1. В чем смысл использования шаблонов?

Шаблоны классов, так же как и шаблоны функций, поддерживают в C++ парадигму

обобщенного программирования, то есть программирования с использованием типов в

качестве параметров.

Шаблоны функций предназначены для того чтобы отделить алгоритм обработки

данных, независимый от конкретных типов данных, с которыми он работает, передавая тип в качестве параметра. Шаблоны классов предоставляют аналогичную возможность, позволяя создавать

параметризованные классы.

Шаблон класса (иначе параметризованный класс) используется для построения

родового класса. Создавая родовой класс, вы создаете целое семейство родственных классов, которые можно применять к любому типу данных.

2. Синтаксис шаблонов функций?

template<class тип\_данных>

тип\_воз\_значения имя\_функции(список\_параметров){тело\_функции}

3. Синтаксис шаблонов классов?

template <class тип\_данных> class имя\_класса {...}; или

template < typename Т> class point {/\*... \*/};

4. Определить параметризованную функцию сортировки массива методом обмена.

5. Определить шаблон класса “вектор” - одномерный массив.

6. Что такое параметры шаблона функции?

Параметрами шаблонов могут быть абстрактные типы или переменные встроенных

типов, таких как int.

Первый вид параметров в котором при инстанцировании на место параметров шаблонов подставляются аргументы либо встроенных типов, либо типов, определенных программистом.

Второй вид используется, когда для шаблона предусматривается его настройка

некоторой константой.

7. Перечислите основные свойства параметров шаблона функции.

- Имена параметров шаблона должны быть уникальными во всем определении

шаблона.

- Список параметров шаблона не может быть пустым.

- В списке параметров шаблона может быть несколько параметров, и каждому из

них должно предшествовать ключевое слово class.

- Имя параметра шаблона имеет все права имени типа в определенной шаблоном

функции.

- Определенная с помощью шаблона функция может иметь любое количество не

параметризованных формальных параметров. Может быть не параметризованное и

возвращаемое функцией значение.

- В списке параметров прототипа шаблона имена параметров не обязаны

совпадать с именами тех же параметров в определении шаблона.

- При конкретизации параметризованной функции необходимо, чтобы при вызове

функции типы фактических параметров, соответствующие одинаково параметризованным

формальным параметрам, были одинаковы.

8. Можно ли перегружать параметризованные функции?

Перегружать параметризованные функции можно, меняя тип данных параметра, либо меняя параметры местами, в том случае, если они разного типа, а также можно выполнять перегрузку функции добавляя или исключая параметры.

9. Перечислите основные свойства параметризованных классов.

- Компонентные функции параметризованного класса автоматически являются

параметризованными. Их не обязательно объявлять как параметризованные с помощью

template.

- Дружественные функции, которые описываются в параметризованном классе, не

являются автоматически параметризованными функциями, т.е. по умолчанию такие функции

являются дружественными для всех классов, которые организуются по данному шаблону.

- Если friend-функция содержит в своем описании параметр типа

параметризованного класса, то для каждого созданного по данному шаблону класса имеется

собственная friend-функция.

- В рамках параметризованного класса нельзя определить friend-шаблоны

(дружественные параметризованные классы).

- С одной стороны, шаблоны могут быть производными (наследоваться) как от

шаблонов, так и от обычных классов, с другой стороны, они могут использоваться в качестве

базовых для других шаблонов или классов.

- Шаблоны функций, которые являются членами классов, нельзя описывать как

virtual.

- Локальные классы не могут содержать шаблоны в качестве своих элементов.

10. Может ли быть пустым список параметров шаблона? Объясните.

Список параметров шаблона функций не может быть пустым, так как при этом теряется возможность параметризации и шаблон функций становится обычным определением конкретной функции.

11. Как вызвать параметризованную функцию без параметров?

В процедуре определяется локальная переменная, в которой хранится вычисляемый в этой процедуре параметр. Эта переменная в процедуре заменяет соответствующую ей глобальную переменную. Слово Procedure заменяется словом Function. После имени функции вставляется двоеточие и тип возвращаемого значения. В конце тела функции необходимо присвоить имени функции значение локальной переменной, хранящей вычисленный параметр.

12. Все ли компонентные функции параметризованного класса являются параметризованными?

Компонентные функции параметризованного класса автоматически являются параметризованными. Их не обязательно объявлять как параметризованные с помощью template.