

Содержание

классы хранения	3
<i>auto</i>	3
<i>extern</i>	3
<i>static</i>	4
<i>register</i>	4

классы хранения

В Си существует четыре основных класса хранения: auto, extern, static, register.

auto

Класс хранения auto является классом по умолчанию для всех **локальных**¹⁾ переменных.

- Область видимости: действие переменной ограничена блоком, в котором она объявлена (локальная видимость).
- Жизненный цикл: переменная создается при входе в блок и уничтожается при выходе из него.
- Начальное значение: если переменная не инициализирована явно, то она содержит «мусорное» неопределенное значение.

```
void myFunction() {  
    int a; //То же самое, что и "auto int a;"  
    a = 10;  
    // a доступна только внутри myFunction  
}
```

extern

Используется для объявления глобальной переменной или функции, которая определена в другом файле.

- Область видимости: глобальная (доступна во всем проекте).
- Жизненный цикл: существует в течение всего времени выполнения программы.
- Начальное значение: определяется в месте ее инициализации. Если глобальная переменная не инициализирована явно, она получает значение 0.

```
// main.c  
#include <stdio.h>  
  
char message[] = "Hello";  
  
void print() {  
    printf("Hello from extern function\n");  
}  
int sum(int x, int y) {  
    return x + y;  
}
```

```
//app.c  
#include <stdio.h>
```

```
extern char message[]; // переменная message из файла main.c
void print(void); // функция print из main.c
int sum(int, int); // функция sum из main.c

int main(void){
    // выводим на консоль внешнюю переменную message
    printf("%s \n", message);

    // вызываем внешнюю функцию print
    print();

    // вызываем другую внешнюю функцию - sum
    printf("4 + 5 = %d \n", sum(5, 4));

    return 0;
}
```

static

Ключевое слово `static` имеет два основных применения в зависимости от контекста: для локальных переменных и для глобальных переменных/функций.

Когда `static` применяется к **локальной** переменной, он изменяет ее жизненный цикл, сохраняя ее значение между вызовами функции. Основные характеристики:

- Область видимости: Локальная, как у `auto` переменных.
- Жизненный цикл: Существует в течение всего времени выполнения программы, а не только внутри функции.
- Начальное значение: Инициализируется только один раз при первом вызове функции. Если не инициализирована явно, получает значение 0.

Когда `static` применяется к **глобальной** переменной или функции, он ограничивает их область видимости текущим файлом. Это означает, что к такой переменной или функции нельзя будет обратиться из другого файла с помощью `extern`. В этом случае действуют следующие характеристики:

- Область видимости: действие переменной/функции ограничено файлом, в котором они объявлены.
- Жизненный цикл: переменная/функция существует в течение всего времени выполнения программы.

register

Класс хранения `register` является рекомендацией для компилятора поместить переменную не в оперативную память, а в регистр процессора. Компилятор может проигнорировать это указание, если нет свободных регистров или если тип данных не подходит для хранения в регистре.



Однако в современных компиляторах register почти не используется, так как они самостоятельно выполняют очень сложную оптимизацию и часто лучше программиста решают, какие переменные стоит помещать в регистры.

- Область видимости: Локальная.
- Жизненный цикл: Локальный.



К register-переменной нельзя применить оператор взятия адреса &, так как у нее нет адреса в памяти.

1)

Это означает, что если вы объявляете переменную внутри функции или блока без указания класса хранения, она автоматически принимает тип хранения auto.

From:
<https://wiki.radi0.cc/> - radi0wiki

Permanent link:
https://wiki.radi0.cc/c:c_ultimate_guide:storage_classes

Last update: **2025/11/09 12:07**

