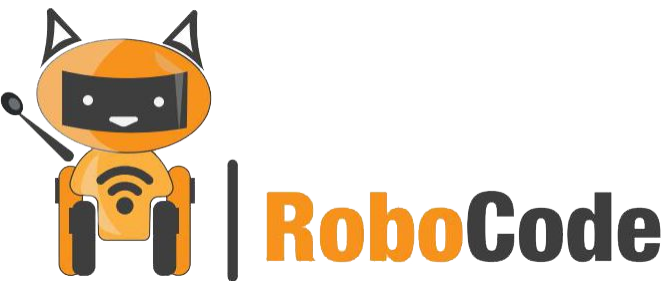


Pro 1.0

Типы данных и циклы C++



План занятия



Повторяем известные нам типы данных и что это такое



Что такое цикл и какие они бывают?



Выводим на экран значения

Английские термины

ENGLISH	РУССКИЙ
Символ	Character
Переменная	Variable
Беззнаковый	Unsigned
Поплавок	Float
Целое число	Integer



Типы данных

Типы данных – это возможность программе поделить различные данные на классы, каждый из которых занимает ограниченное количество памяти (размер ячейки).

Разные типы данных занимают разное количество памяти, что позволяет её

Char, Bool 1 byte	Int, 2 byte	Float, 4byte	Double 8 byte
-------------------------	----------------	-----------------	------------------

Известные нам типы данных

Назовите что это за тип данных, приведенных ниже, и его особенности:

```
1 | int
2 | float
3 | char
4 | String
5 | boolean
6 | long
7 | unsigned int
```

Объявление и использование переменных

Конструкция правильного объявления переменной выглядит так:

Тип данных	Название	Присваивание значение (необязательно)
float	variable	=0;

Циклы

- Для выполнения некоторых действий множество раз в зависимости от определенного условия используются циклы.

В языке C++ имеются следующие виды циклов:

- for
- while
- do while

for

- Цикл, позволяющий повторять действие определенное количество раз.

Цикл будет выполняться, пока выполняется условие указанное в его инициализации

Структура цикла for

начальное значение	конечное значение	Параметр цикла (шаг)
for(int i=0; i<10; i+=2.5)		

Пример использования

```
void setup() {  
  for(int i=0; i<13; i++){  
    pinMode(i, OUTPUT);  
  }  
}
```


while

- Цикл `while` выполняется пока его условие будет истинным, поэтому количество раз может быть бесконечным или каждый раз разным.

Структура цикла `while`

условие выполнения

```
while( a<10)
```

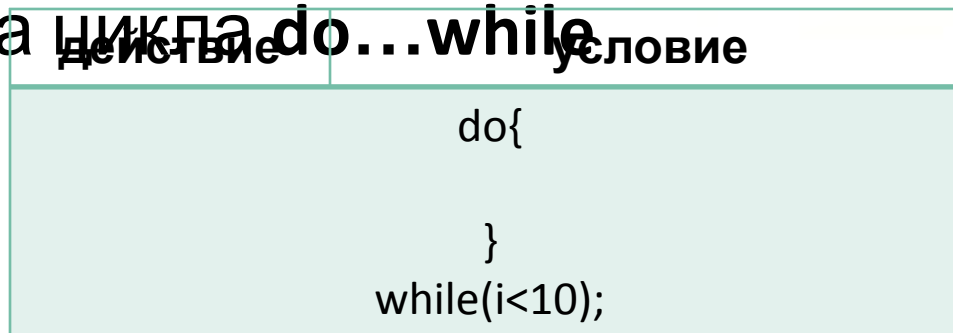
Пример использования

```
int i=0;
void setup() {
  while(i<13){
    i++
    pinMode(i, OUTPUT);
  }
}
```

do...while

- Цикл отличается тем, что прежде всего идет действие, а потом проверка условия выполнения. Таким образом, если условие не верное, то цикл все равно выполнится один раз.

Структура цикла **do...while**



Пример использования

```
int i=0;  
void setup() {  
do{  
    showNubmer(i);  
}  
while(i>0 && i<10);  
}
```

Практические умения

Задание

1. Подключить LCD экран и датчик температуры DHT-11
2. Вывести значение температуры на экран используя тип данных int, далее – float

Решение задачи

```
#include <DHT.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

DHT dht(2, DHT11);
LiquidCrystal_I2C lcd(0x3F, 16, 2);
int temperature=0;
void setup()
{
  dht.begin();
  lcd.begin ();
  lcd.backlight();
}

void loop()
{
  temperature=dht.readTemperature();
  lcd.setCursor(1, 1);
  lcd.print(temperature);
}
```

Вторая задача

- Подключить кнопку к Арудино
- Используя цикл **for**. При нажатии на кнопку сделать счетчик чисел от 0 до 10
- Используя цикл **while**. При нажатии на кнопку сделать счетчик чисел от 0 до 10
- Используя цикл **do...while**. При нажатии на кнопку сделать счетчик чисел от 0 до 10

Проанализировать полученный результат

Решение задачи для for

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x3F, 16, 2);
int counter = 0;
void setup()
{
  pinMode(2, INPUT);
  lcd.begin ();
  lcd.backlight ();
}

void loop()
{
  if (digitalRead(2)) {
    for (int i = 0; i <= 10; i++) {
      counter++;
      lcd.setCursor(1, 1);
      lcd.print(counter);
      delay(1000);
    }
  }
}
```

Решение задачи для while

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x3F, 16, 2);
int counter = 0;
void setup()
{
  pinMode(2, INPUT);
  lcd.begin ();
  lcd.backlight();
}
void loop()
{

  while(digitalRead(2) and counter<=10) {
    counter++;
    lcd.setCursor(1, 1);
    lcd.print(counter);
    delay(1000);
  }
}
```

Решение задачи для do...while

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x3F, 16, 2);
int counter = 0;
void setup()
{
  pinMode(2, INPUT);
  lcd.begin ();
  lcd.backlight();
}
void loop()
{

  while(digitalRead(2) and counter<=10) {
    counter++;
    lcd.setCursor(1, 1);
    lcd.print(counter);
    delay(1000);
  }
}
```