Некоторые замечания, предваряющие план метод пакетов для модулей:

* Основы программирования. Язык С++.
* Объектно-ориентированное программирование. Язык С++.
* Объектно-ориентированное моделирование. Язык UML.
* Паттерны-проектирования. Язык С++.

Голубым цветом будет указываться материал, который предполагает изучение на очных лекциях (то есть при участии преподавателя).

Красным цветом будет указываться материал для самостоятельного прочтения и изучения студентами. В основном это будет дополнительная информация, материал связанный с математикой и информатикой, вспомогательная информация. В принципе красных пунктов может быть гораздо больше — думаю у каждого преподавателя есть что дополнительно рассказать по темам. Это можно оформить в доп. материал.

|  |  |  |
| --- | --- | --- |
|  | СТ | ПС |
| Основы программирования | 69 | 28 |
| ООП | 60 | 6 + 28 |
| ЮМЛ | 18 | 5 |
| Паттерны |  | 5 |

# История Вычислительной Техники

# Введение в алгоритмы и программирование

## Основы алгоритмизации

### Алгоритмы.

### Составные команды: следования, ветвления, цикла

### Свойства и способы описания алгоритмов.

### Переменная. Операция присваивания.

### Вспомогательные алгоритмы. Подпрограммы. Процедуры и функции.

### Тестирование ПО

## Введение в языки программирования

### Языки программирования: определение, классификация.

История языков программирования

### Классификации языков программирования

### Введение в C/C++

## Введение в разработку на C++

### Введение в компилятор, компоновщик и библиотеки

### Установка интегрированной среды разработки (IDE)

### Компиляция вашей первой программы

### Вывод информации на консоль.

Несколько распространенных проблем C++

# Процедурное программирование. Язык С++

## Основы языка С++

### Инструкции (statement) и структура программы

### Комментарии

### Введение в объекты и переменные

### Переменные. Присваивание и инициализация.

### Введение в iostream: cout, cin и endl

Форматированный ввод и вывод на языке C

### Неинициализированные переменные и неопределенное поведение

### Ключевые слова и идентификаторы именования

### Пробелы и базовое форматирование

### Введение в литералы и операторы

### Введение в выражения

### Разработка вашей первой программы

## Основные Типы Данных

### Введение в основные типы данных

### Преобразование между двоичной и десятичной системами счисления

Подробно о различных системах счисления. Возможности и варианты использования.

### Представление числа в памяти

### Размеры объектов и оператор sizeof

### Целые числа со знаком

### Целые числа без знака

Целые числа фиксированной ширины и size\_the

### Числа с плавающей запятой

### Логические значения

### Введение в операторы if

### Символьный тип

### Невозвращаемый (неопределенный) тип void

### Введение в преобразование типов

### Константные переменные и символьные константы

### Неоптимизированные переменные (volatile)

### Литералы

### Системы счисления (десятичная, двоичная, шестнадцатеричная и восьмеричная)

Теория по системам счисления

## Операторы

### Приоритет и ассоциативность операторов

### Арифметические операторы

### Деление по модулю (остаток от целочисленного деления)

### Операторы инкремента/декремента

### Операторы присваивания и модификации

### Оператор запятая

### Операторы сравнения (тонкости сравнения для чисел с плавающей запятой)

### Логические операторы

### Библиотека cmath

Булева алгебра.

Теория множеств

## Область видимости, продолжительность «жизни», связи переменных и объектов.

### Составные операторы (блоки фигурных скобочек)

### Локальные переменные. Локальная область видимости переменных.

### Затенение переменных (скрытие имен)

### Внутренняя связь

### Использование деклараций и использование директив (using)

### Определяемые пользователем пространства имен и оператор разрешения области

## Программный поток, управление потоком и обработка ошибок

### Введение в поток управления

### Операторы ветвления If, if-else, if-else if-else.

### Вложенное ветвление.

### Инструкция switch

### Введение в циклы и цикл с предусловием while

### Цикл с постусловием do while

### Цикл for. Восходящий, нисходящий, переменного шага.

### Команды break, continue, goto

### Вложенные циклы

### Остановки (halts — досрочный выход из программы)

### Введение в отладку и тестирование вашего кода

### Обнаружение и обработка ошибок

### assert и static\_assert

## Подробно о преобразовании типов и значений

### Явное и неявное преобразование типов

### Преобразования с плавающей запятой и интегралом

### Числовые преобразования

### Арифметические преобразования

### Явное преобразование типов (приведение) и static\_cast

### typedef и псевдонимы типов

## Функции. Основы.

### Введение в функции

### Функция, возвращающая значения (функции, возвращающие значения)

### Функции void (функции, не возвращающие значения)

### Введение в параметры или аргументы функции

### Введение в локальную область функций

### Глобальные переменные

### Статические переменные.

### Аргументы по умолчанию

### Прототип функции, сигнатура. Объявление и реализация.

Структура процедурной программы. Использование функций в решении сложных задач.

## Введение в структуры данных.

### Общее понятие о структурах данных. Форма хранения и правила работы с ними.

### Одномерный массив константной длины.

### Введение в генерацию случайных чисел

Алгоритмы генерации случайных чисел

### Базовые принципы работы с одномерными массивами.

### Массивы и циклы.

### Массивы и функции.

### Обработка массивов. Поиск экстремумов (минимального, максимального), отзеркаливание массива, сдвиг влево, вправо.

### Многомерные массивы константной длины.

### Обработка многомерных массивов.

## Введение в процесс отладки программы.

### Средства отладки в Visual Studio.

### Пошаговое исполнение.

### Точки остановки

### Просмотр памяти.

## Битовые манипуляции.

### Преобразование между двоичной и десятичной системами счисления

### Представление чисел в памяти. Продвинутые сведения. Отрицательные числа. Вещественные числа.

### Арифметические операции над двоичными числами.

### Побитовые операции и операторы

### Битовые флаги и битовые манипуляции. std::bitset

### Битовые манипуляции с побитовыми операторами и битовыми масками

## Составные типы: Ссылки и Указатели

### Введение в составные типы данных. Категории значений (значения lvalues и rvalues)

### Ссылки на значения lvalue, ссылки на const

### Введение в указатели

### Нулевые указатели

### Указатели и константы

### Передача аргументов копированием, по указателю, по ссылке.

### Возврат значений по ссылке и возврат по адресу

## Введение в динамическое выделение памяти.

### Массивы и указатели. Индексы и адресная арифметика.

### Понятие приложение (сборка), запуск приложения, процесс.

### Поток, стек вызова функций, куча.

### Динамическое выделение и освобождение памяти в куче. new и delete

### Указатель типа void

Динамическое выделение и освобождение памяти на языке С (alloc/malloc, free)

### Динамическое создание и удаление массива.

### Цикл foreach и автоматическое определение типа auto

### Указатель на указатель.

### Динамические многомерные массивы. Зубчатые массивы.

## Функции. Продвинутые возможности.

### Функции с переменным количеством аргументов.

### Рекурсивные функции.

Рекуррентные соотношения. Числовые ряды.

### Введение в перегрузку функций

### Разрешение перегрузки функций и неоднозначных совпадений

### Шаблоны функций. Создание экземпляра шаблона функции

### Шаблоны функций с несколькими типами

## Структуры данных. Продвинутые возможности.

### Введение в поиск. Линейный поиск — реализация. Бинарный поиск.

### Простейшие алгоритмы сортировки: выбором, пузырьком, вставками.

### Реализация бинарного поиска.

### Продвинутые сортировки: шейкерная, расческой, Шелла.

### Быстрая сортировка.

### Сортировка слиянием.

### Пирамидальная сортировка.

### Введение в алгоритмическую сложность. O(N). Сложность по скорости и по памяти.

Продвинутые сведения об алгоритмической сложности.

## C-style строки.

### Символы в С/С++

### Представление строки в оперативной памяти.

### Константные и динамические строки.

### Обработка строк. Базовые функции языка C/С++й

Широкие символы и функции работы со строками широких символов.

### Библиотеки cctype, cstdlib, cstring

## Структуры данных. Пользовательские (составные) типы.

### Введение в программно-определяемые (определяемые пользователем) типы. Необходимость таких типов.

### Перечисления. Обычные и классы перечислений.

### Введение в структуры, поля структур.

### Инициализация полей структуры

### Передача и возврат структур в функции.

### Указатели на структуры

### Шаблоны структур

## Файлы.

### Файлы как хранилище данных. Текстовые и бинарные файлы.

### Работа с файлами на базовых функциях языка С/С++.

### Текстовые файлы. Создание, запись, чтение.

### Бинарные файлы. Создание, запись, чтение.

Текстовые файлы с широкими символами. Создание, запись, чтение.

### Файлы как сущность файловой системы.

### Базовые функции языка С/С++ для работы с файловой системой.

## Структуры данных. Продвинутые возможности. Связные списки.

### Введение в связные списки на основе структур.

### Однонаправленный список.

### Двунаправленный список.

Кольцевые однонаправленный и двунаправленный списки.

### Организация порядка работы со структурами данных. Стек, Очередь. Дек.

## Функции. Продвинутые возможности.

### Указатель на функцию.

### Функции как аргументы для других функций (функции обратного вызова).

### Аргументы командной строки

### Шаблоны функций и auto

### Явная специализация шаблонной функции

### Анонимные (лямбда) функции. Лямбда-захваты

## Препроцессор и многофайловый проект С++

### Процесс сборки проекта.

### Директивы препроцессора.

### Заголовочный файл и файл реализации.

### Особенности работы с заголовочными файлами для шаблонных сущностей.

### Подробнее о пространствах имен. Неназванные и встроенные пространства имен

### Тонкие настройки компиляции в Visual Studio

# Введение в объектно-ориентированное программирование.

### Основы объектно-ориентированного программирования

### Классы и элементы класса: поля и методы. Состояние и поведение.

### Спецификаторы публичного и частного доступа

### Инкапсуляция. Методы аксессоры (геттер и сеттер)

### Конструкторы. Конструкторы по умолчанию и параметрические.

### Инициализация элементов класса.

### Инициализация нестатического элемента

### Перекрывающиеся и делегирующие конструкторы

### Деструкторы.

### Классы, объекты и скрытый указатель this

### Анонимные объекты. Методы, возвращающие объект класса. Цепочки методов.

### Классы, константные поля и константные методы.

### Статические поля класса.

### Статические методы класса.

### Дружественные функции и классы.

### Вложенные пользовательские типы в классах

### Классы, файлы заголовков и файлы реализации.

### Тонкий момент заголовочных файлов для шаблонных классов.

## Отладка программного кода.

### Внутренний отладчик Visual Studio

### Просмотр памяти, стека вызова функций, значений и адресов переменных, значений контрольных значений.

### Измерение времени выполнения кода.

### Дополнительные возможности. Потоки, регистры и пр.

## ООП. Перегрузка операторов

### Введение в перегрузку оператора.

### Три варианта перегрузки операторов. Технологии перегрузки для бинарных у унарных операторов.

### Перегрузка арифметических операторов с использованием дружественных функций

### Перегрузка операторов с использованием сторонних функций

### Перегрузка операторов ввода-вывода

### Перегрузка операторов с использованием функций-членов

### Перегрузка унарных операторов

### Перегрузка операторов сравнения

### Перегрузка операторов инкремента и декремента

### Перегрузка оператора индекса. Индексаторы.

### Перегрузка оператора круглых скобок. Функторы.

### Функторы возвращающие логическое значение. Предикаты.

### Функторы — алгоритмы.

### Перегрузка приведения типов

### Конструктор копирования. Константные аргументы по ссылке.

### Инициализация поверхностным копированием.

### Конструкторы преобразования. explicit, implecit

### Перегрузка оператора присваивания

### Поверхностное и глубокое копирование

### Использование классов с перегруженными операторами. Примеры.

## Введение в связи между классами и объектами.

### Объектные отношения («является частью», «содержит», «зависит», «использует» и пр)

### Композиция объектов. Сильная связь части и целого.

### Агрегация объектов. Слабая связь части и целого.

### Ассоциация. Слабая связь двух частей.

### Зависимости. Поверхностная связь двух частей.

## Введение в контейнеры.

### Классы контейнеров

### std::initializer\_list

### Собственные классы динамического массива, строки, связного списка и бинарного дерева.

## Наследование

### Введение в наследование

### Базовое наследование в C++. Самая сильная связь между классами и объектами.

### Порядок построения производных классов

### Конструкторы и инициализация производных классов

### Спецификаторы наследования и доступа. Защищенные элементы класса.

### Добавление новой функциональности в производный класс

### Вызов унаследованных функций и переопределение поведения

### Скрытие унаследованной функциональности

### Множественное наследование

## Полиморфизм. Виртуальные методы

### Указатели и ссылки на базовый класс производных объектов

### Виртуальные функции и полиморфизм

### Спецификаторы override и final, а также ковариантность возвращаемых типов

### Виртуальные деструкторы, виртуальное назначение и переопределение виртуализации

### Раннее и позднее связывание

### Таблица виртуальных методов

### Чисто виртуальные функции, абстрактные базовые классы и интерфейсные классы

### Виртуальные базовые классы

### Обрезка объектов при присваивании, передачи в функцию.

### Динамическое приведение типов. dynamic\_cast

### Проблема наследственной перегрузки оператора вывода <<. Решение.

## ООП. Шаблоные классы

### Шаблоны в применении к классам

### Параметры non-type.

### Явная специализация шаблона класса

### Частичная специализация шаблонов

### Частичная специализация шаблона для указателей

## Семантики копирования и перемещения. Умные указатели

### Введение в умные указатели и семантику перемещения

### Ссылки на l-value. Ссылки на r-value.

### Конструкторы перемещения и принцип перемещения

### std::move и std::move\_if\_noexcept

### std::unique\_ptr, std::shared\_ptr, std::weak\_ptr

### Создание итераторов для классов собственных контейнеров.

## Исключения.

### Смысл и необходимость исключений

### Базовая обработка исключений

### Исключения, функции и разматывание стека

### Неперехваченые исключения и обработчики catch\_all

### Исключения stl, классы и наследование

### Выбрасывание исключений throw

### try блоки

### Опасности и недостатки исключений

### Спецификации исключений и noexcept

## STL. Строковой тип.

### string, string\_view

## STL. Файловый ввод и вывод

## STL. Базовые функции, структуры и классы

### Трехстороннее сравнение

### Утилиты общего назначения (обмен, присваивание, вывод)

### Пары и кортежи.

## STL. Контейнеры.

### Последовательные контейнеры

### array, vector, deque, forward\_list, list

### Ассоциативные контейнеры

### set, map, multiset, multimap

### Несортированные ассоциативные контейнеры

### unordered\_set, unordered\_map, unordered\_multiset, unordered\_multimap

### Контейнерные адаптеры

### stack, queue, priority\_queue

### Концепция и реализация итераторов в контейнерах.

## STL. Библиотека алгоритмов.

### Немодифицирующие алгоритмы.

### for\_each, find, count и пр.

### Модифицирующие алгоритмы.

### copy, fill, generate, reverse и пр.

### Алгоритмы разделений

### partition и пр.

### Сортирующие алгоритмы и бинарного поиска

### Прочие алгоритмы.

## STL. Библиотеки

### Библиотека форматирования.

### Библиотека работы с датой и временем

### Библиотека диапазонов

### Библиотека файловой системы

### Библиотека регулярных выражений.

Регулярные выражения. Подробно.

## STL. Параллельное программирование

### Потоки. Многопоточность.

### Синхронизация между потоками.

### Асинхронные задачи и взаимодействие между ними.

## C++. Сборки, библиотеки, модули

### Понятие сборки. Структура исполняемого файла.

### Статические и динамические библиотеки.

### Модули (C++20)

## WinAPI и C++

Типы данных в Win32 API

Создание оконной процедуры в Win32 API

Ввод и вывод данных Win32 API