін қайтарып тұратын PosK (subs, S, k) бүтін типті функциясын сипаттау. Егер S жолындағы subsS ішкі жолының кірістерінің саны k-дан аз болса, ол функция 0 қайтарылады. Берілген subS, k және S5 мәндеріне бұл функцияның мәндерін шығару.

7. S жолының k-сөзін қайталайтын (сөз дегеніміз пробелмен немесе жолдың басы/аяғымен шектелінген пробелсыз символдың жиыны болып табылады) WordN(S,k) жолдық типті функциясын сипаттау. Егер жолдағы сөздердің саны k-дан аз болса ол функция бос жолды қайтарылады. Сол функцияны пайдалана отырып берілген жолынан k1, k2, k3 нөмірлі сөздерді меншіктеу.

8. Берілген S жолмен W сөздердің жиынын қалыптастыратын процедурасын сипаттау (W -шығыс жолдық массив; N-оның өлшемі, $N \leq 10$) Сөз дегеніміз пробелмен немесе жолдың басы/аяғымен шектелінген пробелсыз символдың жиыны болып табылады. Сол жолды қайтарады. Сол функцияның пайдалана отырып берілген S жолдағы сөздердің санын шығару (N) және сол сөздерді шығару.

№ 14- Зертханалық жұмыс.

Файлдармен жалпы жұмыс принципі

Жұмыстың мақсаты: құрылған файлға амалдарды пайдалана білу іскерлігін қалыптастыру

Бақылау мысалы.

Мәтіндік файл берілген. Файл көшірмесін алу.

```
Program primer10;
```

```
  Var f;text;
```

```
  S:string;
```

```
  Begin
```

```
    {файлдық айнымалылары мен файлдың ағтарын  
    арасына байланыс қондыру}
```

```
    assign (f, data1.txt.)
```

```
    assign(g,data2.txt.)
```

```
    {файлдарды жазу және оқу үшін ашу}
```

```
    reset(f);rewrite(g);
```

```
    while not (eof(f))do
```

```
  begin
```

```

readln(f,s) {қатарға файлдан оқу}
writeln(g,s) {қатардан файлға жазу}
                                                    end
{файлды жабу}
                                                    close(f);close(g)
                                                    end.

```

Төмендегі тапсырмаларда кезкелген символдан құралған тізбектерін өңдеуін және оның мәтіндік файлдан енгізуін жүзеге асыру керек. Программада өз нұсқасына кіріс мәліметтерін нақты өңдейтін сәйкес процедуралар мен функцияларды қарастыру керек.

Жұмысты орындауға арналған тапсырма нұсқалары

1. Мәтіндік файл берілген. Оның символдар және жолдар санын шығару.
2. S жолы және мәтіндік файл берілген. Файлдың басы мен соңына S қойып шығу.
3. Мәтіндік файлдар берілген Оның бірінші және соңғы жолдарын алып тастау.
4. Name1 және Name2 мәтіндік файлдар берілген. Name1 және Name2 мәтіндік файлдарды біріктіріп жаңа Name3 файл құру.
5. Name1 және Name2 мәтіндік файлдар бар. Name2 файлды Name1 соңына қою
6. Үш жолдан кем емес жолдан тұратын мәтіндік файл берілген. Берілген файлдың 3 соңғы жолдарын алып тастау.
7. $k(<10)$ сан және k жолынан көп жолдан тұратын мәтіндік файл берілген. Берілген файлдың K-соңғы жолынан тұратын жаңа файл құру.
8. $K (<10)$ сан және k жолынан көп жолдан тұратын мәтіндік файл берілген. Берілген файлдың k-соңғы жолынан тұратын жаңа файл құру.
9. Мәтіндік файл және K саны берілген. K нөмірлі жолды файлдан алып тастау. Егер сол нөмірлі жол табылмаса онда файл өзгертуеу.
10. Мәтіндік файл және K саны берілген. Файлдағы бар бос жолдарды S жолына ауыстыру.

11. Мәтіндік файл берілген. Одан басқа бос жолдарын алып тастау.

12. S жолы және мәтіндік файл берілген. Бар бос жолды S қа болса жолына ауыстыру.

13. Мәтіндік файл берілген. Қатарынан бірнеше тұрған пробельді бір пробелге ауыстыру.

14. Мәтіндік файл берілген, ішіндегі мәтіннің сол жағы тегістелген. Бос емес жолдың басына керекті пробельдің санын қойып отырып мәтінді оң жағы және центр бойынша тегістеу.

15. Мәтіндік файл берілген ішіндегі мәннің сол жағы тегістелген. Абзацтары бір бірінен бір бос жол арқылы бөлінген. Әр бос емес жолдың бірінші пробелінен санын көбейте отырып мәтінді ені бойынша тегістеу.

16. Мәтіндік файл берілген. Егер әр абзац бір бірінен немесе бірнеше жолдар арқылы бөлінсе сол абзац снын табу.

17. Мәтіндік файл берілген. Егер әр абзац 5 пробелі бар азат жолдан басталса сол абзацтардың санын табу Абзац арасындағы бос жол есептемеу.

18. Мәтіндік файл берілген. Абзацтары 5 пробелі бар азат жолдан басталады бос жұп жол. Әр абзац арасына бір бос жол қойып шығу.

19. Мәтіндік файл берілген. Мәтіндегі кездескен тыныс белгілерінен тұратын символдық файл құру.

20. Мәтіндік файл берілген. Ең үлкен ұзындығы бар бірінші сөзді және минималды ұзындығымен соңғы сөз шығару.

21. N саны және мәтіндік файл берілген. Берілген файлдан N ұзындығы бар барлық сөздерден тұратын жолдық файл құру. Егер берілген файлда ұзындығы бар сөздер табылмаса онда жана файл бос қалдыру.

22. C-символ және мәтіндік файл берілген. Сол әріпке аяқталатын берілген файлдың сөздерінен тұратын жолдық файл құру.

23. N саны және мәтіндік файл берілген. Файлдан N нөмірлі абзац алып тастау. Абзацтардың алдында және соңыда тұрған бос жол алмау керек. Егер берілген нөмірлі абзац болмаса –файлды өзгертпей қалдыру керек.

24. N саны және мәтіндік файл берілген. Файлдан N нөмірлі абзац алып тастау. Абзацтардың арасынағы бос жол

есептемеу. Егер берілген нөмірлі абзац болмаса – файлды өзгертпей қалдыру керек.

25. Мәтіндік файл берілген оның әр жолында бүтін саны бар. Одан басқа бос жолдарын алып тастау.

26. Мәтіндік файл берілген оның әр жолында бүтін саны немесе нақты саны бар. Сандардың бүтін және нақты бөліктерінің қосындысын табу.

27. Мәтіндік файл берілген оның әр жолында бірнеше сандардың бейнелеуі бар. Сандарды сол ретімен орналастырып нақты сан файлын құру.

28. Name1 және Name 2 скі текстік файл берілген. Name1 файлдың әр жолының басына Name2 файлдың сәйкесті жолын қосу. Егер name2 файлы Name1 файлдан қысқа болса онда қалған Name1 файлдың жолдарын өзгертпей қалдыру.

29. NameT атты мәтіндік файл және бүтін сандық файл NameN берілген. name T файлдың әр жолының соңына NameN файлдың сәйкесті санын қосу егер Name N файлы NameT файл жолын өзгертпей қалдыру.

30. NameT атты мәтіндік файл берілген. Оның әр жолында бірінші 60 позиция тексте берілген ал қалған орын-нақты сандарға. Екі файлды құру; Name S атына ие жолдық және NameR атына ие нақты сандар файлы

№ 15- Зертханалық жұмыс.

Графикалық режимдегі жұмыс негіздері

Жұмыстың мақсаты: графикалық режимде қолданылатын процедураларды қолданып есептердің программасын құруға үйрену.

Төмен істі тапсырмаларда Паскаль тілінің графикалық мүмкіндіктерін пайдаланып программа құрыңыздар.

Жұмысты орындауға арналған тапсырма нұсқалары

1. Уақытты есептеп отыратын сиқыршымен кездестіңдер делік және ол сендерге жұлдыздар сырын ашып береді деп ойлайық. Неге аспанда жұлдыздар жымыңдайды? Аспандағы әрбір жұлдыз — бір адамның өмірі. Сендерге көне жұлдыздарды дәл уақытында өшіріп, жаңасын дер кезінде жағып отыру керек. Уақыт есептеушінің жұмысын көрсететін программа жазу керек. Есеп модель құрудан басталады: экрандағы әрбір жұлдыз нүктемен көрсетіледі және барлығы 20 жұлдыз жанып тұрсын делік (кездейсоқ түрде таңдап алынған): жанып тұрған бірінші нүктені өшіріп, оны кез келген басқа

орыннан жандыру керек, содан кейін екіншісін. т.с.с. Жиырмасыншы жұлдыз өшкеннен кейін қайтадан біріншісіне көшу керек.

2. “Жаңбыр” программасын жазындар — экранда (кездейсоқ күйде) 50 нүкте жоғарыдан төменге қарай қозғалып келе жатқандай болуы тиіс. Нүктелердің бірі экранның төменгі шекарасына жеткенде, ол қайтадан экранның жоғарғы бөлігінде пайда болуы керек.

3. “Жаңбыр” программасын кейбір нүктелер шапшаң, кейбіреулері баяу қозғалатындай етіп өзгертіндер.

4. “Жаңбыр” программасын нүктелер өз қозғалысын төменгі сол жақ бұрыштан бастап, жоғарғы оң жақ бұрышта аяқталатындай етіп өзгертіндер.

5. Қағазға оралған “кәмпіттің” бейнесін: диагональдары жүргізілген горизонталь тіктөртбұрыш пен оның екі жағына жалғасып жатқан тең қабырғалы үшбұрыштар салындар. Тіктөртбұрыштың сол жақ төменгі төбесінің координатасы (x, y) , биіктігі a -ға, ұзындығы $2a$ -ға тең болсын. Үшбұрыштың биіктіктері $a/2$. Тіктөртбұрыштың диагональдары мен қабырғаларынан пайда болған қарама-қарсы екі үшбұрышты бояндар.

6. Үш тісті және жарты дөңгелек пішінді арқасы бар мұнара салындар. Мұнараның сол жақ төменгі төбесінің координатасы (x, y) , табанының ұзындығы a . Қалған өлшемдерін a арқылы өрнектеп табындар. Мұнараны бояндар.

7. Куб салындар. Кубтың алдыңғы сол жақ төменгі төбесінің координатасы (x, y) , бүйір қыры a . Жоғарғы бүйір бетін бояндар, оң жақ бүйір бетінде диагональ жүргізіндер.

8. Алтыбұрышты тік пирамида салындар. Табанының алдыңғы сол жақ төбесінің координатасы (x, y) , табанындағы қыры a . Қалған өлшемдерін a арқылы өрнектеп табындар. Пирамиданың екі бүйір жақтарын бояндар.

9. Тіктөртбұрышты қиық пирамида салындар. Пирамида табанының төменгі сол жақ төбесінің координатасы (x, y) , табанындағы қыры a , жоғарғы жақтың қыры — $a/2$. Қалған өлшемдерін a арқылы өрнектеп табындар. Пирамиданың оң жақ бүйір бетін бояндар.

10. Қарайтылған “Е” әрпі тәрізді фигура салатын программа құрындар. Әріп құрайтын нүктелердің абсциссалары мен ординаталары жиым элементтері ретінде берілген. Сурет салу үшін циклге енгізілетін тек қана бір LINE операторын пайдаланындар.

11. Экранда боялған шырыша салатын программа жазындар. Сурет элементтерін жиымда сактап, өшіріп және содан кейін қайтадан экранға шығарындар.

12. Боялған 7 шырышаның суретін салатын программа жазындар. Шыршалар бір-біріне ұксас, горизонталь бойында бір қалыпта орналасады, шыршалардың биіктігі солдан оңға қарай сызықтық түрде өсуі керек. Шыршаларды бойлап Қызыл Телпек оларды уақытша “үсіп қалмасын” деп бір затпен жауып шығатын болсын.

13. Графикалық режимде ESC пернесін басқанша монитор экранына кездейсоқ күйде бірсыпыра нүктелер шығаратын программа құрындар.

14. Графикалық режимде ESC пернесін басқанша монитор экранында (100, 100) — (300, 200) тіктөртбұрышының нүктелерін кездейсоқ күйде жоятын программа құрындар.

15. Графикалық режимде ESC пернесін басқанша монитор экранының центрі (200, 200) нүктесімен дәл келетін және радиусы 80 болатын дөңгелек нүктелерін кездейсоқ күйде жоятын программа құрындар.

16. Шахмат тақтасының бейнесін салындар.

17. Экранда қарайтылған M әрпі пішіндес фигура салатын және бағыттауыш тілсызық (→, ←, ↑, ↓) пернелер арқылы басқарылып, фигураның көлемін + және – пернелерінің көмегімен өзгертетін программа құрындар.

18. Циклдік операторды пайдаланып 10 басқышы бар саты салындар, оның жұп және тақ нөмірлі басқыштарының көлемі әр түрлі болатын программа құрындар.

19. Шеңберге іштей бесбұрышты жұлдыз салындар.

20. Қызыл экранға 0..9 аралығындағы 1000 кездейсоқ санды ақ түспен шығару қажет. Мұнан соң экран түсін жасылға бояп, сандарды сары түспен шығарыңыздар.

21. Экранды барлық (8) фон түстеріне 5 секунд кідіріспен бояп шығарыңыздар. Экранның сол жақ жоғарғы бұрышына оның нөмірін жазып қою керек.

22. Қара экранға 1-ден 16-ға дейінгі сандарды 16 түрлі түске бояп шығару қажет. Әрбір сан бөлек жолда орналасатын болсын.

23. Алдын ала тағарылған экранды ақ түске бояп, жеті атаңыздың аттарын жеті түрлі түспен шығарыңыздар.

24. Қара экранға өзгеріп отыратын кездейсоқ түстер арқылы 200 “+” белгісін кездейсоқ түрде берілетін координаталық нүктелерге басып шығару керек.

25. Экранның жоғарғы жағына екі терезе салып, оның біріншісінің ішіне өз атыңызды, екіншісіне – фамилияңызды жазып қойыңыз. 10 секундтан кейін терезелер ішін тазартып, өз аты-жөніңіздің орнына досыңыздың аты мен фамилиясын орналастырыңыз.

26. Экранның сол жақ шетіне көк түсті терезе салып, оны оң жаққа қарай қадамын бір символ етіп жылжытып отыратын программа құру керек.

27. Экранға біртіндеп үлкейе бастайтын шағын қызыл терезе салу қажет. Үлкею қадамын X координатасы үшін 3, ал Y координатасы үшін 1 етіп алыңыздар.

28. Экранға біртіндеп бір символ аумағына дейін кішірейе бастайтын барынша үлкен сары терезе салу қажет. Кішірею қадамы X координатасы үшін 3, ал Y координатасы үшін 1.

29. Экранға сынып тұратындай етіп 9 терезе салыңыздар да, олардың ішіне әртүрлі физика формулаларын жазыңдар.

Емтиханға дайындалуға және өзін-өзі тексеруге арналған тапсырмалар, соның ішінде тесттер

1 Жиын берілуі:

A) (өрнек, өрнек) B) [өрнек... өрнек] C) [өрнек..өрнек] D) {өрнек,...,өрнек}

E) (оператор, өрнек)

2 Жиын базалық типі болып табылуы мүмкін:

A) real B) string C) 1..10 D) longint E) array

3 Графикалық модуль инициализациясы

A) assign() B) uses graph C) unit D) initgraph(айнымалы, айнымалы, өрнек)

E) reset(айнымалы)

4 Сызықтар сызу процедурасы:

A) line(айнымалы, айнымалы) B) line(айнымалы) C) lineto(айнымалы)

D) line(graph) E) linc(өрнек, өрнек, өрнек, өрнек)

5 Pascal тіліндегі негізгі программа түрі:

A) include B) retutn() C) function D) program {операторы} E) END

6 Pascal тіліндегі меншіктеу операторлық белгісі:

A) = B) # C) <> D) := E) =:

7 Pascal тіліндегі теңдік операциялық белгісі:

A) $= = B) \# C) \diamond D) \geq = E) =$

8 Pascal тіліндегі теңсіздік операциялық белгісі:

A) $= = B) != C) \diamond D) = E) ><$

9 Pascal тіліндегі инкремента операциялық белгісі:

A) ++айнымалы B) inc(айнымалы) C) ++өрнек D) inc(өрнек) E) add(өрнек)

10 ASCII-код дегеніміз не?!

A) ОЗУ ішкі қондырғылар арасында ақпарат алмасу коды: B) Мәліметтер оқу мүмкіндігі C) Мәліметтер беру ортасы D) АҚШ ақпараттық коды E) Код алмасу ортасы

11 Компьютер жадындағы ақпарат берілуі:

A) Екілік санау жүйесі түрінде B) Ондық санау жүйесі түрінде C) Мәліметтер қағаздағыдай түрінде D) Мәліметтер монитордағыдай түрінде E) Сегіздік санау жүйесі түрінде

12 Төмендегілердің қайсысы инструменталды программаларға жатпайды?

A) Трансляторлар B) Компиляторлар C) Интерпретаторлар D) Драйверлер

E) Утилиталар

13 Алгоритмдік және есептеу процесі ... түсінік.

A) әртүрлі B) ұқсас C) ұйқас D) виртуалды E) эквивалентті

14 Алгоритмдік процесс ол –

A) берілген есептің шығару жолын реттелген амалдар тізбегіне келтіру. B) алгоритм процесі бойынша сан мәндерді түрлендіру C) өзгерту коды. D) әртүрлі тәуелділікті моделдеу. E) Жүйе моделі

15 Есептеу процесі ол –

A) Мәліметтер беру. B) Берілген есептің шығару жолын реттелген амалдар тізбегіне келтіру. C) Алгоритм процесі бойынша сан мәндерді табу. D) Өзгерту коды. E) Өзгерту саны.

16 Есептеу класының үш класы бар:

A) Иерархиялық, параллелді, бірінен соң бірі. B) Сызықты, параллелді, бірінен соң бірі. C) Сызықты, параллелді, циклды. D) Мәліметтер беру. E) Тармақталу, параллелді, бірінен соң бірі.

17 Топтағы жазылған процестің бірнеше рет қайталануы қалай аталады?

A)сызықты. B) тармақталған. C) циклды. D) қайталанатын. E) біртіндеп.

18 1 ден N санына дейінгі сандар қосындысын анықтау үшін алгоритм типі:

A) Циклды. B) Тармақталған. C) **Сызықты**. D) Қайталанатын. E) Параллелді.

19 Синус функциясын есептеу кезіндегі алгоритм типін анықта.

A) Тармақталған. B) **Сызықты**. C) Циклды D) Қайталанатын. E) Біртіндеп.

20 $Y=a*X+b$ формуласымен есептеу үшін алгоритм типін тап:
A) **Сызықты**. B) Циклды. C) Тармақталған. D) Қайталанатын. E) Параллелді.

21 Косинус функциясын есептеу кезіндегі алгоритм типін анықта.

A) Тармақталған. B) Циклды. C) **Сызықты**. D) Біртіндеп E) Қайталанатын.

22 Алгоритм жазу түрлері (дұрыс емес жауапты көрсет).

A) Біріңғай жүйе жок. B) Графикалық. C) Сөздік. D) Программалау тілдері көмегімен. E) **Функционалды**.

23 Алгоритм ұғымы ЭЕМ ...шықты.

A) пайда болғаннан кейін B) пайда болғаннан дейін C) бір мезгілде D) негізінде E) параллелді

24 Алгоритм соңының қасиеті:

A) Алгоритмдегі торлар саны санмен шектелген. B) **Этап саны алгоритмін шектеу** C) Есептеуіш процесі шексіз қайталануы мүмкін емес D) Қадамдар саны белгілі бір санға тең болуы тиіс. E) Есепті шешу үшін минималды этап саны.

25 Алгоритм жасау кезіндегі рациональдылық:

A) Есепті шешу үшін минималды этап саны. B) **Минималды жады** орнын алатын мүмкіндік. C) Есепті көпнұсқалы **жолмен** шешу кезіндегі эффективтісін таңдау. D) Есепті көпнұсқалы жолмен шешу кезіндегі эффективті емесін таңдау. E) Алгоритм орындау кезіндегі этап санының.

26 Алгоритм негізі оны ... есептеулерге қолдануға болатынын көрсетеді.

A) әртүрлі типті B) бір типті C) кез-келген D) **структурлы** E) біртекті

27 Алгоритмді ұсынудың белгілі түрі:

A) Сөзбе сөз B) **Графикалық**, бағдарламалы. C) Тек **графикалық**. D) Тек бағдарламалы. E) Однородный

28 Блокты символ алгоритмдемынадай шартты орындауда

Қосымша әдебиеттер:

1. Абрамов С.А., Гнездилова Г.Г., Капустина Е.И., Селюн М.И. Задачи по программированию. –М.:Наука, 1988
2. Юркин А.Г. Задачник по программированию. –СПб.:Питер, 2002
3. Немнюгин С.А. Turbo Pascal: практикум. –СПб.: Питер, 2000
4. Гуденко Д., Петроченко Д. Сборник задач по программированию –СПб.:Питер, 2003
5. Климова Л.М. PASCAL 7.0 Практическое программирование. Решение типовых задач. –М.: КУДИЦ-ОБРАЗ, 2000
6. Бәрібаев Б. Компьютердің арифметикалық негіздері: Оқу құралы. –Алматы: Қазақ университеті, 2009. -80 б.
7. Анисимов А.Е., Пупышев В.В. Сборник заданий по основам программирования. 2007. // www.intuit.ru

Интернет-ресурстардың тізімі

1. <http://oop.rosweb.ru/cetus/software.html>
2. <http://www.unn.ru/vmk/graphmod/>
3. http://pmi.ulstu.ru/new_project/korobov/index.htm

Глоссарий

Алгоритм – бастапқы айнымалы түрде берілген мәліметтерден қажетті нәтижеге қол жеткізу жолында атқарылатын есептеу процесін анықтайтын дәлме-дәл нұсқаулар жиыны.

Алгоритмдеу – *есепті шығару алгоритмін құрастыру процесі*, мұның нәтижесінде мәліметтерді өңдеу процесінің кезеңдері айқындалады да, кезеңдер мазмұны формальды (жасанды) түрде жазылып, солардың орындалу реттілігі анықталады.

Блок-схема – алгоритмнің логикалық құрылымын графикалық түрде бейнелеу. Мұнда мәліметтерді өңдеудің әрбір кезеңі атқарылатын операцияға сәйкес әр түрлі геометриялық фигуралар (блоктар) түрінде көрсетіледі.

Жиын (массив) – қасиеттері ұқсас болып келетін бір типтегі айнымалылардың реттелген жиыны. Элементтер арасындағы олардың реттілігі жиым индекстері арқылы анықталады.

Интерпретатор – бір тілден екінші тілге командаларды аударып, оларды бірден орындауға арналған трансляторлық программа.

Компилятор – бұл алгоритмдік тілдің конструкцияларын толығымен машиналық кодқа түрлендіретін программа. Есептің нәтижесін алу үшін машиналық кодты орындау керек.

Объектілік модуль – алгоритмдік тілде бастанқы жазылған программа мәтіні машина тіліне аударылғаннан кейін шығатын программалық код.

Оператор – операциялар мен мәндерді көрсететін немесе солардың элементтерінің қай жерде орналасқанын білдіретін символдар жиыны.

Программа – машинаға түсінікті түрде жазылған алгоритм. Программада берілген мәліметтердің сипаттамаларымен бірге оларды өңдейтін командалар болады. Командалар қандай мәліметтер қандай операцияларға қатынасатынын, олар қандай реттілікпен орындалатынын және нәтиженің қандай түрде шығарылатынын көрсетеді. Бұлар операторлар арқылы жүзеге асырылады.

Программалау жүйесі – программалауды автоматтандыру құралдары. Олар программалау тілінен, осы тілдің трансляторынан, құжаттамаларынан және де программаларды дайындау, әрі орындау құралдарынан тұрады.

Программалау тілдері – программаларды ЭЕМ-де тікелей орындауға арналған алгоритмдерді жазу тәсілі. Программалау тілі – ЭЕМ-дерде программаларды орындау ісін атқарады

Псевдокодтар – нақты программалау тілінің синтаксистік ерекшеліктерін есепке алмай, тек формальды түрде программа логикасын бейнелеуге мүмкіндігі. Бұл тәсіл программалау тілінің операторлары мен табиғи тіл сөздерінің араласуы арқылы құрастырылып, блок-схема орнына пайдалануға болатын программа логикасын бейнелеу құралы болып табылады.