# Лекции

## 1. Выбор структур данных

1.1. Линейные структуры

1.1.1. Массив

1.1.1.1. Недостатки простого массива

1.1.2. Динамический массив

1.1.2.1. Добавление к динамическому массиву

1.1.2.2. Удаление из динамического массива

1.1.2.3. Недостатки динамических массивов

1.1.3. Связные списки

1.1.3.1. Добавление к связанному списку

1.1.3.2. Удаление из связанного списка

1.1.3.3. Преимущества связанного списка

1.1.3.4. Недостатки связанного списка

1.1.4. Заключение по линейным структурам

1.2. Линейные структуры данных с конечными точками

1.2.1.1. Стек

1.2.1.2. Очередь

1.2.1.3. Очередь с приоритетом

1.2.1.4. Заключение по структурам данных с конечными точками

1.3. Деревья и кучи

1.3.1.1. Структуры данных «деревья»

1.3.1.2. Простое дерево

1.3.1.3. Куча

1.3.1.4. Почему их нет в стандартных библиотеках

1.3.1.5. Заключение по деревьям и кучам

1.4. Нелинейные структуры данных

1.4.1. Словарь данных

1.4.2. Упорядоченное и неупорядоченное множество

1.4.3. Заключение по нелинейным структурам данных

1.5. Правильный выбор структур данных.

1.5.1. Эффект правильного выбора

1.5.2. Выбор из паттернов

1.5.2.1. Динамический массив — выбор по умолчанию

1.5.2.2. Стек — только один конец

1.5.2.3. Семейство очередей — первый вошёл, первый вышел.

1.5.2.4. Нелинейные структуры — быстрый поиск.

1.5.2.5. Связанный список — частые изменения с сохранением порядка

## 2. C++. Массивы строк типа string???

1.1. Создание массива строк типа string. Статический и динамический массив строк

1.2. Инициализация массива строк типа string. Пример

1.3. Пример создания динамического массива строк заданного размера

1.4. Пример ввода строк с клавиатуры и формирование массива этих строк

1.5. Пример сортировки массива строк методом вставки

1.6. Пример поиска заданной строки в массиве строк

1.7. Пример определения количества строк в массиве строк в соответствии с заданным условием

## 3. Перечисление. Объединение. STL контейнеры. Массив. Дек. Одно/двух связный список. Вектор.

1.1. Использование enum

1.2. Объединения union

1.2.1. Обычные и анонимные объединения.

1.3. Тип данных - массив

1.4. Динамические массивы

1.5. Гетерогенные массивы

1.6. Ассоциативный массив

1.7. std::array

1.7.1. Разница между std :: array и std :: vector

1.8. std::deque

1.8.1. Создание deque

1.8.2. Получение элементов очереди

1.8.3. Размер очереди

1.8.4. Изменение элементов очереди

1.8.5. Добавление элементов

1.8.6. Удаление элементов

1.9. std::forward\_list

1.9.1. Создание односвязного списка

1.9.2. Получение элементов

1.9.3. Размер списка

1.9.4. Изменение элементов списка

1.9.5. Добавление элементов

1.9.6. Удаление элементов

1.9.7. Разница между std::forward\_list и std::list

1.10. std::list

1.10.1. Создание списка

1.10.2. Получение элементов

1.10.3. Размер списка

1.10.4. Изменение элементов списка

1.10.5. Добавление элементов

1.10.6. Удаление элементов

1.11. map

1.11.1. Понятие map

1.11.2. Создание map

1.11.3. Итераторы для map

1.11.4. Вывод контейнера

1.11.5. Методы map

1.11.6. Плюсы и минусы: использования map

1.12. Вектор (vector)

1.12.1. Создание вектора (vector) в C++

1.12.2. Обращение к ячейке/элементу вектора

1.12.3. Количество ячеек для вектора

1.12.4. Сравнение двух векторов

1.12.5. Вектор векторов

1.12.6. Методы для векторов:

1.12.6.1. size() и empty()

1.12.6.2. push\_back() и pop\_back()

1.12.6.3. insert()

1.12.6.4. front() и back()

## 4. База данных на файлах

1.1. Схема базы данных

1.2. Меню манипулирования данными в базе данных на файлах

1.3. Манипуляции с данными файла

1.3.1. Контроль данных в файле

1.3.2. Функция CountLinesInFile(). Подсчет количества строк в текстовом файле

1.3.3. Функция GetStringsFromFileC(). Получить массив (список) строк типа char\* из текстового файла

1.3.4. Функция GetStringsFromFileS(). Получить массив строк типа string из текстового файла

1.3.5. Функция SetStringsToFileS(). Записать массив (список) строк типа string в текстовый файл

1.3.5.1. Пример 1. Запись данных в файл базы данных

1.3.6. Функция ChangeStringInFileC(). Замена строки в текстовом файле

1.3.7. Функция RemoveStringFromFileByIndex(). Удаление строки из файла по его номеру

1.3.8. Функция InsertStringToFile(). Вставка строки в заданную позицию в файле

1.3.9. Функция SwapStringsInFile(). Обмен местами двух строк в файле

1.3.10. Функция ReverseStringsInFile(). Реверсирования строк файла (перестановка строк файла в обратном порядке)

1.3.11. Функция SortStringsInFile(). Сортировка строк в файле

1.3.11.1. Пример 1. Сортировка. Латиница

1.3.11.2. Пример 2. Сортировка. Латиница+Кирилица

1.3.11.3. Пример 3. Сортировка по выбранному полю (SortBySelectedField)

1.3.11.4. Пример 4. MatricaFileSort

1.4. Просмотр файла

1.5. Создание индексов

1.6. Поиск записи по индексу

1.7. Поиск записи по значению

1.8. Сортировка в файле

1.9. Сортировка по индексу

1.10. ДопПримеры

1.10.1. FilterSort.cpp

1.10.2. MergeSort.cpp

1.10.3. ReadPrintSortVector

1.10.4. ReadStringFromFile

1.10.5. ArbitraryPositioningInFile

1.10.6. CreateReadChange

1.10.7. ReadFromFile

## 5. Манипуляции со структурами данных

1.1. Сортировка

1.1.1. Массив

1.1.2. Однонаправленный связанный список

1.1.3. Двунаправленный связанный список

1.1.4. Стек

1.1.5. Очередь

1.1.6. Дерево

1.1.7. Куча

1.2. Поиск

1.2.1. Массив

1.2.2. Однонаправленный связанный список

1.2.3. Двунаправленный связанный список

1.2.4. Стек

1.2.5. Очередь

1.2.6. Дерево

1.2.7. Куча

1.3. Фильтрация

1.3.1. Массив

1.3.2. Однонаправленный связанный список

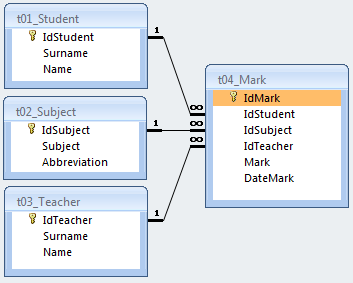
1.3.3. Двунаправленный связанный список

1.3.4. Стек

1.3.5. Очередь

1.3.6. Дерево

1.3.7. Куча



|  |  |  |
| --- | --- | --- |
|  | **ЛК\_Тема** | **Чс** |
| 1 | Выбор структур данных | 4 |
| 2 | Массивы строк типа string | 2 |
| 3 | Перечисление. Объединение. STL контейнеры | 10 |
| 4 | Использование контейнеров | 4 |
| 5 | База данных на файлах | 12 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | **32** | **32** |

# Лабараторные

|  |  |  |
| --- | --- | --- |
|  | **ЛР\_Тема** | **Чс** |
| 1 | Перечисление. Объединение. STL контейнеры. Массив. Дек. Однодвух связный список. Вектор | 4 |
| 2 | Схема БД. Меню программы | 4 |
| 3 | Двоичные и текстовые файлы | 2 |
| 4 | Индексировние записей. Простой/сложный индекс | 4 |
| 5 | Запись/чтение массива структур в файл | 4 |
| 6 | Редактирование файлов: удаление, изменение поля | 4 |
| 7 | Сортировка записей | 4 |
| 8 | Фильтрация данных | 2 |
| 9 | Поиск записи по значению/индексу | 4 |
| 10 |  |  |
| 11 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | **32** | **32** |

# Шаблон вопросов/ответов теста

**<p><b>Какие категории ошибок различает Access во время выполнения запроса на изменение?</b></p>**

1. <p><b>Дубликаты первичного ключа</b></p>

2. <p><b>Ошибки преобразования данных</b></p>

3. <p><b>Заблокированные записи</b></p>

4. <p><b>Нарушение условий на значение</b></p>

Ответ: 1, 2, 3, 4

Комментарий: DB005003

**<p><b>Какая ошибка возникает при попытке добавления или изменения некоторой записи в таблице, приводящей к появлению дублирующего значения первичного ключа или уникального индекса?</b></p>**

1. <p><b>Ошибки преобразования данных</b></p>

2. <p><b>Заблокированные записи</b></p>

3. <p><b>Дубликаты первичного ключа</b></p>

Ответ: 3

Комментарий: DB005004

Комментарий должен содержать

**SD00X0Y**

SD - название дисциплины (структуры данных)

00X - номер темы

0Y - номер вопроса в теме