24. Классы памяти.

auto
класс автоматических переменных, создаваемых при вызове функции и удаляемых после выхода из неё.
Применим только к переменным, определенным в блоке.

локальное время жизни,
видимость в пределах блока,
не имеет связывания.

register
компилятору предъявляется пожелание о том, чтобы переменная использовала какой-нибудь регистр процессора для хранения. Как правило, компилятор игнорирует это резервное слово и способен сам решать, какой переменной можно выделять регистр процессора.
Используется только для переменных, определенных в блоке.
Задает:
локальное время жизни,
видимость в блоке
отсутствие связывания.
К переменным с классом памяти register нельзя применять операцию получения адреса &.

static
класс статических переменных, создаётся при первом вызове функции, а удаляется только при завершении работы программы.
Компилятор хранит значения таких переменных от одного вызова функции до другого.

Для переменной вне какого-либо блока:
static изменяет связывание этой переменной на внутреннее,
глобальное время жизни,
обсласть видимости в пределах файла
Для переменной в блоке,
static изменяет время жизни с автоматического на глобальное,
область видимости в пределах блока,
отсутствие связывания.

```
void f(void)
{
        static int j;
        ...
}
```

Такая переменная сохраняет свое значение после выхода из блока.
Инициализируется только один раз.
Если функция вызывается рекурсивно, это порождает новый набор локальных переменных, в то время как статическая переменная разделяется между всеми вызовами.

extern
класс внешних переменных, память под них не выделяется. Это означает объявление переменной, которая может быть объявлена или в текущем, или в некотором внешнем файле.

```
extern int i;
//Глобальное время жизни, файловая область видимости, связывание непонятное

{
        extern int i;
        ...
        //Глобальное время жизни, видимость в блоке, связывание непонятное
```

Связывание определяется по определению переменной.

Объявлений (extern int number;) может быть сколько угодно.
Определение (int number;) должно быть только одно.
Объявления и определение должны быть одинакового типа.
НО extern int number = 5; // определение

По умолчанию считается:
если переменная объявлена в теле функции или в блоке без спецификатора класса памяти, по умолчанию он равен auto;
если переменная объявлена вне всех функций, она считается внешней и может быть использована в любом смежном файле, в т. ч. и текущем;
все функции внешние.

ФУНКЦИИ:
К функциям могут применяться классы памяти static и extern.

extern int f(int i);
f имеет внешнее связывание. Может вызываться из других файлов.

static int g(int i);
g имеет внутренне связывание. Из других файлов вызываться не может.

int h(int i);
h имеет внешнее связывание (по умолчанию). Может вызываться из других файлов.

Аргументы функций по умолчанию имеют класс памяти auto.
Единственный другой класс памяти, который может использоваться с параметрами функций, - register.