1-Дәріс. Алгоритм және оларды бейнелеу жолдары

Дәрістің мақсаты:

- Алгоритм мен программаның мәнін, олардың өзара байланысын түсіндіру.
- Алгоритмнің негізгі қасиеттерін (анықтылық, нәтижелілік, жалпылық, модульдік) меңгеру.
- Алгоритмнің өрнектелу жолдарын (табиғи тілде, псевдокодта, блок-схема түрінде) түсіндіру.
- Алгоритмдердің бірыңғай құрылымдарын (сызықтық, тармақталу, циклдік) қолдана білу.
- Алгоритмдерді программа түрінде жазудың логикасын және құрылымдық программалау қағидаларын қалыптастыру.

Негізгі тақырыптар

1. Алгоритм ұғымы және тарихы
2. Программа және оның алгоритммен байланысы
3. Алгоритм қасиеттері:
4. Алгоритмнің өрнектелу жолдары:
5. Алгоритмнің негізгі құрылымдары:
6. Циклдердің түрлері:
7. Блок-схемалар және олардың элементтері:
8. Программадағы басқару конструкциялары:

Алгоритм, программа ұғымдары

Алгоритм атауы атақты шығыс математигі абу Жафар Мұхаммед ибн Мұса әл-Хорезми (763-850 ж.ж.) есімінің латынша Algorithmi (Алгоритми) болып жазылуынан шыққан.

Ол санаудың ондық жүйесінде көпорынды сандармен арифметикалық амалдарды бағана түрінде орындау ережесін (алгоритмді) алғаш рет ойлап тапқан. Сол ереже осы күнге дейін қолданылып келеді.

Алгоритм – берілген есептің шығару жолын реттелген амалдар тізбегі түріне келтіру. Алгоритмді орындаушының (атқарушының) рөлін негізінен адам, компьютер, робот, т. б. атқарады.

Мысалы, y = (ax+b)(cx-d) функциясын есептеу төмендегі қарапайым іс-әрекеттерден тұрады:

1) a-ны x-ке көбейту, оны R1 деп белгілеу;
2) оған b-ны қосу, нәтижесін R2 деп белгілеу;
3) c-ны x-ке көбейту, оны R3 деп белгілеу;
4) одан d-ны алу, оны R4 деп белгілеу;
5) R2-ні R4-ке көбейту, оны y деп белгілеу

Алгоритмге тұрмыстан алып бір мысал келтіре кетейік. Студент болу үшін алгоритмнің мынандай қадамдарын орындау керек.

1. Орта мектепті бітіріп, тест (ҰБТ) тапсыру
2. Конкурстан өту
3. Керекті құжаттарды тест нәтижесімен бірге белгілі оқу орнының таңдап алынған мамандығына (колледжге, институтқа, академияға, университетке) өткізу.

Компьютерде орындау үшін есептің шешу жолы, яғни орындалатын әрекеттердің тізбегі әрі түсінікті, әрі дәл болуы қажет. Берілген есептің шешу жолдарының түсініктілігін оның алгоритмінің түсініктілігі деп қарастырады. Алгоритмде алдыңғы әрекеттің нәтижесі келесі әрекетте пайдаланылады. Алға қойған мақсатқа жету немесе берілген есепті шешу бағытында атқарушыға біртіндеп қандай әрекеттер жасау қажеттігін әрі түсінікті, әрі дәл етіп көрсететін нұсқаулар тобын алгоритм деп атайды. Төмендегі алгоритм анықтамаларын салыстырыңыздар:

1. Алгоритм – алғашқы берілген мәліметтерді пайда- ланып нақты нәтижеге қол жеткізу мақсатында шектелген командалар тізбегін орындау үшін атқарушыға берілетін түсінікті және нақты нұсқаулар тізбегі.
2. Алгоритм дегеніміз – берілген мәндерді пайдаланып қажетті нәтижеге жетуді жүзеге асыратын әрекет- тердің орындалу ережесі.
3. Алгоритм дегеніміз – алғашқы берілген мәліметтерді пайдаланып, қойылған мақсатқа жетуге немесе мәселені шешуге (есепті шығаруға) бағытталған әрекеттердің орындалуын жүзеге асыратын атқару- шыға түсінікті және нақты нұсқаулар тізбегі.

Алгоритмді компьютерде орындау үшін оны программа түрінде жазып шығу керек. Программа – алгоритмді машинаға түсінікті нұсқаулар тізімі (командалар) ретінде жазу. Программа машинаға түсінікті командалардан тұрады. Осы командаларды орындау барысында есептің нәтижесі шығады. Әрбір компьютер алдын ала жазылған программамен жұмыс істейді. Программа дегеніміз – белгілі бір нәтиже алу үшін орындалатын командалдардың айқындалған тізбегі. Процессор программаның құрамындағы командаларды біртіндеп кезекпен орындап отырады. Команда бір ғана қарапайым амалды орындау үшін берілген бұйрық ретінде беріледі. Командалар: арифметикалық немесе логикалық амал; ақпаратты тасымалдау командасы; берілген сандарды салыстыру командасы; нәтижені экранға, қағазға басып шығару командасы; келесі командаларға көшу тәртібін орындау, т.с.с. Компьютердің жұмысы программалық принципке негізделген, яғни ол өзінің жадында сақталатын командалар тізбегін автоматты түрде орындау арқылы есеп шығарады. Компьютер берілген тапсырманы орындауға дайын тұрған техникалық аспап болғандықтан, әрбір тапсырманы түсінікті түрде қысқаша жаза білу қажет. Команда бір ғана қарапайым амалды орындау үшін берілген бұйрық ретінде беріледі. Командалар: арифметикалық немесе логикалық амал; ақпаратты тасымалдау командасы; берілген сандарды салыстыру командасы; нәтижені экранға, қағазға басып шығару командасы; келесі командаларға көшу тәртібін орындау, т.с.с. Компьютердің жұмысы программалық принципке негізделген, яғни ол өзінің жадында сақталатын командалар тізбегін автоматты түрде орындау арқылы есеп шығарады. Тапсырма жекеленген командалардан тұрады.

Программа – арнайы мәтін арқылы компьютерге тапсырманың ретті кезегін хабарлайтын ережелер мен нұсқаулар тізбегі. Алгоритмдік тіл – алгоритмдерді жазуға арналған символдар мен сол символдардан тұратын конструкцияларды құрастыру және түсіндіру ережелерінің жиыны.

Алгоритмдеу дегеніміз – есепті шығару алгоритмін құрастыру процесі (үдерісі).

Мәліметтер дегеніміз – белгілі бір процесс көмегімен тасымалдап, өңдеуге болатын, формальды (жасанды) түрде бейнеленген фактілер мен идеялар. Программалау тілі компьютерлерде программаларды орындау ісін атқарады.

Алгоритм қасиеттері

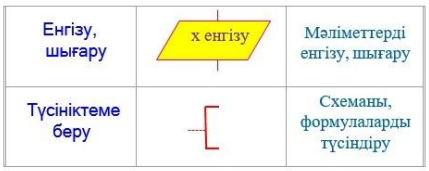
Алгоритмнің мәнін ашатын негізгі қасиеттері немесе оған қойылатын талаптар болады. Олар:
1) детерминділік (анықтылық, бір мәнділік) – басқаша түсінуге жол бермей, тек қана көрсетілген әрекеттерді айқын түрде орындауға арналған нұсқаулар дәлдігі, яғни алгоритм анық, әрі дәл өрнектелуі тиіс;
2) оның модульдік (бөлікке бөліну) қасиеті, яғни алгоритмді шағын бөліктерге бөлу мүмкіндігі болуы қажет, яғни есептеу процесін жекеленген қарапайым операцияларға бөлу қасиетінің болуы;
3) оның нәтижелілік (шектеулілік) қасиеті алгоритм шектелген уақыттан соң нәтиже беруі тиіс, яғни алгоритм қадамдары- ның саны шексіз болмауы керек;
4) бір типтегі (біртектес) есептерге жалпы бір ғана алгоритм қолданылуы тиіс – жалпылық қасиеті.

Алгоритмдерді компьютерде орындау үшін оларды алдын ала жазып алу керек, яғни ол белгілі бір заңдылықпен өрнектелуі тиіс. Жалпы алгоритмді өрнектеу түрлеріне:
1) табиғи тіл арқылы жазу;
2) белгілі бір түйінді сөздер – терминдер (псевдокод- тар - жалған кодтар) арқылы қысқаша тізбекті түрде жазу, мұны қарапайым алгоритмдік тіл деп те айтады;
3)график жолымен (блок-схема арқылы) жазу
4)программалау тілдерінде жазу жолдары жатады.
Алгоритмді табиғи тілде өрнектеу компьютерде қолданылмайды, өйткені онда дәлдік, нақтылық болмайды.
Ал алгоритмді екінші көрсетілген жолмен өрнектеу қарапайым алгоритмдік тіл деп аталып кеңінен қолданылып жүр. Мұны олардың ағылшын тіліне негізделіп жасалған программалау тілдеріне жақындығымен түсіндіруге болады.
Алгоритмдерді график жолымен жазу, онан кейін оны программалау тіліндегі программаға айнал- дыру істері мемлекеттік стандартпен бекітіліп ақпа- рат өңдеу жұмысында кеңінен қолданылып келеді.

1.1 Алгоритмдерді график жолымен жазу

Іс-әрекеттің аты	Блоктың түрі	Атқаратын жұмысы
Процесс (үдеріс)	y := a+b	Математикалық өрнектерді есептеу
Бастау, аяқтау	басы	Алгоритмдерді бастау, аяқтау
Қосалқы программа	E15	Қосалқы программаларға кіру және шығу

Таңдау	жоқ a > b? иә	Есеп шығару жолын таңдау
Модификация (цикл басы)	i := 1,10,1	Цикл (қайталау) басы
Құжат	x, y	Нәтижені баспаға (қағазға) шығару



Алгоритмдерді график арқылы бейнелеу түсінікті, анық, көрнекті түр болып есептеледі. Тек оларды сызу көбірек еңбекті талап етеді. Графикалық жолмен алгоритмдерді жазу үшін мемлекеттік стандарт белгіленген, онда кез келген амал белгілі бір геометриялық фигурамен (блоктармен) өрнектеледі. Ол фигуралар операциялар символы деп те аталады. Блоктар бағытталған сызықтармен байланысып, бірінен соң бірі орналасады.

Алгоритмдердің бірыңғай құрылымдары

Кез келген алгоритмді (программаны) блоктардың өзара байланысуына қарай төмендегідей үш түрлі басқару құрылымын пайдалану арқылы
жазып шығуға болатындығы дәлелденген:

• сызықтық құрылым немесе тізбектелген әрекеттер тізбегі;
• тармақты құрылым немесе шартты тексеру;
• қайталау немесе циклдік құрылым.

Осы үшеуі құрылымдық программалаудың негізгі конструкциялары, яғни құраушылары болып саналады. Программадағы кез келген әрекетті (операторды) оның кіру нүктесі арқылы тауып орындауға болады (осы тәсілмен табылмайтын операторлар және шексіз циклдер болмауы тиіс). Мұндай алгоритмді – программаны басқару ісі жоғарыдан төмен қарай жүргізіледі. Түсініктеме мәтін (комментарий) қосылған осындай программалар оқуға және түсінуге жеңіл болып есептеледі.

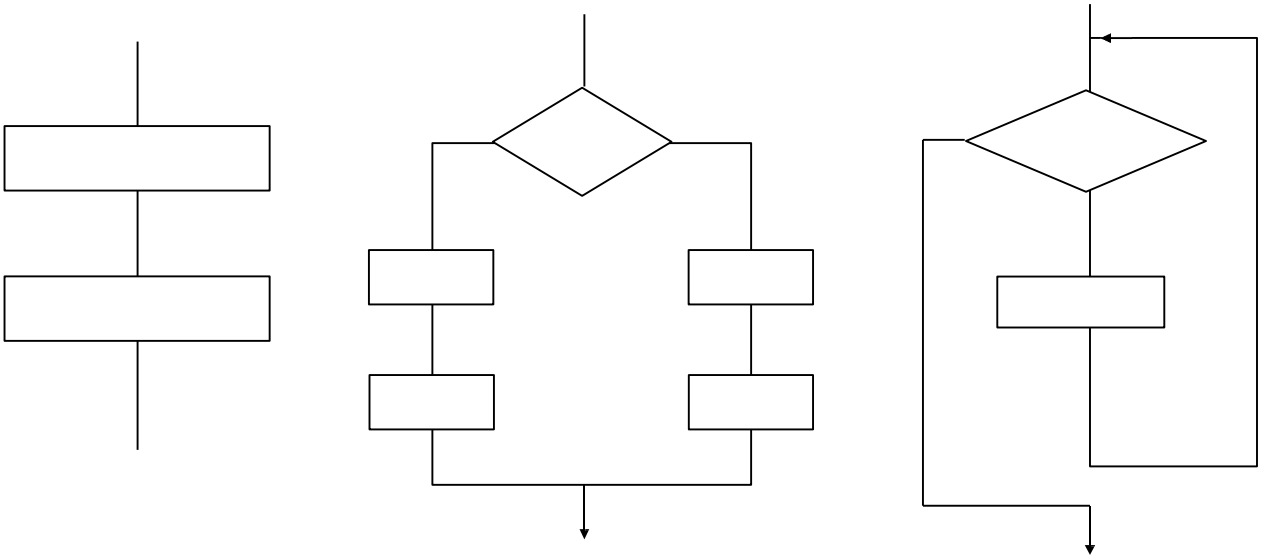
Оператор – тілдің қарапайым сөйлемі, ол белгілі бір әрекет немесе амал орындап, ; таңбасымен аяқталады.

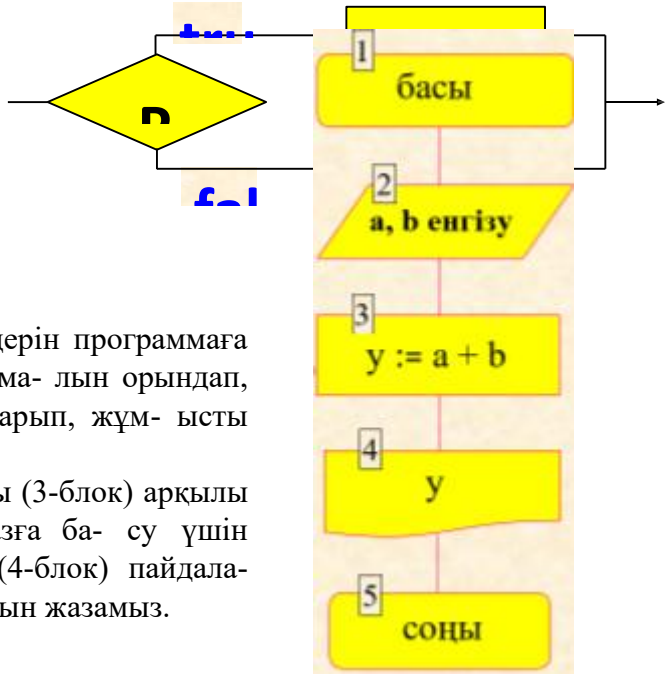
Сызықтық құрылым бірінен кейін бірі орындалып тізбектеле орналасқан бірне-ше операторлардан тұрады.

Тармақты құрылым – шартқа байланысты екі оператордың бірінің орындалуы.

Цикл – операторлар бөлігінің бірнеше рет қайталана орындалуы.

Төменде алгоритмдердің бірыңғай құрылымдарының сызбалық (схемалық) бейнеленуі көрсетілген





Сызықтықалгоритм

Мұнда *a* және *b*-ның сандық мәндерін программаға енгізіп (2-блок), содан кейін қосу ама- лын орындап, ақырында *y*-ті қағазға басып шығарып, жұм- ысты тоқтатамыз.

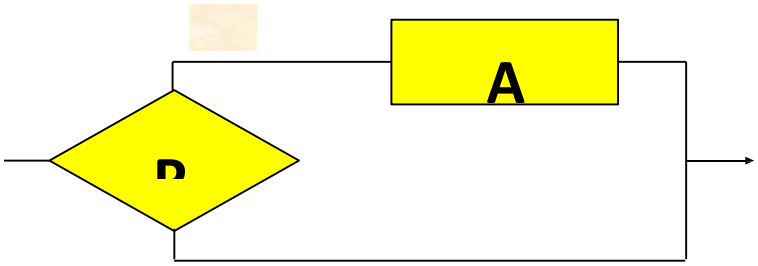
y = a + b формуласы есептеу блогы (3-блок) арқылы өрнек- теледі. Ал нәтижені қағазға ба- су үшін көпбұрышты құжат алу блогын (4-блок) пайдала- нып, оның ішіне нәтиженің атауларын жазамыз.

Тармақталу алгоритмдері

Тармақталу алгоритмінде арифметикалық теңсіздік (теңдік) түрінде берілген логикалық шарт тексеріледі. **P** шар- тының мәні ақиқат (true) немесе жалған (false) бола ала- тын логикалық өрнек түрінде болады. Егер ол орындалса – ақиқат болса, онда алгоритм бір жолмен, ал орындалмаса – екінші жолмен жүзеге асырылады, яғни есепті шы- ғару жолы тармақталып екіге бөлініп кетеді.

If (P) A; else B;

Бұл құрылым толымсыз (қысқаша) түрде болуы мүмкін, онда логикалық өрнектің мәні жалған болғанда ешқандай әрекет орындалмайды.



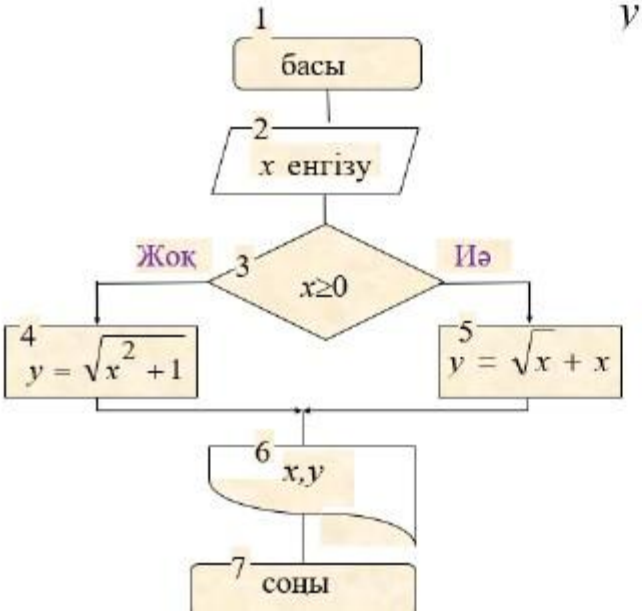
if (P) A;

Тармақталу алгоритмдері осы екі түрде кездеседі, олар "таңдау" және "аттап өту" мүмкіндіктерін іске асыруға көмектеседі.



1-мысал. y функциясын төмендегі формула бойынша есептеу керек

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

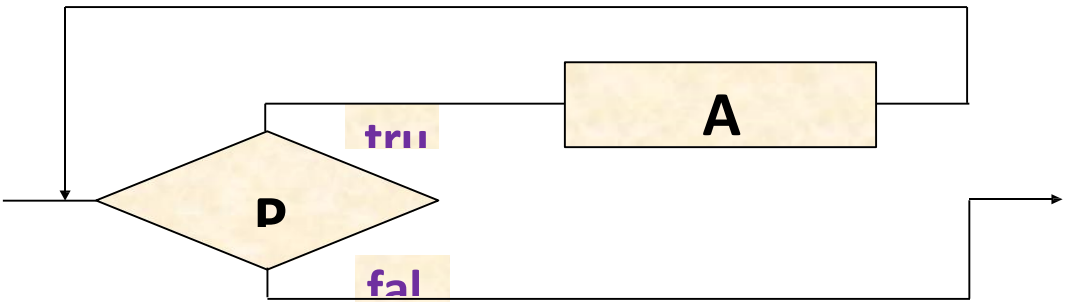


Циклдік алгоритмдер

Көптеген есептерді шығару кезеңінде бір теңдеуді пайдаланып, ондағы айнымалының өзгеруіне байланысты оны бірнеше рет қайталап есептеуге тура келетін сәттер де жиі кездеседі. Осындай қайталап орындалатын есептеу процесінің белгілі бір бөліктерін цикл деп атайды. Бірнеше рет қайталанатын бөлігі бар алгоритм- дер тобы циклдік алгоритмдерге жатады.Циклдік алгоритмдерді пайдалану оларды кейіннен программаларда цикл операторы түрінде қысқартып жазу мүмкіндігін береді.

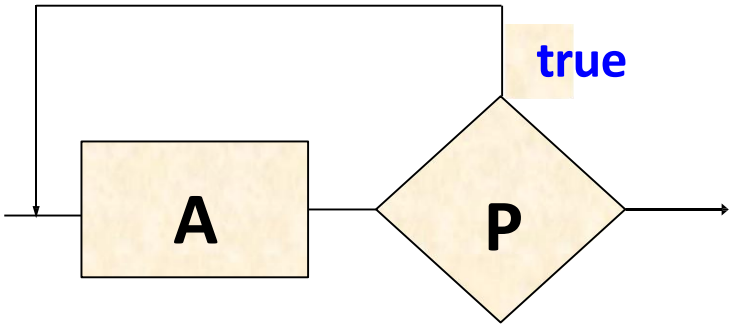
цикл – әзірше (алғы шартты цикл)

while (P) A ;



Мұнда **A** әрекеті **P** шарт мәні ақиқат болып тұрса, қай- талана береді. Сондықтан **A** әрекеті орындалуы кезінде **P**- ға әсер ететін айнымалылар мәні өзгеруі тиіс. Бұлай болмаған жағдайда шексіз цикл орын алады. Шарт мәні **A** әрекетіне дейін анықталады, сол себепті кейде шарт бірден жалған болса, **A** әрекеті бір де бір рет орындал- мауы да мүмкін.

цикл – дейін (соңғы шартты цикл)



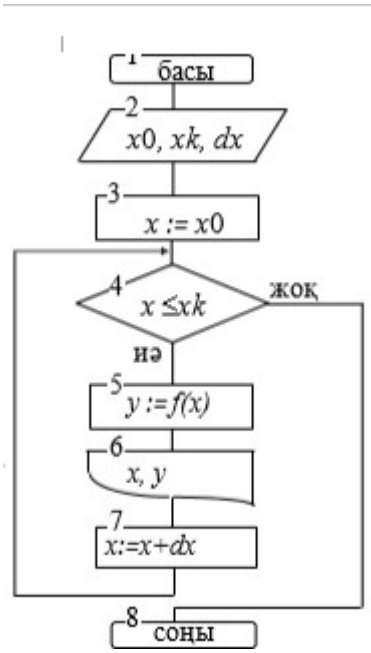
do A while (P);

Цикл – дейін түріндегі қайталау кем дегенде бір рет орындалады, өйткені шарт **A** әрекетінен кейін тексеріледі. **A** әрекеті **P** мәні жалған болған кезде орындалмайтын болады.

Циклдер қайталану санының алдын ала белгілі және белгісіз болуына байланысты екі топқа бөлінеді.

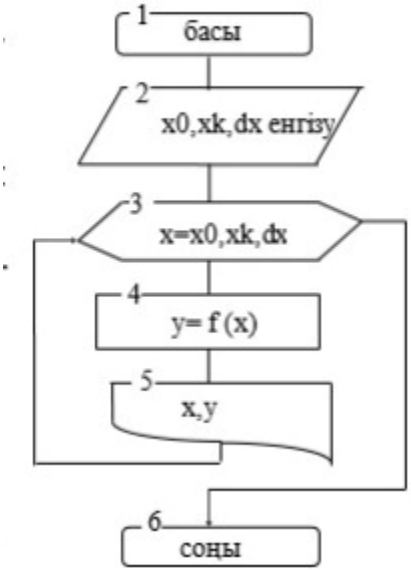
Қайталану сандары алдын ала белгілі болып келетін циклдер тобы арифметикалық цикл болып, ал орындалу саны белгісіз циклдер — қадамдық (итерациялық) цикл болып аталады.
Практикада белгілі бір айнымалының сандық мәніне байланысты орындалатын арифметикалық циклдер жиі кездеседі. Мұнда арифметикалық прогрессияға ұқсас болып келетін циклдер ең қарапайым арифме- тикалық (параметрлі) цикл болып табылады. Оны басқару прогрессияның заңына сәйкес цикл пара- метрінің сандық мәнімен байланысты болады.
Цикл орындалуы алдында оның айнымалы аргументі – параметрі алғашқы мәнге ие болуы керек, сонан соң қайталау кезінде цикл параметрі белгілі бір шамаға (қадамға) өзгере отырып, ол алдын ала берілген ең соңғы мәнге дейін жетуі қажет.

Алгоритмнің орындалу барысында цикл параметрі, мысалы, x өзінің ең алғашқы x0 мәнінен ең соңғы xk мәніне дейін тұрақты шамаға (dx) өзгеріп отырады. Осының нәтижесінде x мынадай мәндерді қабылдай- ды: x0, x0+dx, x0+2dx, ..., x0+(n-1)dx, xk, мұндағы n – циклдің қайталану саны.



Қарапайым циклдік алгоритм

Арифметикалық цикл үшін y=f(x) функциясының есеп- телу жолы алгоритм ретінде суретте көрсетілген. Мұндағы 3-ші, 4-ші, 7-блоктар циклді ұйымдастыру үшін қажет. Олар цикл параметрінің алға- шқы мәнін, өзгеру қадамын белгілеп және оның ең соңғы мәнінежеткен-жетпегенінтек- середі. Ал 5- және 6-блоктар бірнеше рет қайталанып цикл- дің өзін құрайды. 4-блок шартты тексеріп қайталану процесін ұйымдастырады.



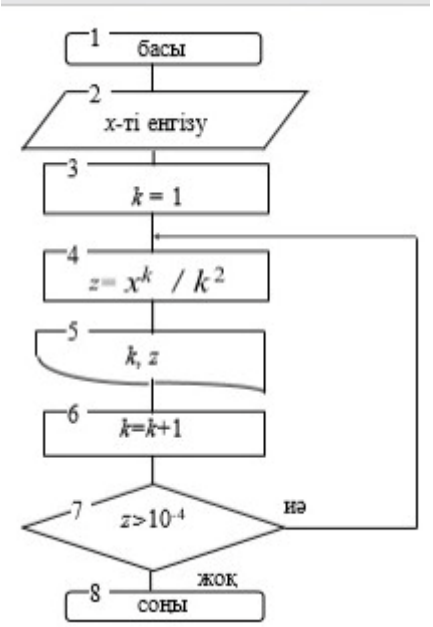
Модификаторлы циклдік алгоритм

Алгоритм салуды және прог- рамманы жазуды жеңілдету үшін цикл алгоритмдерін "модификатор" немесе "цикл басы" блогын пайдалану арқылы жазылады. Онда алд- ыңғы көрсетілген 3-ші, 4-ші, 7-блоктардың орнына "цикл басы" блогы орналасады. Ол алтыбұрышты фигурадан тұрады және оның міндеміндетті түрде екі кіру және екі шығу сызығы болуға тиіс.

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.

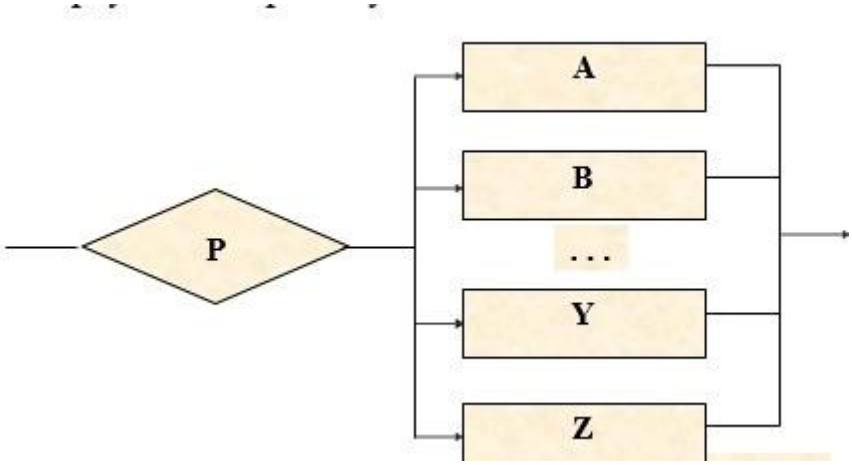
Қадамдық циклдердің схемасын сызғанда модификаторды (алтыбұрышты) қолдана алмаймыз, себебі алдын ала циклдің неше рет қайталанатыны бізге белгісіз. Енді осындай циклдер жұмысына мысал келтірейік.



3-мысал. $z = \frac{x^k}{k^2}$функциясының мәндерін k = 1, 2, 3, ... және Z
0.0001-ден артық болған жағдай- да есептейік, мұндағы $0 \leq x \leq 10$. Бұл мысалда алдын ала цикл не- ше рет қайталанатынын айта ал- маймыз, өйткені бізде тек k параметрінің алғашқы мәні мен қа- дамы ғана белгілі. Сонымен қатар Z функциясының 0.0001-ден артық болуы циклді қайталау шарты болып есептеледі (Z > 0.001). Суретте осы есептің алгоритм схемасы көрсетілген.

Қадамдық цикл алгоритмі

Таңдау – **case** ауыстырғыш (көп тармақты) құрылы- мы программалауды жеңілдететін мүмкіндік болып табылады. Таңдау құрылымы бірнеше мүмкіндіктер- дің біреуін ғана орындау кезінде өте қолайлы.



Программа жұмысын басқару операторларын программаның басқарушы конструкциясы деп атайды. Олар:

- құрама операторлар;
- таңдау операторлары;
- цикл операторлары;
- көшу операторлары болып бөлінеді.

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

Ұсынылатын әдебиеттер

1. Колдаев, В.Д. Основы алгоритмизации и программирования: Уч. пособие/ под ред. проф. Л.Г.Гагариной. -М.: ИД «ФОРУМ»: ИНФРА-М,2015.-416 с.
2. Бөрібаев Б. Компьютердің арифметикалық негіздері: Жоғары оқу орындары студенттеріне арналған оқу құралы. -Алматы: Қазақ университеті, 2009. -80 б.
3. Бөрібаев Б. Алгоритмдеу, мәліметтер құрылымы және программалау тілдері: оқулық. – Алматы: Қазақ университеті, 2012. -235 б.
4. Вирт Н. Алгоритмы и структуры данных: Пер. с англ. -М.: Мир, 2011.-360с.
5. Бөрібаев Б., Абдрахманова М. C# тілінде программалау (практикалық курс): оқу құралы –Алматы: Қазақ университеті, 2018. -258 б.
6. Troy Dimes. C# Programming for Beginners. Paperback – January 25, 2015.
7. Шилдт Г. C# 4.0: полное руководство. М.: ООО ”И.Д.Вильямс”, 2017. -1056 с.
8. Джуст В. Разработка обслуживаемых программ на языке C#. СПб.: Изд. дом «Питер», 2017.
9. Шарп Д. Microsoft Visual C#. Подроб. руководство. 8-е изд.С-Петербург: Изд. дом «Питер», 2016.
10. A. Troelsen, P. Japikse. Pro C# 7: With .NET and .NET Core. – Apress, 2017. -1372 p.