**Введение**

**/\***

**Плавно привести к теме курсовой. Пучем была выбрана именно она**

**\*/**

Язык программирования C (рус. *Си*) — компилируемый статически типизированный язык программирования общего назначения, разработанный в 1969—1973 годах как развитие языка Би, когда Кен Томпсон и Дэннис Ритчи из Bell Labs разрабатывали операционную систему UNDC. Первоначально был разработан для реализации операционной системы UNIX, но, впоследствии, был перенесён на множество других платформ. Согласно дизайну языка Си, его конструкции близко сопоставляются типичным машинным инструкциям, благодаря чему он нашёл применение в проектах, для которых был свойственен язык ассемблера, в том числе как в операционных системах, так и в различном прикладном ПО для множества устройств — от суперкомпьютеров до встраиваемых систем. Язык программирования Си оказал существенное влияние на развитие индустрии программного обеспечения, а его синтаксис стал основой для таких языков программирования, как C++, C#, Java и Objective-C.

Данная работа посвящена изучению языку С/С++ и созданию баз данных. База данных является неотъемлемой частью каждой программы. Будь это сложная программа банковского учета или простая программа, содержащая имена студентов, которые учатся в университете.

Цель курсовой работы:

* приобрести навыки формальной математической постановки задач;
* алгоритмизация задач и программирование, отладка и выполнение на ЭВМ конкретных задач с использованием современных методов программирования;
* научиться характеризовать исходные данные;
* уметь анализировать решение задач;
* овладеть теоретическими знаниями основ программирования и алгоритмизации.

В течении курса по основам алгоритмизации и программированию были изучены следующие темы: структуры, объединения, указатели и динамическая память, объявление указателей; процедуры и функции для работы с динамической памятью; файлы: понятие файла, виды файлов, функции для работы с файлами; алгоритмы сортировки и поиска; списки с разными видами хранения.

По завершению двух семестров студенту предоставляется возможность реализовать накопленные знания в своей курсовой работе. Курсовая работа должна содержать базу данных на произвольную тему. База данных содержит 10 полей. При создании программы требуется создать максимально дружественный интерфейс программы, с которым будет приятно работать пользователю. При выполнении курсовой работы развиваются навыки создания и откладки программы.

1. **Анализ задания и постановка задач**

Цель курсовой работы является создание программы, которая содержала бы базу данных популярных репозиториев сервиса GitHub. Основная функция программы – работа с одно файловой базой данных. Для описания каждого репозитория используются следующие поля (примеры содержания некоторых полей приведены в скобках):

/\*

Добавить описание конкретного поля

\*/

1. Название репозитория
2. Язык программирования в репозиторие
3. Автор репозитория
4. Рейтинг
5. Дата создания
6. Дата последнего изменения
7. Количество коммитов
8. Количество найденных ошибок
9. Количество исправленных ошибок
10. Число веток репозитория

Программа должна иметь дружественный интерфейс и проводить проверку на правильность ввода данных. Программа должна осуществлять:

* Запись и загрузку файла базы данных (текстовый файл).
* Добавление новых записей, удаление старых.
* Сортировать записи по любому из полей базы данных.
* Перемещать запись из старой позиции в новую.
* Отображать весь список записей.
* Отображать информацию по одной записи.

1. **Теоретические сведения**

**/\***

**Описать други типы списков, привести + -**

**Описать стек своими словами**

**\*/**

**Стек** (англ. ***stack*** — стопка; читается стэк) — абстрактный тип данных, представляющий собой список элементов, организованных по принципу LIFO (англ. last in — first out, «последним пришёл — первым вышел»).

Чаще всего принцип работы стека сравнивают со стопкой тарелок: чтобы взять вторую сверху, нужно снять верхнюю. В цифровом вычислительном комплексе стек называется магазином — по аналогии с магазином в огнестрельном оружии (стрельба начнётся с патрона, заряженного последним).

В 1946 Алан Тьюринг ввёл понятие стека. А в 1957 году немцы Клаус Самельсон и Фридрих Л. Бауэр запатентовали идею Тьюринга.

Зачастую стек реализуется в виде однонаправленного списка (каждый элемент в списке содержит помимо хранимой информации в стеке указатель на следующий элемент стека).

Но также часто стек располагается в одномерном массиве с упорядоченными адресами. Такая организация стека удобна, если элемент информации занимает в памяти фиксированное количество слов, например, 1 слово. При этом отпадает необходимость хранения в элементе стека явного указателя на следующий элемент стека, что экономит память. При этом указатель стека (*Stack Pointer*, — SP) обычно является регистром процессора и указывает на адрес головы стека.

Предположим для примера, что голова стека расположена по меньшему адресу, следующие элементы располагаются по нарастающим адресам. При каждом вталкивании слова в стек, SP сначала уменьшается на 1 и затем по адресу из SP производится запись в память. При каждом извлечении слова из стека (выталкивании) сначала производится чтение по текущему адресу из SP и последующее увеличение содержимого SP на 1.

При организации стека в виде однонаправленного списка значением переменной стека является указатель на его вершину — адрес вершины. Если стек пуст, то значение указателя равно NULL.

**Указатель –** этопеременная, значением которой является адрес, по которому располагаются данные.

**Типизированный указатель** – указатель, содержащий адрес данных определенного типа (системного или пользовательского).

**Не типизированный указатель** – указатель, содержащий адрес данных неопределенного типа (просто адрес).

Классификация по области доступа определяется методом адресации принятой для семейства процессоров x86: адрес состоит из двух элементов: сегмент и смещение.

**Адрес** – это номер ячейки памяти, в которой или с которой располагаются данные.

Классифицировать указатели можно:

* по типу данных (типизированные и не типизированные указатели);
* по области доступа (ближние и дальние указатели).

**Динамическая память** – это область (блок) памяти выделенный для нужд программы в процессе работы программы (а не заранее).

Основными двумя действиями над динамической памятью являются: выделение и освобождение. В языке С функции для осуществления этих действий описаны в библиотеке **stdlib.h**.

**Структура** – это сложный тип данных представляющий собой упорядоченное в памяти множество элементов различного типа. Каждый элемент в структуре имеет свое имя и называется полем. Размер структуры определяется суммой размеров всех элементов.

**Файл** – именованная область данных на каком-либо носителе информации (жесткий диск, дискета, компакт-диск и т.д. и т.п.).

Язык С, как и другие языки программирования высокого уровня, позволяет осуществлять операции файлового ввода и вывода. Основной алгоритм обработки файлов выполняется в три действия в следующем порядке: открытие файла; чтение и/или запись данных в файл; закрытие файла.

В языке С все файлы делятся на два вида:

* бинарные;
* текстовые.

**Бинарный файл** – файл, содержащий структурированную или не структурированную информацию, представленную в двоичном (бинарном) виде.

**Текстовый файл** – файл, содержащий структурированную или не структурированную информацию, представленную в текстовом (ASCII символы) виде.

Операции чтения и записи для текстовых и бинарных файлов отличаются друг от друга (в языке С реализованы в виде различных функций). В языке С для работы с файлами набор функций реализован в библиотеке stdio.h.

**Функция** – это синтаксически выделенный именованный программный модуль, выполняющий определенное действие или группу действий. Каждая функция имеет свой интерфейс и реализацию.

**Реализация функции** – тело функции, содержащее внутренние (локальные) данные функции и программный код, выполняющий действия согласно переданным в функцию параметрам и возвращающий значение, соответствующего интерфейсу функции типа.