Программные объекты. Продолжительность хранения объекта. Классы памяти. Примеры.

Объект - описание некоторого участка физической памяти определенного размера, который может хранить одно или более значений. В зависимости от момента, объект может содержать или не содержать сохраненного значения.
Каждый объект имеет:

- **Имя** (идентификатор), если он не является анонимным (например, `int x;`).
- **Тип данных** (например, `int`, `float`, `char`).
- **Адрес в памяти**.

```
int initVal = 0, val[5];
//4byte           20byte
```

L-значение - выражение, которое обозначает объект. Чаще всего L-значение представлено идентификатором или выражением.

```
int val = 10;
int *ptr = &val;
```

Модифицируемое L-значение - L-значение, которое может быть использовано для изменения значения внутри объекта.

```
const char * text = "Text";
```

Продолжительность хранения (ПХ) - регламентирует постоянство объектов, доступных через эти идентификаторы.
Выделяют:

- статическое
- потоковое
- автоматическое
- выделенное

Статическая

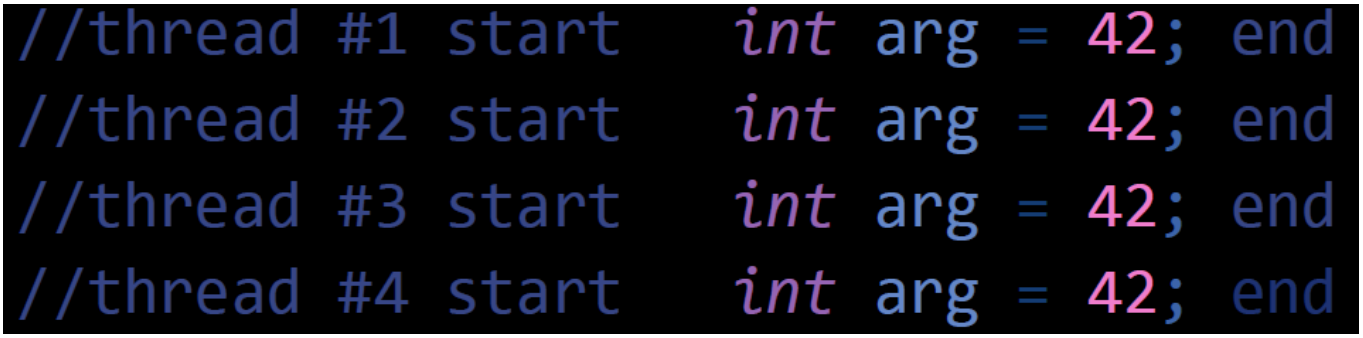
- Объекты существуют на протяжении всей работы программы.
- Инициализируются один раз (при первом вызове или запуске программы).

```
int units =0;
int main (void ){
    /* code */
    return 0;
}
```

Потоковая

- Объект существует ровно с момента его объявления и до завершения потока.
- Поток - часть программы, которая выполняется внутри процесса

```
void *threadFunc(void) {
    int arg = 42;
    return NULL;}
for (size_t i = 0; i < 4; i++)
    pthread_create(&thread, NULL, thread_function, &arg);
```



Автоматическая

- Объект существует в пределах блоках или в пределах функции

```
for (size_t i = 0; i < 4; i++){
    int otherArg =0;
}
void someFunct (void){
    int arg = 0;
}
```

- Если возникает потребность изменить автоматическую ПХ на статическую следует использовать ключевое слово static

```
void someFunct (void){
    int arg = 0;
```

```
    static int otherArg = 0;
}
```

Выделенное

- Память выделяется и освобождается вручную с помощью `malloc`, `calloc`, `realloc` и `free`.

```
int *arr = malloc(10 * sizeof(int)); // Динамический массив
free(arr); // Освобождение памяти
```

Классы памяти

Класс памяти определяет область видимости, связывание и продолжительность хранения объекта.

Класс хранения	Продолжительность хранения	Область видимости	Связывание	Объявление
Автоматический	Автоматическая	В пределах блока	Нет	В блоке
Регистровый	Автоматическая	В пределах блока	Нет	В блоке с указанием ключевого слова `register`
Статический с внешним связыванием	Статическая	В пределах файла	Внешнее	За рамками всех функций
Статический с внутренним связыванием	Статическая	В пределах файла	Внутреннее	За рамками всех функций с указанием ключевого слова `static`
Статический без связывания	Статическая	В пределах файла	Нет	В блоке с указанием ключевого слова `static`

В Си используются следующие спецификаторы:

`auto`

- Любая переменная, объявленная в блоке или заголовке функции, относится к автоматическому классу хранения.
- Указывает на автоматическую продолжительность (по умолчанию для локальных переменных).
- Пример:

```
auto int x = 5;
```

`register`

- Си дает возможность разместить значение переменной в регистровой памяти процессора, при благоприятных условиях. У такой переменной нельзя взять адрес

```
register int quick;
```

`static`

- Для **локальных переменных**: сохраняет значение между вызовами функции.
- Для **глобальных переменных**: ограничивает видимость файлом (внутреннее связывание).

```
static int fileLocal = 42;
```

`extern`

- Указывает, что переменная объявлена в другом файле (внешнее связывание).

```
extern int globalVar; // Определена в другом файле
```