

Ответы на контрольные вопросы:

1 вопрос: 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 языке c++ различают:

→ Шаблоны функций - обобщенное описание поведения функций, которые могут вызываться для объектов разных типов.

→ Шаблоны классов - обобщенное описание пользовательского типа, в котором могут быть параметризованы атрибуты и операции типа.

2 вопрос: Каким образом осуществляется двухэтапная трансляция шаблона

Шаблоны "компилируются" в два этапа:

1. Во время определения (definition time) код шаблона проверяется на корректность без инстанцирования, с игнорированием параметров шаблонов.

Этот процесс включает:

- выявление таких синтаксических ошибок, как отсутствующие точки с запятой;
- выявление применения неизвестных имен (имен типов, функций и т.гг), которые не зависят от параметров шаблона;
- выполнение проверок статических утверждений, не зависящих от параметров шаблонов.

2. Во время инстанцирования код шаблона вновь проверяется на корректность. Таким образом, все части, которые зависят от параметров шаблонов, подвергаются двойной проверке.

Проверка имен, выполняемая дважды, называется двухэтапным (двухфазным) поиском (two-phase, lookup)

3 вопрос: Что предпочитает компилятор при перегрузке шаблона функции?

Более специализированный шаблон функции.

4 вопрос: В чём заключается особенность инстанцирования шаблонов классов?

Инстанцирование шаблона – это генерация кода функции или класса по шаблону для конкретных параметров. Различают неявное инстанцирование, которое происходит при вызове функции или создании объекта класса, и явное инстанцирование с помощью резервированного слова `template`. Инстанцирование можно делать только в точке программы, где доступна реализация шаблона функции или методов шаблонного класса.

Если мы реализуем шаблон в заголовочном файле, то компилятор сможет выполнить инстанцирование шаблона для любых аргументов, однако есть ещё второй способ: мы можем только объявить шаблон в заголовочном файле, а реализовать его уже в главном, тогда будет применено явное инстанцирование с помощью `template`.

5 вопрос: Когда необходимы полная или частичная специализации шаблона?

Вообще сначала надо разобраться, что такое специализация: допустим, у нас есть шаблон функции или класса, всё здорово, он для всех типов работает одинаково. Теперь нам вдруг понадобилось, чтобы для какого-то одного типа эта функция или класс работали по-другому: супер, тогда пишем специализацию, то есть *выделяем* этот тип и теперь можем писать специальную реализацию для этого типа.

→ Полная специализация необходима, если мы хотим, чтобы методы класса или функции для какого-то особенного типа работали по-другому. При этом все параметры шаблона должны быть явно определены.

→ Частичную специализацию можно применять только для классов, когда явно определены лишь

некоторые, но не все, параметры шаблона.