

## Вопросы к экзамену по курсу программирование (1 курс)

**2020/2021**

№	Вопрос	Уровень сложности
1	Аппаратные и программные средства компьютеров. Гарвардская и принстонская архитектуры.	I
2	Понятие о программном обеспечении. Предмет программирования. Системное, прикладное и инструментальное программное обеспечение (ПО). Основные технологические этапы разработки программ.	I
3	Организация памяти компьютера. Оперативная память: уровни, адресация. Внешняя память: файлы, прямой и последовательный доступ к данным.	I
4	Информация и её представление в ЭВМ. Информационное сообщение. Дискретные и аналоговые информационные сообщения. Дискретизация и кодирование. Алфавит. Двоичный алфавит ЭВМ.	I
5	Представление в ЭВМ символьной информации. Однобайтовые, двухбайтовые и многобайтовые коды (ASCII, UNICODE). Логические данные и их представление в ЭВМ.	I
6	Представление в памяти ЭВМ целых чисел. Двоично-десятичный, прямой и дополнительный двоичный коды.	II
7	Особенности представления целых чисел на ограниченной разрядной сетке. Переполнение в целочисленной арифметике: арифметика по модулю.	I
8	Представление в памяти ЭВМ вещественных чисел. Мантисса и порядок. Представление по основаниям 2 и 16. Основные свойства представления вещественных чисел на ограниченной разрядной сетке.	II
9	Арифметические операции над вещественными числами на ограниченной разрядной сетке. Сравнение вещественных чисел на точное равенство.	I
10	Понятие языка программирования. Алгоритмы, структуры данных и программы. Основы классификации языков программирования. Язык программирования C++, его соотношение с современными стандартами языка C.	I
11	Основные элементы программы на C++. Символы, ключевые слова, идентификаторы, комментарии. Предложения (инструкции, операторы) C++.	I
12	Функции как структурные компоненты программы на C++. Библиотечные функции.	I
13	Неименованные (литералы) и именованные константы. Символьные и строковые константы.	I

14	Переменные. Понятие типа переменной. Области видимости переменных.	II
15	Выражения, операции и операторы. Приоритет операторов.	I
16	Типизация данных. Концепция типа данных. Статическая и динамическая типизация в языках программирования.	II
17	Базовые (простые) и структурированные типы данных. Классификация типов данных. Простые стандартные типы данных в C++.	I
18	Операции. Операция присваивания и основные правила преобразования типов. Операция присваивания в правой части выражения. Множественное присваивание.	II
19	Арифметические операции. Приоритеты арифметических операций. Префиксная и постфиксная формы операций инкремента и декремента. Арифметические операции с присваиванием.	II
20	Операции сравнения и логические операции. Побитовые операции. Операция последовательного вычисления “,”.	II
21	Приоритеты операций. Явные и неявные преобразования типов в выражениях. Иерархия типов данных. Ошибки при преобразовании типов.	II
22	Ввод-вывод с использованием стандартных потоков ввода-вывода. Использование управляющих последовательностей и манипуляторов.	I
23	Цели и принципы структурного программирования.	I
24	Базовые управляющие структуры. Изображение управляющих структур на блок-схемах. Построение структурированных диаграмм. Конструирование структурных алгоритмов.	I
25	Ветвления. Реализация ветвлений в C++.	I
26	Побочные эффекты и точки последовательности.	III
27	Условный оператор. Оператор выбора <b>switch</b> .	I
28	Повторения. Цикл «пока» ( <b>while</b> ). Цикл «до тех пор» ( <b>do while</b> ).	I
29	Цикл со счётчиком ( <b>for</b> ).	I
30	Взаимозаменяемость конструкций циклов. Примеры циклических алгоритмов.	II
31	Приёмы программирования циклов: циклы <b>for</b> с двумя счётчиками, циклы <b>for</b> без тела, бесконечные циклы.	II
32	Инструкции перехода <b>break</b> и <b>continue</b> , особенности их работы в различных конструкциях циклов.	I
33	Структуры. Объединения.	I
34	Массивы. Размещение массивов в памяти. Одномерные и многомерные массивы. Описание одномерных массивов. Операция индексирования.	II
35	Строки символов в C++. Строки символов, завершающиеся нулевым байтом.	I
36	Двумерные массивы. Размещение в памяти двумерных массивов. Представление матриц в виде двумерных массивов. Элементарные алгоритмы обработки двумерных массивов.	II
37	Оптимизация программ. Методы оптимизации циклов.	II

38	Многомерные массивы. Массивы структур. Операция <a href="#">sizeof</a> .	I
39	Сортировка: постановка задачи сортировки массивов. Бинарный поиск в отсортированном массиве.	III
40	Асимптотическая сложность алгоритмов.	III
41	Теорема о предельной сложности сортировки, основанной на попарном сравнении элементов.	III
42	Сортировка выбором (выделением). Оценка сложности сортировки выбором.	III
43	Сортировка вставкой (включениями). Оценка сложности сортировки вставкой.	III
44	Сортировка обменом (метод «пузырька»). Оценка сложности сортировки обменом.	III
45	Указатели. Объявление указателя. Операция взятия адреса. Операция разыменования. Инициализация указателей. Присваивание указателей.	I
46	Неявное и явное приведение указательных типов с помощью <a href="#">reinterpret_cast</a> .	II
47	Арифметические действия с указателями. Сравнение указателей.	II
48	Указатели и одномерные массивы. Операция индексации как сочетание смещения в памяти и последующего разыменования.	II
49	Указатели и многомерные массивы. Интерпретация идентификатора многомерного массива как указателя на массив.	II
50	Массивы указателей. Указатели на структуры. Указатели на указатели.	I
51	Указатели и строки. Инициализация указателей на <a href="#">char</a> .	I
52	Ссылки, L-value и R-value.	I
53	Статическое и динамическое выделение памяти. Операторы <a href="#">new</a> и <a href="#">delete</a> .	II
54	Динамическое выделение памяти для переменных и структур.	I
55	Динамическое выделение памяти для одномерных и двумерных массивов.	I
56	Функции как средство структурной декомпозиции программ. Структура функций. Прототипы функций.	I
57	Способы передачи данных в вызываемую функцию и из вызываемой функции. Передача аргументов в функцию по значению и по ссылке.	II
58	Передача массивов по ссылке. Передача указателей.	I
59	Передача указателей на функции.	II
60	Значения параметров функции по умолчанию. Инициализация параметров в прототипе функции.	I
61	Перегрузка функций.	I
62	Возвращаемые значения. Структуры как возвращаемые значения.	I
63	Возврат ссылок и использование функций в левой части оператора присваивания.	III
64	Области видимости и классы памяти переменных.	II

65	Рекурсия. Рекурсивные определения. Сравнение рекурсивной и итерационной реализаций алгоритмов. Глубина и текущий уровень рекурсии. Конечность рекурсии.	II
66	Структура рекурсивных процедур: вычисления на рекурсивном спуске и рекурсивном возврате.	II
67	Рекурсивный алгоритм быстрой сортировки.	III
68	Повторное использование исходного кода. Шаблоны функций.	II
69	Класс сложности алгоритма: амортизированная константная.	III
70	Декомпозиция как средство преодоления сложности. Алгоритмическая и объектная декомпозиции.	III
71	Инкапсуляция. Классы. Идиома RAII: выделение ресурса есть инициализация.	II
72	Конструкторы. Перегрузка конструкторов. Деструкторы.	II
73	Перегрузка конструктора копирования при работе с указателями членами класса.	III
74	Семантика перемещения.	III
75	Размещение объектов в памяти. Указатель <a href="#">this</a> .	II
76	Статические переменные и константы – члены класса. Статические функции – члены класса.	III
77	Константные объекты. Константные функции.	III
78	Связанные динамические структуры данных. Организация связанных динамических структур данных. Основные виды связанных структур данных.	III
79	Перегрузка операций. Абстрактные типы данных. Паттерн итератор. Примеры абстрактных типов данных.	III
80	Дружественные классы и дружественные функции.	III
81	Шаблоны классов. Реализация стека как шаблона классов.	III
82	Использование умного указателя <code>unique_ptr</code> .	III
83	Библиотека STL. Примеры использования стандартных контейнеров.	III
84	Наследование классов. Наследование членов базового класса в производном классе. Добавление полей, методов и переопределение методов в производных классах.	III
85	Управление доступом производного класса к унаследованным членам.	II
86	Конструкторы производных классов. Деструкторы производных классов.	III
87	Перегрузка оператора присваивания в производных классах.	II
88	Виртуальные функции и позднее связывание. Полиморфизм.	III
89	Абстрактные классы и чистые виртуальные функции. Организация доступа к данным через явный интерфейс.	II
90	Удаление объекта производного класса через указатель на базовый класс. Виртуальные деструкторы.	II
91	Динамическая информация о типах (RTTI). Операторы <code>typeid</code> и <code>dynamic_cast</code> .	II

92	Этапы разработки объектно-ориентированных программных систем: объектно-ориентированный анализ, объектно-ориентированное проектирование, объектно-ориентированное программирование.	III
93	Композиция классов и её реализация.	III
94	Формирование требований к системе: диаграммы использования. Диаграммы классов.	III
95	Понятие об архитектуре объектно-ориентированной программы: модель-представление-контроллер, трёхзвенная архитектура (клиент, сервер, база данных)	III