Переменные и их типы

Консольный ввод / вывод, переменные и их типы, арифметические операции

Михаил Марков
C++ - Разработчик

Проверка связи

Поставьте “+”, если меня видно и слышно

Если у вас нет звука:

- убедитесь, что на вашем устройстве и на колонках включен звук
- обновите страницу вебинара (или закройте страницу и заново присоединитесь к вебинару)
- откройте вебинар в другом браузере
- перезагрузите компьютер (ноутбук) и заново попытайтесь зайти

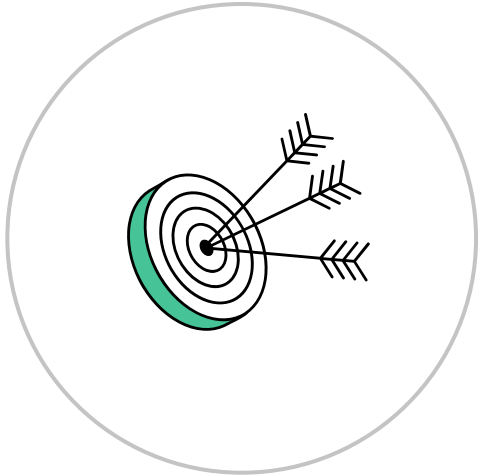
Михаил Марков

О спикере:
C++-разработчик, фрилансер

- Разработка алгоритма для релевантной выдачи объявлений.
- Разработка эмуляторов оборудования

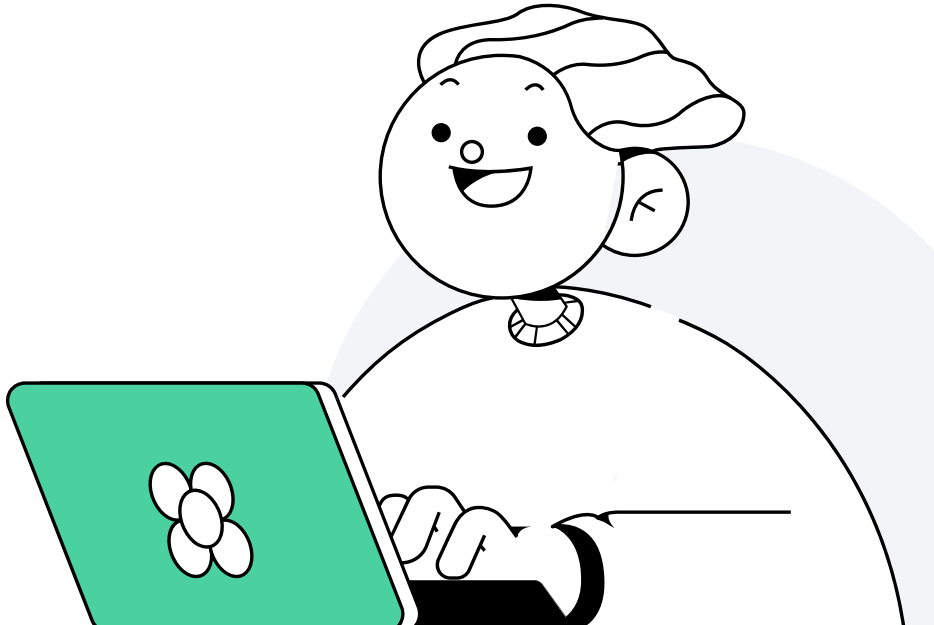
Цели занятия

- Рассмотрим ввод-вывод из консоли
- Узнаем, что такое переменная и её тип
- Научимся приводить типы переменных
- Познакомимся с базовыми арифметическими операциями



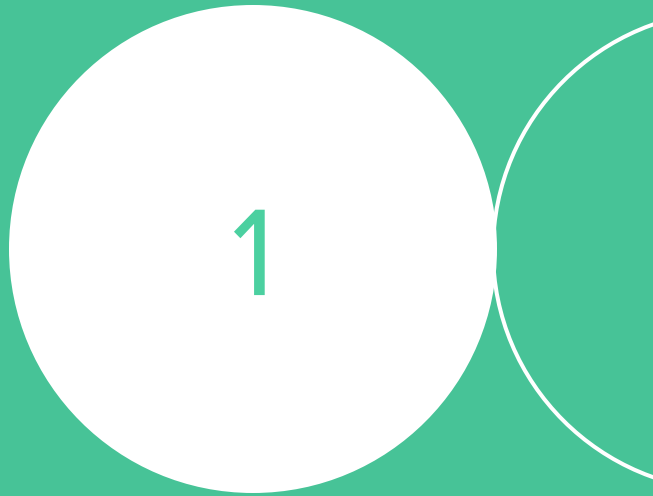
План занятия

1. Вывод в консоль
2. Переменные и типы
3. Ввод с консоли
4. Арифметические операции
5. Приведение типа
6. Итоги
7. Домашнее задание



*Нажми на нужный раздел для перехода

Вывод в консоль



Вывод в консоль

```
#include <iostream> // это подключение заголовочного файла
// пока что нам стоит это понимать как подключение библиотеки
// библиотеки добавляют удобные команды
// например, команду для вывода информации на экран:
// std::cout

int main()
{
     std::cout << "Hello World!" << std::endl;
     std::cout << std::endl;
     std::cout << 42 << std::endl;
     std::cout << -3.14 << std::endl;
}
```

Ссылка на готовый код в Repl

Переменные и типы

2

Переменные

```
int main()
{
// тип имя_переменной_1, имя_переменной_2, …;
int  a, b, c;
char CharName;
}
```

Переменная — это контейнер, в котором будет находиться какое-нибудь значение.

C++ — это строго типизированный язык. Это значит, что у любой переменной есть тип и этот тип нельзя изменить. Создали переменную определенного типа — переменная будет содержать значения только этого типа до конца своей жизни.

Напоминание: не забудьте в конце поставить точку с запятой. Объявление переменной — это тоже команда.

Основные типы переменных

Тип	Диапазон	Примечание
int	-2 147 483 648 … 2 147 483 647	целые числа
long	-9 223 372 036 854 775 808 … 9 223 372 036 854 775 807	
double	±5,0 * 10^−324 … ±1,7 * 10^308	дробные числа
bool	true / false	логический тип: истина или ложь
char	U+00 … U+FF	символ ASCII
void	—	ничего, «пустота»

Документация по типам

Имена переменных

- имя может содержать цифры, английские буквы и символ подчеркивания
- первый символ должен быть буквой или символом подчеркивания
- имя не может быть ключевым словом языка C++
- имя переменной должно отображать то, что в ней хранится
- максимальный размер имени - 255 символов

Имена переменных

Язык C++ — регистрозависимый:

```
char name;
char Name;
```
две переменные с разными именами

Сообщество C++-разработчиков придерживается нескольких стилей именования переменных:

- **CamelCase** (несколько слов пишутся слитно без пробелов, при этом каждое слово внутри фразы пишется с прописной буквы);
- **Венгерская нотация** (имена предваряются заранее оговоренными префиксами, состоящими из одного или нескольких символов).

Проверим себя

Вопрос: что лучше в качестве имени переменной `creditCardNumber` или `l`?

Проверим себя

Вопрос: что лучше в качестве имени переменной `creditCardNumber` или `l`?

Ответ: конечно, `creditCardNumber`! По одному имени мы сможем понять что именно эта переменная описывает.

Проверим себя

Вопрос: что лучше в качестве имени переменной `foo` или `void`?

Проверим себя

Вопрос: что лучше в качестве имени переменной `foo` или `void`?

Ответ: честно говоря, оба варианта не очень :(`foo` нам ничего не говорит, а `void` является зарезервированным словом и мы не можем его использовать в качестве имени переменной.

Проверим себя

Вопрос: что насчет `firstBit` и `1Bit`?

Проверим себя

Вопрос: что насчет `firstBit` и `1Bit`?

Ответ: второй вариант смотрится красивее, но его не получится использовать — имя переменной должно начинаться с буквы английского алфавита (или символа _).

Проверим себя

Вопрос: можно ли написать вот так — `SPEEDLIMIT_100`?

Проверим себя

Вопрос: можно ли написать вот так — `SPEEDLIMIT_100`?

Ответ: да, если это константа speedLimit = 100, в этом случае мы можем её именовать большими буквами.

Инициализация

```
int main()
{
	int a; // объявление неинициализированной переменной целого типа
	// если переменная не инициализирована, то в ней будет находиться какое-то
	// случайное значение

	a = 8; // в переменной a теперь находится число 8

	int a = 10; // ошибка!
	// нельзя сделать ещё одну переменную с таким же именем

	a = 10; // вот так можно
	// меняем значение в существующей переменной

	double pi = 3.14; // объявление и заполнение переменной
	// программисты такое называют «инициализацией»
}
```

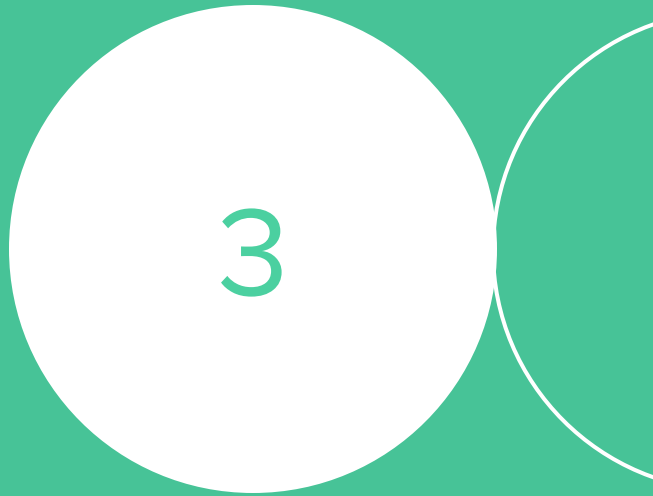
Рассмотрим инициализацию переменных

Инициализация

```
int main()
{
int intVar = 12;

long longVar = -12147483648;

double doubleVar = 2.7;

bool boolVar; // переменная не инициализирована!
boolVar = true;

char charVar = 'c';

std::string strVar = "Hello world";

strVar = 12; // ошибка!
// в тип string нельзя положить целое число (int)
}
```

Ссылка на готовый код в Repl

Ввод с консоли



Рассмотрим ввод с консоли

[Ссылка на готовый код в Repl]

Ввод числовых данных с консоли

```
#include <iostream>

int main()
{
  int num;

  std::cout << "Hello!" << std::endl;
  std::cout << "Enter number: ";
  std::cin >> num;
  std::cout << "Number is: " << num << std::endl;
}
```

В данном примере мы сначала используем оператор `std::cout <<` для вывода сообщения пользователю, затем с помощью аналогичного оператора `std::cin >>` получаем пользовательский ввод.

Ввод символьных данных с консоли

```
#include <iostream>

int main()
{
  std::string str;

  std::cout << "Enter string: ";
  std::cin >> str;
  std::cout << "Your string is: " << str << std::endl;
}
```

В прошлом примере мы рассмотрели ввод с консоли числового значения. Если же требуется ввести не число, а произвольные символы, необходимо объявить переменную подходящего типа (например, std::string) и вводить данные в неё. В отличие от int, в переменную типа std::string можно ввести любые символы, не только цифры.

Ввод строки текста с консоли

```
#include <iostream>

int main()
{
  std::string str;

  std::cout << "Enter some text: ";
  std::getline(std::cin, str);
  std::cout << "Your text is: " << str << std::endl;
}
```

Теперь мы умеем вводить с консоли произвольные символы. Но если среди них встретится пробел (“ “), он будет рассмотрен как разделитель, и в целевую переменную попадут только символы, следующие до него. Чтобы ввести, например, фразу “Мама мыла раму”, следует воспользоваться функцией std::getline().

Арифметические операции

4

Базовые арифметические операции

+	сложение
-	вычитание
*	умножение
/	деление
%	остаток от деления

Базовые операции. Пример

```
#include <iostream>

int main()
{
std::cout << 1 + 6 << std::endl;
std::cout << 4 - 9 << std::endl;
std::cout << -3.5 * 3 << std::endl;
std::cout << 7 / 5 << std::endl;
std::cout << 7.0 / 5.0 << std::endl;
std::cout << 5 % 3 << std::endl;
std::cout << 2 + 2 * 2 << std::endl;
std::cout << 1 + 6 << std::endl;
}
```

Ссылка на Repl

Сокращенные арифметические операции

Операция	Название	Пояснение
x += N	сложение	x = x + N
x -= N	вычитание	x = x - N
x *= N	умножение	x = x * N
x /= N	деление	x = x / N
x %= N	остаток от деления	x = x % N
x++ **++x**	инкремент	увеличивает значение на 1, то же что и += 1
x-- **--x**	декремент	уменьшает значение на 1, то же что и -= 1

Эти операции можно применять только к переменным!

Рассмотрим операции на примере

Пример

```
#include <iostream>

int main()
{
int num = 10;
std::cout << num++ << std::endl;
std::cout << ++num << std::endl;

std::cout << num-- << std::endl;
std::cout << --num << std::endl;

std::cout << (num += 5) << std::endl;
std::cout << (num -= 6) << std::endl;
std::cout << (num *= 9) << std::endl;
std::cout << (num /= 10) << std::endl;
std::cout << (num %= 3) << std::endl;
}
```

Ссылка на Repl

Приведение типа

5

Рассмотрим такой пример:

```
char num = 4;
char num1 = '4';
std::cout << num + num1 << std::endl;
```

Вопрос: Что будет выведено на экран?

Вспоминаем основные типы переменных

Рассмотрим такой пример:

```
char num = 4;
char num1 = '4';
std::cout << num + num1 << std::endl;
```

Вопрос: Что будет выведено на экран?

Ответ: 56

Вспоминаем основные типы переменных

Приведение типа

Преведение типа (typecast) — преобразование значения одного типа в значение другого типа.

Преобразование типа может быть:

- **явным** (указывается программистом как отдельная команда);
- **не явным** (выполняется самим компилятором).

Явное приведение типа

- круглые скобки
- static_cast
- dynamic_cast
- const_cast
- reinterpret_cast

Сейчас мы рассмотрим только первые два (остальные будут рассмотрены в лекциях по STL и boost).

```
(int) num                 // старый вариант, оставлен для совместимости с языком C
static_cast< int >(num)   // современный вариант, входит в стандарт C++
```

Пример

Допустим, нам нужно преобразовать тип char к типу int, тогда:

```
char num = 4
(int) num                 // старый вариант, оставлен для совместимости с языком C
static_cast< int >(num)   // современный вариант, входит в стандарт C++
```

Далее, рассмотрим только использование **static_cast**< тип >(объект):

- тип — тип переменной который нужно получить;
- объект — переменная, тип которой нужно преобразовать.

Рассмотрим преобразования типов

Пример

```
#include <iostream>

int main()
{
      char num = 4, num1 = '4';
      double fl = 5.6, fl1 = 5.7;

      std::cout << (num + num1) << std::endl;                                  // 56
      std::cout << num << std::endl;                                           //
      std::cout << static_cast< int >(num) << std::endl;                       // 4
      std::cout << fl << std::endl;                                            // 5.6
      std::cout << fl1 << std::endl;                                           // 5.7
      std::cout << fl + fl1 << std::endl;                                      // 11.3

      std::cout << static_cast<int>(fl + fl1) << std::endl;                    // 11

      std::cout << static_cast<int>(fl)+static_cast<int>(fl1)<< std::endl;     // 10

      std::cout << static_cast<int>(num + num1) << std::endl;                  // ? Напишите в чат
}
```

Ссылка на Repl

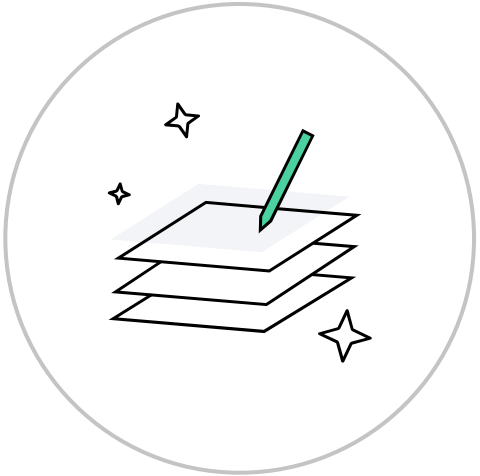
Итоги

1. Научились выводить информацию на консоль и получать информацию из консоли
2. Рассмотрели базовые арифметические операции
3. Разобрались с переменными и типами

Домашнее задание

Давайте посмотрим ваше домашнее задание

1. Вопросы по домашней работе задавайте в чате группы
2. Задачи можно сдавать по частям
3. Зачёт по домашней работе ставят после того, как приняты все задачи



Задавайте вопросы и пишите отзыв о лекции

Михаил Марков
C++ - разработчик