1. Что такое класс? Для чего описываются классы?

Класс - это определяемый пользователем тип. Описание класса очень похоже на описание структуры в Си. В этом смысле класс является расширением понятия структуры.

2. Чем отличается класса от структуры?

**Класс** - это данные и методы для работы с этими данными. Структра схожа с **классом**, она так же может иметь в себе перечень неких данных и даже методов. Но: **структура** не поддерживает наследование, имплементирование, ее можно объявить внутри **класса**, **структура** является дополнительным сборным типом данных.

3. Какие модификаторы используются для ограничения доступа к членам класса?

**private**: закрытый класс или член класса. Представляет полную противоположность модификатору public. Такой закрытый класс или член класса доступен только из кода в том же классе или контексте.

4. Для чего в классе определяется конструктор? Сколько может быть конструкторов в классе? Когда вызывается конструктор?

***Конструктор*** класса – это специальный метод (функция) класса. Конструктор вызывается при создании объекта класса. Как правило, конструктор используется для:

* выделения памяти под объект класса;
* начальной инициализации внутренних данных класса.

Конструктор предназначен для формирования экземпляра объекта класса. Имя конструктора класса совпадает с именем класса.

перегружать можно только при наследовании  
конструкторов же сколько хотите, но естественно с разным списком аргументов

Вызов конструктора осуществляется при создании объекта класса. Конструктор класса вызывается компилятором.

5. Может ли быть конструктор описан в области доступа private?

Да, может. Такой конструктор называется приватным конструктором (private constructor).

6. Что будет, если в классе не определен конструктор?ъ

 Если в классе нет ни одного конструктора, то его объекты можно создавать - и это правильно, в С++ нет обязательных методов, класс может быть вообще пустым (иногда это полезно).

7. Для чего в классе определяется деструктор? Сколько может быть деструкторов в классе? Когда вызывается деструктор?

Деструктор, как и [**конструктор**](https://www.bestprog.net/ru/2018/04/02/c-class-constructor-features-of-use-constructors-in-the-classes-the-default-constructor-parameterized-constructors-examples-of-classes-that-contain-constructors_ru/), относится к специальным функциям класса. **Деструктор** – это специальный метод, который вызывается при удалении объекта. Как правило, деструктор используется для освобождения памяти, динамично выделенной под внутренние данные класса. Могут быть и другие случаи применения деструктора.

Деструктор – это обратная по отношению к конструктору функция.

Имя деструктора совпадает с именем класса, перед которым следует символ ‘~’. Например, если класс имеет имя CMyClass, тогда имя деструктора будет ~CMyClass().

Явно — один. Но компилятор может неявно создать второй, в определённых случаях.

**Деструктор** **вызывается** автоматически, когда объект выходит из области видимости

8. Может ли быть деструктор описан в области доступа private?

**Да, можно.** Такой деструктор называется приватным деструктором

9. Что будет, если в классе не определен деструктор?

Если не создавать ничего, то компилятор создаст оба. Но если создать конструктор с параметром/параметрами, то неявных конструкторов уже не будет, если при этом не будет явного контруктора оп умолчанию, то вообще не будет конструктора по умолчанию. Тогда нельзя будет декларировать экземпляры, не указывая конкретное значение параметра, который следует передать в конструктор. А вот деструктор всегда ровно один, или явный, или неявный. Неявное - это, что в программе есть, но чего в её тексте нет.

10. Что такое конструктор, деструктор по умолчанию? Когда они вызываются?

***Конструктор по умолчанию*** – это конструктор класса, который объявляется без параметров. Если класс не содержит явным образом определенный конструктор, тогда при создании объекта автоматически вызывается конструктор по умолчанию. Конструктор по умолчанию просто выделяет память для объекта класса, если он объявляется.

Конструктор по умолчанию автоматически вызовется только тогда, если в классе не объявлено ни одного конструктора. Как только в классе объявить любой другой конструктор с параметрами, то при объявлении

Если **деструктор** не определен, компилятор будет предоставлять его **по** **умолчанию**

11. Что такое экземпляр класса?

**Экземпляр** **класса** — это конкретный объект данного **класса**, за которым закрепляется определенная память. А **класс** - это только описание свойств и способов выполнения над его объектами различных операций (методов). Каждый **класс** может иметь одновременно несколько объектов (**экземпляров**), каждый из которых имеет одни и те же свойства и методы.

13. Для чего в классе задаются статические члены? В чем отличие их использования от обычных членов класса?

Класс – это тип, а не объект данных, и в каждом объекте класса имеется своя собственная копия данных – членов этого класса. Однако некоторые типы требуется реализовать так, что все объекты этого типа могут совместно использовать некоторые данные. Такие совместно используемые данные должны быть описаны как часть класса.  
  
Статические данные относятся ко всем объектам класса. Такие данные используются, если

* требуется контроль общего количества объектов класса;
* требуется одновременный доступ ко всем объектам или части их;
* требуется разделение объектами общих ресурсов.

В этом случае в определение класса могут быть введены статические члены.  
  
Статические члены описываются с помощью ключевого слова static, которое может использоваться при объявлении член-данных и член-функций.  
  
Такие члены классов называются статическими, и независимо от количества объектов данного класса, существует только одна копия статического элемента.