СЫЗЫҚТЫ ҚҰРЫЛЫМДАР, STL ЖӘНЕ БАЗАЛЫҚ КОНТЕЙНЕРЛЕР

СЫЗЫҚТЫ ҚҰРЫЛЫМДАР

Сызықты құрылымдар — деректердің белгілі бір тәртіппен орналасқан түрлері. Оларда деректер бірінен соң бірі тізбектеліп орналасады, және әр элементтің нақты алдыңғы және кейінгі элементтері болады. Сызықты құрылымдар деректерді сақтау, басқару және өңдеу үшін кеңінен қолданылады. Сызықты құрылымдар программалау тілдерінде деректерді ұйымдастырудың тиімді тәсілдерін ұсынады.

Олардың басты мақсаттары:

- *Деректерді сақтау*: Сызықты құрылымдар деректерді жүйелендіру арқылы оларды сақтау мен ұйымдастыруды жеңілдетеді.
- *Жылдам қол жеткізу*: Массивтерде индекс бойынша элементтерге жылдам қол жеткізу мүмкіндігі бар.
- Динамикалық өзгерістер: Тізімдер мен стектер элементтерді оңай қосып немесе жоюға мүмкіндік береді, бұл программаның жұмысын тиімді етеді.
- *Реттеу және өңдеу*: Деректерді сұрыптау, іздеу және өңдеу алгоритмдерін қолдану арқылы структуралардың көмегімен оларды тиімді басқаруға болады.
- *Кодтың оқылымдылығы*: Сызықты құрылымдар кодыңыздың құрылымын жеңілдетіп, оның оқылымдылығын арттырады.

СЫЗЫҚТЫ ҚҰРЫЛЫМДАРДЫҢ ТҮРЛЕРІ

01

Массив
Өлшемі белгілі деректерді сақтау үшін қолданылатын тұрақты өлшемді құрылым.

02

Тізім
Құрамдас бөліктерінің орындары өзгеруі мүмкін деректер жиынтығы.

03

Стек
LIFO (Last In, First Out) принципімен жұмыс істейтін құрылым.

04

Кезек
FIFO (First In, First Out) принципімен жұмыс істейтін құрылым.

МАССИВТЕР ЖӘНЕ ДИНАМИКАЛЫҚ МАССИВТЕР

Массив

Өлшемі белгілі, деректердің типтері бірдей элементтер жиынтығы. Массивке жаңа элемент қосу немесе одан элементті алып тастау, массивтің өлшемін өзгертуге әкеледі. Массивтің өлшемі компиляция кезінде белгілі болуы керек.

Динамикалық Массив

Өлшемі өзгертілетін, деректердің типтері бірдей элементтер жиынтығы. Массивтің өлшемі жүгіру уақытында өзгертілуі мүмкін. Динамикалық массивтер жадыны тиімді пайдалануға мүмкіндік береді, өйткені олар тек қажетті мөлшерде жадыны бөледі.

ТІЗІМДЕР ЖӘНЕ ВЕКТОРЛАР

Тізім

1.Тізімнің элементтері бір-біріне байланысқан (байланысты тізім) және әр элемент өзінен кейінгі элементке сілтеме береді.
2.Тізімге элемент қосу және одан элементті алып тастау тиімді.
3.Тізімнің элементтеріне кездейсоқ қол жеткізу тиімсіз.

Вектор

1.Тізім сияқты, бірақ элементтер жадыда бір-бірінің жанында орналасады, біркелкі орналасқан.
2.Векторға элемент қосу және одан элементті алып тастау тиімді.
3.Векторға элементтерге кездейсоқ қол жеткізу тиімді.

СТЕКТТЕР ЖӘНЕ КЕЗЕКТЕР

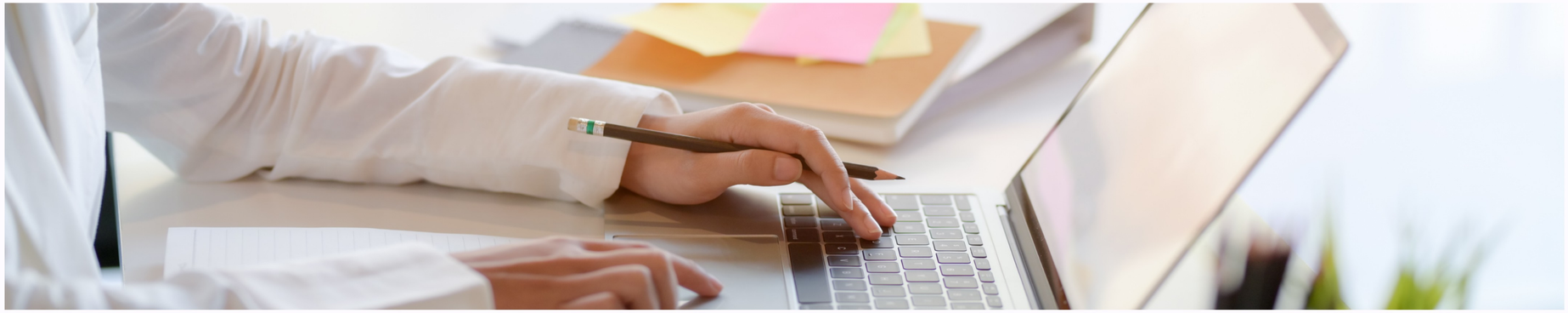
- Стек

LIFO (Last In, First Out) принципімен жұмыс істейтін деректер құрылымы. Соңғы қосылған элемент ең алдымен алынады. Стек қоңырау стектерін басқаруда, тереңдікке байланысты іздеу алгоритмдерінде қолданылады.

- Кезек

FIFO (First In, First Out) принципімен жұмыс істейтін деректер құрылымы. Ең алдымен қосылған элемент ең алдымен алынады. Кезектер жұмыс кезектерін басқаруда, принтерде басып шығару кезектерін басқаруда қолданылады.

КОНТЕЙНЕР ҰҒЫМЫ ЖӘНЕ КОНТЕЙНЕРЛЕРДІҢ ТҮРЛЕРІ



Контейнер

Деректерді сақтау және басқару үшін қолданылатын деректер құрылымы. Контейнерлер әртүрлі деректер типтерін сақтауға мүмкіндік береді. STL кітапханасы контейнерлердің кең ауқымын ұсынады.

Контейнерлердің түрлері

Контейнерлердің екі негізгі түрі бар: сызықты контейнерлер және жиынтық контейнерлер. Сызықты контейнерлер элементтерді ретпен сақтайды, ал жиынтық контейнерлер элементтерді ретсіз сақтайды.

STL (STANDARD TEMPLATE LIBRARY)

Стандартты шаблон кітапханасы

C++ тілінің бөлігі болып табылатын кітапхана. STL шаблондарды қолдана отырып, деректер құрылымдары мен алгоритмдердің кең ауқымын қамтиды.

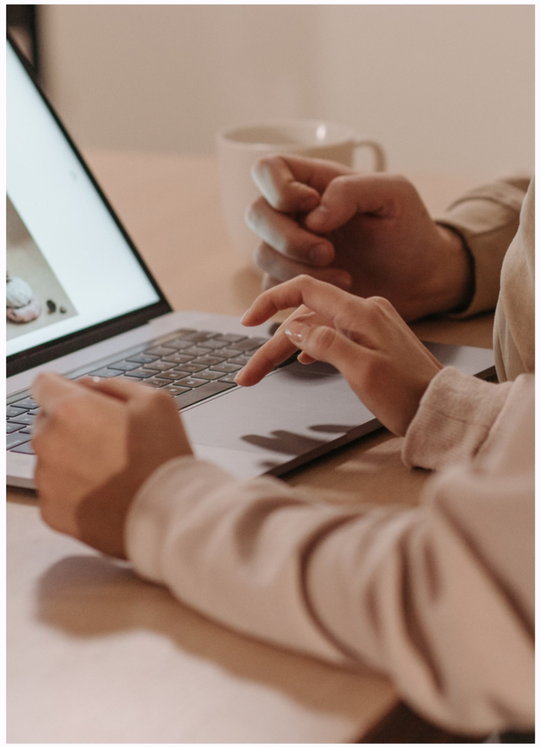
Қайта пайдалану

STL-дегі контейнерлер мен алгоритмдерді әртүрлі бағдарламаларда қайта пайдалануға болады. Бұл кодтың қайта пайдалануын арттырады және бағдарламалауды жеңілдетеді.

Жадыны басқару

STL жадыны тиімді басқаруға мүмкіндік береді, өйткені ол динамикалық жадыны бөлудің және босатудың жауапкершілігін өз мойнына алады.

БАЗАЛЫҚ КОНТЕЙНЕРЛЕР:
VECTOR, LIST, DEQUE



1.**Вектор**

-Тізім сияқты, бірақ элементтер жадыда бір-бірінің жанында орналасады. Векторға элемент қосу және одан элементті алып тастау тиімді.

2.**Тізім**

-Тізімнің элементтері бір-біріне байланысқан және әр элемент өзінен кейінгі элементке сілтеме береді. Тізімге элемент қосу және одан элементті алып тастау тиімді.

3.**Deque**

-Вектор мен тізімнің артықшылықтарын біріктіреді. Деққа екі жағынан да элементтерді тиімді қосу және одан элементтерді алып тастауға болады.

ТАҚЫРЫП БОЙЫНША МЫСАЛ