**Вопросы для подготовки к экзамену**

**по дисциплине «Алгоритмизация и программирование» (1 семестр)**

|  |  |
| --- | --- |
| 1 | Схемы алгоритмов программ: обозначения и правила составления. Структурная и неструктурная схемы алгоритма вычисления определенного интеграла методом прямоугольников. |
| 2 | Основные и дополнительные структурные алгоритмические конструкции и их особенности. Примеры их использования в схемах алгоритмов. |
| 3 | Синтаксис и семантика языков программирования. Способы описания синтаксиса языка. Расширенная форма Бэкуса-Наура. Алфавит языка С++. |
| 4 | Способы представления данных разных типов в памяти компьютера. Примеры внутреннего представления данных различных типов. |
| 5 | Скалярные типы данных языка С++ и основные операции над ними. Совместимость и несовместимость типов данных. Операции преобразования типов данных. |
| 6 | Два способа организации операций ввода-вывода в языке С++. Примеры. |
| 7 | Основные управляющие операторы языка С++, используемые для программирования разветвленных процессов: синтаксис операторов, их особенности и примеры использования. |
| 8 | Основные управляющие операторы языка С++, используемые для программирования циклических процессов: синтаксис операторов, их особенности и примеры использования. |
| 9 | Поисковый цикл. Неструктурная и структурная реализации поискового цикла в алгоритме и программе. |
| 10 | Статические и автоматические массивы языка С++ и их особенности. Основное правило адресной арифметики. Примеры различных способов обращения к элементам массивов. |
| 11 | Адреса данных в памяти. Типы данных, используемые для работы с адресами. Операции получения адреса и разыменования, а также процедуры получения и освобождения памяти и область их применения. |
| 12 | Отличие статических и автоматических массивов от динамических. Примеры объявления массивов каждого типа и особенности работы с ними. |
| 13 | Структурный тип данных С++. Примеры статических и динамических структур, а также операций над их элементами. |
| 14 | Способы передачи данных в подпрограмму и особенности их применения. Примеры. |
| 15 | Классы памяти переменных, примеры их объявления и особенности переменных перечисленных классов. |
| 16 | Подпрограммы языка С++. Способы передачи параметров в подпрограмму. Примеры. Параметры-массивы, параметры-строки и параметры-структуры. |
| 17 | Способы разработки универсальных подпрограмм. «Развертки» многомерных массивов и особенности их применения. |
| 18 | Способы разработки универсальных подпрограмм. Параметры процедурного типа: область применения и пример описания. |
| 19 | Структура многофайловой программы, назначение и особенности описания h-файлов, диаграммы компоновки. |
| 20 | Рекурсия: особенности программирования с использованием рекурсии, фрейм активации, достоинства и недостатки рекурсивных подпрограмм. |
| 21 | Списковые структуры данных: создание элемента, добавление элемента к списку, удаление элемента из списка. Область применения списковых структур данных. |
| 22 | Понятия «файл», «каталог», «дисковод», «полное имя файла». Представление информации в файлах различных типов и особенности работы с файлами. |

|  |  |
| --- | --- |
| 23 | Двоичный файл: отличия от текстового, пример создания и обработки двоичного файла. |
| 24 | Текстовый файл: отличия от двоичного, пример создания и обработки текстового файла. |
| 25 | Структурный и объектный подходы к программированию. Декомпозиция программного обеспечения. Области применения подходов. |
| 26 | Класс в языке С++: описания класса, его поля и методы, объявление объектов класса, ограничение доступа к полям и методам объекта. Неявный параметр this и пример его использования. |
| 27 | Конструктор, типы конструкторов. Инициализация полей объектов при наличии и при отсутствии конструкторов. |
| 28 | Основные отношения между классами. Примеры диаграмм классов. |
| 29 | Наследование в объектно-ориентированном программировании: способ реализации, пример. Особенности описания конструкторов производных классов. |
| 30 | Композиция в объектно-ориентированном программировании: способ реализации, пример. Особенности описания конструкторов классов-агрегатов. |
| 31 | Наполнение в объектно-ориентированном программировании: способ реализации, пример. Особенности описания конструкторов и деструкторов классов-агрегатов. |
| 32 | Полиморфное наследование в языке С++. Раннее и позднее связывание: описание в программе и механизмы реализации. |

**2 вопрос - теория объектно-ориентированного программирования на примере** **задачи**.

Разработать диаграмму классов для реализации двух заданных объектов. При построении классов необходимо выбрать и реализовать отношение между классами: наследование, композицию или наполнение. Классы должны включать методы инициализации значений и их вывода. Для проверки работоспособности реализации необходимо составить тестирующую программу.

*Примеры пар объектов*: буква и слово, число и массив чисел, число и строка, длина которой задана числом и т.п.