

Контрольные вопросы по языку C++

Здесь собраны вопросы по понятиям и конструкциям языка C++, которые явно или неявно возникали при обзоре языка. Не все эти понятия подробно обсуждались на лекциях. Некоторые только упоминались, некоторые присутствовали в примерах кода, а некоторые, возможно, и не упоминались вовсе. Тем не менее, эти понятия необходимы для понимания общей идеологии языка и также способов его использования при написании программ. Поэтому всем предлагается найти и аккуратно сформулировать ответы на приведенные ниже вопросы. В некоторых случаях это можно сделать по материалам лекций, в некоторых случаях придется заглянуть в справочные руководства по языку C++. На основе этих вопросов будет составлена часть заданий экзамена в зимнюю сессию.

Какой смысл имеют секции `private` и `public` в описании класса?

Чем отличаются объявления `struct` и `class` с точки зрения последующего использования таких объектов?

В каких случаях необходим конструктор без параметров?

В каких случаях необходим конструктор копирования?

Чем принципиально отличаются реализации конструктора копирования и оператора присваивания?

Как работает конструктор копирования по умолчанию?

В каких случаях достаточен конструктор копирования по умолчанию, а в каких нет?

Каким требованиям должен удовлетворять класс, чтобы можно было создавать массив таких классов?

Может ли один класс содержать внутри своего объявления объявление некоторого другого класса?

Для чего нужно определять деструктор класса?

В какой момент активируется деструктор класса?

Как работает деструктор по умолчанию?

В какой ситуации может возникнуть запись `A(int x) : b(x)` ? В каких случаях она необходима?

Для чего в программе может потребоваться запись вида `class A;` в отдельной строке перед фрагментом кода?

Что означает ключевое слово `const`, в каких случаях оно применяется?

Какие действия запрещено выполнять с константным указателем?

Можно ли изменять константный указатель после его инициализации?

Какой смысл имеет запись `const int x = 1;` ?

Какой смысл имеет запись `const int *x;` ?

Какой смысл имеет запись `int * const x;` ?

Имеет ли смысл запись `const char * const s;` ?

Что такое константный метод класса, в чем его специфика?

Что такое ссылочный тип данных?

Можно ли изменять ссылку после ее инициализации?

В каких случаях следует использовать ссылки?

В чем состоит отличие константной и неконстантной ссылок?

Что такое статический член класса?

Может ли метод класса быть статическим?

Как нужно объявлять и определять статические члены класса?

Как можно обратиться к статическому члену класса?

Какие ограничения накладываются на перегрузку операций для данного класса?

В каком случае перегруженная операция может являться членом класса?

В каком случае перегруженная операция не может являться членом класса?

Чем отличаются сигнатуры перегрузки операторов `a++` и `++a` ?

Что такое параметры по умолчанию при описании функций?

Какие ограничения накладываются на параметры по умолчанию при описании функций?

Какие свойства операции остаются неизменными после ее перегрузки для конкретного класса?

Что означает ключевое слово `template`, в каких случаях оно применяется?

Может ли ключевое слово `template` применяться не к классам или членам классов?

Как описать `template` реализацию для определенного конкретного типа данных?

В какой момент компилятор реализует `template` функции, описанные в коде программы?

Как компилятор выбирает какой тип использовать для реализации функции при компиляции ее вызова в конкретном месте программы?

Как вызвать в программе `template` функцию для конкретного типа данных?

Что означает запись `::`, в каких случаях она применяется?

Что означает ключевое слово `namespace`, в каких случаях оно применяется?

Что означает конструкция `#pragma once`, в каких случаях она применяется?

Что означает тип `auto`, когда его можно и когда нельзя использовать?

Что означает спецификация `friend`, когда она применяется?

Что означает ключевое слово `typename`, в каких случаях оно применяется?

Что означает ключевое слово `inline`, в каких случаях оно применяется?

Можно ли использовать объект, объявленный в рамках некоторого пространства имен `A`, если в коде не указана директива `using namespace A` ; ?

Что такое исключение? В каких ситуациях оно возникает?

Что означают ключевые слова `try` и `catch`, в каких случаях они применяются?

Что означает запись `catch(...)` ?

Как перехватывать исключения нескольких разных типов в одном блоке программы?

Чем отличается выделение памяти при использовании `new` и `malloc` ?

Чем отличается освобождение памяти при использовании `delete` и `free` ?

Как работает оператор `new` в случае отказа в выделении памяти?

Какой конструктор объекта срабатывает при выполнении оператора `new`

Чем отличаются операторы `delete` и `delete []` ?

Как сработает инструкция `delete p` если указатель `p` равен `0` ?