



СЕВЕРНЫЙ (АРКТИЧЕСКИЙ)
ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ
ИМЕНИ М.В. ЛОМОНОСОВА

Введение в язык программирования C++

Латухина Екатерина Александровна,
старший преподаватель кафедры прикладной математики
и высокопроизводительных вычислений

Содержание

- С++. Основные понятия языков программирования
- Создание простых программ
- Типы данных
- Операции, переменные и выражения
- Список источников



История, состав, типы данных, операции, переменные и выражения

C++. Основные понятия языков программирования

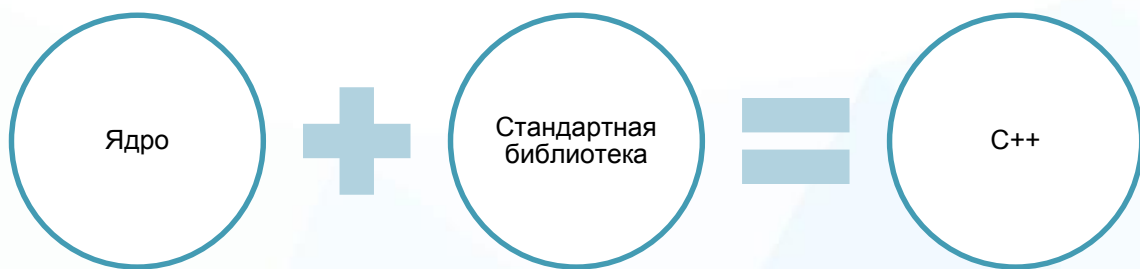
История языка C++

- Появился в начале 1980х на базе языка C в компании Bell Labs
- Автор языка — Бьёрн Страустрап — добавил к C возможности из объектно-ориентированного языка Симула
- C + классы → «C с классами» → C++
- 1998 — первый международный стандарт языка C++, де-факто являлся основным до появления C++11
- Каждые три года новый стандарт. Сейчас принят C++20, ведется работа над C++23
- На занятиях мы будем ориентироваться на C++11, т.к. большая часть добавленных позднее возможностей требуют профессионального владения языком C++. Но вы можете изучать их самостоятельно

Характеристики языка C++

- **Компилируемый** — программа превращается в исполняемый код и может запускаться операционной системой без использования дополнительного ПО
- **Статическая типизация** — тип переменной указывается при объявлении и не может быть изменен
- **Слабая (нестрогая) типизация** — есть преобразование типов
- **Мультипарадигменный** — поддерживает несколько парадигм программирования
- **Общего назначения** — можно писать программы различного предназначения

Состав C++



- **Ядро** — очень минималистичное, только базовый функционал, нет даже ввода-вывода
- **Стандартная библиотека** — все остальное, для использования нужно подключать

Состав языка



Алфавит

Большие и маленькие английские буквы A..Za..z (регистр имеет значение!)

Цифры 0..9

Специальные символы + – / * % < > = ! ; : . , \ | и другие

Разделители (пробел, табуляция, конец строки)

Идентификаторы

- Идентификатор — это имя программного объекта (переменной, функции, класса и т.д.)
- Может содержать только английские буквы, цифры, _
- Не может начинаться с цифры
- Не должны совпадать с ключевыми словами

Ключевые слова

- `auto, bool, break, case, catch, char, class, const, continue, default, delete, do, double, else, enum, false, float, for, friend, if, inline, int, long, namespace, new, operator, private, protected, public, return, short, signed, sizeof, static, struct, switch, template, this, throw, true, try, typedef, using, void, volatile, while` и другие

Константы

Числовые

- 42
- -1
- 27.23f
- -1.278

Символьные

- 'a'

Строковые

- "Hello"

Управляющие последовательности

Символы, которые можно задать внутри символьных и строковых констант

- `\'` одинарная кавычка
- `\"` двойная кавычка
- `\?` вопросительный знак
- `\\` обратный слеш
- `\0` нулевой символ
- `\n` перевод строки — новая строка
- `\t` горизонтальная табуляция
- `\v` вертикальная табуляция



Ввод-вывод, Hello world, запуск программы, среды разработки

Создание простых программ

Ввод и вывод

- В C++ есть два способа ввода-вывода — потоковый и унаследованный от C (операции printf и scanf)
- Для отправки в поток используется команда **<<**, для чтения из потока **>>**
 - Для вывода на экран используется cout
 - Для чтения с клавиатуры используется cin
 - При желании можно перенаправить cout и cin в текстовые файлы
- Команда отправки в поток/извлечения из потока пишется перед каждой отправкой или извлечением (примеры см. далее)
- Способ, унаследованный от C, рассматриваться на занятиях не будет



Hello world

```
#include <iostream>
using namespace std;
int main()
{
    cout<<"Hello world"<<endl;
    return 0;
}
```

Простая программа, выводящая на экран строку "Hello world"



Hello world

```
#include <iostream>
using namespace std;
int main()
{
    cout<<"Hello world"<<endl;
    return 0;
}
```

Подключаем заголовочный файл
стандартной библиотеки,
отвечающий за ВВОД-ВЫВОД



Hello world

```
#include <iostream>
using namespace std;
int main()
{
    cout<<"Hello world"<<endl;
    return 0;
}
```

Подключаем стандартное пространство имен (подробные объяснения см. в следующих лекциях)



Hello world

```
#include <iostream>
using namespace std;
int main()
{
    cout<<"Hello world"<<endl;
    return 0;
}
```

Выполнение программы начинается с
вызова главной функции, т.е.
функции main()



Hello world

```
#include <iostream>
using namespace std;
int main()
{
    cout<<"Hello world"<<endl;
    return 0;
}
```

{ } — обозначают границы блока, в данном случае тело главной функции



Hello world

```
#include <iostream>
using namespace std;
int main()
{
    cout<<"Hello world"<<endl;
    return 0;
}
```

Вывод строки на экран



Hello world

```
#include <iostream>
using namespace std;
int main()
{
    cout<<"Hello world"<<endl;
    return 0;
}
```

Завершение работы программы, возвращение значения. Ноль для функции `main()` означает, что программа завершилась корректно.

Код, написанный после этой команды, выполняться не будет!

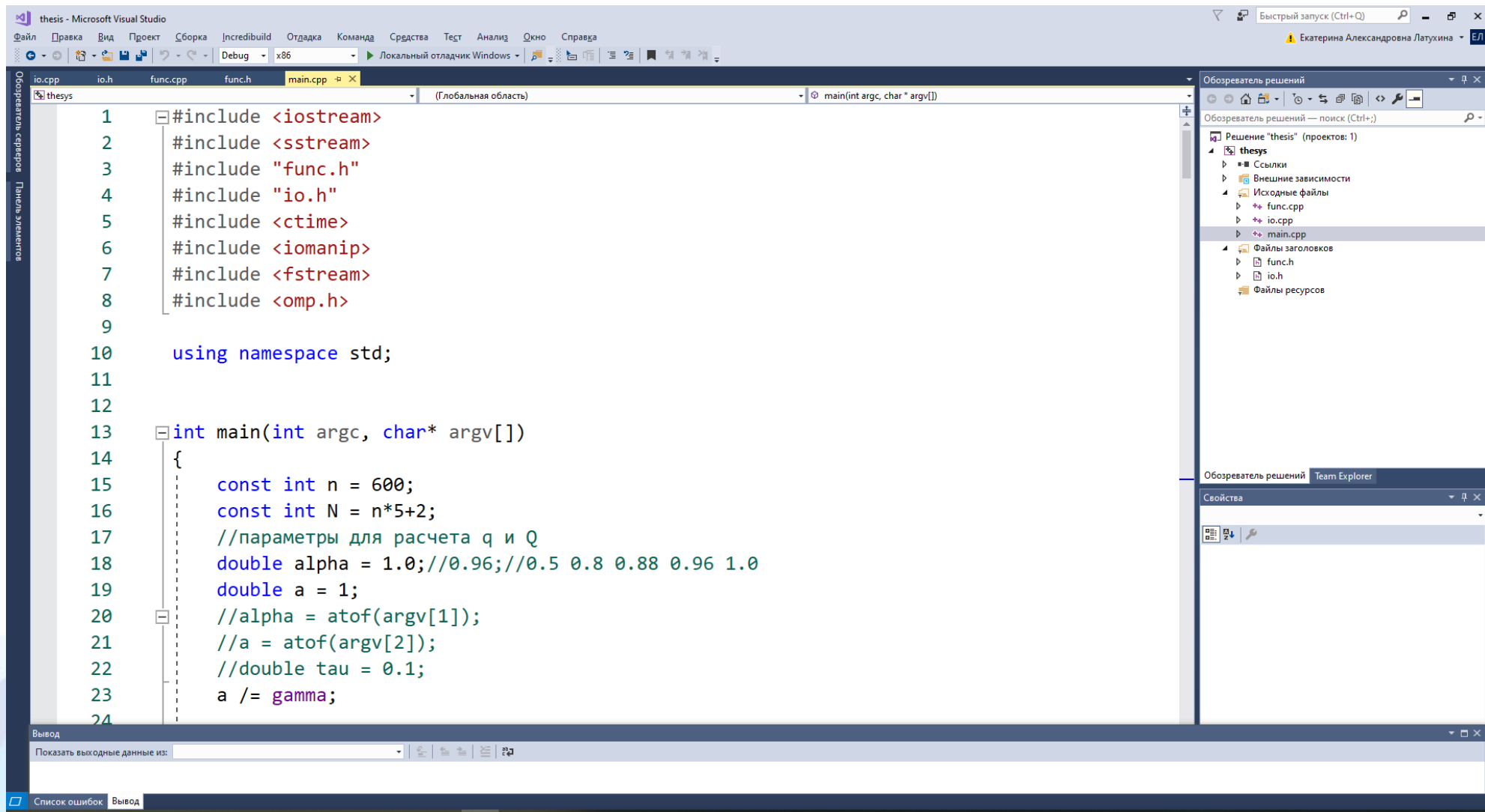


Запуск программы

- В онлайн-песочнице, например, <https://ideone.com/>
- В среде разработки

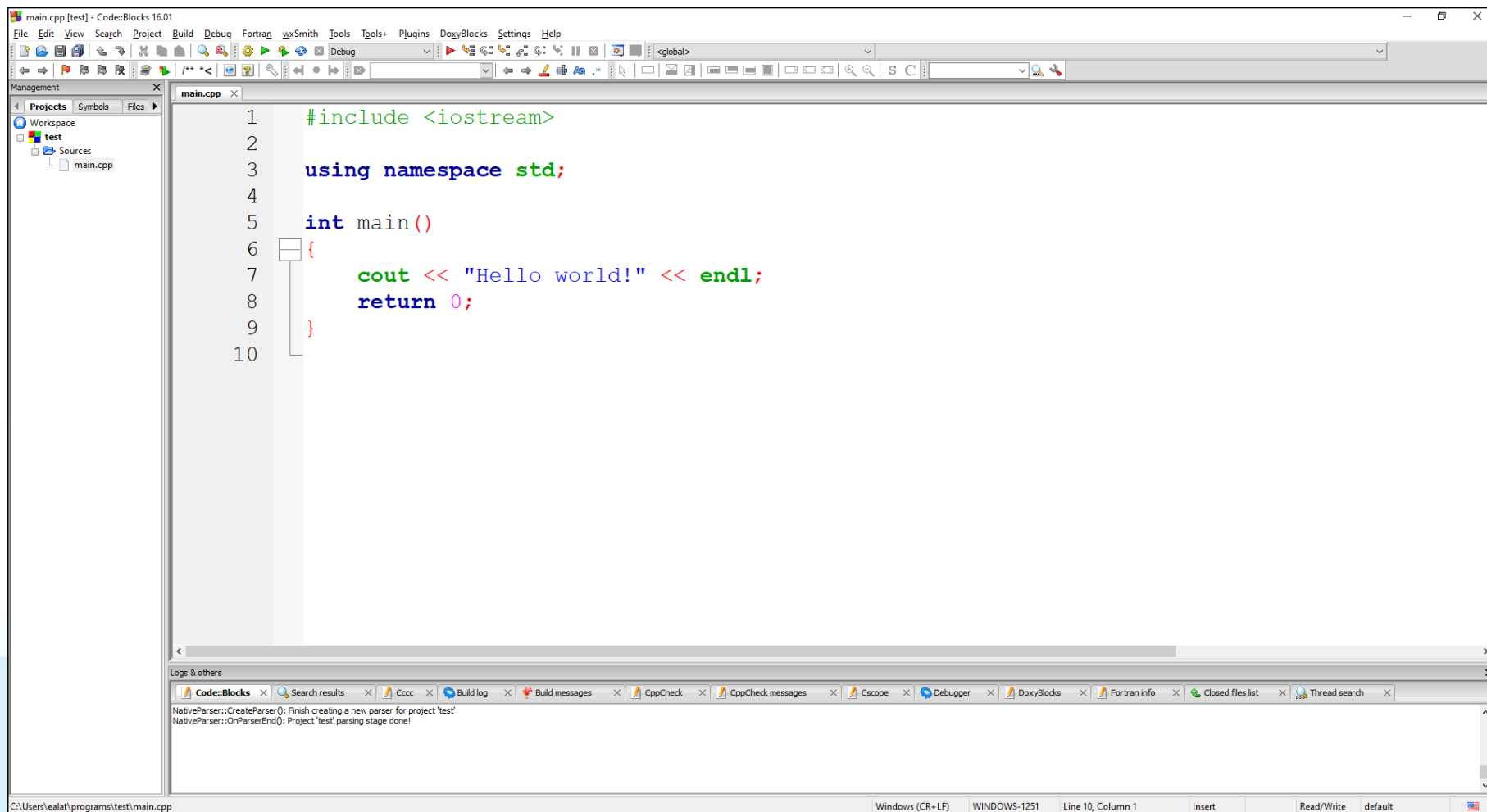


MS Visual Studio





Code::Blocks





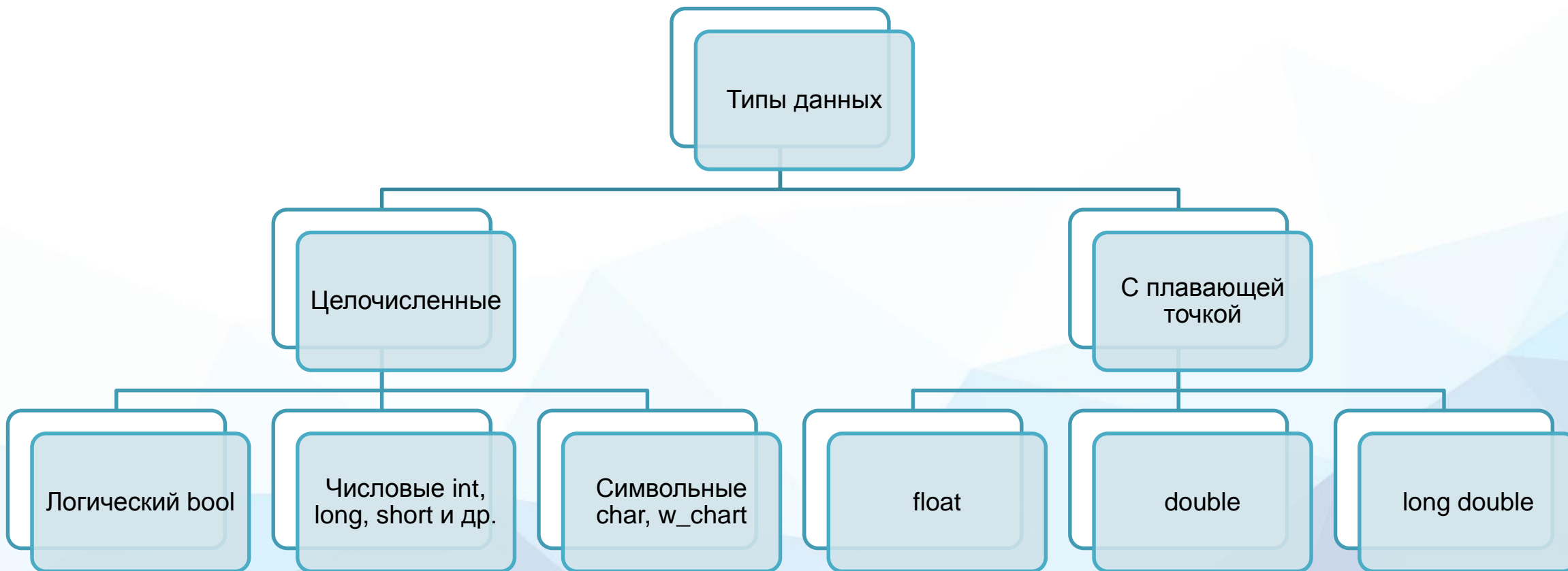
Простые типы данных (целочисленные, с плавающей точкой, логический, символьный)

Типы данных

Тип данных определяет

- Множество допустимых значений, которые могут принимать данные, принадлежащие к этому типу
- Набор операций, которые можно осуществлять над данными, принадлежащими к этому типу
- Размер памяти, занимаемый переменной этого типа

Простые (примитивные) типы данных



Числовые целые типы

- Базовый целочисленный тип `int`
- Размер зависит от реализации, обычно `int` 4 байта
- Модификаторы `signed` (по умолчанию, писать не нужно) и `unsigned` (беззнаковый, только неотрицательные числа)
- Модификаторы
 - `short` (2 байта)
 - `long` (4 байта)
 - `long long` (8 байт)
- Какие диапазоны значений у каждого из этих типов?



Числовые целые типы. Примеры

```
short a = -230;
```

```
int b = 38893;
```

```
long c = 3998823L;
```

```
long long d = 3434353534LL;
```

```
unsigned long q = 34030403ul;
```

Логический тип

- `bool`
- Занимает 1 байт, хотя достаточно 1 бита
- `true` и `false`
- Преобразование в логический тип и обратно выполняется по следующему правилу:

`0 → false → 0`

`не 0 → true → 1`

Символьный тип

- Символьный `char` (1 байт) и расширенный символьный `w_char` (2 байта, не будем использовать)
- Числовое значение от -128 до 127
- Беззнаковый тип `unsigned char` (от 0 до 255)

```
char a = 65;
```

```
char b = 'b';
```

```
cout << a << " " << b << endl;
```

Типы с плавающей точкой

- Точка (запятая) может быть помещена где угодно относительно цифр в строке
- Положение точки указывается отдельно
- Позволяет использовать существенно больший диапазон значений, чем у типов с фиксированной точкой
- FLOPS (floating-point operations per second) — скорость выполнения компьютером операций с плавающей точкой

Типы с плавающей точкой

Число с плавающей точкой состоит из следующих частей:

- знак мантиссы (знак числа)
- мантисса (значение числа без учёта порядка), [1; 10)
- знак порядка
- порядок (степень основания числа, на которое умножается мантисса)

$1,528535047 \times 10^{-25}$ записывается как **1.528535047E-25**



Диапазоны значений

Точность	Одинарная	Двойная	Расширенная
Размер (байты)	4	8	10
Число десятичных знаков	~7.2	~15.9	~19.2
Наименьшее значение (>0)	$1,2 \cdot 10^{-38}$	$2,3 \cdot 10^{-308}$	$3,4 \cdot 10^{-4932}$
Наибольшее значение	$3,4 \times 10^{+38}$	$1,7 \times 10^{+308}$	$1,1 \times 10^{+4932}$
Поля	S-E-F	S-E-F	S-E-I-F
Размеры полей	1-8-23	1-11-52	1-15-1-63

- S — знак, E — показатель степени, I — целая часть, F — дробная часть
- Так же, как и для целых, знаковый бит — старший.

Типы с плавающей точкой

- Одинарной точности float (4 байта)
- Двойной точности double (8 байт)
- Расширенный двойной точности long double (10 байт)

```
double a, b = 4.12;  
a = 22.1 + b;  
float pi = 3.14;
```

Недостатки

- Числа хранятся в двоичной системе, поэтому многие десятичные дроби являются периодическими
- Машинный эпсилон — наименьшее положительное число, добавление которого к данному числу не равно этому числу
- Для float это примерно $5.96e-8$, для double $1.11e-16$
- Числа с плавающей точкой нельзя сравнивать напрямую!

```
float a = 0.1f, b = 0.1f;  
if (a + b == 0.2)  
    cout << "yeah" << endl;  
else  
    cout << "No" << endl;
```

Операции, переменные и выражения

Переменные

- Переменная — это именованная область памяти, в которой хранятся данные определенного типа
- У переменной есть имя и значение
- Во время выполнения программы значение переменной можно изменять
- Перед использованием любая переменная должна быть описана
- Константа должна быть инициализирована при объявлении

Примеры

```
short int a = 1; // целая переменная a
const char C = 'C'; // символьная константа C
char s, sf = 'f'; // инициализация относится только к sf
char t(54);
float c = 0.22, x(3), sum;
```

Операции

<p>Унарные</p> <p>- !</p> <p>++ --</p> <p>* & new delete</p>	<p>Бинарные</p> <p>+ - / * %</p> <p>< > <= >= ==</p> <p>!=</p> <p><< >></p> <p>= += -= *=</p> <p>& ^ && </p>	<p>Тернарная</p> <p>? :</p>
--	--	-----------------------------

Унарные операции

- Операции с одним операндом (участником операции)
- В порядке уменьшения приоритета:
 - **++** (инкремент) и **--** (декремент) — есть в двух формах
 - Отрицание (**~** побитовое, **!** логическое, **-** унарный минус)
 - Взятие адреса **&** и разадресация ***** (будут рассмотрены позже)
 - Выделение памяти **new** и высвобождение памяти **delete**

Бинарные операции

- Операции с двумя операндами
- В порядке уменьшения приоритета:
 - Умножение *****, деление **/**, остаток от деления **%**
 - Сложение **+** и вычитание **-**
 - Сдвиги влево **<<** и вправо **>>**
 - Операции сравнения **<**, **<=**, **>**, **>=**
 - Проверка на равенство **==** (неравенство **!=**)
 - Поразрядные конъюнкция **&**, исключающее или **^**, дизъюнкция **|**
 - Логические конъюнкция **&&**, дизъюнкция **||**

Тернарная операция и операции присваивания

- Тернарная операция **?!** используется как замена простых ветвлений
- Операции присваивания имеют более низкий приоритет, чем тернарная операция:
 - Обычное присваивание **=**
 - Умножение с присваиванием ***=**, деление с присваиванием **/=** и т.д.

Выражения

- Состоят из операндов, знаков операций и скобок
- Используются для вычисления некоторого значения определенного типа
- Каждый операнд является выражением, константой или переменной

```
(a + 0.12) / 6
```

```
x && y || !z
```

```
(t * sin(x)) / ((2 * k + 2) * (2 * k + 3))
```



Дополнительные материалы по C++

Список источников

Практикум с задачиком

- Павловская, Т.А. С/С++. Структурное программирование : практикум / Т. А. Павловская, Ю. А. Щупак. - СПб. : Питер, 2007 и другие издания



Литература по C++

- Павловская, Т. А. С/C++. Программирование на языке высокого уровня : для магистров и бакалавров : учебник для студ. вузов, обуч. по напр. подгот. дипломированных спец. "Информатика и вычислительная техника" / Т. А. Павловская. - М. ; СПб. ; Нижний Новгород [и др.] : Питер, 2012. - 460 с. и другие издания
- <http://www.cplusplus.com/reference/>