

Что такое шаблон и какие разновидности шаблонов существуют?

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

Шаблоны функций - это обобщенное описание поведения функций, которые могут вызываться для объектов разных типов (фактически, задается семейство функций). По описанию шаблон функции похож на обычную функцию: разница в том, что некоторые элементы не определены (типы, константы) и являются параметризованными.

Шаблоны классов - обобщенное описание пользовательского типа, в котором могут быть параметризованы атрибуты и операции типа. Представляют собой конструкции, по которым могут быть сгенерированы действительные классы путём подстановки вместо параметров конкретных аргументов.

Также на семинаре говорили про шаблоны псевдонимов и переменных.

Каким образом осуществляется двухэтапная трансляция шаблона?

- 1) Без подстановки конкретного типа
выявляются ошибки: отсутствие ;
- 2) Проверки с подстановкой конкретного типа
ошибки: нет оператора меньше/больше для сравнения
проверки делаются компилятором

Из-за наличие двухэтапной проверки нельзя разбивать определение и объявление по разным файлам, так как проверка происходит до линковки.

Что предпочитает компилятор при перегрузке шаблона функции?

Если программист сам перегрузил шаблонную функцию, то компилятор будет использовать именно её. Он не будет создавать новый экземпляр шаблона с таким типом

В чём заключается особенность инстанцирования шаблонов классов?

Инстанцирование выполняется только для тех функций-членов, которые используются в программе, не нужное компилятор просто не создает

Когда необходимы полная и частичная специализации шаблонов?

Частичная специализация может быть сделана только для класса. Она используется в том случае, если отдельные функции члены должны работать по-другому для определенного типа данных, а остальные функции-члена подходят и без переработки. Полная специализация может быть сделана как для классов, так и для функций. Это удобнее чем перегружать функцию или класс без шаблона, так как при необходимости компилятор сам решит, создавать ему такую версию или нет.