

Добрый вечер, Михаил Иванович!

Эссе на тему: «Что такое “template” на C++?»

Шабло́ны (template) — средство языка C++, предназначенное для кодирования обобщённых алгоритмов, без привязки к некоторым параметрам (например, типам данных, размерам буферов, значениям по умолчанию). В C++ возможно создание шаблонов функций и классов.

Шаблоны позволяют создавать параметризованные классы и функции. Параметром может быть любой тип или значение одного из допустимых типов (целое число, enum, указатель на любой объект с глобально доступным именем, ссылка). Например, нам нужен какой-то класс:

```
Class SomeClass{  
  
    int SomeValue;  
  
    int SomeArray[20];  
  
};
```

Для одной конкретной цели мы можем использовать этот класс. Но, вдруг, цель немного изменилась, и нужен еще один класс. Теперь нужно 30 элементов массива и вещественный тип элементов. Тогда мы можем абстрагироваться от конкретных типов и использовать шаблоны с параметрами. Синтаксис: в начале перед объявлением класса декларируем шаблон, то есть template, и укажем параметры в угловых скобках.

```
template <int ArrayLength, typename SomeValueType> class SomeClass{  
  
    SomeValueType SomeValue;  
  
    SomeValueType SomeArray[ArrayLength];  
  
};
```

Тогда для первого случая (с целочисленным SomeValue и SomeArray в 20 элементов):

```
SomeClass <20, int>
```

```
SomeVariable;
```

для второго:

```
SomeClass <30, double>
```

```
SomeVariable2;
```

Хотя шаблоны предоставляют краткую форму записи участка кода, на самом деле их использование не сокращает исполняемый код, так как для каждого набора параметров компилятор создаёт отдельный экземпляр функции или класса. Как следствие, исчезает возможность совместного использования скомпилированного кода в рамках разделяемых библиотек.