

Дәріс-1.

Тақырыбы: Алгоритм түсінігі, қасиеттері, бейнелеу әдістері. Алгоритмді бейнелеудің графиктік әдісі және алгоритмдік тілде бейнелеу

Жоспар:

1. Алгоритм түсінігі
2. Алгоритм қасиеттері
3. Алгоритмнің сөзбен бейнеленуі
4. Алгоритмді бейнелеудің графиктік әдісі
5. Алгоритмдік тілде бейнеленуі

Алгоритм – ол белгіленген ережелер бойынша жазылған есепті шығару әдісі (тәсілі).

Дәлірек айтқанда:

Алгоритм – ол қатан анықталған тізбектілік.

Кез келген есепті шешу - берілген алғашқы мәліметтерді пайдаланып, нәтиже алуды талап етеді, яғни есепті шешу алгоритмі алғашқы мәліметтерді бірте-бірте түрлендіру арқылы нәтиже алу жолын сипаттайды.

Көздеген мақсатқа жетуге, қойылған мәселені шешуге бағытталған әрекеттер тобын іске асыру үшін атқарушыға түсінікті әрі дәл нұсқаулар жиынтығы *алгоритм* деп анықталады.

Алгоритм сөзі IX ғасырда арифметикалық амалдардың орындалу тәртібін тұжырымдаған ұлы математик әл-Хорезми атының латынша *algorithmy* болып жазылуынан шыққан. Алғашында алгоритм тек көп таңбалы сандарға арифметикалық төрт амалды орындау ережесі мағынасында қолданылған. Кейінірек бұл ұғым жалпы қойылған мәселені шешуге әкелетін әрекеттер тізбегін белгілеу үшін қолданылатын болды.

Алгоритм жеке нұсқаулардан тұрады және нақты орындаушыға арналып жазылады. Сондықтан алгоритмде орындаушының мүмкіндіктері ескерілуі керек, біздің жағдайымызда алгоритмдердің орындаушысы - электронды есептеуіш машина. Компьютерге алгоритм белгілі бір программалау тілінде жазылып енгізіледі, компьютер программа нұсқауларын орындайды.

Қойылған мәселені шешудің дұрыс алгоритмі мынадай негізгі қасиеттерге ие болуы тиіс: анықтылық, нәтижелілік, жалпыламалық, дискреттік.

Алгоритмнің *анықтылығы* дегенде оның әрбір адымы әртүрлі түсінікке жол бермейтін, дәл және анық тұжырымдалған ережені түсінеміз.

Алгоритмнің *нәтижелілігі* деп санаулы қадам жасалған соң қарастырылған мәселенің не шешімі табылатынын, не шешімі жоқтығы анықталатынын айтады.

Алгоритмнің *жалпыламалық* қасиеті бойынша дайын алгоритммен жеке бір есепті ғана емес, осы есеп класына тиісті кез-келген есепті шығару мүмкін болуы тиіс.

Алгоритмнің *дискреттілігі* деп алгоритммен анықталған есептеулерді жекеленген кезеңдерге (қарапайым амалдарға) жіктеуді ұғамыз.

Қойылған мәселені шешу үшін осы негізгі қасиеттерді қанағаттандыратын алгоритм түзу мүмкін болмаса, мәселе “алгоритмдік жолмен шешілмейтін” деп аталады да, оны ЭЕМ-де шешу мүмкін болмайды.

Негізгі **қасиеттердің** қатарына келесілер жатады:

Алгоритмнің дискреттілігі. Бұл қасиет келесіні білдіреді. Алгоритм түрінде жазылған есептің шығарылуы жеке қарапайым командаларға бөлінген, ал олар орындалу реті бойынша орналасқан.

Алгоритмнің анықталғандылығы. Бұл қасиет келесіні білдіреді, алгоритмнің әр бір командасы орындаушыға түсінікті болу керек, басқаша талқылауға немесе түсінуге орын қалдырмау керек.

Алгоритмнің нәтижелілігі. Сатылы орындаулар жүргізілгеннен кейін алгоритмнің нәтижеге келтіру қасиеті.

Алгоритмнің жалпылығы. Бұл қасиеттің мәнісі келесіде. Кез келген есепті шығаруға арналып өңделген әр бір алгоритм, осыған типтес есептерді шығаруға қолданыла алыну керек (бастапқы берілгендердің мәндері жуық жағдайда).

Алгоритмдерді бейнелеу әдістері

Көп жағдайда алгоритм түзу дайын алгоритмді орындаудан қиын соғады, есепті шығару жолын білгенмен оны жинақты алгоритм түрінде көрсету оңай емес. Егер есепті шешу алгоритмі әбден пысықталса, оны көпшілікке түсінікті түрде қалай жазуға болады? Алгоритм түсінікті де көрнекі және ықшам болғаны дұрыс.

Алгоритмді

- қарапайым тілмен (сөзбен және формулалар арқылы) жазып көрсетуге
- схема (графиктік жол) арқылы кескіндеуге
- алгоритмдік тілде (программалау тілінде) жазуға болады.

Алгоритмді сөзбен жазып көрсеткенде мәліметтерді өңдеу кезеңдерін өзімізге түсінікті тілмен сипаттаймыз.

Сөздік әдіс мәліметтерді өңдеудің тізбекті этаптарын сипаттау болып табылады. Алгоритм еркін түрде беріледі.

Мысалы:

Шариктерді сорттау туралы есеп

Үш урна бар дейік (ақ, қара және жолақты). Жолақты урнада ақ және қара шариктер бар. Барлық қара шариктерді қара урнаға ал ақ шариктерді ақ урнаға салу керек. Сорттау келесі түрде жүргізіледі: жолақты урнадан бірінен соң бірі шариктер алынып түсіне байланысты ақ немесе қара урнаға салынады.

Алгоритм:

1. шарикті жолақты урнадан алу;
2. егер ол ақ болса, ақ урнаға салу;
3. егер ол қара болса, қара урнаға салу;
4. егер жолақты урна бос болмаса, онда 1 пунктке көшу керек;
5. соңы.

Сөздік әдіс кеңінен таралмаған, себебі:




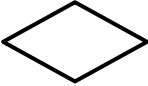
1. қатаң түрде жазылмайды;
2. жазуға көп сөздер қажет;
3. бөлек жазуларды жан жақты талқылауға мүмкіндік тудырады.

Графикалық әдіс сөздікпен салыстырғанда көрнекілігі жоғары және жинақы болып табылады.

Графикалық бейнелеу кезінде алгоритм өз ара тізбекті байланысқан функционалды блоктар түрінде бейнеленеді, олардың әр қайсысы бір немесе бірнеше жұмыстын атқарылуына сәйкес келеді.

Мұндай графикалық сипаттау алгоритмнің схемасы немесе *блок-схема* деп аталады. Блок-схемада әр бір типті құбылысқа (мәліметтерді еңгізуге, мәндердің шамаларын есептеу, шарттарды тексеру, өңдеудің аяқталуына және т.б.) *блоктық символ* түрінде көрсетілген геометриялық фигура сәйкес келеді. Блоктық символдар *өту сызықтарымен* жалғастырылады, олар орындалу реттерің анықтайды.

Кестеде кеңінен қолданылатын символдар келтірілген.

Символдың аталуы	Белгіленулер және оларды толтыру мысалы	Түсініктер
Қосу — тоқтату		Алгоритмнің басы, соңы, ішкі программаға кіру және шығу
Енгізу — шығару		Мәліметтерді аттары берілген сақтау ұяшықтарына енгізу моментің немесе осы ұяшықтардан мәліметтерді мониторға немесе принтерге шығару моментін белгілейді.
Процесс		Есептеу немесе оларды орындалу реті
Шешімі (шарттардың)		Шарттарды тексеру

Псевдокодтар

Псевдокодтар – шартты алгоритмдік тілде алгоритмдерді жартылай формалданған сипаттаулар. Программалау тілінің элементтерімен қатар жалпы математикалық белгілерді, кәдімгі тіл тармақтарын қамтиды.

Псевдокод алгоритмдерді біркелкі жазу үшін арналған ережелердің және белгілердің жүйесі.

Псевдокод формальды тілмен кәдімгі тілдердің орталық орынын алады. Бір жағынан ол кәдімгі тілге жақын, сондықтан ондағы алгоритмдер кәдімгі мәтін ретінде оқылып жазылады. Басқа жағынан онда бірқатар формалды конструкциялар және математикалық символдар қолданылады, ол алгоритмнің жазылуын күнделікті математикалық түрдегі жазуға ұқсастырады.

Псевдокодтарда формалды тілдерге тән біршама конструкциялар бар, сондықтан алгоритмді жазған кезде псевдокодтардан формалды тілге ауысу жеңіл. Формалды тілдегідей, псевдокодтарда *қызметті сөздер* бар, олардың мәндері бірден және өзгеріссіз анықталған. Жазу барысында олар қалың шрифт пен бөлінеді, ал қолман жазылғанда сызықпен белгіленеді.

Псевдокодтардың біркелкі немесе формалды анықтамалары жоқ, сондықтан қызметті сөздер және негізгі (базалық) конструкциялар жиынтығымен түрленетін әртүрлі псевдокодтар болу мүмкін.

Алгоритмнің жалпы түрі

алгоритмнің аты (аргументтер және нәтижелер)

берілгені алгоритмнің қолданылу шарты

қажет алгоритмді орындау мақсаты

бас аралық шамаларды сипаттау командалардың реті (алгоритмнің өзі)

ақталуы

Бақылау сұрақтары:

1. Алгоритм дегеніміз ?
2. Негізгі алгоритм түрлері?
3. Алгоритмнің бірдейлігі?
4. Алгоритмнің нәтижелігі?
5. Алгоритмнің дұрыстығы?
6. Алгоритмнің жалпыға бірдейлігі?
7. Алгоритмдік қате дегеніміз не ?

Әдебиеттер тізімі:

1. С.А.Немнюгин. Программирование на языке высокого уровня. Turbo Pascal: Учебник. 2-е изд.-СПб.: Издательство «Питер»,2003.
2. С.А.Немнюгин. Turbo. Pascal. Практикум. 2-е изд.–СПб.: Издательство «Питер»,2005.
3. Гусева А.И. Учимся программированию.PASCAL.7. Задачи и методы их решения. - 2-е изд, перераб, и доп.-М.; “ Диалог -МИФИ“, 2003.