**Программирование на С++**

**5-8 классы**

**Занятие 2**

**Тема: Переменные. Простые типы данных.**

**Цели и задачи:**

* Познакомиться с стандартным вводом и выводом в С++
* Познакомиться с переменными и типами данных
* Научиться использовать арифметические операции с переменными

**По результатам занятия слушатель будет знать:**

* Что такое переменная
* Что такое стандартный ввод и вывод
* Что такое тип данных
* Что такое комментарий

**По результатам занятия слушатель будет уметь:**

* Создавать переменные в С++
* Работать с стандартным вводом и выводом
* Использовать арифметические операции с переменными

**Теоретическая часть**

**Комментарии**

Комментариями называются пояснительные выражения, которые вы можете включать в ваш код на языке C++, чтобы объяснить, что именно выполняет программа.

Компилятор игнорирует все, что находится в комментариях. Это значит, что их не будет видно в результате выполнения программы**.**

Комментарий, который начинается с двух слэшей (//), называется однострочным комментарием. Комбинация двух слэш символов указывает компилятору игнорировать все, что следует за ними, вплоть до окончания строки.

**Многострочные комментарии**

Комментарии, в которых необходимо использование множества строк начинаются с /\* и заканчиваются \*/

Вы можете поместить их на одной строке, или же поместить одну и более строк между ними.

**Переменные**

Создание переменной резервирует место, или пространство в памяти для хранения значений. Компилятору необходимо, чтобы вы указали тип данных для каждой объявляемой переменной.

C++ предлагает большой ассортимент встроенных типов данных.

Целочисленный тип, встроенный тип, представляет собой целое число. Для определения переменной целочисленного типа используется ключевое слово int (integer).

C++ требует чтобы вы указали тип и идентификатор (имя) для каждой переменной.

Идентификатор это имя для переменной, функции, класса, модуля, или чего-либо другого определенного пользователем. Идентификатор начинается с буквы (A-Z или a-z) или нижнего подчеркивания (\_), с последующими дополнительно буквами, нижними подчеркиваниями, и цифрами (от 0 до 9).

Например, определим переменную под названием myVariable которая может хранить целочисленные значения

**Переменные**

Определяйте все переменные с именем и типом данных до их использования в программе. В случае, если у вас есть несколько переменных одинакового типа, можно определять их в одном объявлении, разделяя их запятыми.

int a, b;

Переменным могут быть присвоены значения и они могут использоваться для выполнения операций.

Например, мы можем дополнительно создать переменную sum, и сложить две переменные.

**Объявляем переменные**

У вас есть возможность присвоить значение переменной во время ее объявления или объявить переменную и присвоить ей значение позже.

Вы также можете изменить значение переменной.

Несколько примеров:

**int a;   
int b = 42;  
  
a = 10;  
b = 3;**

**Ввод пользователем (стандартный поток ввода)**

Чтобы позволить пользователю ввести значение используйте cin вместе с оператором извлечения (>>). Переменная содержащая извлекаемую информацию следует за оператором.

Следующий пример показывает, как принимать введенную пользователем информацию и сохранять ее в переменной num:

**int num;**

**cin >> num;**

**Получение введенной информации**

Следующая программа подсказывает пользователю ввести число и сохраняет его в переменной a:

(пример на слайде 13)

После запуска программы выводится сообщение "Please enter a number", затем ожидается ввод пользователем числа и нажатие кнопки Enter.

Введенное число сохраняется в переменной a.

Программа будет ждать столько времени, сколько необходимо пользователю чтобы ввести число.

**Получение введенной информации**

Вы можете выполнить ввод пользователем информации множество раз, как сделано в следующей программе:

(пример на слайде 14)

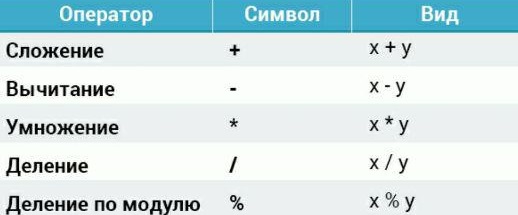
**Получение введенной информации**

Давайте создадим программу, которая позволяет ввести два числа и выводит на экран их сумму.

(пример на слайде 15)

**Арифметические операторы**

C++ поддерживает следующие арифметические операторы.



C++ имеет короткие операторы одновременного выполнения операции и присваивания.

int x = 10;  
**x += 4;** // equivalent to x = x + 4  
**x -= 5;** // equivalent to x = x – 5

**x \*= 3;** // equivalent to x = x \* 3  
**x /= 2;** // equivalent to x = x / 2  
**x %= 4;** // equivalent to x = x % 4

**Оператор инкремента**

Оператор инкремента используется для увеличения целочисленного значения на единицу.

Следующие выражения эквивалентны:

**x++; x +=1; x = x + 1;**

**Оператор декремента**

Оператор декремента (--) работает почти таким же образом, как и оператор инкремента, но вместо увеличения значения, он уменьшает его на единицу.

**x--; x -=1; x = x - 1;**

**Практическая часть**

**Задание**

Пользователь вводит 2 числа. Ваша программа в стандартный вывод (на экран) посылает сумму этих чисел, и разницу и произведение.

\*

Также выведите среднее арифметическое двух чисел (среднее арифметическое нескольких чисел – это их сумма заданных чисел, делённая на их количество)

\*\*

Выведите на экран поясняющие предложение, например, “Hello!” , “Enter first number” и т.д.

\*\*\*

Не используйте промежуточные переменные для хранения результатов арифметических операций.

(Файл с кодом лежит в папке code/)