



Программа курса «Программирование на языке C»

Програмування на мові C
C Programming Language

Для групп стационара. Версия 3.0.1

Объем курса: 92 пары.

Цель курса

Обучить слушателя программированию на языке C. Научить студента мыслить алгоритмически. Научить использовать блок-схемы, условия, циклы, массивы, функции, указатели и другие базовые конструкции языка программирования C.

По окончании курса слушатель будет:

- понимать, что такое алгоритм;
- уметь разрабатывать алгоритмы;
- уметь строить блок-схемы;
- использовать циклы, условия и другие базовые конструкции;
- оперировать основами языка программирования C;
- понимать и использовать отладчик;
- создавать функции;
- использовать одномерные и многомерные массивы;
- применять указатели и разбираться в тонкостях арифметики указателей;
- уметь работать с динамически выделенной памятью;
- использовать алгоритмы поиска и сортировки данных;
- оперировать структурами, объединениями, битовыми полями;
- сохранять и загружать данные из файловой системы;
- применять битовые операции для воздействия на отдельные биты переменных.

По окончании данного курса студент сдаёт экзамен (по материалам курса), содержащий теоретическую и практическую часть. Для допуска к экзамену, должны быть сданы все домашние и практические задания.



Тематический план

Модуль 1. Введение в язык программирования С	(5 пар)
Модуль 2. Переменные и типы данных	(6 пар)
Модуль 3. Логические операторы и операторы ветвлений	(8 пар)
Модуль 4. Циклы, использование отладчика	(10 пар)
Модуль 5. Массивы одномерные и многомерные	(10 пар)
Модуль 6. Функции	(10 пар)
Модуль 7. Указатели	(10 пар)
Модуль 8. Строки	(6 пар)
Модуль 9. Многомерные динамические массивы	(7 пар)
Модуль 10. Структуры	(6 пар)
Модуль 11. Препроцессор в приложениях	(4 пары)
Модуль 12. Использование файловой системы	(8 пар)
Модуль 13. Экзамен	(2 пары)



Модуль 1 (5 пар)

Введение в язык программирования C

1. Введение:
 - история и этапы развития языка C;
 - сравнительный анализ языка C с другими языками программирования.
2. Алгоритм:
 - понятие алгоритма;
 - примеры использования алгоритмов в реальной жизни;
 - типы алгоритмов: линейный, разветвлённый, циклический;
3. Понятие блок-схемы.
 - базовые обозначения в блок-схемах;
 - блок начала алгоритма;
 - блок завершения алгоритма;
 - блок ввода данных;
 - блок вывода данных;
 - блок вычислений;
 - простейшие примеры использования блок-схем.
4. Программная среда Microsoft Visual C++;
 - инсталляция;
 - основы работы с IDE VC++;
 - создание проекта;
 - добавление файла к проекту;
 - обзор альтернативных средств разработки.
5. Первая программа:
 - первой программы на языке блок-схем;
 - анализ первой программы.
6. Классификация символов языка.
7. Лексемы.
8. Понятие библиотеки.
9. Анализ понятий: компилятор, линковщик, интерпретатор.
10. Организация вывода данных в консоль.
11. Понятие ESCAPE-последовательности:
 - ESCAPE последовательность \n;
 - ESCAPE последовательность \t;
 - ESCAPE последовательность \b;
 - ESCAPE последовательность \»;
 - ESCAPE последовательность \\;
 - ESCAPE последовательность \a.



12. “Raw” строки:
 - что такое “raw” строки;
 - примеры использования “raw” строк.
13. Комментарии:
 - однострочные;
 - многострочные.

Модуль 2 (6 пар)

Переменные и типы данных

1. Типы данных:
 - понятие типа данных. Размер, диапазон значений;
 - целые типы данных. Типы данных long long, unsigned long long;
 - типы данных для работы с дробными числами;
 - символьный тип данных. Типы char16_t, char32_t;
 - логический тип данных;
 - перечислимый тип данных (enum).
2. Переменная:
 - необходимость использования переменных;
 - идентификаторы;
 - ключевые слова;
 - синтаксис объявления переменных;
 - инициализация переменной. Списковая инициализация переменной.
3. Организация ввода данных с консоли.
4. Константы и литералы.
 - необходимость применения;
 - синтаксис объявления.
5. Операторы:
 - понятие оператора;
 - типы операторов;
 - арифметические операторы;
 - логические операторы;
 - операторы ветвлений;
 - унарные операторы;
 - бинарные операторы;
 - тернарный оператор;
 - оператор присваивания;
 - арифметические операторы;
 - оператор сложения;
 - оператор вычитания;
 - оператор умножения;



- инкремент. Постфиксная и префиксная форма;
 - декремент. Постфиксная и префиксная форма;
 - сокращенные формы.
6. Примеры построения программ с использованием блок-схем.

Модуль 3 (8 пар)

Логические операторы и операторы ветвлений

1. Преобразование типов данных:
 - необходимость использования;
 - неявное преобразование типов;
 - списковая инициализация, сужение и неявное преобразование типов;
 - явное преобразование типов.
2. Логические операторы:
 - знакомство с логическими операциями;
 - таблица результатов применения логических операций;
 - «логическое отрицание». Оператор !;
 - «логическое И». Оператор &&;
 - «логическое ИЛИ». Оператор ||.
3. Таблица приоритетов операторов.
4. Конструкции логического выбора. Операторы ветвлений:
 - оператор ветвления if;
 - оператор ветвления if – else;
 - лестница if - else if;
 - обозначение условий в блок-схемах. Блок условия;
 - обозначение объединения ветвей в блок-схемах;
 - примеры построения программ с использованием операторов ветвлений на языке блок-схем;
 - понятие составного оператора;
 - тернарный оператор;
 - оператор множественного выбора – switch.
5. Понятие enum:
 - понятие enum, как перечислимого типа;
 - синтаксис объявления enum;
 - использование enum для switch-конструкций.



Модуль 4 (10 пар)

Циклы, использование отладчика

1. Циклы:
 - необходимость использования циклов. Примеры использования;
 - цикл while;
 - цикл for;
 - цикл do-while;
 - обозначение циклов в блок-схемах. Блок цикла;
 - операторы break и continue;
 - примеры построения программ с использованием циклов на языке блок-схем;
 - вложенные циклы. Примеры использования.
2. Работа с интегрированным отладчиком в Microsoft Visual C++:
 - что такое отладчик. Цели и задачи отладчика;
 - запуск программы по шагам;
 - окна для работы с отладчиком. Окна переменных, локальных переменных, памяти;
 - исполнение одного шага;
 - установка точки останова (breakpoint);
 - установка умной точки останова (smart breakpoint).

Модуль 5 (10 пар)

Массивы одномерные и многомерные

1. Массивы:
 - что такое массивы. Необходимость их использования;
 - синтаксис объявления одномерного массива;
 - способы инициализации массива;
 - схема размещения массивов в памяти;
 - индексация элементов массива;
 - Примеры использования массивов на языке блок-схем.
2. Алгоритмы суммирования.
3. Алгоритмы поиска (линейный, бинарный).
4. Алгоритмы сортировки:
 - пузырьковая сортировка;
 - сортировка выбором;
 - сортировка вставками.
5. Многомерные массивы:
 - многомерные массивы. цели и задачи их использования;
 - двумерные массивы, как частный случай многомерных;



- синтаксис объявления многомерного массива;
 - примеры использования многомерных массивов.
6. Понятие статического выделения памяти.

Модуль 6 (10 пар)

Функции

1. Функции:
- необходимость использования функций;
 - синтаксис объявления функции;
 - использование ключевого слова void при работе с функциями;
 - вызов функции;
 - аргументы функции;
 - возврат значения из функции (оператор return);
 - хвостовой возвращаемый тип;
 - понятие области видимости. Локальные и глобальные переменные. Классы памяти;
 - передача массива в функцию;
 - прототип функции;
 - аргументы по умолчанию;
 - перегрузка функций;
 - рекурсия;
 - функция, принимающая неограниченное количество элементов.

Модуль 7 (10 пар)

Указатели

1. Указатели
2. Необходимость использования указателей.
3. Адрес переменной. Оператор &.
4. Косвенная адресация или оператор разыменования.
5. Принцип работы оператора присвоения для указателей.
6. Анализ использования NULL и nullptr.
7. Связь массивов и указателей.
8. Операции над указателями:
 - арифметические операции;
 - логические операции.
9. Примеры работы с указателями.
10. Константный указатель и указатель на константу.
11. Понятие стека и динамической памяти.



12. Средства языка для работы с динамической памятью.
 - оператор new;
 - оператор delete.
13. Понятие ссылки &:
 - что такое ссылка;
 - синтаксис объявления ссылки;
 - синтаксис объявления const ссылки;
 - примеры.
14. Понятие &&.
15. Передача аргументов внутрь функции:
 - передача по значению;
 - передача по ссылке;
 - передача по указателю.
16. Ключевые слова auto и decltype. Автоматическое выведение типа.
17. Указатель на функцию:
 - понятие адреса функции;
 - необходимость использования указателя на функцию;
 - синтаксис объявления указателя на функцию;
 - примеры использования указателя на функцию, массива указателей на функции.

Модуль 8 (6 пар)

Строки

1. Понятие строки как массива символов.
2. Знакомство с нультерминированными строками.
3. Варианты инициализации строки при объявлении.
4. Тонкости ввода, вывода строк.
5. Алгоритмы, используемые при работе со строками.
6. Анализ функций из библиотеки string.h.:
 - длина строки;
 - копирование строк;
 - конкатенация строк;
 - поиск символов в строке;
 - поиск подстроки в строке;
 - работа с различным регистром символов в строке;
 - замена символов и подстрок в строке.



Модуль 9 (7 пар)

Многомерные динамические массивы

1. Указатель на указатель:
 - необходимость использования указателя на указатель;
 - синтаксис объявления указателя на указатель;
 - примеры использования;
 - использование указателя на указатель для передачи указателя внутрь функции.
2. Многомерные динамические массивы:
 - необходимость использования многомерного динамического массива;
 - двумерный, как частный случай многомерного массива;
 - схема расположения двумерного динамического массива в оперативной памяти;
 - примеры использования двумерного динамического массива.

Модуль 10 (6 пар)

Структуры

1. Структуры:
 - необходимость использования структур;
 - синтаксис объявления структур;
 - инициализация и доступ к элементам структуры;
 - массивы структур;
 - вложенные структуры;
 - структуры, как аргументы функций;
 - ключевое слово `typedef`;
 - применение `typedef` для структур.
2. Объединения:
 - необходимость использования объединений;
 - синтаксис объявления объединений;
 - примеры использования объединений.
3. Битовые поля:
 - необходимость использования битовых полей;
 - синтаксис объявления битовых полей;
 - примеры использования битовых полей.



Модуль 11 (4 пары)

Препроцессор в приложениях

1. Директивы препроцессора:
 - понятие препроцессора;
 - директивы препроцессора;
 - препроцессорная директива `#include`;
 - препроцессорная директива `#define` для создания констант;
 - макроопределения с параметрами;
 - условная компиляция:
 - понятие условной компиляции;
 - директива `#if`;
 - директива `#if #else`;
 - директива `#ifdef`;
 - директива `#ifndef`;
 - директива `#undef`.

Модуль 12 (8 пар)

Использование файловой системы

1. Стандартная библиотека ввода-вывода в языке C. Функции `scanf`, `printf`.
2. Выделение и очистка памяти в языке C:
 - функция `malloc`;
 - функция `calloc`;
 - функция `realloc`;
 - функция `free`.
3. Перенаправление ввода-вывода.
4. Использование аргументов командной строки.
5. Работа с файлами:
 - понятие файла;
 - понятие дескриптора файла;
 - текстовые и двоичные файлы;
 - текстовый и двоичный режим открытия файлов;
 - открытие файлов;
 - сохранение данных в файл;
 - чтение данных из файла;
 - закрытие файла;
 - понятие буфера при работе с файлами;
 - текущая позиция в файле. Позиционирование по файлу.
6. Битовые операции:
 - системы исчисления двоичная, восьмеричная, шестнадцатеричная;



- цели и задачи битовых операций;
 - битовое «И»;
 - битовое «ИЛИ»;
 - битовое «ИСКЛЮЧАЮЩЕЕ ИЛИ»;
 - битовое отрицание;
 - битовые сдвиги.
7. Поиск файлов:
- поиск файлов по заданной маске и пути;
 - использование битовых операций при поиске файлов.

Модуль 13 (2 пары)

Экзамен