

Шаблоны являются основой для универсального программирования на C++. Как строго типизированный язык C++ требует, чтобы все переменные имели определенный тип, явно объявленный программистом или выведенный компилятором. Однако многие структуры данных и алгоритмы выглядят одинаково независимо от типа, с каким типом они работают. Шаблоны позволяют определять операции класса или функции и позволяют пользователю указать, с какими конкретными типами должны работать эти операции. Шаблон — это конструкция, которая создает обычный тип или функцию во время компиляции на основе аргументов, которые пользователь предоставляет для параметров шаблона. В качестве аргумента типа можно использовать любой встроенный или определяемый пользователем тип. Например, можно использовать `std::vector` в стандартной библиотеке для хранения переменных типа `int`, `double`, `std::string`, `class`, и т. д. Основное ограничение при использовании шаблонов заключается в том, что аргумент типа должен поддерживать любые операции, применяемые к параметрам типа.

В некоторых случаях шаблон не может или не рекомендуется определять точно такой же код для любого типа. Например, можно определить путь к коду, который будет выполняться, только если аргумент типа является указателем, `std::wstring` или типом, производным от определенного базового класса. В таких случаях можно определить *специализацию* шаблона для конкретного типа. Когда пользователь создает экземпляр шаблона с этим типом, компилятор использует специализацию для создания класса, а для всех остальных типов компилятор выбирает более общий шаблон. Специализации, в которых все параметры являются специализированными, являются *полными специализациями*. Если только некоторые параметры являются специализированными, это называется *частичной специализацией*. Шаблон может иметь любое количество специализаций, если каждый параметр специализированного типа уникален. Только шаблоны классов могут быть частично специализированными. Все полные и частичные специализации шаблона должны быть объявлены в том же пространстве имен, что и исходный шаблон.

