



СИНТАКСИС ЯЗЫКА C++

СИНТАКСИС ЯЗЫКА C++

Общие положения



Прописные и строчные буквы считаются разными символами, поэтому идентификаторы DATABASE, DataBase, database относятся к разным переменным, константам, объектам и т.п. При записи идентификатора могут использоваться латинские буквы, арабские цифры, символ подчеркивания. Идентификатор не может начинаться с цифры и не должен содержать пробельных символов.



Пробельные символы могут размещаться в любом месте текста, но не внутри идентификатора.



Комментарии в тексте заключаются в скобки вида `/* */`.

Строка текста, перед которой помещен символ двойной слеш `«//»`, также считается комментарием.



Каждое предложение языка заканчивается точкой с запятой `«;»`, кроме комментариев и особо оговоренных случаев.

СИНТАКСИС ЯЗЫКА C++

Общие положения



В строке может размещаться несколько операторов, разделенных точкой с запятой. Этих случаев следует избегать, так как размещение нескольких операторов в одной строке снижает читаемость программы.



Фигурные скобки { } выделяют составной оператор. Все операторы, помещенные между ними, воспринимаются синтаксически как один оператор. Точка с запятой после фигурных скобок не ставится.



Все используемые типы, константы, переменные, функции должны быть объявлены или описаны до их первого использования. Объявления могут быть вставлены в любом месте текста.

ОБЗОР ЯЗЫКА C++

Типы данных C++

В C++ *тип данных (data type)* определяет способ хранения информации в памяти.

Тип данных	Размер в байтах	Диапазон допустимых значений
char	1	от -128 до 127
unsigned char	1	от 0 до 255
short	2	от -32768 до 32767
unsigned short	2	от 0 до 65535
long	4	от -2147483648 до 2147483647

ОБЗОР ЯЗЫКА C++

Типы данных C++

Тип данных	Размер в байтах	Диапазон допустимых значений
unsigned long	4	от 0 до 4294967295
int	4	совпадает с long
unsigned int	4	совпадает с unsigned long
float	4	от 1.2E-38 до 3.4 E38
double	8	от 2.2E-308 до 1.8E308
long double	10	от 3.4E-4932 до 1.1E4932
bool	1	true или false

ОБЗОР ЯЗЫКА C++

Переменная (variable) - именованная, либо адресуемая иным способом абстрактная/виртуальная или физическая память.

Адрес, на который указывает переменная используется для доступа к данным.

Данные, находящиеся в переменной, называются значением этой переменной.

Свойства значений в переменной явно или неявно определяются *типом* переменной.

```
int x;           //объявлена переменная «x» целого типа
x = 100;         // «x» инициализирована значением 100
x += 50;         //теперь «x» содержит значение 150
int y=150;       //«у» объявлена и инициализирована
                 //значением 150
```

Константа - способ адресации данных, изменение которых рассматриваемой программой не предполагается или запрещается

Именованные константы

const тип имя_константы = значение;

const float PI = 3.14159;

const float PI2 = 2*PI;

Константы перечислимого типа

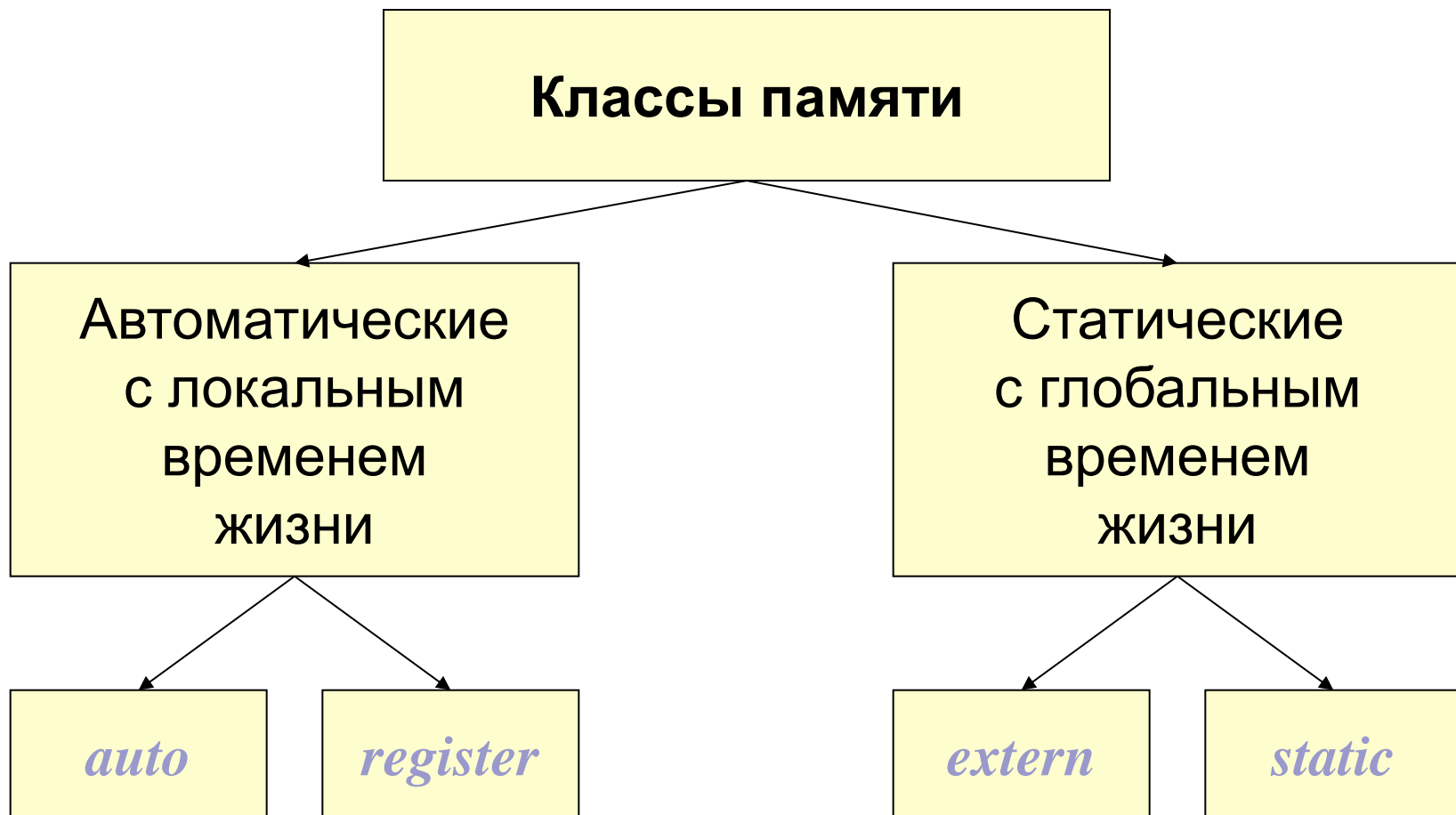
enum имя { значения};

enum color {red, yellow, green};

enum color {red, yellow = 3, green = red + 1};

//доступ red или color(red) или color{red}

Классы памяти



ОПЕРАЦИИ

Классы памяти

```
int i = 1, k = 4;    //объявлены глобальные переменные
{
    int i = 5, j = 2;    //объявлены переменные внешнего блока
    .....              //видны перемен. j,k и перемен. i этого блока
    {
        int i = 7;      //объявлена перемен. i внутреннего блока
    }                  //видны перемен. j,k и перемен. i внутр. блока
    .....              // видны перемен. j,k и перемен. i внешнего блока
}
```

ОПЕРАЦИИ

Арифметические операции

Унарные арифметические операции

Обозначение	Операция	Пример
+	унарный плюс	+7
-	унарный минус	-X
++	инкремент	i++; ++i
--	декремент	i--; --i

$$a += 1 \equiv a = a + 1$$

ОПЕРАЦИИ

Арифметические операции

Постфиксная запись:

$i++$

Префиксная запись:

$++i$

```
int i = 1, j;  
j = i++ * i++;  
//i = 3, j = 1
```

```
int i = 1, j;  
j = ++i * ++i;  
//i = 3, j = 9
```

ОПЕРАЦИИ

Арифметические операции

Бинарные арифметические операции

Обозначение	Операция	Пример
$+$	сложение	$X + Y$
$-$	вычитание	$X - Y$
$*$	умножение	$X * Y$
$/$	деление	X / Y
$\%$	остаток целочисленного деления	$X \% Y$

ОПЕРАЦИИ

Арифметические операции

Операции присваивания

Обозначение	Операция	Пример
=	присваивание	$X = Y$
+=	присваивание со сложением	$X += Y \ (X = X + Y)$
-=	присваивание с вычитанием	$X -= Y \ (X = X - Y)$
/=	присваивание с делением	$X /= Y \ (X = X / Y)$
%=	присваивание остатка целочисленного деления	$X \% = Y \ (X = X \% Y)$

ОПЕРАЦИИ

Арифметические операции

Операции отношения и эквивалентности

Обозначение	Операция	Пример
<code>==</code>	равно	<code>1 == Max</code>
<code>!=</code>	не равно	<code>X != Y</code>
<code><</code>	меньше чем	<code>X < Y</code>
<code>></code>	больше чем	<code>Len > 0</code>
<code><=</code>	меньше или равно	<code>Cnt <= 1</code>
<code>>=</code>	больше или равно	<code>I >= 1</code>

ОПЕРАЦИИ

Арифметические операции

Логические операции

Обозначение	Операция	Пример
!	отрицание	!A
&&	логическое И	A && B
	логическое ИЛИ	A B
~	побитовое отрицание	~X
&	побитовое И	X & Y
	побитовое ИЛИ	X Y

ОПЕРАЦИИ

Арифметические операции

Приоритет операций и ассоциативность

Приоритет	Операция	Ассоциативность
1	() [] -> ::	слева направо
2	! ~ + - ++ --	справа налево
3	* / %	слева направо
4	+ -	слева направо
5	<< >>	слева направо
6	< <= > >=	слева направо

ОПЕРАЦИИ

Арифметические операции

Приоритет операций и ассоциативность

Приоритет	Операция	Ассоциативность
7	<code>==</code> <code>!=</code>	слева направо
8	<code> </code>	слева направо
9	<code>&&</code>	слева направо
10	<code> </code>	слева направо
11	<code>=</code> <code>+=</code> <code>/=</code> <code>%=</code> <code>+=</code> <code>-=</code>	слева направо

ОПЕРАЦИИ

Директива препроцессора #define

Символические константы

#define идентификатор_константы замещающий_текст

#define PI 3.14159

Макросы с параметрами

#define идентификатор_макроса (аргументы) замещающий текст

#define CIRC(x) (3.14159*(x)*(x)) //точка с запятой после
объявления макроса не
ставиться