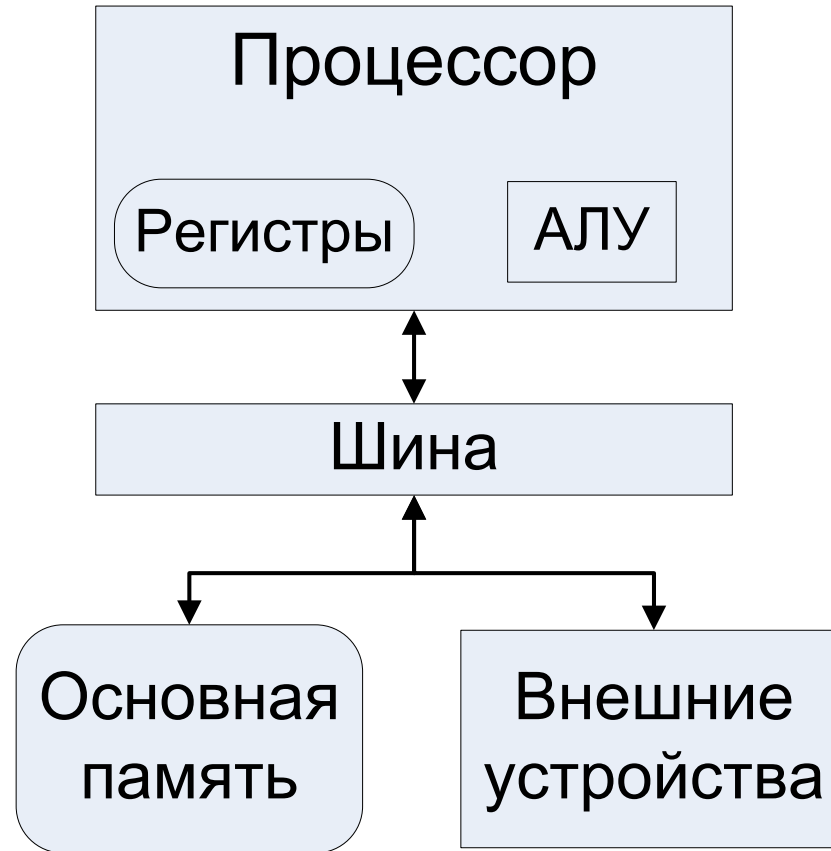


**Курс «Алгоритмы и алгоритмические языки»  
1 семестр 2015/2016**

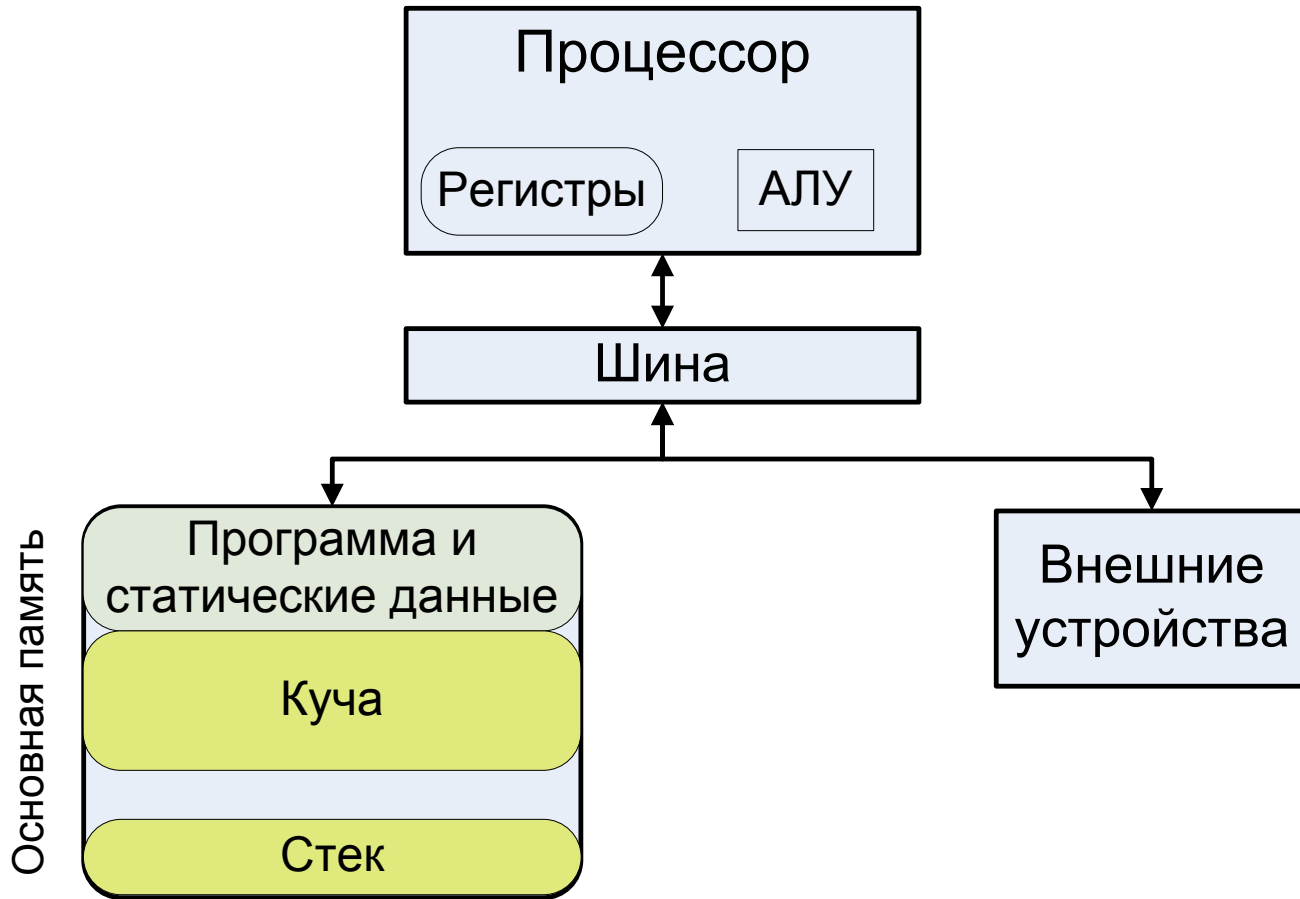
**Лекция 5**

# *Введение в язык программирования Си*

## *Схема простейшего компьютера*



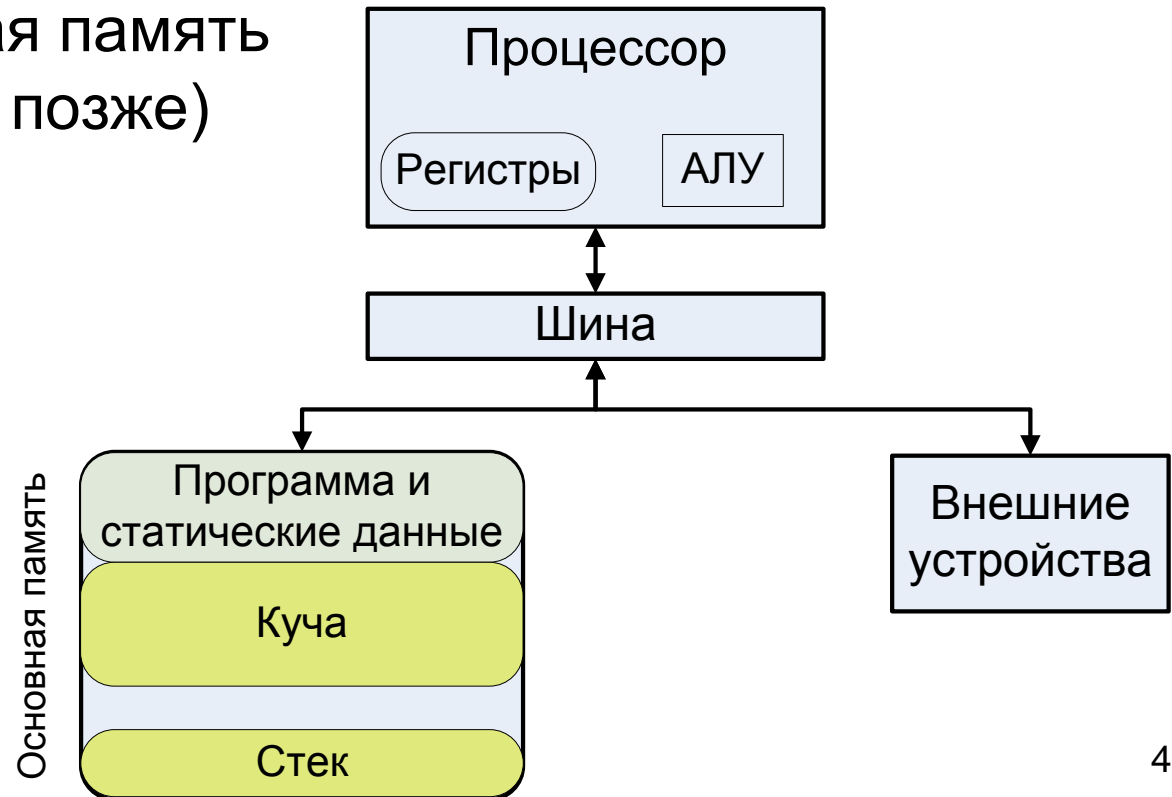
# Си-машина



# Си-машина

## Классы памяти

- ◆ Регистровые переменные
- ◆ Автоматические переменные
- ◆ Статические переменные
  
- ◆ Динамическая память (рассмотрим позже)



## Типы данных

- ◆ Базовые типы данных: **char** (*символьный*), **int** (*целый*), **float** (*с плавающей точкой*), **double** (*двойной точности*), **\_Complex** (C99, *комплексный*)
- ◆ Тип **void** (*без значения*)
- ◆ Модификаторы базовых типов: **signed**, **unsigned**, **long**, **short**, **long long** (C99)
  - ◆ к типу **int** применимы все модификаторы
  - ◆ к типу **char** – только **signed** и **unsigned**
  - ◆ к типу **double** – только **long** (C99)

## Типы данных

- ◆ Представление целых чисел: позиционная двоичная система
  - ◆ Байты в представлении числа идут подряд
  - ◆ Порядок байт не гарантируется, то есть зависит от аппаратуры (big/little endian)
  - ◆ Порядок бит в байте также не гарантируется (и его может быть невозможно узнать)
  - ◆ Отрицательные числа *часто* представляются в дополнительном коде ( $n$  бит):
    - самый значащий бит ( $n-1$ ) является знаковым
    - биты от 0 до  $n-2$  – значения
    - положительные значения – как обычно
    - отрицательные значения:  $2^n - |x|$

## Типы данных

- ◆ `sizeof` – размер типа (любого объекта типа)
  - ◆ `int x -> sizeof(x) == sizeof(int)`
  - ◆ Файл `limits.h` задает минимальные и максимальные значения целых типов
  - ◆
    - `sizeof(char) == 1`
    - `sizeof(short) ≥ 2`
    - `sizeof(int) ≥ 2`
    - `sizeof(long) ≥ 4`
    - `sizeof(long long) ≥ 8`
  - ◆ Файл `inttypes.h` задает знаковые и беззнаковые целые типы фиксированных размеров (8, 16, 32, 64 бита)

## Типы данных

- ◆ Тип `_Bool` (C99, значения 0/1, целый беззнаковый)
  - ◆ Необходимо включить `stdbool.h` для объявлений `bool`, `true`, `false`
- ◆ Тип `_Complex` (C99, `float/double/long double`)
  - ◆ Необходимо включить `complex.h` для объявлений `complex`, `I` и т.п.
  - ◆ Тип `_Imaginary` (C99) является необязательным



## ***Переменные***

- ◆ **Переменная = тип + имя + *значение***  
Каждая переменная является *объектом* программы
- ◆ **Ключевые слова** (C89 – 32, C99 – C89 + 5) не могут быть именами переменных
- ◆ **Объявление переменной:**  
*тип список\_переменных*  
Можно задать класс памяти и начальное значение переменной

## ***Область действия переменной***

- ◆ **Переменная может быть объявлена:**
  - (1) внутри функции или блока (локальная);
  - (2) в объявлении функции (параметр функции);
  - (3) вне всех функций (глобальная).
- ◆ **Область действия (видимости)**
  - ◆ локальной переменной – блок, в котором она объявлена (C99 – начиная со строки объявления)
  - ◆ глобальной переменной – программный файл, начиная со строки объявления
- ◆ **В одной области действия нельзя объявлять более одной переменной с одним и тем же именем**

## ***Область действия переменной и классы памяти***

```
#include <stdio.h>
int count;                                /* глобальная переменная */
void func (void)
{
    int count;                            /* автоматическая переменная */
    count = count - 2;
}
static int mult = 0;                      /* статическая переменная */
int sum (int x, int y)
{
    count++;
    return (x + y) * (++mult);
}
int main (void)
{
    register int s = 0;                   /* регистровая переменная */
    count = 0;
    s += sum (5, 7);
    s += sum (9, 4);
    func ();
    printf ("Сумма равна %d, вызвали функцию %d раз\n", s, count);
    return 0;
}
```