

Кроме переменных и методов, которые относятся непосредственно к объекту, C++ позволяет определять переменные и методы, которые относятся непосредственно к классу или иначе говоря статические члены класса. Статические переменные и методы относят в целом ко всему классу. Для их определения используется ключевое слово **static**.

## Статические поля

Статические переменные обычно применяются для хранения значений, специфичных для класса, для всех объектов класса в целом. То есть статические поля хранят состояние всего класса. Статическая переменная определяется только один раз и будет существовать, даже если объекты класса не были созданы.

Показательным примером статических переменных являются различные счетчики.

```
3 class Person
4 {
5 public:
6     Person(std::string p_name, unsigned p_age)
7     {
8         ++count; // при создании нового объекта увеличиваем счетчик
9         name = p_name;
10        age = p_age;
11    }
12    void print_count()
13    {
14        std::cout << "Created " << count << " objects" << std::endl;
15    }
16 private:
17     std::string name;
18     unsigned age;
19     static inline unsigned count{}; // статическое поле - счетчик объектов Person
20 }
```

```
22 int main()
23 {
24     Person tom{"Tom", 38};
25     Person bob{"Bob", 42};
26     Person sam{"Sam", 25};
27     tom.print_count();
28     bob.print_count();
29     sam.print_count();
30 }
```

в классе Person определена статическая переменная **count** с помощью ключевого слова **static**

При создании каждого нового объекта в конструкторе увеличиваем счетчик на единицу:

```
1 Person(std::string p_name, unsigned p_age)
2 {
3     ++count; // при создании нового объекта увеличиваем счетчик
```

Для теста в функции main создаем три объекта Person и затем у каждого вызываем функцию print\_count:

```
1 tom.print_count();
2 bob.print_count();
3 sam.print_count();
```

Но поскольку переменная count статическая и относится ко всему классу в целом и не зависит от конкретного объекта, во всех трех случаях будет выведено число 3

## Статические функции

Статические функции также принадлежат классу в целом и не зависят от любого отдельного объекта класса.

Статические функции-члены используются для работы со статическими переменными.

Для определения статической функции перед ней указывается ключевое слово **static**

К подобным функциям также можно обращаться через имя объекта `tom.print_count();`

C++ также поддерживает и другой синтаксис:

`класс::член_класса`

`Person::print_count();`

```
1 #include <iostream>
2
3 class Person
4 {
5 public:
6     Person(std::string p_name, unsigned p_age)
7     {
8         ++count; // при создании нового объекта увеличиваем счетчик
9         name = p_name;
10        age = p_age;
11    }
12    // статическая функция
13    static void print_count()
14    {
15        std::cout << "Created " << count << " objects" << std::endl;
16    }
17 private:
18     std::string name;
19     unsigned age;
20     static inline unsigned count{}; // статическое поле - счетчик объектов Person
21 };
22
23 int main()
24 {
25     Person tom{"Tom", 38};
26     Person bob{"Bob", 42};
27     Person sam{"Sam", 25};
28     tom.print_count(); // Created 3 objects
29 }
```