2. Сравнение линейных и динамических структур

	Линейные структуры	Динамические структуры
1	Базисное множество – множество элементов	Элементы базисного множества являются структурами ("ДС – структуры структур")
2	Базисное отношение – отношение следования	Базисное отношение – отношение включения
3	В структуре хранения хранятся все элементы структуры	В структуре хранения хранится только текущий элемент структуры
4	Отношение следования реализуется при помощи адресной арифметики	Отношение включения реализуется при помощи программ

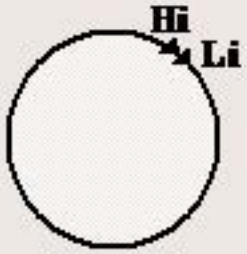
Определение1.10. Программы, реализующие отношение включения называются *средствами поддержания динамической структуры*

4. Использование нескольких структур и необходимость динамического распределения памяти

Пример 1.7. Стек для хранения фиксированного количества последних записанных значений

Стек подобного вида может быть использован, например, для запоминания данных, достаточных для восстановления текущего состояния программы перед выполнением очередной операции обработки (для реализации процедуры типа операции отмены в редакторе текстов). При реализации данного механизма обычно фиксируется максимальная глубина отката; при заполнении памяти из стека удается наиболее старая записанная информация.

↳ Наиболее простой способ реализации такого стека – использование кольцевого буфера (при заполнении стека следующий для использования по кольцу элемент памяти содержит подлежащее удалению значение данных)



☑ Как реализовать процедуру отмены операции отката ?

4. Использование нескольких структур и необходимость динамического распределения памяти

Пример 1.8. Задача использования двух стеков

☑ При использовании двух стеков может возникнуть ситуация переполнения одного стека, в то время как в другом стеке свободная память может еще присутствовать – эффективность использования памяти в этом случае не будет являться максимально-возможной

↳ Возможное решение проблемы – использование общей памяти для этих стеков, в которой один стек будет "расти" слева направо, второй стек – справа налево.

☑ При таком подходе свободная память является общей и ситуация переполнения стеков возникает только при полном исчерпании памяти

1. Разработка спецификаций...

- S - стек, $|S|$ - количество элементов в стеке
- S_n^k - стек, n – размер памяти, k – количество значений,
- δ - код выполнения операции (=1 – стек пуст, =2 – стек полон)

Операции

- $S(n) \rightarrow S_n^0$ - создание стека
- $S<<x \rightarrow S'$ – вставка в стека (при $k=n \Rightarrow S' = S$ и $\delta=2$)
- $S>>x \rightarrow S'$ - исключение из стека (при $k=0 \Rightarrow S' = S$ и $\delta=1$)
- $\alpha_0(S)$ - предикат проверки пустоты стека ($\alpha_0(S)=1$ при $k=0$)
- $\alpha(S)$ - предикат проверки переполнения памяти ($\alpha(S)=1$ при $k=n$)

Аксиомы

- $S=S(n) \Rightarrow S = S_n^0$ и $\alpha_0(S)=1$ – созданный стек является пустым
- $S'=S<<x \Rightarrow \alpha_0(S)=0$ – стек, в который выполнена вставка, не пуст
- $S'=S>>x \Rightarrow \alpha(S)=0$ – стек после операции исключения не полон
- $S'=(S<<x)>>y \Rightarrow x=y$ – при выборке значения из стека извлекается последнее вставленное значение
- $S'=S>>x$, $\alpha_0(S)=1 \Rightarrow S' = S$ и $\delta=1$ – выборка значения из пустого стека устанавливает код завершения $\delta=1$
- $S'=S<<x$, $\alpha(S)=1 \Rightarrow S' = S$ и $\delta=2$ – вставка значения в полный стек устанавливает код завершения $\delta=2$

3.2. Расширенный вариант реализации – схема наследования

☑ Класс TDataRoot является абстрактным базовым классом и обеспечивает управление памятью и определяет спецификацию операций, которые должны обеспечиваться структурой данных:

Управление памятью – память для хранения элементов структуры может выделяться двумя различными способами (использованный вариант фиксируется в переменной MemType):

- память выделяется конструктором из динамически-распределяемой области (MemType=MEM_HOLDER)
- память передается объекту при помощи метода SetMem (MemType=MEM_RENTER)

Поля

- pMem - указатель на память для хранения значений
- MemSize – размер памяти
- DataCount – количество значений, хранимых в структуре

Методы

- IsEmpty – проверка пустоты структуры
- IsFull – проверка переполнения памяти
- Put – операция вставки нового значения в структуру
- Get – операция выборки очередного значения из структуры

3.2. Расширенный вариант реализации – схема наследования

☑ Класс TStack обеспечивает реализацию динамической структуры стек:

Управление памятью – хранение значений осуществляется в векторе памяти от младших элементов к старших; индекс последнего занятого элемента в векторе памяти фиксируется в переменной Hi

Методы

- GetNextIndex – метод для получения следующего значения индекса (скрытие способа реализации отношения следования)
- Put – операция вставки нового значения в структуру
- Get – операция выборки очередного значения из структуры

Программирование с защитой от ошибок

☑ При разработке программного кода проверяется выполнимость всех условий, которые априори должны иметь место (допустимость значений переменных, истинность тех или предикатов и т.п.). При обнаружении ошибок:

- аварийное завершение программы (с той или иной диагностикой)
- формирование специальных кодов завершения, которые могут быть проанализированы на более высоком уровне управления

↳ Для управления кодами завершения может быть разработан общий базовый класс TDataCom

- при завершении любого метода производного класса должна вызываться процедура SetRetCode(int RetCode)
- для получения кода завершения последней выполненной программы используется метод int GetRetCode()

Программа

1. Проблема эффективного использования памяти

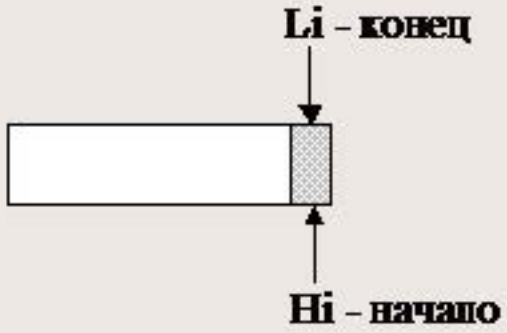
Очередь

☑ Вставка значений происходит в начало очереди, исключение значений – с конца очереди ⇒ для индикации начала и конца очереди требуется два индекса

Выборка из элемента с индексом Li и увеличение Li

Вставка - увеличение Hi и запись в элемент с индексом Hi

В ходе вычислений может возникнуть ситуация Li= Hi=n-1. Тогда вставка нового значения невозможна, а $E_{mem} \approx 0$ (!)

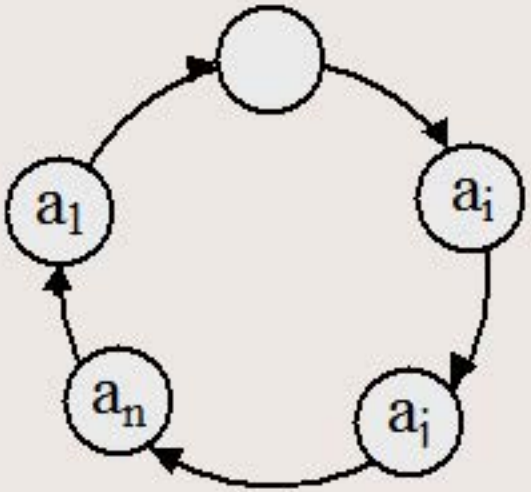


2. Способы достижения полного использования памяти…

- сдвиг значений очереди после выборки очередного значения (т.е. обеспечение Li=0) – возрастание накладных расходов

– использование левого участка свободной области при достижении Hi=n-1 (т.е. при отсутствии свободной памяти справа)

Определение 1.12. Структура хранения, получаемая из вектора расширением отношения следования парой $p(a_n, a_1)$, называется *циклическим или кольцевым буфером*.



2. Способы достижения полного использования памяти…

☑ Реализация кольцевого буфера логически может быть обеспечена переходом индексов Li и Hi при достижении граничного значения MemSize-1 на индекс первого элемента вектора памяти

Ситуации для структуры хранения очереди