

## 10. КЛАССЫ ПАМЯТИ

В C++ существуют 3 класса памяти, или хранения:

- 1) статическая память – статические данные, размещаемые в сегменте данных;
- 2) автоматические данные, размещаемые в специальном стеке (сегмент стека) или, как частный случай, в регистрах процессора;
- 3) динамические данные, явно размещаемые в динамической памяти с помощью операций **new** и **delete**.

Статические объекты существуют в течение всего времени выполнения программы. К ним относятся глобальные и локальные переменные, объявленные со служебным словом **static**:

```
int i = 3, j; // глобальные переменные. Класс памяти – static; void
main( ){
    int a; // Автоматическая переменная;
    static float b [1000], c = 2.3; // статические переменные;
    ...
}
int f(){int d; // автоматическая переменная; static
int m = 2, k; // статические переменные m, k.
...
}
```

Статические и глобальные переменные, если они не инициализированы явно, инициализируются нулевыми значениями. В **любом случае** инициализация статических переменных осуществляется только **один раз**.

Локальные переменные, не объявленные как **static**, являются автоматическими. Такие объекты начинают свое существование при объявлении его имени в блоке и заканчивают его при завершении этого блока. Если автоматический объект явно не инициализирован, то его значение до присвоения не определено.

```
void f();
void main(){
    for (int i = 3; i > 0; i --)f();
} void f(){
    static int i;
    int j = 0;
```

```
cout << "i =" << i++ << " j =" << j++ << "\n"; }
```

Здесь будет напечатано:

i = 0	j = 0
i = 1	j = 0
i = 2	j = 0

Заметим, что если служебное слово `static` применено к глобальной переменной или к имени функции, то оно имеет другой смысл. В этом случае и глобальная переменная, и функция становятся видимыми только в пределах файла, где они определены, и невидимыми из других файлов.