**На 3:**

"1) Краткий обзор основных парадигм программирования. Основы объектноориентированного программирования на С++.

2) Основные принципы ООП: инкапсуляция, наследование, полиморфизм.

3) Описание класса. Определение объектов с помощью классов. Общий синтаксис

определения класса. Управление доступом к элементам класса. Классы и структуры С++.

Привести примеры программ.

4) Объявление и определение методов класса. Манипуляции с состоянием объекта. Класс

как область видимости. Спецификатор inline. Указатель this . Привести примеры

программ.

5) Область действия класса и доступ к элементам класса. Отделение интерфейса от

реализации. Привести примеры программ.

6) Управление доступом к элементам класса. Привести примеры программ.

7) Конструкторы класса. Конструктор умолчания. Конструкторы с двумя и более

параметрами. Привести примеры программ.

8) Конструкторы класса. Конструктор копирования. Конструктор преобразования.

Деструкторы. Автоматическая генерация конструкторов и деструкторов. Список

инициализации. Привести примеры программ.

9) Порядок вызова конструкторов и деструкторов. Динамическое распределение памяти

при создании объектов класса. Привести примеры программ.

10) Константные объекты и константные элементы-функции. Статические элементы

класса. Организация сцепленного вызова функций. Привести примеры программ.

11) Композиция или вложенные классы. Привести примеры программ.

12) Статический полиморфизм. Описание друзей классов. Дружественные функции и

дружественные классы. Привести примеры программ.

13) Понятие перегрузки операций. Основные принципы перегрузки операций. Запреты на

перегрузку операций. Основные правила перегрузки операций.

14) Перегрузка операций передачи в поток и извлечения из потока. Привести примеры

программ.

15) Перегрузка бинарных операций. Перегрузка унарных операций. Привести примеры

программ.

16) Перегрузка операций декремента и инкремента. Преобразование типов. Привести

примеры программ.

17) Виды отношений между классами. Наследование и полиморфизм. Базовые и

производные классы.

18) Одиночное наследование. Правила наследования. Переопределение методов в

производном классе, применение конструкторов и деструкторов. Привести примеры

программ.

19) Правила видимости при наследовании. Открытое (public) наследование. Привести

примеры программ.

20) Правила видимости при наследовании. Закрытое (private) наследование. Привести

примеры программ.

21) Правила видимости при наследовании. Защищенное (protected) наследование.

Привести примеры программ.

22) Преобразование указателей при выполнении одиночного наследования. Приведение

указателей при открытом, защищеном и закрытом наследовании.

23) Преобразование указателей при выполнении множественного наследования.

Перекрытие имен при преобразовании указателей. Привести примеры программ.

24) Динамический полиморфизм, механизм виртуальных функций. Абстрактные базовые

классы и конкретные классы. Интерфейсы. Привести примеры программ.

25) Виртуальные деструкторы. Реализация виртуальных функций. Динамическое

связывание. Таблица виртуальных методов. Привести примеры программ.

26) Множественное наследование и интерфейсы. Видимость при множественном

наследовании. Виртуальные базовые классы. Привести примеры программ.

27) Параметрический полиморфизм. Понятие и синтаксис шаблона. Параметры шаблона. Шаблонны класса с одним параметром, синтаксис. Привести примеры программ.

28) Параметрический полиморфизм. Шаблонны класса с несколькими типами, синтаксис.

Аргументы шаблона без типа. Шаблон класса в качестве аргумента для шаблона класса.

Аргументы по умолчанию. Привести примеры программ.

29) Параметрический полиморфизм. Шаблоны класса и методы класса, как шаблоны

функций. Требования к шаблонам функций и шаблонам классов. Отделение декларации

от реализации в шаблонах класса. Раздельная реализация класса. Привести примеры

программ.

30) Параметрический полиморфизм. Шаблоны класса и друзья, перегрузка операторов. Шаблоны класса и наследование. Шаблоны и виртуальные функции. Привести примеры

программ.

**На 4:**

1) Описывать механизм обработки исключительных ситуаций. Понятие исключения.

Синтаксис механизма обработки исключительных ситуаций. Правила выбора обработчика

исключений.

2) Описывать блоки обработки исключительных ситуаций: блок try, catch, throw. Правила

и синтаксис. Привести примеры программ.

3) Описывать исключения при одиночном и множественном наследовании. Перехват

исключений. Спецификации исключительных ситуаций. Привести примеры программ.

4) Описывать механизм обработки исключительных ситуаций при обработке потоков.

Генерация исключений в конструкторах. Перенаправление исключительных ситуаций.

Привести примеры программ.

5) Описывать последовательность действий при возникновении исключительных

ситуаций. Стандартные исключения.

6) Описывать порядок записи обработчиков исключений. Исключительная ситуация,

генерируемая оператором new. Задание собственной функции завершения. Задание

собственного неожиданного обработчика. Привести примеры программ.

7) Описывать классы с самоадресацией. Описание динамических структур с помощью

классов. Синтаксис и определение. Привести примеры программ.

8) Описывать стандартную библиотеку шаблонов. Общее понятие о контейнере. Группы

контейнеров.

9) Описывать итераторы. Общее понятие об итераторе. Категории итераторов. Итераторы

ввода и вывода. Итераторы произвольного доступа. Привести примеры программ.

10) Описывать итераторы. Операции с итераторами. Однонаправленные и

двунаправленные итераторы. Вспомогательные итераторы. Привести примеры программ.

11) Разрабатывать программы с использованием контейнера последовательностей Vector -

векторы. Приводить примеры программ.

12) Разрабатывать программы с использованием класса List – двухсвязных списков.

Приводить примеры программ.

13) Разрабатывать программы с использованием класса Deque – очередей. Приводить

примеры программ.

14) Разрабатывать программы с использованием массивов значений в С++. Приводить

примеры программ.

15) Разрабатывать программы с использованием ассоциативного контейнера multiset

(мультимножество). Приводить примеры программ.

16) Разрабатывать программы с использованием ассоциативного контейнера set

(множество). Приводить примеры программ.

17) Разрабатывать программы с использованием ассоциативного контейнера multimap.

Приводить примеры программ.

18) Разрабатывать программы с использованием ассоциативного контейнера map.

Приводить примеры программ.

19) Разрабатывать программы с использованием адаптера контейнеров stack. Приводить

примеры программ.

20) Разрабатывать программы с использованием адаптера queue. Приводить примеры

программ.

21) Разрабатывать программы с использованием адаптера priority\_queue. Приводить

примеры программ.

22) Разрабатывать программы с использованием стандартных алгоритмов библиотеки

STL. Приводить примеры программ.

23) Разрабатывать программы с использованием алгоритмов сортировки библиотеки STL.

Приводить примеры программ.

24) Разрабатывать программы с использованием алгоритмов поиска библиотеки STL.

Приводить примеры программ.

25) Разрабатывать программы с использованием алгоритмов для заполнения

определенным значением некоторого диапазона библиотеки STL. Приводить примеры

программ.

26) Разрабатывать программы с использованием алгоритмов для сравнения на равенство

последовательностей значений библиотеки STL. Приводить примеры программ.

27) Разрабатывать программы с использованием математических алгоритмов библиотеки

STL. Приводить примеры программ.

28) Разрабатывать программы с использованием алгоритмов для работы с множествами

библиотеки STL. Приводить примеры программ.

**На 5:**

1) Методами решения задач по теме ""Объектно-ориентированное программирование"" (по

вариантам в приложении).

2) Методами решения задач по теме ""Наследование и полиморфизм"" (по вариантам в

приложении).

3) Методами решения задач по теме ""Шаблоны классов"" (по вариантам в приложении).

4) Методами решения задач по теме ""Обработка исключительных ситуаций"" (по

вариантам в приложении).

5) Методами решения задач по теме ""Стандартная библиотека шаблонов STL"" (по

вариантам в приложении).