

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. Н.И. ЛОБАЧЕВСКОГО
ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МАТЕМАТИКИ И МЕХАНИКИ

Учебный курс

МЕТОДЫ ПРОГРАММИРОВАНИЯ - 2





Нижегородский государственный университет им. Н.И. Лобачевского
Институт информационных технологий, математики и механики

Учебный курс:

Методы программирования - 2

Практическая работа 3:

Структуры хранения стека

Гергель В.П., профессор ,
директор института ИТММ

Содержание

1. Разработка спецификаций

2. Начальный вариант реализации

3. Расширенный вариант реализации

- Контроль ошибок
- Схема наследования
- Базовый класс управления памятью TDataRoot
- Класс реализации стека TStack

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



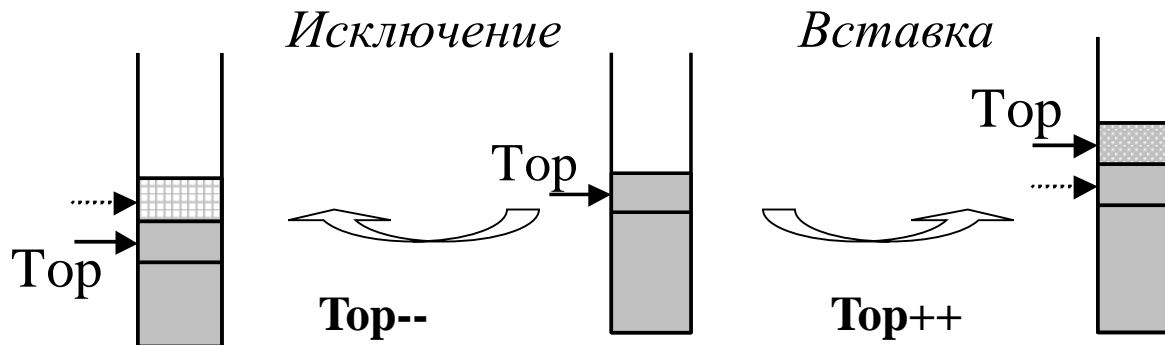
1. Разработка спецификаций

Тесты:

- общий (вставка значений определенной последовательности значений и проверка при выборке),
- программы проверки выполнения аксиом (регулярное или периодическое выполнение),
- проверка исключительных ситуаций (пустота, переполнение),
- визуализация структуры или содержимого стека,
- печать значений стека,
- копирование значений стека (в массив или файл),
- проверка корректности структуры хранения.

2.1. Начальный вариант реализации – структура хранения

- ☑ Для хранения элементов базисного множества выделяется вектор памяти размера, достаточного для хранения максимально-возможного количества (!?) значения в стеке
- ☑ Хранение значений в памяти осуществляется последовательно от младшего элемента вектора к старшему
- ☑ Для запоминания количества хранимых в стеке значений используется индекс последнего занятого элемента в векторе



2.2. Начальный вариант реализации – операции

- **IsEmpty** – проверка пустоты
- **IsFull** – проверка переполнения
- **Put** – добавить значение
- **Get** – извлечь значение

Программа:



3.1. Расширенный вариант реализации – контроль ошибок

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:



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 старшим; индекс последнего занятого элемента в векторе памяти фиксируется в переменной N_i

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

Заключение

- Разработка спецификаций
- Выбор структуры хранения стека
- Программирование с защитой от ошибок
- Определение схемы наследования

Вопросы для обсуждения

- Управление памятью в структуре хранения стека (выделение и передача памяти)
- Реализация отношения следования в виде виртуального метода

Темы занятий для самостоятельной работы

- Сравнение способов организации динамического распределения памяти (перепакровка, списки)
- Проблема статического определения максимального размера памяти

Следующая тема

- Структура хранения очереди

Контакты

Нижегородский государственный университет им.
Н.И. Лобачевского (www.unn.ru)

Институт информационных технологий, математики
и механики (www.itmm.unn.ru)

603950, Нижний Новгород, пр. Гагарина, 23,
р.т.: (831) 462-33-56,

Гергель Виктор Павлович

(<http://www.software.unn.ru/?dir=17>)

E-mail: gergel@unn.ru

