16. Динамикалық деректер құрылымы, стек, кезек, декті сипаттап беріңіз

Динамикалық деректер құрылымдары-бұл жад қажет болған жағдайда бөлінетін және босатылатын мәліметтер құрылымы.   
Жадта өмір сүру процесінде деректердің динамикалық құрылымдары олардың құраушы элементтерінің санын ғана емес, сонымен қатар элементтер арасындағы қатынастардың сипатын да өзгерте алады. Бұл деректер элементтерінің мазмұнын өзгертуді ескермейді. Динамикалық құрылымдардың олардың мөлшерінің және элементтер арасындағы қатынастардың табиғатының сәйкес келмеуі сияқты ерекшелігі, машиналық кодты құру кезеңінде компилятор бағдарламасы бүкіл құрылым үшін белгіленген көлемдегі жад бөлігін бөле алмайды, сонымен қатар нақты мекен-жайларды құрылымның жеке компоненттерімен салыстыра алмайды. .

Динамикалық деректер құрылымы мыналармен сипатталады:

* оның аты жоқ;
* бағдарламаны орындау барысында оған жад бөлінеді;
* Құрылым элементтерінің саны бекітілмеуі мүмкін;
* Бағдарламаның орындалу барысында құрылымның өлшемі өзгеруі мүмкін;
* Бағдарламаны орындау барысында құрылым элементтері арасындағы өзара байланыстың сипаты өзгеруі мүмкін.

Деректердің әрбір динамикалық құрылымы көрсеткіш түрінің статикалық айнымалысын (оның мәні-осы объектінің мекен-жайы) салыстырады, ол арқылы динамикалық құрылымға қол жеткізуге болады.   
Динамикалық құрылымдар, анықтама бойынша, жадтағы Құрылым элементтерінің физикалық іргелес болмауымен, оны өңдеу процесінде құрылым мөлшерінің (элементтер санының) тұрақсыздығымен және болжанбауымен сипатталады.  
Динамикалық деректер құрылымын жіктеу   
көптеген тапсырмалар бағдарламаны орындау барысында конфигурациясы, өлшемдері мен құрамы өзгеруі мүмкін деректерді пайдалануды талап етеді. Оларды ұсыну үшін динамикалық ақпараттық құрылымдар қолданылады. Мұндай құрылымдарға мыналар жатады:

* бір бағытты тізім;
* көп бағытты тізім;
* циклдік тізімдер;
* стек;
* дек;
* кезек;
* екілік ағаштар.

Олар жеке элементтерді және рұқсат етілген операцияларды байланыстыру әдісімен ерекшеленеді. Динамикалық құрылым жедел жадтың іргелес емес бөліктерін иелене алады.  
  
 **Стек** (ағылш. stack-стек) - Бұл элементтердің реттелген жиынтығы болып табылатын мәліметтер құрылымы, онда жаңа элементтерді қосу және барын жою стек шыңы деп аталатын бір ұшынан жасалады. Сонымен қатар, стектің біріншісі оған соңғы орналастырылған элементті алып тастайды, яғни стекте "соңғы кірді — бірінші шықты" стратегиясы жүзеге асырылады (соңғы-in, бірінші-out — LIFO). Шынайы өмірдегі стектің мысалы табақтар жиынтығы болуы мүмкін: табақты шығарғымыз келгенде, жоғарыдағы барлық тақтайшаларды алып тастауымыз керек. Стек операцияларының сипаттамасына оралыңыз:

1. emptyempty-стек элементтерін тексеру,
2. pushpush (стекке жазу) — жаңа элементті енгізу әрекеті,
3. poppop (дестеден шығару) — жаңа элементті жою әрекеті.

**Дек** (ағылш. deque-double ended queue) — жаңа элементтерді қосу және бар элементтерді жою екі ұшынан жасалатын элементтер тізімін білдіретін деректер құрылымы. Бұл құрылым FIFO-ны да, LIFO-ны да қолдайды, сондықтан оны стек те, кезек те жүзеге асыра алады. Бірінші жағдайда сіз тек бастың немесе құйрықтың әдістерін, екіншісінде — екі түрлі ұштың итеру және pop әдістерін қолдануыңыз керек. Деканы екі жақты кезек ретінде қабылдауға болады. Оның келесі операциялары бар:

1. emptyempty-элементтерді тексеру,
2. pushBackpushBack — соңына жазу) - жаңа элементті соңына енгізу операциясы,
3. popBackpopBack (соңынан шығару) — соңғы элементті жою әрекеті,
4. pushFrontpushFront — басына жазу) - жаңа элементті басына енгізу операциясы,
5. popFrontpopFront (басынан бастап алып тастау) — бастапқы элементті жою әрекеті.

**Кезек** (ағыл. queue) - бұл мәліметтер құрылымы, элементтерді қосу және жою сәйкесінше pushpush және poppop операциялары арқылы жүзеге асырылады. Сонымен қатар, бірінші кезекте алдымен орналастырылған элемент жойылады, яғни кезекте "бірінші кірді — бірінші шықты" принципі жүзеге асырылады (ағылш. first-in, first-out — FIFO). Кезектің басы бар (ағылш. head) және құйрық (ағылш. tail). Элемент кезекке қойылған кезде, ол құйрығында орын алады. Оның басында тұрған элемент әрқашан кезектен шығады. Кезек келесі әрекеттерді қолдайды:

1. emptyempty-ондағы элементтердің кезегін тексеру,
2. pushpush (кезекке жазу) - жаңа элементті енгізу әрекеті,
3. poppop (кезектен шығару) - жаңа элементті жою әрекеті,
4. sizesize-кезекте элементтер санын алу операциясы.