**Глоссарий: Обзорная лекция**

**Программирование** — это процесс создания компьютерных программ с использованием языка программирования. Оно включает в себя разработку алгоритмов, написание и тестирование программного кода, а также отладку и оптимизацию программ. Программирование позволяет компьютеру выполнять различные задачи, автоматизировать процессы и обрабатывать данные.

**C++** — это высокоуровневый язык программирования, разработанный как расширение языка программирования C. Он является мощным и универсальным языком, который широко применяется для разработки различных типов программ, включая системное программирование, создание приложений, игр, веб-сервисов и многое другое.

**Visual Studio** - это интегрированная среда разработки (Integrated Development Environment, IDE), разработанная компанией Microsoft. Она предоставляет разработчикам широкий набор инструментов и функций для создания, отладки и развертывания различных типов приложений.

Проект - это организованная совокупность файлов и ресурсов, которые вместе составляют программу или приложение. Проект включает в себя исходный код программы, зависимости, настройки компиляции, конфигурацию и другие файлы, необходимые для разработки и сборки программы.

**Консоль** — это текстовый интерфейс, через который пользователь или программист может взаимодействовать с операционной системой или программами, используя команды или текстовый ввод и вывод. Консоль обеспечивает возможность выполнения команд, отображения текстовых сообщений и приема текстового ввода от пользователя.

**Приложение** — это программное обеспечение, разработанное для решения определенных задач или выполнения определенных функций на компьютере или другом устройстве. Приложения могут быть различными: от текстовых редакторов и интернет-браузеров до компьютерных игр и мобильных приложений.

**Переменная в программировании** - это именованное место в памяти компьютера, которое используется для хранения и обработки данных.

**Объявление переменной** - это процесс создания переменной в программе путем указания ее имени и типа данных. Объявление переменной сообщает компилятору о существовании переменной определенного типа данных, но не выделяет фактическую память для хранения значения переменной.

**Присваивание значений переменным** - это процесс присвоения конкретного значения переменной. При присваивании значения переменной, вы устанавливаете ее начальное значение или изменяете текущее значение на новое.

**Типы данных в C++** определяют, какие виды значений могут быть сохранены и обрабатываться в программе. В C++ существуют различные типы данных, которые предоставляются языком и могут быть использованы для разных целей.

1. Целочисленные типы данных:

- `int`: Целочисленный тип данных `int` используется для хранения целых чисел. Он обычно занимает 4 байта и может представлять значения в диапазоне от -2,147,483,648 до 2,147,483,647.

- `short`: Целочисленный тип данных `short` используется для хранения целых чисел с меньшим диапазоном значений, чем `int`. Он обычно занимает 2 байта.

- `long`: Целочисленный тип данных `long` используется для хранения целых чисел с большим диапазоном значений, чем `int`. Он обычно занимает 4 или 8 байт, в зависимости от платформы.

- `long long`: Целочисленный тип данных `long long` используется для хранения очень больших целых чисел. Он обычно занимает 8 байт.

2. Вещественные типы данных:

- `float`: Вещественный тип данных `float` используется для хранения чисел с плавающей точкой одинарной точности. Он обычно занимает 4 байта.

- `double`: Вещественный тип данных `double` используется для хранения чисел с плавающей точкой двойной точности. Он обычно занимает 8 байт.

- `long double`: Вещественный тип данных `long double` используется для хранения чисел с плавающей точкой повышенной точности. Размер этого типа данных зависит от платформы и компилятора.

3. Символьный тип данных:

- `char`: Символьный тип данных `char` используется для хранения отдельных символов. Он занимает 1 байт и может представлять символы из набора ASCII или широкий спектр символов Unicode.

4. Логический тип данных:

- `bool`: Логический тип данных `bool` используется для хранения логических значений `true` (истина) или `false` (ложь). Он обычно занимает 1 байт.

**Булевый тип данных** - это основной тип данных в компьютерном программировании, который может принимать два возможных значения, обычно обозначаемые как "истина" и "ложь". В разных языках программирования эти значения могут обозначаться по-разному, но сама суть остается неизменной. Булев тип используется для управления условными и циклическими конструкциями.

**Логические выражения в программировании** - это выражения, которые в результате своего выполнения возвращают значение булевого типа (истина или ложь). Они обычно используются для проверки некоторых условий. Например, "x > y", "name == 'Alex'", "a != b" - все это примеры логических выражений.

**Условные операторы** - это конструкции в программировании, которые позволяют выполнить различные действия в зависимости от выполнения определенного условия (логического выражения). Наиболее известные условные операторы - это "if", "else", "elif" (или "else if").

**Логические операторы** - это операторы, которые применяются к булевым значениям или выражениям. Они включают операторы "И" (AND), "ИЛИ" (OR) и "НЕ" (NOT). Они используются для создания сложных условий.

**Условные выражения с тернарным оператором** - это способ записи простых условных выражений более компактно. В разных языках программирования формат может отличаться, но общая структура выглядит примерно так: "условие ? выражение\_если\_истина : выражение\_если\_ложь".

**Вложенные условия** - это условия, размещенные внутри других условий. Они позволяют создавать более сложные логические конструкции, обрабатывая различные сценарии в зависимости от нескольких условий.

**Цикл** - конструкция потока управления, позволяющая выполнять фрагмент кода до тех пор, пока не будет выполнено некоторое условие.

**Оператор в программировании** - это команда, обозначающая определенное математическое или логическое действие, выполняемое с данными (операндами).

**Тело цикла** - это набор действий (программных операций), которые подразумеваются к повторению раз за разом при работе цикла. Если проще, то это то, что находится между фигурными скобками относительно цикла.

**Условие цикла** - выражение, определяющее, будет в очередной раз выполняться итерация или цикл завершится.

**Итерация цикла** - одно выполнение тела цикла называется одной итерацией.

**Бесконечный цикл** - цикл, в котором условие всегда равно истине (true).

**Вложенный цикл** - это цикл, который находится внутри другого цикла. Цикл внутри цикла называется внутренним (вложенным), другой внешним.

**Строки** - это основной способ работы с текстом в C++, а std::string упрощает работу с ними.

**std::string** — класс стандартной библиотеки языка C++, реализующий понятие "строка". Описание этого класса содержится в заголовочном файле iostream, внутри которого заголовочный файл string. Но при использовании строк желательно подключать явно #include <string>.

**std::cout** - это поток для вывода информации (cOUT = вывод).

**std::cin** - это поток для ввода информации (cIN = ввод).

**std::getline()** - функция, которая предназначена для ввода данных из потока, например, для ввода данных из консольного окна.

**std::ws** - манипулятор, который указывает потоку на то, что перед извлечением следует игнорировать все пробельные символы.

**Конкатенация строк** - процесс сложения строк.

**Массив** — это совокупность однотипных элементов, которые расположены в определённом порядке и обладают одним именем.

**Элемент массива** — это переменная, имя которой совпадает с именем массива и у которой есть индекс. Чтобы вызвать элемент массива, нужно написать имя массива и указать в квадратных скобках индекс этого элемента.

**Индекс — э**то номер элемента в массиве.

**Инициализация массива** — это объявление массива, при которой происходит выделение памяти на определённое количество элементов определённого типа.

**Размерность массива** — это количество индексов, которое необходимо для работы с массивом.

**Одномерный массив** — это массив, в котором для однозначного обращения к элементу нужен только один индекс.

**Размер массива** — это количество элементов в массиве.

**Событие —-** это любое событие, которое может произойти или не произойти в результате опыта.

**Опыт** — это действие, которое можно повторить множество раз в одинаковых условиях, результат которого непредсказуем.

**Невозможное событие —** событие, которое не произойдёт ни при данных условиях.

**Достоверное событие —** событие, которое точно произойдёт при данных условиях.

**Случайное событие —** событие, которое при одних и тех же условиях может произойти, а может и не произойти.

**Генератор псевдослучайных чисел** — это программа, которая принимает некоторое число (стартовое значение), проводит с ним сложные математические вычисления и возвращает результат, который практически невозможно предугадать.

**Стартовое значение** (семя, сид) — начальное число в последовательности псевдослучайных чисел, с которым проводятся вычисления, чтобы получить все дальнейшие псевдослучайные числа.

**Условные конструкции** –- это конструкции, которые направляют программу по определённому пути.

**Условные операторы** — это операторы, которые позволяют реализовать условные конструкции.

**Оператор if** — это оператор, который позволяет проверить условие и выполнить определённый блок в случае, если условие истинно.

**Оператор if-else** — это оператор, который позволяет проверить условие и выполнить одно, если условие истинно, или другое, если условие ложно.

**Оператор switch-case** — это оператор множественного выбора, который позволяет проверить условие на соответствие некоторому значению и выполнить определённый блок для конкретного значения.

### 

**Циклическая конструкция** — это конструкция, которая позволяет несколько раз выполнить один и тот же блок.

**Цикл** — многократное повторение последовательности действий.

**Тело цикла** — последовательность действий, которую повторяем.

**Итерация** — однократное выполнение тела цикла.

**Цикл с предусловием** — это цикл, перед выполнением каждой итерации которого проверяется условие на истинность.

**Цикл с постусловием** — это цикл, после выполнения каждой итерации которого проверяется условие на истинность.

**Оператор while** — оператор, который позволяет реализовать цикл с предусловием.

**Оператор do-while** — оператор, который позволяет реализовать цикл с постусловием.

**Оператор for** — оператор, который позволяет реализовать цикл с предусловием. Обязательно включает переменную-счётчик с начальным значением, конечное значение для счётчика и шаг изменения счётчика за одну итерацию.