

Вопросы к коллоквиуму

Телематика 2 курс

2020/2021

Технология ООП

1. Различие в подходах к разработке компьютерных приложений (Структурное программирование и Объектно-ориентированное программирование). Классы: определение, область действия. Структуры в С и С++.
2. Последовательность преобразований исходного кода на языке С++ в исполняемый (машинный) код. Основные концепции ООП. Понятие класса в С++. Инкапсуляция, наследование, полиморфизм.
3. Доступ к элементам класса: управление доступом, функции доступа. Инициализация объектов класса: конструкторы. деструкторы. Константные объекты и функции-элементы класса, вложенность классов, указатель this. Динамическое распределение памяти.
4. Указатели и массивы. Многомерные массивы. Массивы указателей. Указатели и ссылки на объекты. Операции приведения типов. Способы обработки исключительных ситуаций.
5. Передача объектов в функции. Возвращение объектов. Модификаторы доступа и дружественные функции, дружественные классы. Переопределение операций. Операции в виде методов класса и в виде дружественных функций.
6. Особенности переопределения унарных и бинарных операций. Наследование: базовые и производные классы. Переопределение элементов базового класса. Модификаторы наследования и модификаторы доступа, их сочетание.
7. Прямые и косвенные базовые классы, множественное наследование. Конструкторы и деструкторы производных классов, неявный и явный вызовы. Виртуальные функции, динамическое связывание, деструкторы. Абстрактные классы.
8. Поточный ввод-вывод, функции работы с потоками. Классы и объекты потоков ввода-вывода (библиотеки STL). Ввод-вывод типов данных, определенных пользователем. Методы работы с файлами.
9. Особенности работы со строками текста. Класс string библиотеки STL. Пример разбора текста. Шаблоны функций, переопределение функций шаблона.
10. Шаблоны классов. Пример разработки шаблона универсальной коллекции типа array (list).
11. Статические члены класса. Особенности определения. Пространства имен в С++. Компоненты библиотеки STL. Контейнеры STL.
12. Методы обработки ошибок в С++. Генерация и перехват исключений, обработчики исключений. Классы-индикаторы исключений, иерархия исключений.
13. Итераторы, их типы и применение (библиотека STL). Примеры использования класса vector (библиотека STL). Типизированные и обобщенные последовательности (коллекции) данных.

14. Лямбда-выражения. Специфика лямбда-выражений. Использование λ -выражений вместо функциональных объектов. Модификация переменных анонимного функционального объекта внутри λ -функции.
15. Семантика перемещения, move-конструктор копирования, move-оператор присваивания. Глобальные функции как предикаты в шаблонных функциях в библиотеке STL. Функциональные объекты и шаблоны классов как предикаты в шаблонных функциях STL. Ключевое слово auto при объявлении типов, примеры.
16. Динамические структуры данных (ассоциативные). Сферы использования и особенности функционирования ассоциативных контейнеров.
17. Основные модули в Qt. Иерархия объектов в Qt. Концепция сигналов и слотов. Преимущества их использования. Подключение сигналов и слотов.
18. Система сборки qmake. Пользовательские и встроенные переменные. Операторы. Комментарии. Области видимости. Циклы. Встроенные и условные функции. Шаблоны проектов. Meta Object Compiler.
19. Виджеты. Свойства виджетов. Взаимоотношения виджетов. Особенности управление памятью в Qt. Создание и удаление объектов.
20. Глобальные определения. Методы отладки в Qt. Интернационализация приложений. Способы изменения геометрии виджета. Менеджеры компоновки. Ресурсы в Qt.
21. Работа со строками в Qt. Проверка ввода. Регулярные выражения.
22. Контейнерные классы в Qt. Итераторы. Последовательные контейнеры.
23. Контейнерные классы в Qt. Алгоритмы работы с контейнерными классами. Ассоциативные контейнеры.
24. События и таймеры в Qt. Перехват и обработка событий. Фильтры событий. Отправка событий.
25. Вывод графики в Qt. Рисование на виджетах. Модель графического представления.
26. Работа с базами данных в Qt. Уровень драйверов. Программный уровень. Уровень пользовательского интерфейса. Архитектура модель-представление.