

Практическое занятие №1

Густов Владимир Владимирович
gutstuf@gmail.com

Цитата дня: Веселое время наступает друзья. (с) Егор Летов

Системное программирование – реализация функционирования системы. Пример:

- компиляторы;
- операционные системы;
- драйверы/библиотеки.

Прикладное программирование – реализация конкретных задач. Пример:

- браузер;
- мобильный банкинг;
- Adobe Reader, etc.

Типы

характеризует:

- допустимые значения;
- набор операций;
- примитивные (встроенные), комплексные (структуры/пользовательские)

Структуры

частный случай типов.

- предоставляет набор функций/интерфейс.

Типизация

- **Статическая** – тип «закрепляется» за переменной в момент объявления и не может быть изменён позже;
 - **Динамическая** – тип «закрепляется» за переменной на этапе присвоения значения. В различных участках кода одна и та же переменная может принимать значения разных типов;
-
- **Слабая** – проверка типов производится во время выполнения (runtime).
 - **Сильная** (строгая) – проверка типов производится на этапе компиляции (compile time);

Структуры данных

- массив
- строка
- очередь и стек
- вектор
- список
- таблица (словарь, хеш-таблица)
- множество
- дерево
- граф*

Массив

Хранит **упорядоченный набор** значений только **одного типа**. **Элементы** массива **идентифицируются по индексу**. Одномерный массив можно рассматривать как реализацию вектора.

Свойства:

- обращение к элементу по индексу – $O(1)$;
- вставка/удаление элемента в конец – амортизируемое $O(1)$;
- вставка/удаление элемента в начало – $O(n)$;
- вставка/удаление не в конец/начало требует сдвиг элементов;
- не имеет лишней информации, т.к. просто хранит данные;
- динамическая реализация работает медленнее статической.

STL реализация: `std::array`

Очередь/стек

Одномерные динамически изменяемые упорядоченные наборы элементов.

Очередь: новый элемент всегда **добавляется в конец, выталкивается из начала (FIFO)**.

Стек: новый элемент всегда **добавляется в конец, выталкивается из конца (LIFO)**.

Свойства:

- нет возможности обращаться к элементам по индексу;
- вставка/удаление элемента (всегда только в конец) – $O(1)$;

STL реализация: `std::stack`, `std::queue`, `std::deque`,
`std::priority_queue`

Вектор

Одномерный массив с динамическим расширением памяти.

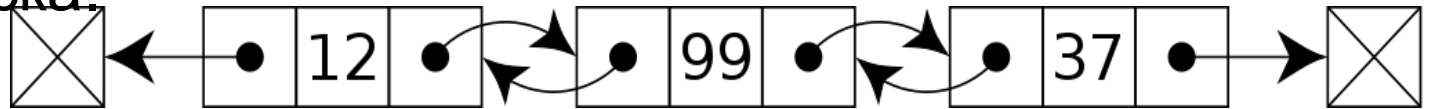
Свойства:

- обращение к элементу по индексу – $O(1)$;
- вставка/удаление в конец – $O(1)$;
- вставка/удаление не в конец – $O(n)$;
- данные расположены в памяти последовательно, за счёт чего они хорошо кешируются;
- разрешает произвольный доступ;

STL реализация: `std::vector`

(Связный) Список

Набор элементов состоящий из двух (трёх для двусвязного) полей. **Одно** поле содержит **элемент данных**, а **другое** – **указатель** на следующий (и предыдущий – для двусвязного) элемент списка.



Свойства:

- индексирование – $O(n)$
- вставка/удаление в начало – $O(1)$;
- вставка/удаление в конец – амортизированное $O(1)$;
- вставка/удаление в произвольное место – $O(1)$
- структурная гибкость – порядок элементов может не совпадать с порядком расположения данных в памяти;

STL реализация: `std::list`, `std::forward_list`

Список пустоты

Таблица (словарь, хэш-таблица)

Ассоциативный массив (ключ + значение), предоставляющий основной интерфейс:

- добавление пары;
- поиск пары по ключу
- удаление пары по ключу

Свойства (словарь):

- вставка/удаление элемента — $O(\log n)$;
- поиск элемента — $O(\log n)$

Свойства (хеш-таблица):

- вставка/удаление элемента — амортизированное $O(1)$;
- поиск элемента — амортизированное $O(1)$

STL реализация: `std::map`, `std::unordered_map`

Множество

Набор уникальных значений определённого типа.

Свойства:

- вставка/удаление элемента — $O(\log n)$;
- поиск элемента — $O(\log n)$;
- повторение значений не допустимо (исключение для мультимножества).

STL реализация: `std::set`, `std::multiset`

Git(Hub)

Git - распределённая система контроля версий.

GitHub - сервис-хостинг предоставляющий возможность публикации репозиториев/проектов.

Аналоги:

Bitbucket (Github): права доступа на ветви;

Mercurial (Git): нельзя отменять коммиты;

SVN (Git): централизованная система.

<https://youtu.be/Nz9SiF0QVKY>

<https://cppconf-moscow.ru/> - ежегодная
российская конференция по C++.