

УОТ-1 Тагирчев

06.09.2022

1. Типы данных C++

`int result = 0;` Указываем тип и значение `Integer`
`double coefficient = 10.81;`

`auto name = "Valerii";` Анализ кода за исключением
`auto address;` ~~error~~ компилятор не может определить
тип без значения.

`age = 12` error При указании необходимо указать
тип или `auto`

Вспомогательные типы: `int`, `double`, `long`, `bool`, `char`, `wchar_t` для
ASCII и Юникода соответственно.

Большинство целочисленных функциональных типов
(за исключением `bool`, `double`, `wchar_t` и строковых типов) имеют
unsigned версию, которая позволяет хранить значения
которые не могут хранить отрицательные.

Буде количество символов, пишется только
quantum.

int	4 байт	8
double	8 байт	8 083
bool	1 байт	T/F
char	1 байт	'a'
wchar_t	2 байт	символы юникода
unsigned char	1 байт	В C++ нем. полностью мило. char используется unsigned char где нужно.
unsigned int	4 байт	Вариант по умолчанию где Symbol size
long long	8 байт	Представляет очень большие значения.

адреса (*). Тип указателя хранит в памяти адрес расположения, в котором хранятся структурные др.

2. Классы памяти

В C++

автоматический (auto)

регистрационный (register)

внешний (extern)

статический (static)

изменяемый (mutable)

Класс auto определяет автоматическую переменную.

Если в методе функции объявлен объект класса, то этот объект имеет класс auto.

Register - указывает компилятору положить где-то хранение данных объекта не в ячейке памяти, а в регистрах процессора.

Extern: Если класс объявлен в одном месте, а нужно объявить, что этот класс есть внешний в другом месте.

Пример

```
#include "MyClass.h"
extern class CMyClass; - since MyClass.h
```

Static можно где угодно объявлять в программе
общими переменными

Mutable используется с const
показывает, что переменная не будет изменена

3. Указатели
int x;

int *y = &x; - это указатель на переменную x
адрес переменной x. Возвращает указатель

int z = *y; - Указатель можно разыменовать и
получить значение "x" - Возвращает значение

Ссылки тоже ссылки, что указывается, но в
структуре ссылок и некоторые ссылки
`int x;`

`int &y = x;` (Присваивание адреса ссылки, а не ссылки)
`int z = y;`

Указатель есть указатель на структуру или
на доступный элемент структуры с указателем на
структуру. Ссылка доступна только через указатель

4. Адресная арифметика - способ вычисления
адреса какого-либо объекта при помощи ссылки
структуры. Мы указываем, а также используем
указатель в структуре структуры.

5. Арифметика переменных - указывается к адресу
структуры с переменными. Можно работать
Меморизация

```
char* str = new char [30];
```

```
// delete [ ] str;
```

```
str = new char [60]; - указатель на новую память
```

```
delete [ ] str; освобождение 60
```


Методу как - некоторая операция помет.

6. Функция - блок кода, выполняющий определенную операцию.

Тип возвращаемого значения Имя (Применение зн.) {

Тело функции

return значение

}

Синтаксис функции - часть кода программы, позволяющая использовать функцию повторно.

7. Static предоставляет возможность использовать переменные.

8. Функция ^{возвращает значение} может вернуть return

Пример: `int sum(int a, int b) {
 return a + b;
}`

9.

```
std::cout << "Bucuresti are un";  
std::cin >> i;  
std::cout << "Bucuresti are un: " << i << "\n";
```

Plac nuca nu nouca cstdio
cout - cu aym (cout)
cin - cu un (cin)

10. Block - block io stream

include <io stream>

Obrem cett boloin yvome blyper nemer
extern ostream vett;

ynpalueem boloin dez byperuay b omangapm boloz
b luy daimetore nemera.

extern ostream cout;

ynpalueem veebom nemer cout

Obvemo std::cout - obvemo boloz

std::cin - boloz chetk

std::clog - que nemera.

vcerr - boloin dez byperuay

vcin - vglueme v by b luy daimetore

nemera.