

# Первое знакомство с алгоритмами



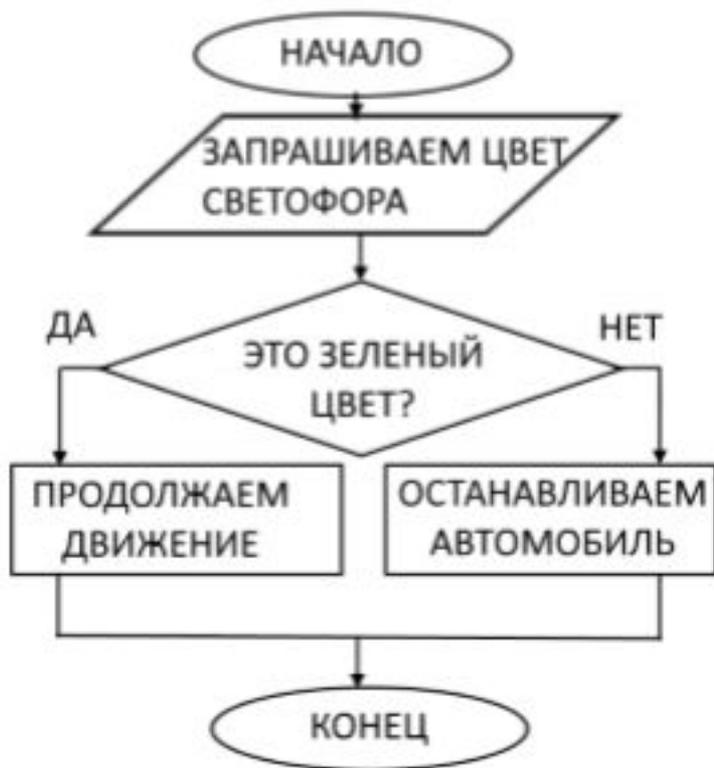
# Графический способ описания алгоритма

Блок-схема — графический способ описания алгоритма. Она состоит из нескольких функциональных блоков, каждый из которых в свою очередь выполняет свое четкое назначение.



Логическая головоломка про Козу, Волка и Капусту. Крестьянину необходимо всех перевезти на противоположный берег. На берегу нельзя одновременно оставить козу с капустой или волка с козой.

Автомобиль движется по дороге. Вдали водитель видит светофор, необходимо определить действия водителя в зависимости от горящего цвета светофора.



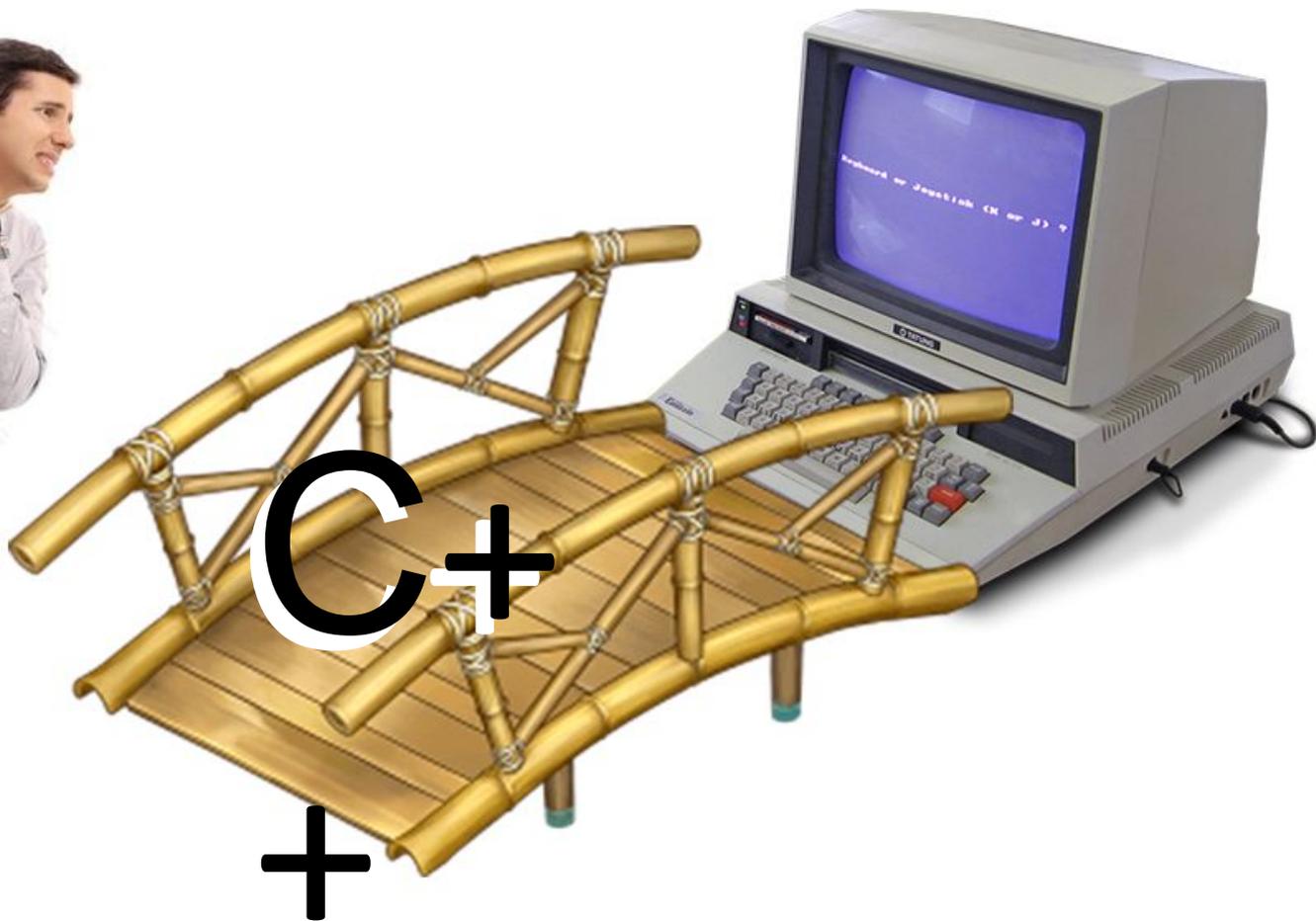


Первый  
проект  
Компьютер  
понимает только  
один язык — язык  
машинных кодов.

# «Hello, world!»

```
_YH|ÿ+. _5lч№+f-H|ÿ+ +f+ ∞д-ы+fO_fO!∞+эм№∞+∞+|Ф№+∞-;-
wГГ=xФ№+MZR __ + @ A -! | -!+ L-!T
u . _5lч№+f-&ЛZ¶QW&бФЬ№+&Л &ЛJf&OB_Ф№
+f;sллФ№+Л&lt; _t=ИФ№+_G¶u_O¶WfM+-чfЛ№3+M-Ас°
_>] Л!Л+ЛМЛm _сбшЬ№+Л 9шФ№+t$Л$рФ№+ш№) fd! __+RPбшЬ
№+Л 9шФ№+tЛ$рФ№+шС) + +Э
тAc° -щЗ-Л_Ф№+f;s $lФ№+Л +u"бАФ№+∞+MAc°ÿD At Э+- 3+
ыÿы•ы°ыÿыІЛD$П$. _4ч№++ . _ dЛ4 ÿ- ubСЛ dr4 +__e --fУЙf-+
-+ЙА+@У№+Л-[Xf +tRPRVh•ÿ+ш_S Z+efO-fЛцf-d∞+$ RfRfh ш4
Л+Зы!SшС∞д-ы!∞+ьМ-∞++∞_e fM+fO_fO!f∞бЛĒdÿ t3+ОшAc°Л$АФ№+-
dб4 ЙdЙ$4 f_f _t dЛ4 ÿ- uSSL dr4 +__e --fУЙf-+-+ЙА+@У№+Л-[
+eЙ]№fM+fO-fЛц+yf-d∞+$ RfRfh шз Л+Зы+Sш!
$ PE L !7 p ∞!
' P OK P ÿ+ 0
```

Вывод фразы на машинном языке



Языки программирования делятся  
на две основные группы  
**ИНТЕРПРЕТИРУЕМЫЕ**  
**и КОМПИЛИРУЕМЫЕ.**

# ИНТЕРПРЕТИРУЕМЫЕ

- 1) Каждый раз при запуске программы интерпретатор осуществляет проверку кода построчно.
- 2) Если ошибок в синтаксисе нет, команды будут преобразованы в машинный код.
- 3) Программа запустится на выполнение.

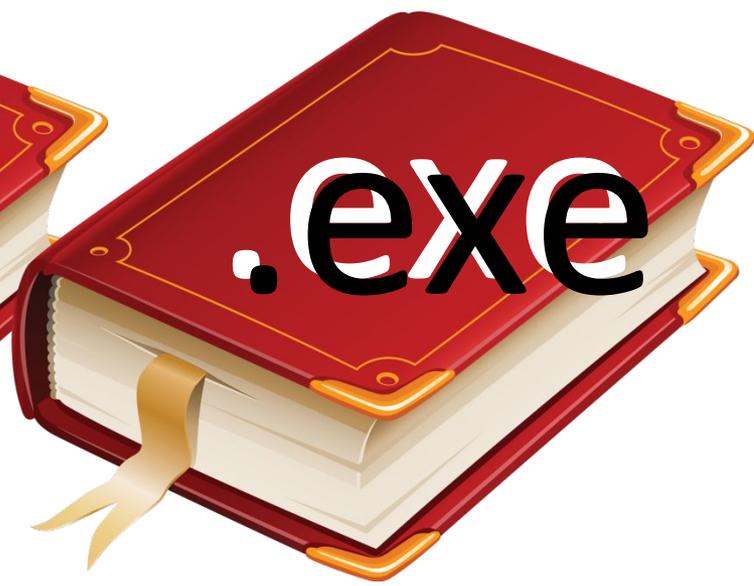
При каждом запуске будет срабатывать интерпретатор и опять осуществлять проверку кода. Таким образом, можно сделать выводы, что машинная версия кода нигде не сохраняется.

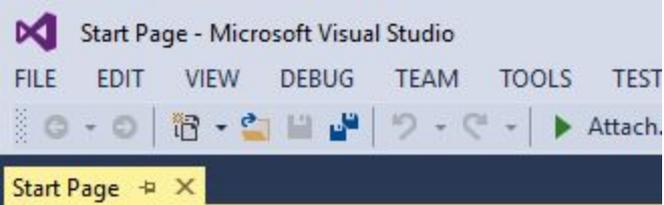
Минусы такого подхода состоят в том, что скорость запуска программы снижается.



# КОМПИЛИРУЕМ

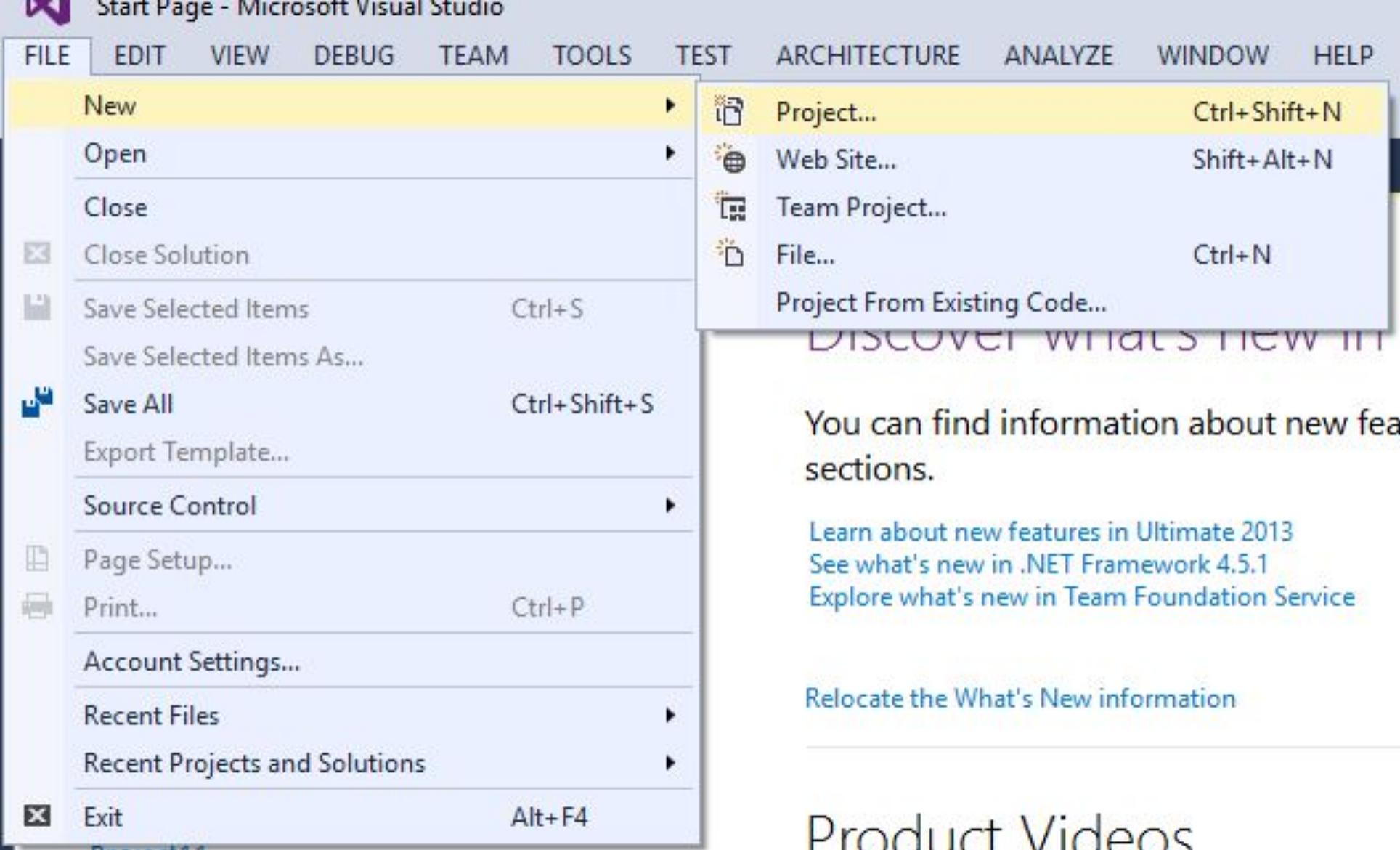
Компилятор действует почти также, как и интерпретатор, т.е. проверяет код построчно. Но если встречается ошибку, то не останавливается, а исследует код до конца, выявляя все последующие ошибки и выдавая о них сообщения. Кроме того компилятор формирует специальный объектный файл с расширением .OBJ. В этом файле хранится текст программы, переведенный на машинный язык.





# Microsoft Visual Studio 2017

2013



**File -> New -> Project или Ctrl+Shift+N**

- Recent
- Installed
  - Templates
    - Visual Basic
    - Visual C#
    - Visual C++**
      - Windows Store
      - ATL
      - CLR
      - General
      - MFC
      - Test
      - Win32
    - Visual F#
    - SQL Server
    - TypeScript
    - JavaScript
    - Python
    - Other Project Types
    - Modeling Projects
    - Samples
- Online

.NET Framework 4.5    Sort by: Default

	Win32 Console Application	Visual C++
	MFC Application	Visual C++
	Win32 Project	Visual C++
	Blank App (XAML)	Visual C++
	Empty Project	Visual C++
	Grid App (XAML)	Visual C++
	Hub App (XAML)	Visual C++
	Split App (XAML)	Visual C++
	DirectX App	Visual C++
	DirectX App (XAML)	Visual C++
	DLL (Windows Store apps)	Visual C++

[Click here to go online and find templates.](#)

Search Installed Templates (Ctrl+E)

**Type:** Visual C++

An empty project for creating a local application

Name:

Location:

Solution name:

Create directory for solution

Add to source control

Solution Explorer

Search Solution Explorer (Ctrl+ж)

Solution 'NameYouProjects' (1 project)

- NameYouProjects
  - External Dependencies
  - Header Files
  - Resource Files
  - Source Files

- New Item... Ctrl+Shift+A
- Existing Item... Shift+Alt+A
- New Filter
- Class...
- Resource...

- Add
  - Class Wizard... Ctrl+Shift+X
  - Scope to This
  - New Solution Explorer View
  - Cut Ctrl+X
  - Copy Ctrl+C
  - Paste Ctrl+V
  - Delete Del
  - Rename
  - Properties Alt+Enter

ПК  
М

# Рабочая область

# “Корневые файлы проекта”

# Errors list

The screenshot displays the Visual Studio IDE interface. The main editor window shows the source file `PageOne.cpp` with the following code:

```
#include <iostream>
using namespace std;

int main()
{
    cout >> i;
    return 0;
}
```

The Solution Explorer on the right shows the project structure for 'NameYouProjects', with 'PageOne.cpp' listed under 'Source Files'. The Properties window at the bottom right shows the properties for the 'main' function, including its name, file path, and data type 'int'.

The Error List at the bottom left shows two errors:

Description	File	Line	Column	Project
1 error C2059: syntax error: ';'	pageone.cpp	6	1	NameYouProjects
2 IntelliSense: expected an expression	PageOne.cpp	6	10	NameYouProjects

```
PageOne.cpp*  X  
(Global Scope)  
#include <iostream>
```

**1**

```
PageOne.cpp*  X  
(Global Scope)  
#include <iostream>  
using namespace std;
```

**2**

## Список ключевых слов языка C++:

asm	char	delete	extern
auto	class	do	float
break	const	double	for
case	continue	else	friend
catch	default	enum	goto
if	protected	static	typedef
inline	public	struct	union
int	register	switch	unsigned
long	return	template	virtual
new	short	this	void
operator	signed	throw	volatile
private	sizeof	try	while

```
#include <iostream>
using namespace std;
```

```
int main()
```

```
{
```

```
    //Тут код программы
```

```
}
```

```
#include <iostream>
using namespace std;
```

```
int main()
{
    cout << "Hello, world!!!";

    system("pause");
}
```

```
#include <string>
using namespace std;

int main()
{
    cout << "Hello, World!";
    system("pause");
}
```

- Build Solution Ctrl+Shift+B
- Rebuild Solution
- Clean Solution
- Run Code Analysis on Solution Alt+F11
- Build NameYouProjects
- Rebuild NameYouProjects...
- Clean NameYouProjects
- Project Only
- Profile Guided Optimization
- Batch Build...
- Configuration Manager...
- Compile

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Hello, World!";
    system("pause");
}
```

- Windows
- Graphics
- Start Debugging F5
- Start Without Debugging Ctrl+F5
- Attach Unity Debugger
- Attach to Process...
- Debug Installed App Package...
- Exceptions... Ctrl+Alt+E
- Performance and Diagnostics Alt+F2
- Step Into F11
- Step Over F10
- Toggle Breakpoint F9
- New Breakpoint
- Delete All Breakpoints Ctrl+Shift+F9
- Options and Settings...
- NameYouProjects Properties...

```
Hello, world!!!Для продолжения
```

## Escape sequence

`\b` - Delete the last output symbol

`\n` - Go to the beginning of a new line

`\t` - Go to the next tab position

`\\` - Output backslash

`\"` - Output double quote

`\'` - Output single quote

```
cout << "Hello, world!!! \n How are you? \t \\ norm \/ \"bad\" \n";
```

```
Hello, world!!!
```

```
How are you?    \ norm / "bad"
```

```
Для продолжения нажмите любую клавишу . . .
```

## 1 Задача.

**Вывод в консоль рассказ о себе используя  
Escapeпоследовательность .**

`\b` - Удаление последнего выведенного  
символа

`\n` - Перейти на начало новой строки

`\t` - Перейти к следующей позиции  
табуляции

`\\` - Вывести обратную черту

`\"` - Вывести двойную кавычку

`\'` - Вывести одинарную кавычку

```
tПривет всем.  
Меня зовут Александр.  
Мне 27 лет и я живу в городе "N".
```

**2 Задача.  
Вывод  
таблицы.**

```
|      Hello world      |  
|      How are you?    |  
|  Norm.   |   Bad    |  
|  Yes    |   No     |
```

```
int main()
{

    cout << "-----\n";
    cout << "|          Hello world          |\n";
    cout << "-----\n";
    cout << "|          How are you?          |\n";
    cout << "-----\n";
    cout << "|  Norm.    |    Bad    |\n";
    cout << "-----\n";
    cout << "|  Yes      |    No      |\n";
    cout << "-----\n";

    system("pause");
}
```

**Тип данных** — понятие, определяющее максимальный размер (в байтах) и тип информации, которая будет использоваться программой.

1. Числовые. 2. Символьные. 3.  
Логические.

# Числовые ТИПЫ С плавающей точкой

(3.14)

Пояснение	Тип	Размер в байтах
Описывает вещественные числа одинарной точности	float	4
Описывает вещественные числа двойной точности	double	8

## Целочисленные

Описывает целые числа	int	4	от -2147483648 до 2147483647
Описывает короткие целые числа	short	2	от -32768 до 32767
Описывает длинные целые числа	long	4	от -2147483648 до 2147483647
Описывает длинные целые числа	long long	8	от -9,223,372,036,854,775,808 до 9,223,372,036,854,775,807

# Символьный

Пояснение	Тип	Размер в байтах
Описывает символы	char	1

# Логический

## тип

Пояснение	Тип	Размер в байтах	Значения
Описывает логические значения	bool	1	true false

Тип\_данных имя\_переменной;

Тип\_данных  
имя\_переменной=значение;

const тип\_данных имя\_переменной=значение;

```
int main()
{
    int Age;
    int age = 7;
    const double val = 3.14;
    const int x = 5;
    char simbol = '$';
    bool YouAgeMore18 = true;
}
```

```
int main()
{
    int Age=100;
    int age = 7;
    const double val = 3.14;
    const int x = 5;
    char simbol = '$';
    bool YouAgeMore18 = true;

    cout << "Age =" << Age << "\n"; //тут будет ошибка
    cout << "age =" << age << "\n";
    cout << "val =" << val << "\n";
    cout << "x =" << x << "\n";
    cout << "Sim =" << simbol << "\n";
    cout << "You... =" << YouAgeMore18 << "\n";

    system("pause");
}
```

```
Age =100
age =7
val =3.14
x =5
Sim =$
You... =1
Для продолжения н
```

```
int main()
{
    int x = 5;
    int y = 10;
    int result = x + y;
    cout << result;

    system("pause");
}
```

Что будет в  
консоле?

```
int main()
{
    double x = 5;
    double y = 10;
    int result = x / y;
    cout << result;

    system("pause");
}
```

Что будет в  
консоле?

```
#include <iostream>
using namespace std;

int main()
{
    double Kilometers = 500;
    double meters = Kilometers*1000;

    cout << meters<<"\n";

    system("pause");
}
```

500000

Для продолже

```
#include <iostream>
using namespace std;

int main()
{
    int Kilometers = 500;
    int meters = Kilometers*1000;

    cout << meters<<"\n";

    int meters2 = 50;
    int Kilometers2 = meters2 / 1000;
    cout << Kilometers2 << "\n";

    system("pause");
}
```

500000

0

Для продолж

```
= int main()
{
    int Kilometers = 500;
    int meters = Kilometers*1000;

    cout << meters<<"\n";

    double meters2 = 50;
    double Kilometers2 = meters2 / 1000;
    cout << Kilometers2 << "\n";

    system("pause");
}
```

500000

0.05

Для продолжен

1с

3

Задача.

$$X=50 \quad Y=100$$

$$X*Y+5=?$$

$$X/Y=?$$

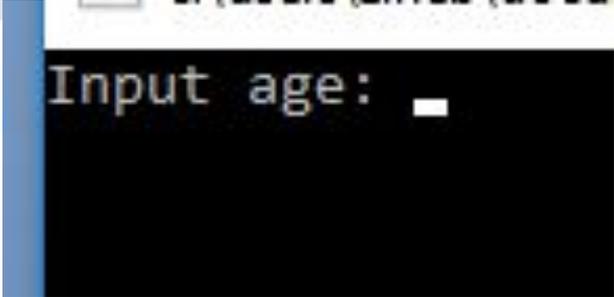
$$X-Y*6=?$$

$$X+Y-Y/5=?$$

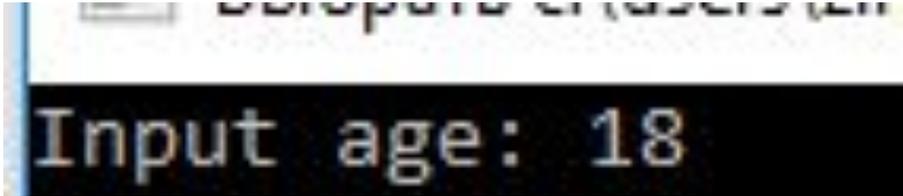
# Ввод данных

```
cin>>имя_переменной;
```

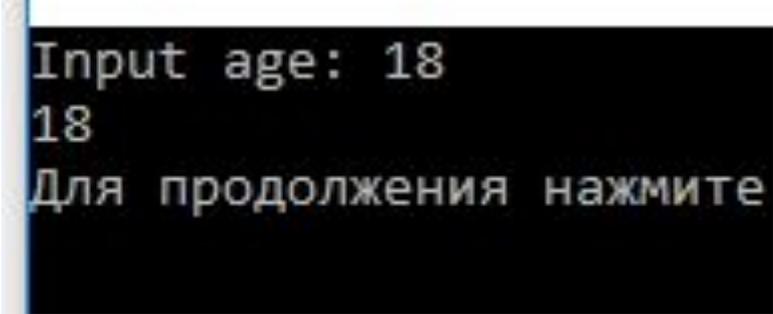
```
int main()  
{  
    int age;  
    cout << "Input age: ";  
    cin >> age;  
    cout << age<<" \n";  
}
```



Input age: \_



Input age: 18



Input age: 18  
18  
Для продолжения нажмите

```
int main()
{
    int age,number,years;
    cin >> age >> number >> years;

    system("pause");
}
```

18

895388811

2019

## 4 Задача.

- Введите с клавиатуры  $X$  и  $Y$
- Сложить  $X$  и  $Y$
- Вывести результат на консоль

# 11. Домашнее задание

1. Напишите программу, которая выводила бы на экран текстовую таблицу (используйте escape-последовательности):



```
C:\Windows\system32\cmd.exe
X | Y | X AND Y | X OR Y | NOT X
0 | 0 | 0 | 0 | 1
0 | 1 | 0 | 1 | 1
1 | 0 | 0 | 1 | 0
1 | 1 | 1 | 1 | 0
```

Рисунок 30

2. Дана диагональ телевизора в дюймах. Написать программу, определяющую эту же диагональ в сантиметрах (1 дюйм = 2.54 сантиметра).

Ниже представлен пример для проверки программы на корректность.

Входными данными являются:

Диагональ телевизора (inch) – > 42

Выходные данные:

Диагональ телевизора: 107 см.

3. Емкость аккумулятора смартфона составляет  $N$  мАч. Написать программу, определяющую минимальную емкость переносного зарядного устройства (*powerbank*), если необходимо 3 полных заряда смартфона.

Ниже представлен пример для проверки программы на корректность.

Входными данными являются:

Емкость аккумулятора смартфона (мАч.) – > 3000

Выходные данные:

Минимальная емкость: 9000 мАч.

4. За один день хомячок съедает  $K$  грамм корма. Написать программу, определяющую закупку корма в килограммах на 30 дней.

Ниже представлен пример для проверки программы на корректность.

Входными данными являются:

Расход корма за 1 день (г.) – > 20

Выходные данные:

Объем корма на 30 дней: 0.6 кг.