

# Операционные системы и языки

## Темы лекций

1. Особенности языка C++. Стандарт ANSI. Отличия от C.
2. Парадигма ООП. Специфика интерфейса ОО программ.
3. Основные виды программирования. Их недостатки и достоинства. Преимущества ООП.
4. Понятие класса. Отношение "объект - класс".
5. Доступ к членам класса. Закрытые и открытые члены класса.
6. Реализация методов класса. Использование inline функций. Соглашения о вызовах функций.
7. Понятие полиморфизма. Проявления полиморфизма.
8. Механизм наследования.
9. Модификаторы доступа и наследования. Как изменяются атрибуты элементов класса при наследовании?
10. Смысл инкапсуляции.
11. Интерфейс и реализация класса. Объявление класса. Реализация его методов. Встраиваемая реализация. Постоянные функции-члены.
12. Конструкторы и деструкторы. Использование стандартного конструктора.
13. Операторы new и delete. Создание объектов в области динамической памяти. Удаление объектов. Утечка памяти.
14. Данные-члены в динамической памяти. Указатель this. Указатели и константы.
15. Ссылки. Их отличие от указателей. Передача функции аргументов по ссылке. Возвращение нескольких значений.
16. Передача ссылок как средство повышения эффективности. Преимущества и недостатки использования ссылок.
17. Перегрузка функций-членов класса. Использование значений по умолчанию.
18. Конструктор по умолчанию. Перегрузка конструкторов. Инициализация объекта. Конструктор копий.
19. Перегрузка операторов. Перегрузка префиксных и постфиксных операторов.
20. Перегрузка оператора суммы. Ограничения на перегрузку операторов.
21. Проблемы при присваивании одного объекта другому. Поверхностное и глубокое копирование данных. Перегрузка оператора присваивания.
22. Дружественные функции. Использование дружественных функций при перегрузке операторов. Перегрузка оператора вывода.
23. Преобразование типов данных. Операторы преобразования.
24. Понятие наследования. Синтаксис происхождения классов. Модификаторы доступа наследования.
25. Конструкторы и деструкторы при наследовании.
26. Переопределение функций при наследовании.
27. Совместимость типов при наследовании.
28. Виртуальные функции. Виртуальные деструкторы и конструкторы копий. Недостатки применения виртуальных методов.
29. Раннее и позднее связывание.
30. Полиморфизм и виртуальные методы.
31. Абстрактный класс
32. Множественное наследование.
33. Шаблоны функций. Требования к фактическим параметрам шаблона. Отождествление типов аргументов.
34. Шаблоны классов. Наследование в шаблонах классов.
35. Стандартная библиотека классов. Vector, List, Map и другие.

36. Исключения в C++. Try, catch, throw.

### **ПЕРЕЧЕНЬ ВОПРОСОВ (дополнительных)**

1. В чем различие между видами и методами (способами) абстракции?
2. Классы операций абстракции данных. Полнота класса операций.
3. Какие существуют механизмы для создания новых типов?
4. В чем смысл декомпозиции при составлении программ?
5. В чем разница между struct, class и union?
6. Обращение к элементам класса («.», «->», this).
7. Динамическое распределение памяти в Си/Си++.
8. Перегрузка унарных операций.
9. Перегрузка бинарных операций.
10. Перегрузка операции присваивания.
11. Перегрузка операций new и delete.
12. Перегрузка вызова функции.
13. Перегрузка операции индексирования.
14. Перегрузка операции постфиксного и префиксного типа.
15. Доступ к членам родительского класса при наследовании.
16. Простое наследование.
17. Что такое абстрактный класс и чем может быть вызвана необходимость построений абстрактного класса?
18. Чем виртуальные функции отличаются от перегружаемых?
19. Совместимость типов. Какую роль в достижении совместимости объектов могут играть абстрактные классы?
20. Почему шаблоны называют параметризованными типами?
21. Когда следует в программе применять шаблоны, а когда нет?
22. Чем шаблоны лучше макроподстановок?
23. Для каких типов данных может применяться конкретный шаблон, а для каких нет?
24. В чем разница между классом и шаблоном класса?
25. Что может выступать в качестве параметра для шаблона класса?
26. В чем основное преимущество использования библиотеки потокового ввода-вывода?
27. Как работает операция внесения (на примере)?
28. Что такое манипулятор ввода - вывода?