

Ответы на контрольные вопросы:

1 вопрос: Перечислите все специальные функции-члены класса, включая перемещающие операции

К специальным функциям-членам относятся члены класса, которые в некоторых случаях (при отсутствии их самостоятельного определения пользователем) автоматически генерируются компилятором:

- Конструктор по умолчанию
- Копирующий конструктор
- Оператор копирующего присваивания
- Деструктор
- Конструктор перемещения
- Оператор присваивания перемещения

2 вопрос: Приведите примеры операторов, которые можно, нельзя и не рекомендуется перегружать

→ Можно:

→ `[] ()`

→ Не рекомендуется:

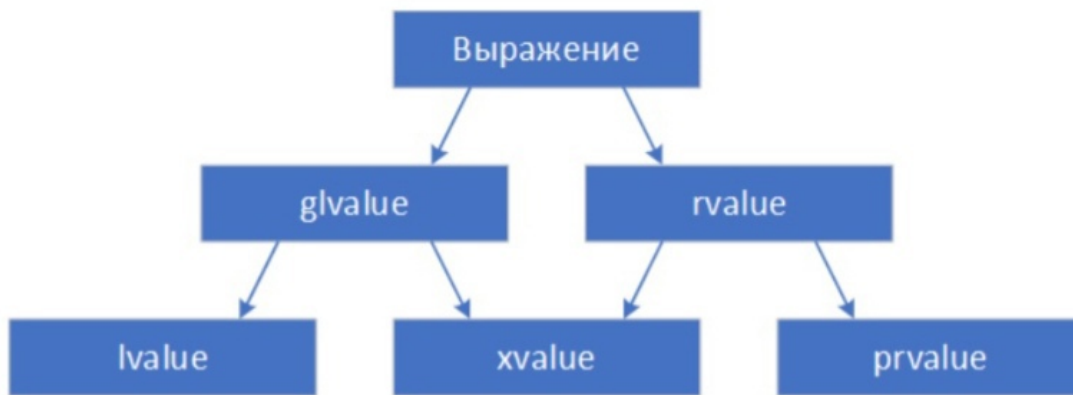
& ,

→ Нельзя:

`:: . .*`

3 вопрос: О каких преобразованиях следует помнить при проектировании операторов?

4 вопрос: Опишите классификацию выражений на основе перемещаемости и идентифицируемости



lvalue: изменяемые инициализируются, неизменяемые - нет. Ссылки lvalue на const могут быть инициализированы всегда.

rvalue: инициализируются только значениями r-values.

5 вопрос: Зачем нужны rvalue-ссылки?

→ Они увеличивают продолжительность жизни объекта, которым инициализируются, до продолжительности жизни ссылки r-value (до конца `main()` обычно).

→ Неконстантные ссылки r-value позволяют нам изменять значения r-values, на которые указывают ссылки r-value.

6 вопрос: Почему семантика перемещения лучше копирования?

Если инициализировать одну из переменных класса значением другой переменной, то оба объекта класса будут указывать на один и тот же объект. Когда же один из объектов выходит из области видимости, он удаляет объект, на который указывает, оставляя второй объект класса с висячим указателем, поэтому попытка удаления второго объекта ни к чему не приводит, потому что фактически он уже ни на

что не указывает.

Другое дело получится, если использовать именно семантику перемещения, при которой конструктор копирования и оператор присваивания не копируют указатель (семантика копирования), а передают владение указателем из источника в объект назначения. *Семантика перемещения* означает, что класс, вместо копирования, передает право собственности на объект.

7 вопрос: Что делает функция `std::move` и когда нет необходимости явно её вызывать?

Функция `std::move()` — это стандартная библиотечная функция, которая конвертирует передаваемый аргумент в r-value. Мы можем передать l-value в функцию `std::move()`, и `std::move()` вернет нам ссылку r-value.

Нет необходимости её вызывать при возврате объектов, если тип переменной совпадает с типом возвращаемого объекта. Коротко: нужно использовать `std::move`, когда есть lvalue, которое хочет быть rvalue.

8 вопрос: Кем выполняется непосредственная работа по перемещению?

Конструктором перемещения и оператором присваивания перемещением

9 вопрос: Когда может потребоваться пользовательская реализация специальных функций-членов класса?

Возможно, когда мы пишем хороший интерфейс класса в заголовочном файле, чтобы его можно было использовать в различных ситуациях, а не только непосредственно в данной программе.

10 вопрос: Для чего нужны ключевые слова `default` и `delete` в объявлении специальных функций-членов класса?

Default - для явного объявления специальной функции-члена по умолчанию, это приводит к тому, что компилятор определит функцию только при необходимости, так же, как если бы функция не была объявлена вообще.

В некоторых случаях компилятор может создавать удаленные специальные функции элементов, которые не определены и поэтому не вызываемы. Это может произойти в тех случаях, когда вызов определенной специальной функции-члена в классе не имеет смысла, учитывая другие свойства класса. Чтобы явно запретить автоматическое создание специальной функции-члена, можно объявить ее как удаленную с помощью ключевого слова `Delete`.