На содержание

КОНТРОЛЬНЫЕ ВОПРОСЫ (2 СЕМЕСТР)

1. Указатели на функцию. Определение указателей на функции с инициализацией. Присваивание значений указателю на функции. Вызов функции с помощью указателя на функцию. Указатель на функцию – параметр функции. Фактические параметры для формального параметра – указателя на функции. Определение типа указателя на функции.
2. Массив указателей на функции. Использование массива указателей на функции при разработке программы, управление которой, осуществляется с помощью меню. Указатель на функцию - результат работы функции.
3. Ссылка на функцию. Вызов функции с помощью ссылки на функцию. Ссылка - возвращаемый результат функции.
4. Рекурсивные функции. В качестве примера, определить рекурсивную функцию вычисления математической функции F(N) = 1*3*5*7*…*(2*N+1) , функция определена при N>0.Подставляемые функции.
5. Перегрузка функций. Шаблоны функций. Привести примеры перегрузки функций и шаблона функций.
6. Структура как тип и совокупность данных. Инициализация. Внутреннее представление. Указатели и ссылки на структуру. Обращение к элементам структуры.
7. Структуры и функций. Возвращаемые функциями значения и параметры функций структурного типа.
8. Вложенные структуры. Массивы структур. Указатели на структуры. Выделение динамической памяти.
9. Объединения. Назначение объединений. Определение типа и переменных – объединений. Инициализация. Объединения и битовые поля. Внутреннее представление объединений. Динамическое выделение памяти.
10. Связанные линейные динамические структуры. Стек.
11. Очередь. Базовые функции и алгоритмы.
12. Список. Базовые функции и алгоритмы.
13. Препроцессорная обработка программы. Директивы препроцессора. Директива #define. Макроподстановки средствами препроцессора. Макросы и шаблоны функций.
14. Директивы препроцессора условной компиляции. Директивы #include, #line, #error, # , #pragma, а также операции # и ##.
15. Объектно-ориентированный подход к программированию. Понятие класса как структурированного типа. Определение класса (два способа описания методов класса). Создание объектов (экземпляров класса) по имени и с помощью указателя и динамического выделения памяти. Инициализация. Обращение к компонентным данным и функциям.
16. Статический компонент класса. Инициализация, обращение к статическому элементу Статическая компонентная функция, назначение, вызов.
17. Доступ к членам класса. Спецификаторы доступа. Друзья классов: дружественные функции и дружественные классы.

18. Конструкторы и деструкторы. Назначение, формат определения, свойства. Создание объекта с помощью вызова конструктора с параметрами по имени и с помощью указателя.
19. Конструктор с параметрами. Конструктор с аргументами, задаваемыми по умолчанию. Конструктор по умолчанию.
20. Конструктор копирования. Назначение. Копирование по умолчанию. Определение конструктора копирования при наличии в классе указателя на динамический участок памяти.
21. Компонентные данные и компонентные функции. Статические компоненты класса. Внешнее и внутреннее определение компонентных функций.
22. Указатель this . Применение в связанных списках объектов класса.
23. Перегрузка функций. Перегрузка конструкторов.
24. Перегрузка стандартных операций. Два варианта перегрузки: операция функция является компонентной функцией класса, и операция функция является дружественной функцией класса.
25. Перегрузка арифметических операций (+, -, *) и операции включения в поток << для пользовательского класса комплексного числа.
26. Перегрузка арифметической операции '+ ' и операции включения в поток << для пользовательского класса "строка".
27. Перегрузка операций инкремента и декремента на примере класса , который имеет данные (два) числовых типов.
28. Перегрузка операции присваивания. Перегрузка присваивания при наличии в классе указателя на динамический участок памяти. Различие между копированием и присваиванием. Блокировка копирования и присваивания.
29. Наследование. Суть метода. Определение производного класса. Влияния формата определения производного классов и спецификаторов доступа на доступ наследуемых элементов.
30. Наследование. Передача параметров конструктора в базовый класс. Конструкторы с инициализацией по умолчанию в иерархии классов.
31. Множественное наследование. Прямое и косвенное наследование. Иерархия производных классов в виде графа (НАГ). Порядок вызовов конструкторов и деструкторов базовых классов при множественном наследовании.
32. Множественное наследование на примере графического класса "окружность, вписанная в квадрат", производного от базовых классов" окружность" и "квадрат".
33. Дублирование объектов базового класса, косвенно наследуемого при множественном наследовании. Виртуальные базовые классы. Примеры иерархии классов (НАГ), с участием виртуальных базовых классов.
34. Полиморфизма. Понятие виртуальной функции. Режимы раннего и позднего (динамического) связывания. Полиморфные классы.
35. Преобразование типов указателей в иерархии классов. Работа с виртуальными функциями. Пустая и чистая виртуальные функции.

36. Абстрактный класс. Работа с абстрактным классом на примере построение производных классов конкретных фигур (“окружность”, “эллипс”, “квадрат”) на базе абстрактного класса “фигура”.

//38. Работа с виртуальными функциями на примере построения и использования производных классов от класса кнопка- кнопки с номером клавиши, в которых замещена чистая виртуальная функция базового класса - функция отклика на нажатие клавиши.

37. Классы и шаблоны.

38. Библиотека классов ввода – вывода. Понятия потока. Иерархия потоковых классов.

39. Стандартные потоки ввода – вывода (объекты потоковых классов). Операции вставки в поток и извлечения из потока. Особенности операций вставки и извлечения.

40. Форматирование данных при обмене с потоком. Флаги, компонентные функции для форматирования и манипуляторы.

41. Перегрузка операций >> и << для типов пользователя (на примере перегрузки операций ввода- вывода данных некоторого структурного типа и для ввода-вывода объектов некоторого пользовательского класса) .

42. Компонентные функции неформатного ввода – вывода (put( ) , write ( ) , get ( ), getline ( ), read( ))

43. Компонентные функции анализа и коррекции очередного символа потока peek( ) , putback() , ignore( ) и функции анализа и установки позиции ввода вывода.

44. Строковые потоки (обмены в оперативной памяти). Объявление входных и выходных строковых потоков.

45. Копирование строк при использованием входных строковых потоков. Конкатенация строк в выходных строковых потоках. Двунаправленные строковые потоки. Безымянные строковые потоки.

46. Средства языка С++ для работы с файлами и логическими устройствами. Ввод-вывод нижнего уровня. Создание файла. Текстовой и бинарный режим обмена с файлом. Открытие и закрытие файла.

47. Функции работы с файлами на нижнем уровне(чтение и запись в файл, функции анализа и установки позиции в файле и др.).

48. Библиотека файловых потоковых классов. Определение файловых потоков - объектов классов, используя, конструктор по умолчанию. Использование компонентной функции open() для связи физического файла с потоком.

49. Проверка удачности завершения функции open( ) (перегруженная операция !, примененная к потоку). Компонентная функция eof(). Закрытие файлов.

50. Определение потока и присоединение к нему физического файла с помощью конструктора класса с параметрами.

51. Создание файла на нижнем уровне и подсоединение файла к потоку, используя дескриптор файла.

52. Перегрузка операций ввода-вывода данных произвольного структурного типа в файловый поток при текстовом и бинарном обмене с файлом.

53. Стандартная библиотека ввода-вывода верхнего уровня языка Си. Доступ к файлам: открытие и закрытие файлов.

54. Функции бесформатного ввода-вывода в текстовые файлы: посимвольного и построчного.
55. Функции форматного ввода-вывода в текстовой файл.
56. Функции ввода-вывода записями. Функции положения и управление положением указателя файла.