



Введение в объектно-ориентированное программирование.ои(dor)_211115

- 1 Метод, принадлежащий классу и соотнесенный с классом (статический метод), может быть вызван сам по себе и имеет доступ к ...
- 2 Модификаторы доступа в объектно-ориентированных языках являются ...
- 3 Классы в программе могут соответствовать:
- 4 Класс – это ...
- 5 Объект - это ...
- 6 Идея объектно-ориентированного программирования была основана на ...
- 7 Определите соответствие между терминами и определениями:
- 8 Определите соответствие между открытиями и их создателями:
- 9 Определите соответствия между языками и видами программирования.
- 10 Определите в исторической последовательности открытия в программировании:
- 11 Определите создание языков программирования в исторической последовательности (от старого до нового):
- 12 Определите последовательность, в которой развивалось направление объектно-ориентированного программирования:
- 13 На практике, объектно-ориентированное программирование сводится к созданию некоторого количества классов, включая интерфейс и реализацию, и последующему их использованию. Графическое представление некоторого количества классов и связей между ними называется диаграммой классов. Объектно-ориентированный подход, за время своего развития, накопил множество рекомендаций (паттернов) по созданию классов и иерархий классов. Предсказать результат следующей программы. На экран выведется строка «Compiler Error», т.к. компилятор увидит ошибку в строке 1.



- 14) Как и структуры, классы могут задавать поля, то есть, переменные, принадлежащие либо непосредственно самому классу (статические), либо экземплярам класса (обычные). Статические поля существуют в одном экземпляре на всю программу (или, в более сложном варианте, — в одном экземпляре на процесс или на поток/нить). Обычные поля создаются по одной копии для каждого конкретного объекта — экземпляра класса. Представлен код программы. Какой результат выполнения?
- 15) В объектно-ориентированной программе с применением классов, каждый объект является «экземпляром» некоторого конкретного класса, и других объектов не предусмотрено. То есть, «экземпляр класса» в данном случае означает не «пример некоторого класса» или «отдельно-взятый класс», а «объект, типом которого является какой-то класс». При этом, в разных языках программирования допускается, либо не допускается существование еще каких-то типов данных, экземпляры которых не являются объектами. Перед Вами код. Опишите его работу.
- 16) Конструктор — это...
- 17) Деструктор — это...
- 18) При передаче параметра по значению, конструктор будет вызываться рекурсивно, поэтому он должен иметь в качестве параметра ...
- 19) Перегруженные функции подчиняются тем же правилам относительно типов параметров, что и ...
- 20) В процессе выполнения операции присваивания между существующими объектами одного типа обычно вызывается ...
- 21) Варианты использования больше всего нужны для того, ...
- 22) Конструктор копирования вызывается ...
- 23) Установите соответствие между терминами и определениями
- 24) Когда вызывается конструктор, автоматически создаются объекты при определенных условиях. Определите соответствие между объектами и условиями создания конструктора.
- 25) При автоматическом вызове деструктора, для объектов при определенных условиях, определите соответствие между объектами и условиями уничтожения.



- 26 Проставьте последовательность создания конструктора
- 27 Определите последовательность действий, если есть два класса, один из которых базовый, а другой — унаследованный от базового.
- 28 Конструктор является типом метода класса, который автоматически вызывается при создании объекта этого же класса, а деструктор — это тип метода класса, который выполняется при удалении объекта класса. Ниже представлен код, подумайте какой тип метода класса здесь используется. Опишите код программы:
- 29 Предсказать результат следующей программы:
- 30 В отличие от обычных методов, конструкторы имеют определенные правила их именования: конструкторы всегда должны иметь то же имя, что и класс (учитываются верхний и нижний регистры), конструкторы не имеют типа возврата (даже void). Перед Вами код. Опишите его результат.
- 31 В обычном режиме компилятор автоматически формирует конструктор копирования для каждого класса, однако, в некоторых случаях, программист формирует конструктор копирования, называемый ...
- 32 Конструктор копирования вызывается...
- 33 При передаче параметра по значению, конструктор будет вызываться рекурсивно, поэтому он должен иметь...
- 34 «Закон Большой Тройки» или «Правило трёх» — правило в C++ гласит, что если класс или структура определяет один из следующих методов, то они должны явным образом определить все три метода. Какие три метода нужны?
- 35 Оператор присваивания – это...
- 36 Существует два вида операций присваивания – это...
- 37 Операторы присваивания хранят значение в объекте, указанном левым операндом. Установите соответствие между операторами и их значениями:
- 38 Операторы присваивания хранят значение в объекте, указанном левым операндом. Установите соответствие между операторами и их значениями



- 39) Операторы присваивания хранят значение в объекте, указанном левым операндом. Установите соответствие между операторами и их значениями.
- 40) Проставьте последовательность алгоритма процесса присваивания.
- 41) Оператор простого присваивания (=) вызывает сохранение значения второго операнда в объекте, указанном первым операндом. Если оба объекта имеют арифметические типы, правый операнд преобразуется в тип слева перед сохранением значения. Посмотрите код и напишите алгоритм его создания.
- 42) Оператор присваивания не может быть объявлен как «дружественный» к классу. Если перегрузить оператор присваивания как «дружественный» к классу, то это означает, что будет перегруженный глобальный оператор присваивания, который вызывается для экземпляров классов автоматически. Это, в свою очередь, может привести к путанице в операциях присваивания и увеличении невидимых ошибок. Поэтому, компиляторы языка C++ не допускают перегружать оператор присваивания как «дружественный» к классу. Посмотрите код и напишите алгоритм его создания.
- 43) Описать работу кода программы:
- 44) Функция operator++() может создавать новый объект класса для использования его в качестве возвращаемого значения. Опишите работу кода:
- 45) Операция присваивания копированием отличается от конструктора копирования тем, что должна очищать члены-данные цели присваивания (и правильно обрабатывать самоприсваивание), тогда как конструктор копирования присваивает значения неинициализированным членам-данным. Опишите работу кода:
- 46) Оператор простого присваивания (=) вызывает сохранение значения второго операнда в объекте, указанном первым операндом. Если оба объекта имеют арифметические типы, правый операнд преобразуется в тип слева перед сохранением значения. Посмотрите код и напишите алгоритм его создания.



- 47) Оператор присваивания не может быть объявлен как «дружественный» к классу. Если перегрузить оператор присваивания как «дружественный» к классу, то это означает, что будет перегруженный глобальный оператор присваивания, который вызывается для экземпляров классов автоматически. Это, в свою очередь, может привести к путанице в операциях присваивания и увеличении невидимых ошибок. Поэтому, компиляторы языка C++ не допускают перегружать оператор присваивания как «дружественный» к классу. Посмотрите код и напишите алгоритм его создания.
- 48) Оператор разрешения обычно ...
- 49) К преимуществам использования наследования не относится:
- 50) Назначение наследования состоит в том, чтобы:
- 51) Механизм формирования новых классов из уже существующих или базовых классов – это ...
- 52) Укажите первую строку описания класса Tire, который является производным классом Wheel и Rubber:
- 53) Общий синтаксис механизма наследования выглядит следующим образом:
- 54) Установите соответствие между терминами и определениями:
- 55) Установите соответствие между типами наследования и их значениями
- 56) Установите соответствие между операциями в наследовании и их значениями
- 57) Проставьте последовательность выполнения представленного кода:
- 58) Проставьте последовательность создания кода:
- 59) Проставьте последовательность создания кода простого наследования:
- 60) Описать код программы:



- 61) Важный момент при наследовании – это перегруженные функции-методы класса-потомка. Если в классе-родителе и в его классах-потомках встречаются методы с одинаковым именем, то для объектов класса-потомка компилятор будет использовать методы именно класса-потомка. Перегруженные методы класса-потомка, могут вызывать методы класса-родителя. В таком случае, важно помнить, что необходимо правильно определить область действия с помощью оператора «::». Иначе, компилятор воспримет это, как вызов функцией самой себя. Наглядно, если бы мы перегрузили в классе функцию - это выглядело бы так.
Опишите, что указывает данная запись кода.
- 62) В определенных ситуациях могут появиться проблемы, связанные со множественным наследованием. Допустим, что в обоих базовых классах существуют методы с одинаковыми именами, а в производном классе метода с таким именем нет. Рассмотрим пример. Как в этом случае объект производного класса определит, какой из методов базовых классов выбрать?
- 63) Для объявления функции как виртуальной используется ключевое слово...
- 64) Обычно компиляторы при обработке виртуальных функций добавляют к каждому объекту скрытый элемент, который содержит указатель на массив адресов функций, называемый ...
- 65) Полиморфизм в объектно-ориентированном программировании реализуется:
- 66) Шаблон функции - это...
- 67) Для указания на объект производного класса можно воспользоваться ...
- 68) Множественное наследование – это ...
- 69) Установите соответствие между терминами и определениями:
- 70) Установите соответствие между терминами и определениями:
- 71) Установите соответствие между термином и его определением.
- 72) Установите порядок выполнения конструкторов при порождении из нескольких классов:
- 73) Если есть два класса, один из которых базовый, а другой – унаследованный от базового, то последовательность вызовов следующая:



- (74) Ниже представлен код, проставьте последовательность действий, которая произведена в этом коде:
- (75) Перед Вами код с перегрузкой методов. Выберите правильное описание работы кода.
- (76) Перед Вами код. Выберите, какой результат кода:
- (77) Перед Вами код. Выберите, какой будет результат кода:
- (78) Контейнерные классы – это...
- (79) Библиотека стандартных шаблонов – это ...
- (80) Итераторы – это ...
- (81) Контейнеры делятся на два вида - это ...
- (82) Какой вид контейнеров обеспечивают хранение конечного количества однотипных величин в виде непрерывной последовательности?
- (83) Стандартизованный интерфейс могут предоставлять:
- (84) Контейнерные классы – это классы, предназначенные для хранения данных, организованных определенным образом. Контейнеры бывают двух видов – ассоциативные и последовательные. Установите соответствие между терминами и определениями контейнерных классов.
- (85) Каждый контейнерный класс имеет 4 основных метода для работы с оператором - метод begin(), метод end(), метод cbegin(), метод cend(). Установите соответствие между названием метода контейнерного класса и его описанием.
- (86) Контейнерные классы – это классы, предназначенные для хранения данных, организованных определенным образом. Контейнеры бывают двух видов – ассоциативные и последовательные. Установите соответствие между терминами и определениями контейнерных классов.
- (87) Проставьте порядок выполнения кода поэтапно.
- (88) Проставьте последовательность функций классов-контейнеров в языке C++.



- 89 Пример работы с контейнером приведен ниже. В файле находится произвольное количество целых чисел. Программа считывает их в вектор и выводит на экран. Опишите последовательность кода:
- 90 Векторы — это умные массивы. Они занимаются автоматическим размещением себя в памяти, расширением и сужением своего размера по мере вставки или удаления данных. Векторы можно использовать в какой-то мере как массивы, обращаясь к элементам с помощью привычного оператора []. Случайный доступ выполняется очень быстро в векторах. Также, довольно быстро осуществляется добавление (или проталкиванием) данных в конец вектора. Когда это происходит, размер вектора автоматически увеличивается для того, чтобы было, куда положить новое значение. Рассмотрим код программы.
- 91 Стек — это динамическая структура сохранения данных, которая работает по принципу «последний пришел — первый вышел» (Last-In First-Out). В стеке добавление новых элементов и удаление существующих элементов производится с одного конца, который называется вершиной стека. Организация данных с помощью стека эффективна, когда нужно реализовать: - обмен данными между методами приложения с помощью параметров; - синтаксический анализ разнообразных выражений. Рассмотрим пример. Выберете, что будет результатом программы:
- 92 Для работы с очередью вам понадобится знать функции: push(), pop(), front(), back(), empty(). 1) Для добавления в очередь нового элемента нужно воспользоваться функцией — push(). В круглых скобках должно находиться значение, которое мы хотим добавить. 2) Если нам понадобилось удалить первый элемент, нужно оперировать функцией pop(). В круглых скобках уже ничего не нужно указывать, но по правилам они в обязательном порядке должны присутствовать! Эти функции тоже не нуждаются в указании аргумента: empty(), back() и front(). 3) Если вам понадобилось обратиться к первому элементу очереди, то вам понадобится функция front(). 4) Чтобы обратиться к последнему элементу в очереди, вам поможет функция back(). 5) Чтобы узнать, пуста ли очередь, нужно воспользоваться функцией empty(). - Если ваша очередь пуста — возвратит true. - Если же в ней что-то есть — возвратит false. В коде использовались все перечисленные методы. Выберете, что будет результатом программы.
- 93 Обработка исключительной ситуации — это ...
- 94 Оператор присваивания — это ...
- 95 Исключение — это ...



- (96) В С ++ с помощью ключевого слова «throw» функция может ...
- (97) Исключительные ситуации, возникающие при работе программы, можно разделить на два основных типа:
- (98) Процесс целочисленного деления на ноль является
- (99) Механизм исключений использует три слова С++: catch, throw и try. Установите соответствие между терминами и определениями:
- (100) Алгоритмы в STL выполняют сложные операции, типа сортировки и поиска. Однако, для выполнения более простых операций, специфичных для конкретного контейнера, требуются методы. Установите соответствие между методом и назначением.
- (101) При работе программ возникают исключительные ситуации, когда дальнейшее нормальное выполнение приложения становится невозможным. Чтобы справиться с этими ситуациями, программисту помогают средства С++ catch, try и throw. Установите соответствие между конструкцией и описанием ее функции.
- (102) Проставьте последовательность действий программы при возникновении ошибки.
- (103) Проставьте последовательность действий процесса обработки исключений, возбужденных оператором throw:
- (104) Проставьте последовательность стандартных исключений в иерархии родительско-дочерних классов.
- (105) Рассмотрим программный код, представленный в листинге. Он очень простой. Каким будет результат, если после выполнения программы ввести отрицательное число?
- (106) В представленной программе используются throw, try и catch вместе. Результат выполнения программы: We caught an int exception with value -1. Continuing our way! Опишите, как работают блоки программы:
- (107) Рассмотрим пример использования блока try...catch Текст программы типа Console Application следующий. Какой будет результат программы?
- (108) Модификаторы доступа являются ключевыми слова, которые задают параметры доступа для классов, методов и прочих элементов в ...
- (109) В определении класса члены класса с ключевым словом private доступны:



- (110) Методу класса всегда доступны данные:
- (111) Шаблон для создания объектов, обеспечивающий начальные значения состояний: инициализация полей-переменных и реализация поведения функций или методов – это ...
- (112) Структурированная переменная, содержащая всю информацию о некотором физическом предмете или реализуемом в программе понятии – это ...
- (113) Метод программирования, идея которого была основана на представление программы, как совокупность объектов – это...
- (114) Установите соответствие между терминами и определениями:
- (115) Установите соответствие между открытиями и их создателями:
- (116) Установите соответствия между языками и видами программирования.
- (117) Проставьте открытия в программировании в исторической последовательности:
- (118) Проставьте создание языков программирования в исторической последовательности (от старого до нового):
- (119) Проставьте историческую последовательность: как после появления объектно-ориентированного программирования развивалось это направление:
- (120) Предсказать результат следующей программы:
- (121) Представлен код программы. Какой результат выполнения?
- (122) Перед Вами код. Опишите его работу.
- (123) Специальная функция класса, которая вызывается при создании нового объекта этого класса и предоставляет возможность инициализировать объекты во время их создания и гарантировать, что все данные члены будут иметь корректные значения это ...
- (124) Специальный метод без параметров, имя которого состоит из символа ~ и имени класса это ...
- (125) Конструктор копирования должен иметь в качестве параметра ссылку, потому что ...
- (126) Конструкторы подчиняются тем же правилам относительно типов параметров, что и ...



- (127) При выполнении операции присваивания между существующими объектами одного типа вызывается ...
- (128) Варианты использования (кроме всего прочего) нужны для:
- (129) При описании нового объекта с инициализацией другим объектом этого же типа вызывается ...
- (130) Установите последовательность создания конструктора:
- (131) Если есть два класса, один из которых базовый, а другой - унаследованный от базового, то последовательность вызовов следующая:
- (132) Конструктор — это особый тип метода класса, который автоматически вызывается при создании объекта этого же класса. Деструктор — это специальный тип метода класса, который выполняется при удалении объекта класса. В то время, как конструкторы предназначены для инициализации класса, деструкторы предназначены для очистки памяти после него. Т.о., опишите код программы:
- (133) Предсказать результат следующей программы:
- (134) Перед Вами код. Опишите его результат.
- (135) Обычно, компилятор автоматически создает конструктор копирования для каждого класса, но в некоторых случаях, программист создает конструктор копирования, называемый:
- (136) Специальный конструктор в языке программирования C++, и в некоторых других языках программирования, применяемый для создания нового объекта, как копии уже существующего:
- (137) Конструктор копирования должен иметь в качестве параметра ссылку, потому что ...
- (138) Правило трёх (также известное как «Закон Большой Тройки» или «Большая Тройка») - это правило в C++, гласящее, что если класс или структура определяет один из следующих методов, то они должны явным образом определить все три метода:
- (139) Особый вид оператора присваивания, используемый для присваивания объектов одного класса друг другу ...
- (140) Операторы присваивания хранят значение в объекте, указанном левым операндом. Существует два вида операций присваивания:



- (141) Установите соответствие между операторами и их значениями:
- (142) Установите соответствие между операторами и их значениями
- (143) Установите соответствие между операторами и их значениями
- (144) Проставьте последовательность алгоритма процесса присваивания
- (145) Посмотрите код и напишите алгоритм его создания:
- (146) Посмотрите код и напишите алгоритм его создания
- (147) Описать работу кода программы:
- (148) Описать работу кода программы:
- (149) Операция присваивания копированием – это особый вид операции присваивания, используемый для присваивания объектов одного класса друг другу. Является одним из особых членов-функций и генерируется автоматически компилятором в случае, если нет явного объявления программистом. Код, сгенерированный компилятором и выполняет побитовое копирование. Операция присваивания копированием отличается от конструктора копирования тем, что должна очищать члены-данные цели присваивания (и правильно обрабатывать самоприсваивание), тогда как конструктор копирования присваивает значения неинициализированным членам-данным. Опишите работу кода программы:
- (150) Предположим, что класс Derv является частным производным класса Base. Мы определяем объект класса Derv, расположенный в функции main(). Через него мы можем получить доступ к:
- (151) Иерархия классов ...
- (152) Использование виртуальных функций для полиморфной обработки родственных объектов относят к:
- (153) Процесс создания новых классов, называемых наследниками или производными классами, из уже существующих или базовых классов – это ...
- (154) Роль наследования в объектно-ориентированном программировании заключается в:



- (155) Синтаксис наследования выглядит так:
- (156) Основное преимущество наследования – это возможность повторного использования кода. Укажите последовательность выполнения представленного кода:
- (157) Укажите последовательность написания кода:
- (158) Как решить вопрос о том, какой из спецификаторов использовать при наследовании? В большинстве случаев, производный класс представляет собой улучшенную или более специализированную версию базового класса. В случае, когда объект производного класса предоставляет доступ как к общим методам базового класса, так и к более специализированным методам своего класса, имеет смысл воспользоваться общим наследованием. Проставьте последовательность создания кода простого наследования:
- (159) Ключевыми элементами методологии объектно-ориентированного проектирования, связанного с наследованием относят: 1) выбор надлежащей совокупности типов; 2) проектирование взаимосвязей между типами и применение наследования для использования общего кода; 3) использование виртуальных функций для полиморфной обработки родственных объектов. Опишите код программы, содержащую наследование:
- (160) Важный момент при наследовании - это перегруженные функции - методы класса-потомка. Если в классе-родителе и в его классах-потомках встречаются методы с одинаковым именем, то для объектов класса-потомка, компилятор будет использовать методы именно класса-потомка. Перегруженные методы класса-потомка, могут вызывать методы класса-родителя. В таком случае, важно помнить, что необходимо правильно определить область действия с помощью оператора «::». Иначе, компилятор воспримет это, как вызов функцией самой себя. Наглядно, если бы мы перегрузили в классе функцию - это выглядело бы так. Опишите, что указывает данная запись кода:
- (161) Роль наследования в ООП такая же, как у функций в процедурном языке программирования, т.е. сократить размер кода и упростить связи между элементами программы. В определенных ситуациях могут появиться некоторые проблемы, связанные со множественным наследованием. Рассмотрим наиболее общую проблему. Допустим, что в обоих базовых классах существуют методы с одинаковыми именами, а в производном классе метода с таким именем нет. Рассмотрим пример. Как в этом случае объект производного класса определит, какой из методов базовых классов выбрать?



- (162) ключевое слово `virtual` используется ...
- (163) Когда используется таблица виртуальных функций?
- (164) Что реализуется через механизмы перегрузки (функций и операций), виртуальные функции и шаблоны?
- (165) Определение функции, в которой типу обрабатываемых данных присвоено условное обозначение - это...
- (166) Указателем базового класса можно воспользоваться для ...
- (167) Множественное наследование – это наследование, при котором создание производного класса основывается на ...
- (168) Опишите процесс выполнения конструкторов при порождении из нескольких классов поэтапно.
- (169) Если существует два класса, один базовый, другой унаследованный от базового, то при необходимости вызова, последовательность будет следующая:
- (170) Проставьте последовательность процесса написания представленного кода:
- (171) Перегрузка методов это механизм, который позволяет создавать несколько методов с одним и тем же именем. В результате создается иллюзия, что можно вызывать метод с разными аргументами. Перед Вами код перегрузки функции. Выберите правильное описание работы кода.
- (172) При множественном наследовании может сложиться ситуация, когда один и тот же базовый класс наследуется (через разные цепочки наследования) в производном классе несколько раз. Обычно, в этом ничего хорошего нет, т.к. нередко такие ситуации приводят к ошибкам. С другой стороны, совсем избежать подобных ситуаций бывает проблематично. Выход состоит в том, чтобы при создании цепочки наследования, использовать виртуальные базовые классы. Перед Вами код. Выберите, какой будет результат кода.



- (173) Виртуальные методы – это один из важнейших приёмов реализации полиморфизма. Они позволяют создавать общий код, который может работать как с объектами базового класса, так и с объектами любого его класса-наследника. При этом, базовый класс определяет способ работы с объектами, и любые его наследники могут предоставлять конкретную реализацию этого способа. Общий шаблон объявления чисто виртуального метода следующий:
- (174) Классы, предназначенные для хранения данных, организованных определенным образом:
- (175) Набор согласованных обобщённых алгоритмов, контейнеров, средств доступа к их содержимому и различных вспомогательных функций в C++
- (176) Сущности, использующиеся для получения доступа к отдельным данным в контейнере.
- (177) Контейнеры бывают ...
- (178) Последовательные контейнеры ...
- (179) Все контейнерные классы предоставляют ...
- (180) Проставьте порядок выполнения кода поэтапно.
- (181) Проставьте последовательность функций классов-контейнеров в языке C++
- (182) Пример работы с контейнером приведен ниже. В файле находится произвольное количество целых чисел. Программа считывает их в вектор и выводит на экран. Опишите последовательность кода:
- (183) Векторы — это умные массивы. Они занимаются автоматическим размещением себя в памяти, расширением и сужением своего размера по мере вставки или удаления данных. Векторы можно использовать в какой-то мере как массивы, обращаясь к элементам, с помощью привычного оператора []. Случайный доступ выполняется очень быстро в векторах. Также, довольно быстро осуществляется добавление (или проталкиванием) данных в конец вектора. Когда это происходит, размер вектора автоматически увеличивается для того, чтобы было куда разместить новое значение. Рассмотрим код программы. Выберете, что демонстрирует данный пример.



- 184) Стек – это динамическая структура сохранения данных, которая работает по принципу «последний пришел — первый вышел» (Last-In First-Out). В стеке добавление новых элементов и удаление существующих элементов производится с одного конца, который называется вершиной стека. Организация данных с помощью стека эффективна, когда нужно реализовать: обмен данными между методами приложения с помощью параметров; синтаксический анализ разнообразных выражений. Рассмотрим пример. Выберете, что будет результатом программы:
- 185) Для работы с очередью вам понадобится знать функции: push(), pop(), front(), back(), empty(). 1) Для добавления в очередь нового элемента нужно воспользоваться функцией — push(). В круглых скобках должно находиться значение, которое мы хотим добавить. 2) Если нам понадобилось удалить первый элемент нужно оперировать функцией pop(). В круглых скобках уже ничего не нужно указывать, но по правилам, они должны присутствовать! Эти функции не нуждаются в указании аргумента: empty(), back() и front(). 3) Если вам понадобилось обратиться к первому элементу очереди, то вам понадобится функция front(). 4) Чтобы обратиться к последнему элементу в очереди, вам поможет функция back(). 5) Чтобы узнать, пуста ли очередь, нужно воспользоваться функцией empty(). Если ваша очередь пуста — возвратит true. Если в ней что-то есть — возвратит false. В коде (ниже) использовали все выше перечисленные методы. Выберете, что будет результатом программы:
- 186) Механизм языков программирования, предназначенный для описания реакции программы на ошибки времени выполнения и другие возможные проблемы (исключения), которые могут возникнуть при выполнении программы, и приводят к невозможности дальнейшей отработки программой её базового алгоритма – это:
- 187) Особый вид оператора, используемый для присваивания объектов одного класса другому – это...
- 188) Реакция на исключительное обстоятельство, которое возникает во время работы программы – это ...
- 189) В C ++ функция может указывать исключения, которые она генерирует, с помощью ключевого слова:
- 190) Исключительные ситуации, возникающие при работе программы, можно разделить на два основных типа:
- 191) Классическим примером исключительной ситуации в C++ является:



- (192) Проставьте последовательность работы конструкций исключительных ситуаций:
- (193) Обработка исключений, возбужденных оператором throw, идет по следующей схеме:
- (194) Проставьте последовательность стандартных исключений в иерархии родительско-дочерних классов.
- (195) Мы рассмотрим ситуацию создания динамического массива. В программе для обработки исключительных ситуаций создадим динамический массив. Единственная особенность программы в том, что размер создаваемого массива определяется пользователем уже в процессе выполнения программы: при запуске программы, появляется сообщение с просьбой ввести целое число, число считывается, и в соответствии со считанным значением создается динамический массив. Рассмотрим программный код. Каким будет результат, если после выполнения программы ввести отрицательное число.
- (196) В представленной программе используются throw, try и catch вместе. Результат выполнения программы следующий: We caught an int exception with value -1. Continuing our way! Опишите как работают блоки.
- (197) Чтобы в блоке try сгенерировать исключительную ситуацию, нужно использовать оператор throw. Оператор throw может быть вызван внутри блока try или внутри функции, которая вызывается из блока try. Общая форма оператора throw: throw исключение; В результате выполнения оператора throw генерируется исключение некоторого типа. Это исключение должно быть обработано в блоке catch. Рассмотрим пример использования блока try...catch. Демонстрируется использование блока try...catch для обработки выражения. В данном выражении, в трех случаях может возникнуть исключительная ситуация: - корень из отрицательного числа a, если a