



Санкт-Петербургский
государственный
университет

Вопросы

- 1 КАКИЕ ВИДЫ ТЕСТОВ СУЩЕСТВУЮТ И ДЛЯ ЧЕГО ОНИ ПРИМЕНЯЮТСЯ?
- 2 ЧТО ДАЮТ МОДУЛЬНЫЕ ТЕСТЫ РАЗРАБОТЧИКАМ?
- 3 КАКИЕ КРИТЕРИИ НАЛАГАЮТСЯ НА МОДУЛЬНЫЕ ТЕСТЫ?
- 4 ЧТО ТАКОЕ TDD?
- 5 РАССКАЖИТЕ О СИСТЕМАХ АВТОМАТИЗАЦИИ СБОРКИ И ТЕСТИРОВАНИЯ ПРОЕКТОВ В C++.
- 6 ЧТО ТАКОЕ СТАТИЧЕСКАЯ И ДИНАМИЧЕСКАЯ БИБЛИОТЕКИ?
- 7 В ЧЁМ ПЛЮСЫ И МИНУСЫ СТАТИЧЕСКОЙ И ДИНАМИЧЕСКОЙ БИБЛИОТЕК?
- 8 В ЧЕМ РАЗНИЦА МЕЖДУ ИСПОЛНЯЕМЫМ ФАЙЛОМ И ДИНАМИЧЕСКОЙ БИБЛИОТЕКОЙ?
- 9 ЧТО ТАКОЕ DLL hell?
- 10 ЧТО ДОЛЖНО БЫТЬ РЕАЛИЗОВАНО В КЛАССЕ?



Санкт-Петербургский
государственный
университет

Вопросы

- 11 Что такое конструктор по умолчанию?
- 12 Для чего нужны default и delete с конструкторами и другими методами класса?
- 13 Что такое конструктор копирования и оператор присваивания копированием?
- 14 Что такое lvalue и rvalue?
- 15 Что такое конструктор перемещения и оператор присваивания перемещения?
- 16 В каких случаях надо определять КК, ОПК, КП, ОПП, деструктор?

В каких случаях КК, ОПК, КП, ОПП, деструктор будут созданы компилятором?
- 17 Что такое делегирующий конструктор?



Санкт-Петербургский
государственный
университет

ОТВЕТЫ

1 КАКИЕ ВИДЫ ТЕСТОВ СУЩЕСТВУЮТ И ДЛЯ ЧЕГО ОНИ ПРИМЕНЯЮТСЯ?

Тесты бывают модульные (unit), функциональные и интеграционные.

Модульное (Unit) тестирование — это тестирование на корректность отдельных модулей (hpp + cpp) / классов исходного кода без взаимодействия / зависимости с другими модулями/классами.

Функциональное тестирование — это тестирование отдельных функциональностей в рамках одной независимой части системы. Пример: аутентификация пользователя, запрос в БД, отправка/обработка команды на сервер.

Интеграционное тестирование — это тестирование совместной работы нескольких частей системы (программных модулей, микросервисов, приложений распределённой системы). Пример: сценарий когда пользователь добавляет товар в корзину, оформляет его и оплачивает.

2 ЧТО ДАЮТ МОДУЛЬНЫЕ ТЕСТЫ РАЗРАБОТЧИКАМ?

1. Уменьшают время на общее тестирование (быстро находится большое количество мелких ошибок).
2. Защищают от регрессии.
3. Подтверждают документацию. Показывают как работает код.
4. Показывают связность кода: чем больше % покрытия кода тестами, тем менее связный код системы (и более легко заменяемый).



Санкт-Петербургский
государственный
университет

ОТВЕТЫ

3 КАКИЕ КРИТЕРИИ НАЛАГАЮТСЯ НА МОДУЛЬНЫЕ ТЕСТЫ?

1. Быстрота

Модульных тестов тысячи. Они запускаются регулярно разработчиком для отсеечения всех мелких ошибок.

2. Изолированность

Результат одного теста не должен влиять на результат другого теста. Модульные тесты должны запускаться параллельно.

3. Повтряемость

Не должно возникать ситуаций, когда 1 раз в 100 запусков тест будет падать. Трудно выявляемые и исправляемые ситуации.

4. Самопроверяемый

В тесте для запуска не надо ничего настраивать.

5. Уместный

Тест должен проверять то, что необходимо.

4 ЧТО ТАКОЕ TDD?

TDD (Test Driving Development) — это техника разработки кода, при которой тесты пишутся до реализации:

1. Вначале создаётся структура класса в хедере и пустая реализация в соответствующей единице трансляции (.cpp).

2. Потом пишутся тесты, описывающие все необходимые ситуации с методом, который будет реализован.

3. Реализовывается соответствующий пункту 2 метод и запускаются тесты.

4. Пункты 2 и 3 повторяются пока не будут реализованы все методы класса.



Санкт-Петербургский
государственный
университет

ОТВЕТЫ

5 РАСКАЖИТЕ О СИСТЕМАХ АВТОМАТИЗАЦИИ СБОРКИ И ТЕСТИРОВАНИЯ ПРОЕКТОВ В C++.

Для автоматизирования сборки и тестирования проекта в C++ можно использовать связку Cmake + Ninja.

Cmake — кроссплатформенная автоматизированная система сборки проектов, созданная в 2000 г. для облегчения написания сценариев на скриптовом языке Make. Cmake является свободным и открытым ПО. Cmake не собирает исполняемые файлы, а лишь генерирует файлы сборки из предварительно написанного сценария CmakeLists.txt в проект одного из видов: make, ninja, visual studio и т. д. И только потом задействуется соответствующий собранному проекту инструмент для компиляции кода.

Для компиляции кода на следующем этапе после Cmake может быть использована утилита ninja. Она является улучшенной и доработанной версией make.

Для запуска тестов cmake содержит встроенную утилиту ctest и специальный синтаксис Cmake скриптового языка.

6 ЧТО ТАКОЕ СТАТИЧЕСКАЯ И ДИНАМИЧЕСКАЯ БИБЛИОТЕКИ?

Статическая библиотека — это файл .lib для windows или .a для unix, который содержит набор объектных файлов функции и классов. Она используется в процессе статической линковки в исполняемый файл вместе с другим кодом и статическими библиотеками во время его создания.

Динамическая библиотека — это файл .dll (dynamic link library) для windows или .so для unix, который содержит набор объектных файлов функции и классов. Но линковка динамической библиотеки запускается в момент создания процесса (когда вы запускаете исполняемый файл на выполнение). Также возможна её линковка уже после запуска, т.е. библиотека может быть подгружена в адресное пространство уже работающего процесса.



Санкт-Петербургский
государственный
университет

ОТВЕТЫ

7

В чём плюсы и минусы статической и динамической библиотек?

Плюсы статической библиотеки:

1. Простота установки пользователем

Приложение, использующее статическую библиотеку, может быть запущено на любом компьютере без необходимости установки дополнительных зависимостей.

2. Скорость работы

Т.к. функции и классы статической библиотеки встраиваются в исполняемый файл программы, то процесс выполнения (вызова этих функций и классов) будет более быстрым по сравнению с динамическим подходом.

3. Надёжность

Т.к. все необходимые функции и классы находятся в рамках одного файла, то версии библиотеки могут быть контролируемыми и учитываться во время разработки.

Минусы статической библиотеки:

1. Большой размер исполняемого файла.

2. Требование перекомпиляции зависимых файлов при любом изменении в библиотеке.

Плюсы и минусы динамической библиотеки обратны минусам и плюсам статической библиотеки.

Дополнительным важным плюсом динамической библиотеки является то, что она загружается в память только тогда, когда надо. И также её можно выгрузить из процесса, когда надо.



Санкт-Петербургский
государственный
университет

ОТВЕТЫ

8 В ЧЕМ РАЗНИЦА МЕЖДУ ИСПОЛНЯЕМЫМ ФАЙЛОМ И ДИНАМИЧЕСКОЙ БИБЛИОТЕКОЙ?

1. Файлы exe — это исполняемые файлы, которые можно запускать независимо. А файлы dll — это программы динамически подключаемых библиотек, используемые для совместного использования кода и ресурсов.
2. Файлы exe содержат инструкции, которым компьютер следует, чтобы выполнить программу, а файлы dll содержат код (в виде машинного кода), который может использоваться несколькими программами одновременно.

9 Что такое DLL hell?

Все dll должны быть совместимы от версии к версии и взаимозаменяемы в обе стороны. На практике всё получилось наоборот. Это проблема несовместимости и невзаимозаменяемости в обе стороны разных версии dll и стала называться dll hell.

10 Что должно быть реализовано в классе?

1. Конструкторы.

- 1.1. Конструктор по умолчанию.
- 1.2. Конструкторы с параметрами.
- 1.3. Конструктор копирования.
- 1.4. Конструктор перемещения.

2. Деструктор.

3. Операторы присваивания.

- 3.1. Оператор присваивания копирования.
- 3.2. Оператор присваивания перемещения.

4. Другие методы



Санкт-Петербургский
государственный
университет

ОТВЕТЫ

11 Что такое конструктор по умолчанию?

КпоУ — это конструктор, который вызывается без передачи аргументов (включая конструктор с параметрами, имеющими значение по умолчанию):

```
LongNumber x;
```

```
LongNumber *xptr = new LongNumber();
```

Каждый класс может иметь только один конструктор по умолчанию либо без параметров, либо с параметрами, имеющими значения по умолчанию.

При объявлении массива объектов или объявлении динамически КпоУ инициализирует все элементы:

```
LongNumber arr[10];
```

```
LongNumber *arr = new LongNumber[10];
```

Если в классе не определён КпоУ, компилятор неявно создаст его. Он будет аналогичен явно объявленному конструктору с пустым телом.

Если определены конструкторы для класса, но среди них нет КпоУ, то компилятор его не создаст неявно. Это приведёт к ошибкам.

12 Для чего нужны default и delete с конструкторами и другими методами класса?

Конструкция `LongNumber () = default;` говорит компилятору явно создать КпоУ или другой метод (с реализацией по умолчанию).

Конструкция `LongNumber () = delete;` удаляет КпоУ. Теперь любая ситуация, в которой вызывается КпоУ будет генерировать ошибку компиляции.



Санкт-Петербургский
государственный
университет

ОТВЕТЫ

13 ЧТО ТАКОЕ КОНСТРУКТОР КОПИРОВАНИЯ И ОПЕРАТОР ПРИСВАИВАНИЯ КОПИРОВАНИЕМ?

КК и **ОПК** — это ситуации, когда надо создать копию существующего объекта или через инициализацию, или через присваивание. При этом, будет создано два разных объекта с одинаковыми данными, и эти данные будут храниться в разных областях памяти.

```
LongNumber(const LongNumber& x);
```

```
LongNumber& operator = (const LongNumber& x);
```

КК и ОПК создаются компилятором автоматически. Но если объект содержит указатель или ссылку, то их надо определять самостоятельно. Если этого не сделать, то будет несколько объектов, ссылающихся на одну и ту же область памяти. Что приведёт к неопределённому поведению программы.

Надо различать:

`LongNumber x = y;` — КК

и

```
LongNumber x;
```

`x = y;` — ОПК

Существует три **случая вызова КК**:

1. Когда объект является возвращаемым значением.

```
LongNumber x = func();
```

2. Когда объект передается в функцию по значению в качестве аргумента.

```
func(LongNumber x);
```

...



Санкт-Петербургский
государственный
университет

ОТВЕТЫ

3. Когда объект конструируется на основе другого объекта (того же класса).

```
LongNumber x = y;
```

Вопрос: Чем отличается конструктор копирования от оператора присваивания копирования?

КК применяется тогда, когда объект ещё не существует в памяти. А ОПК применяется к уже существующему в памяти объекту.

14 Что такое lvalue и rvalue?

В C++ используемые значения можно разделить на две группы: lvalue и rvalue.

Lvalue представляем ИМЕНОВАННОЕ значение (переменные, параметры, константы), с которыми ассоциирован адрес памяти для хранения. Lvalue можно ПРИСВОИТЬ значение.

Rvalue — это то, что можно ТОЛЬКО ПРИСВАИВАТЬ, но ему нельзя присвоить (литералы, результаты выражений).

```
int x = 5;
```

x — lvalue

5 — rvalue

rvalue-ссылка — это переменная, которая ссылается на результат выражения rvalue. Привязка продлевает время жизни такого rvalue до окончания области видимости.

```
int&& rvx {5};
```

rvx хранит ссылку на rvalue значение, но само является lvalue значением.



Санкт-Петербургский
государственный
университет

ОТВЕТЫ

15 Что такое конструктор перемещения и оператор присваивания перемещения?

КП позволяет переместить данные, принадлежащие объекту `rvalue`, в `lvalue` без копирования значений из динамической памяти (копируются значения указателей, но при этом надо не забыть присвоить копируемым указателям `nullptr`). Ещё КП применяется, когда копирование данных нежелательно или излишне.

```
LongNumber(LongNumber&& x);
```

ОПП решает ту же задачу, что и КП, но в ситуации, когда объект `lvalue` уже существует в памяти и ему присваивается новое значение.

16 В каких случаях надо определять КК, ОПК, КП, ОПП, деструктор?

В каких случаях КК, ОПК, КП, ОПП, деструктор будут созданы компилятором?

КК, ОПК, КП, ОПП надо определять, если класс содержит среди полей указатели, ссылки.

Компилятор создаст КК, ОПК, деструктор всегда сам, если они не определены пользователем и не удалены с использованием конструкции `<сигнатура> = delete;`

Компилятор создаст КП и ОПП, если они не определены пользователем, не удалены `<сигнатура> = delete;` и при этом не определены пользователем КК, ОПК соответственно. Если КК и ОПК будут определены пользователем, то компилятор не будет неявно создавать КП и ОПП, а при вызове их в коде просто заменит их вызовом КК и ОПК.



Санкт-Петербургский
государственный
университет

ОТВЕТЫ

17 ЧТО ТАКОЕ ДЕЛЕГИРУЮЩИЙ КОНСТРУКТОР?

Начиная с C++11 конструкторам разрешено вызывать другие конструкторы.

Delegate – передавать полномочия.

ДК — это конструктор, который вызывает другой конструктор.

Синтаксис вызова конструктора:

```
LongNumber(int n) : LongNumber() {...}
```

Запомнить!

Синтаксис делегирования конструктора не предотвращает случайное создание рекурсии конструктора: конструктор1 вызывает конструктор2, который вызывает конструктор1, и никаких ошибок не возникает, пока не произойдёт переполнения стека.