



Дз 9

1. Двухэтапная трансляция шаблонов что такое.

Двухэтапная трансляция шаблонов — это процесс, при котором компилятор выполняет два этапа для обработки шаблонных классов и функций. Двухэтапная трансляция используется в тех случаях, когда мы пытаемся инстанцировать шаблон для типа, который не поддерживает все используемые в нем операции (приводит к ошибке времени компиляции).

На первом этапе компилятор анализирует шаблонный код и создает специализированный код для каждого типа данных, который будет использоваться в шаблоне. Этот код не генерируется до тех пор, пока не будет вызвана конкретная специализация шаблона.

На втором этапе компилятор генерирует конкретный код для каждой специализации шаблона, который затем может быть скомпилирован и выполнен.

2. Как происходит разрешение перегрузки как шаблонных, так и обычных функций с одинаковыми именами.

Разрешение перегрузки как шаблонных, так и обычных функций с одинаковыми именами происходит за три шага. При таких преобразованиях будет действовать контроль однозначности интерпретации:

- Если у двух функций одинаковые имена, но разные типы аргументов, то компилятор выбирает ту функцию, которая соответствует переданным аргументам.
- Если две функции имеют одинаковые имена и типы аргументов, но различаются только типом возвращаемого значения или количеством аргументов, то компилятор считает их перегрузками и выбирает ту функцию, которая соответствует переданным аргументам.

- Если две функции имеют одинаковые имена и типы аргументов, и не могут быть отличены по типу возвращаемого значения или количеству аргументов, то компилятор выдаст ошибку "ambiguous call" (неоднозначный вызов) при попытке вызвать такую функцию.

если функция найдена по этим правилам, вызвать ее, иначе вызов является ошибкой.

3. *Что такое специализация шаблона. Для чего необходима?*

Специализация шаблона позволяет настраивать код шаблона для заданного набора аргументов шаблона. Специализация шаблона делится на частичную (позволяет частично настроить код шаблона для определенных типов) и полную (указываются значения для всех параметров шаблона).

4. *Для чего нужны static\dynamic_cast. Преимущества static_cast над "(type)value" конструкцией*

Static_cast и dynamic_cast - это операторы языка C++, которые используются для преобразования типов данных.

Static_cast используется для выполнения явного преобразования типов данных во время компиляции.

Dynamic_cast выполняет проверку на корректность преобразования типов данных во время выполнения программы. Он используется для преобразования указателей и ссылок на базовые классы в указатели и ссылки на производные классы. Если преобразование невозможно, то dynamic_cast вернет нулевой указатель или бросит исключение bad_cast.

Преимущества static_cast над "(type)value" конструкцией заключаются в том, что static_cast выполняет проверку на корректность преобразования типов данных во время компиляции, а "(type)value" конструкция не выполняет такой проверки и может привести к ошибкам во время выполнения программы.

5. *Чем обусловлен, и как можно обойти запрет на частичную специализацию функций.*

Запрет на частичную специализацию функций обусловлен тем, что язык не предоставляет механизма для этого, в отличие от шаблонов классов, где частичная специализация поддерживается. Для того, чтобы обойти этот запрет, можно использовать перегрузку функций и функциональные объекты.