



1. Структура программы на языках C и C++. Функции C и C++. Перегрузка функций в C++. Параметры функций по умолчанию.
2. Ссылки. lvalue и rvalue ссылки. Передача параметров в функции по ссылке. Автоматическое выведение типа.
3. Классы и объекты в C++. Определение класса с помощью class, struct, union. Ограничение доступа к членам класса в C++. Члены класса и объекта. Методы класса и объекта. Константные члены класса. Схемы наследования.
4. Создание и уничтожение объектов в C++. Конструкторы и деструкторы. Раздел инициализации конструкторов. Способы создания объектов. Явный и неявный вызов конструкторов. Приведение типа.
5. Конструкторы копирования и переноса. Модификатор explicit. Удаление конструктора и default конструктор. Делегирующие и унаследованные конструкторы.
6. Наследование в C++. Построение иерархии классов. Выделение общей части группы классов. Расщепление классов.
7. Множественное наследование. Прямая и косвенная базы. Виртуальное наследование. Понятие доминирования. Порядок создания и уничтожения объектов. Проблемы множественного наследования. Неоднозначности при множественном наследовании.
8. Полиморфизм в C++. Виртуальные методы. Чисто виртуальные методы. Виртуальные и чисто виртуальные деструкторы. Понятие абстрактного класса. Ошибки, возникающие при работе с указателем или ссылкой на базовый класс. Дружественные связи.
9. Обработка исключительных ситуаций в C++. Решение проблем структурного программирования. Блоки try и catch. Блоки try и catch методов и конструкторов. Безопасный код относительно исключений. Обертывание исключения в exception\_ptr. Задачи которые может решать исключение. Проблемы с динамической памятью при обработке исключительных ситуаций.
10. Перегрузка операторов в C++. Операторы .\*, ->\*. Правила перегрузки операторов. Перегрузка операторов =, () и [ ]. Перегрузка операторов ->, \* и ->\*.
11. Перегрузка унарных и бинарных операторов. Проблемы с перегрузкой операторов &&, ||, ,, &. Перегрузка операторов ++, --. Перегрузка операторов приведения типов. Тривалентный оператор spaceship.
12. Шаблоны функций, методов классов и классов в C++. Недостатки шаблонов. Параметры шаблонов. Параметры типы и параметры значения. Шаблоны функций и методов классов. Подстановка параметров в шаблон. Выведение типов параметров шаблона. Явное указание значений типов параметров шаблона при вызове функции. Срезание ссылок и модификатора const.
13. Неявные шаблоны. Протаскивание типа передаваемого параметра через шаблон (шаблон std::forward). Определение типа с помощью decltype. decltype(auto).
14. Специализация шаблонов функций. Шаблоны типов. Шаблоны классов. Полная или частичная специализация шаблонов классов. Параметры шаблонов задаваемых по умолчанию. Шаблоны с переменным числом параметров. Пространства имен.
15. Ограничения накладываемые на шаблоны. Требования к шаблонам (requires). Концепты. Типы ограничений. Варианты определения шаблонов функций и классов с концептами.

16. Проблемы с динамическим выделением и освобождением памяти. Шаблон Holder.  
«Умные указатели» в C++: `unique_ptr`, `shared_ptr`, `weak_ptr`. Связь между `shared_ptr` и `weak_ptr`.
17. Приведение типа в C++: `static_cast`, `dynamic_cast`, `const_cast`, `reinterpret_cast`.  
Контейнерные классы и итераторы. Требования к контейнерам и итераторам.  
Категории итераторов. Операции над итераторами. Цикл `for` для работы с контейнерными объектами.

# Теория

1. Структурное программирование: нисходящая разработка, использование базовых логических структур, сквозной структурный контроль.
2. Преимущества и недостатки структурного программирования. Идеи Энтони Хоара. Преимущества и недостатки объектно-ориентированного программирования.
3. Основные понятия ООП: инкапсуляция, наследование, полиморфизм. Понятие объекта. Категории объектов. Отношения между объектами. Понятие класса. Отношения между классами. Понятие домена.
4. Цикл разработки ПО с использованием ООП: анализ, проектирование, эволюция, модификация. Рабочие продукты объектно-ориентированного анализа.
5. Концепции информационного моделирования. Понятие атрибута. Типы атрибутов. Правила атрибутов. Понятие связи. Типы связей. Формализация связей. Композиция связей. Подтипы и супертипы. Диаграмма сущность-связь.
6. Модель поведения объектов. Жизненный цикл и диаграмма перехода в состояния (ДПС). Виды состояний. События, данные событий. Действия состояний. Таблица перехода в состояния (ТПС). Правила переходов.
7. Модель взаимодействия объектов (МВО). Диаграмма взаимодействия объектов в подсистеме. Типы событий. Схемы управления. Имитирование. Каналы управления.
8. Диаграмма потоков данных действий (ДПДД). Типы процессов: аксессоры, генераторы событий, преобразования, проверки. Таблица процессов (ТП). Модель доступа к объектам (МДО).
9. Домены. Модели доменного уровня. Типы доменов. Мосты, клиенты, сервера.
10. Объектно-ориентированное проектирование. Диаграмма класса. Структура класса. Диаграмма зависимостей. Диаграмма наследования.
11. Структурные паттерны: адаптер (Adapter), декоратор (Decorator), компоновщик (Composite), мост (Bridge), заместитель (Proxy), фасад (Facade). Их преимущества и недостатки.
12. Порождающие паттерны: фабричный метод (Factory Method), абстрактная фабрика (Abstract Factory), строитель (Builder). Их преимущества и недостатки.
13. Порождающие паттерны: одиночка (Singleton), прототип (Prototype), пул объектов (Object Pool). Их преимущества и недостатки.
14. Паттерны поведения: стратегия (Strategy), команда (Command), цепочка обязанностей (Chain of Responsibility), подписчик-издатель (Publish-Subscribe), посредник (Mediator). Их преимущества и недостатки.
15. Паттерны поведения: стратегия (Strategy), посетитель (Visitor), опекун (Memento), шаблонный метод (Template Method), хранитель (Holder), итератор (Iterator), свойство (Property). Их преимущества и недостатки.