

Лекция №16 по дисциплине «ТЕОРИЯ АЛГОРИТМОВ»

CTEK

Преподаватель: Золотоверх Д.О.

СТРУКТУРА ДАННЫХ

Способ организации информации: ее формат хранения, способы изменения и доступ.

Такой информацией может быть совокупность и однотипных, и многотипных данных.

Данные могут быть связаны.

Управление данными предоставлено определенным способом.



СТРУКТУРА ДАННЫХ

- имеет внутреннюю форму, данные могут быть связаны;
- может иметь несколько разных типов данных;
- хранит информацию об предмете;
- значение нельзя изменить прямым способом, только с помощью специальной операции;
- нужно учитывать проблему сложности вычислений.



ПРИМЕРЫ СТРУКТУР ДАННЫХ

Примеры структур данных:

- Стек (определенный порядок доступа и модификации);
- Очередь (схож в реализации со Стеком);
- Связанный список (массив, но элементы связаны опр. способом);
- <mark>Множество</mark> (может хранить только уникальные элементы);
- Хеш-таблица (пара ключ-значение);
- Дерево (древовидная структура связанных элементов);
- Граф (связанное множество).



АБСТРАКЦИЯ

Структуры данных являются более абстрактными сущностями, чем массивы и типы данных.

Они определяются, прежде всего, своим интерфейсом: набором разрешенных операций, которые могут выполняться с ними.

Интерфейс этих структур проектируется с расчетом на поддержку ограничений доступа.

Базовый механизм, используемый для их реализации, обычно остается невидимым для пользователя.



CTEK

Структура данных, представляющая из себя упорядоченный набор элементов.

Добавление новых элементов и удаление существующих производится с одного конца — вершина стека.

В отличие от списков, мы не можем получить доступ к произвольному элементу стека.



УСТРОЙСТВО

Принцип LIFO (last-in, first-out) — первым из стека удаляется элемент, который был помещен туда последним.

Структура реализована таким образом, что доступ к элементам, что находятся не на вершине либо скрыт, либо вообще не возможен.

Стек может быть реализован на основе массива или связанного списка, но на его работу это не влияет (абстракция).



СИНТАКСИС

Объявление стека состоит из трех частей:

- указание типа шаблона;
- указание типа данных;
- название переменной.

Стек можно импортировать с помощью директивы #include <stack>

Синтаксис:

```
стек <тип_данных> названиеСтека;
названиеСтека.метод();
```

```
int main() {
    stack <int> testStack;
    testStack.push(42);
    testStack.push(1337);
    printf(
        "Element on top: %d\n",
        testStack.top());
    testStack.pop();
    printf(
        "Element on top: %d\n"
         testStack.top());
    testStack.pop();
```

МЕТОДЫ РАБОТЫ

Стек имеет исчерпывающий список методов:

Название метода	Описание метода	Аргументы	Возврат
empty()	Проверка стека на пустоту	Ничего	Является ли стек пустым
pop()	Убирает последний элемент стека	Ничего	Ничего
push()	Добавляет элемент в стек	Новый элемент	Ничего
size()	Смотрит размер стека	Ничего	Размер стека
top()	Смотрит последний элемент стека	Ничего	Элемент стека

РАБОТА СО СТЕКОМ

Как говорилось ранее, нет возможности иметь доступ к любому элементу, только к верхнему.

Например, чтобы вывести на экран последний элемент стека — необходимо обойти все его элементы.

С помощью метода empty() можно соорудить цикл, поэтапно выводя и извлекая элементы.

```
int main() {
    stack <int> testStack;
    testStack.push(42);
    testStack.push(228);
    testStack.push(1337);
    printf("Start\n");
    while(!testStack.empty()) {
        printf(
            "Elem: %d\n",
             testStack.top());
        testStack.pop();
    printf("Out of elements!\n")
```

СПАСИБО ЗА ВНИМАНИЕ!

