

---

# Язык Си. Структура программы на языке Си. Этапы компиляции. Препроцессор. Директивы препроцессора. Примеры.

---

C - язык программирования, созданный в 1972 году Деннисом Ритчи в США. На данный момент имеет 4 стандарта: C90(1990г), C99(1999г), C11(2011г) и C18(2017г).

## Конструктивные особенности:

- *Мощные управляющие структуры*
- *Высокая скорость*
- *Компактный код – небольшие по объёму программы*
- *Переносимость на другие компьютеры*

В качестве минусов можно выделить сложность в понимании указателей.

---

## Использование языка C:

- *Системное программирование*
  - *Встроенные системы*
  - *Разработка игр*
  - *Высокопроизводительные вычисления*
- 

## Этапы выполнения программы на C

1. *Редактирование* - Написание программы в редакторе.
  2. *Препроцессорная обработка* - Выполнение программы препроцессора.
  3. *Компиляция* - Компиляция исходного кода в объектный код.
  4. *Компоновка* - Связывание объектного кода с библиотеками.
  5. *Загрузка и исполнение* - Исполняемый код помещается в память и выполняется.
-

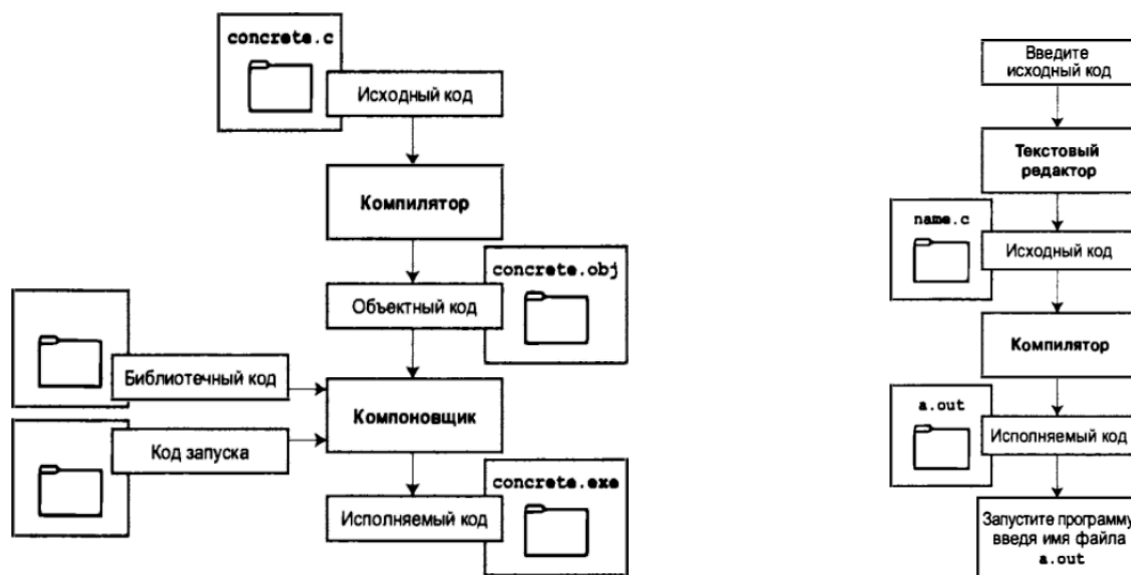
# Язык Си

Реализует **императивную парадигму** программирования: программа представляет собой набор директив предназначенных для исполнителя.

Ориентирован на **структурное программирование**. Данная парадигма предполагает следующее:

- программа состоит из трех базовых конструкций: *следование*, *ветвление*, *цикл*;
- базовые конструкции могут быть вложены друг в друга;
- логически законченные фрагменты программы оформляются в виде подпрограмм.

## Компилятор и Компоновщик



## Пример программы

```
#include <stdio.h>
int main(void)
{
    int num;                // определяем переменную именем num
```

```
    num = 1;           // присваиваем ей значение
    printf("Я простой "); // выводим на экран строки через функцию
printf()
    printf("Компьютер\n");
    printf("Моей любимой цифрой является %d, так как она первая.\n", num);
    return 0;
}
```

---

## Препроцессор

**Препроцессор** — это специальная программа, являющаяся частью компилятора языка Си. Она предназначена для предварительной обработки текста программы.

Препроцессор позволяет включать в текст программы файлы и вводить макроопределения.

Работа препроцессора осуществляется с помощью специальных директив (указаний). Они отмечаются знаком решетки #. По окончании строк, обозначающих директивы в языке Си, точку с запятой можно не ставить.

---

## Основные директивы препроцессора

- **#include** — подключает заданный файл.  
#include <stdio.h>  
#include "somefile.h"
  - **#define** — задаёт макроопределение (макрос) или символическую константу.  
#define AGE 18
  - **#undef** — отменяет определение.  
#undef AGE
- 

## Компиляция и запуск программы на С, используя gcc

**gcc** (GNU Compiler Collection) - Набор компиляторов для разных языков, включая си.

---

Компиляция простой программы:

`$ gcc program.c` - на выходе получаем файл `a.out` или `a.exe` в зависимости от ОС.

Запуск программы после компиляции:

`$ ./a.out` или `./a.exe`

---

Если в нашей программе, помимо основного файла, есть еще заголовочный(в формате `.h`), то делаем так:

1. `$ gcc header.h`

2. `$ gcc main.c`

После чего запускаем программу как обычно.

---

## Прочее

- `$ gcc program.c -o name` вместо `name` можно написать желаемое название файла после компиляции.
- `$ cd folder/cLearn` - не забываем переходить в папку с программой, которую хотим скомпилировать!