**УТВЕРЖДАЮ**

Зав. кафедры программной инженерии БГТУ

к.т.н., доцент\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.В. Смелов

протокол № \_ от \_\_ \_\_\_\_\_\_\_\_ 202\_ г.

**Экзаменационные вопросы**

**летняя экзаменационная сессия**

Кафедра: **программной инженерии**

Дисциплина: **Основы алгоритмизации и программирования**

Курс: **1**

Специальность:

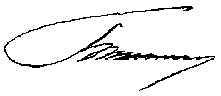
**6-05-0611-01 Информационные системы и технологии. Профилизация:**

**Информационные системы**

**6-05-0612-01 Программная инженерия. Профилизация:**

**Программное обеспечение информационных технологий**

| **Нвп** | **Экзаменационный вопрос** |
| --- | --- |
| 1 | Использование функций. Создание и использование простой функции. Прототипы функций. Вызов по значению и вызов по ссылке. |
| 2 | Использование функций. Использование указателей для связи между функциями. Рекурсия. Равноправность функций в языке Си. |
| 3 | Использование функций. Параметры и аргументы функций. Формальные и фактические параметры. Аргумент типа void. Символьные параметры. Целочисленные параметры. Параметры в формате чисел с плавающей точкой. Параметры в формате чисел двойной длины. Массивы в качестве параметров. Аргументы по умолчанию. |
| 4 | Использование функций. Возвращение значения функцией: оператор return. Типы функций. Функции типа void. Функции типа char. Функции типа int. Функции типа long. Функции типа float. Функции типа double. |
| 5 | Использование функций. Важные возможности C++. Встраивание (inline). Перегрузка (overloading). |
| 6 | Использование функций. Функции с переменным числом параметров. Многоточие (...). Задание числа дополнительных параметров с помощью первого параметра. Определение конца списка параметров с помощью параметра индикатора. Использование специального набора макроопределений. Список указателей переменной длины на char (конкатинация строк). Изменение параметров по числу и по типу. Дополнительные примеры функций с произвольным числом параметров. |
| 7 | Использование функций. Аргументы функции main(). Строки. Целые числа. Числа с плавающей точкой. |
| 8 | Использование функций. Области видимости. Локальные и глобальные переменные. Сложности в правилах области действия (scope rules). Неопределенные символы в программе на С. Использование переменной с файловой областью действия. Приоритет переменных с файловой и локальной областями действия. Проблемы области действия в C++. Операция уточнения области действия в C++. |
| 9 | Использование функций. Математические функции. |
| 10 | Использование функций. Указатель на функцию. Указатели на функции. Указатели на методы. Указатель на функцию и динамическое связывание. Таблицы функций, вызов по имени. Указатель на функцию как средство параметризации алгоритма. |
| 11 | Использование функций. Массив указателей на функции. Шаблоны функций в С++. Основные понятия. Параметры шаблонов функций. |
| 12 | Использование функций. Шаблоны функций. Аргументы по умолчанию. Функции округления. Компиляция программ, состоящих из двух или более функций. |
| 13 | Классы памяти. "Зоопарк" классов памяти. Автоматические переменные. Регистровые переменные. Статические переменные. Внешние переменные. Внешние статические переменные. |
| 14 | Классы памяти. Объявление переменных на внутреннем уровне. Объявление переменных на внешнем уровне. |
| 15 | Классы памяти. Переменные класса volatile. Ключевое слово mutable. Классы памяти и область действия. Правила области действия переменной. Операция уточнения области действия в C++. Выбор класса памяти. |
| 16 | Классы памяти. Пространства имен. Функции и классы памяти. Объявления функций на внешнем уровне. |
| 17 | Дополнительные приемы программирования. Совместимость типов. Определение совместимости типов в ANSI С. |
| 18 | Дополнительные приемы программирования. Идентичный тип. Перечисляемые типы. Типы массивов. Типы функций. Типы структур и объединений. Типы указателей. Совместимость нескольких исходных файлов. |
| 19 | Дополнительные приемы программирования. Макроопределения. Определение макросов. Макросы и параметры. Сложности при раскрытии макросов. Создание и использование собственных макросов. Макросы, поставляемые вместе с компилятором. Выбор макроопределения или функции. |
| 20 | Дополнительные приемы программирования. Директивы препроцессора. Директива #define. Директива #include. Заголовочные файлы. Правильное использование заголовочных файлов. Более эффективное использование заголовочных файлов. Новый стиль заголовков. Предварительная компиляция заголовочных файлов. Файлы limits.h и float.h. |
| 21 | Дополнительные приемы программирования. Директивы #ifdef и #endif. Директива #undef. Директива #ifndef. Директива #if. Директива #else. Директива #elif. |
| 22 | Дополнительные приемы программирования. Директива #line. Директива #error. Директива #pragma. |
| 23 | Дополнительные приемы программирования. Оператор defined. Условная компиляция. |
| 24 | Дополнительные приемы программирования. Дополнительные операции препроцессора. Операция подстановки строки (#). Операция конкатенации (##). Операция подстановки символа (#@). |
| 25 | Дополнительные приемы программирования. Обработка ошибок: perror(). Модели памяти. Модель tiny. Модель small. Модель medium. Модель compact. Модель large. Модель huge. |
| 26 | Дополнительные приемы программирования. Модификаторы функций. Модификаторы cdecl и pascal. |
| 27 | Файлы в C. Файлы и потоки. Связь с файлами. Закрытие потоков. Обработка ошибок в C и C++. Переменная errno и коды ошибок. |
| 28 | Файлы в C. Переключение и работа с файлами. Переключение вывода. Переключение ввода. Комбинированное переключение. Конвейерная пересылка. Текстовые и бинарные (двоичные) файлы. |
| 29 | Файлы в C. Потоковый ввод-вывод. Соединение и отсоединение потока от файла. Функция fopen. Функция fclose. Функция freopen. |
| 30 | Файлы в C. Работа с индикаторами ошибки, позиции и конца файла (ferror, clearerr, feof, rewind, fseek, fsetpos, ftell, fgetpos). Блочный ввод-вывод (fwrite, fread). |
| 31 | Файлы в C. Символьный ввод-вывод. Функции fputc, putc, fgetc, getc, ungetc. Функции fputs и fgets. Функции fprintf, fscanf. |
| 32 | Файлы в C. Работа с буферами (setvbuf, setbuf, fflush). Стандартные потоки (putchar, getchar, puts, gets, perror). Служебные функции для работы с файлами (remove, rename, tmpfile, tmpnam). |
| 33 | Файлы в C. Низкоуровневый ввод и вывод в С. Форматированный вывод. Использование функций printf() и fprintf(). |
| 34 | Файлы в C++. Структура стандартной библиотеки ввода-вывода. Предопределенные потоки. От файла STREAM.H к файлу IOSTREAM.H. Список классов iostream. |
| 35 | Файлы в C++. Простота ввода/вывода в С ++. Потоки cin, cout и сегг. Операции выделения ( >> ) и вставки ( << ). Опции ввода/вывода в С/С++. |
| 36 | Файлы в C++. Операции и методы классов. Вывод символов в C++. Преобразование системы счисления в C++. Форматирование строк в C++. Форматирование чисел в C++. Файловый ввод и вывод в C++. |
| 37 | Файлы в C++. Классы ios\_base и ios. Потоки вывода. Функции open, close и is\_open. Функция flush. Функция put. Функция seekp и tellp. Функция write. Функция str. Классы потокового вывода. |
| 38 | Файлы в C++. Потоки ввода. Функция gcount. Функция get. Функция getline. Функция ignore. Функция peek. Функция putback. Функция read. Функция seekg. Функция sync. Функция unget. Функция str. Классы потокового ввода. |
| 39 | Файлы в C++. Потоки ввода-вывода. Классы буферизированных потоков. Класс строковых потоков. Форматирование потока. Резидентные в памяти потоки. Буферы и синхронизация. |
| 40 | Файлы в C++. Условные признаки файлов в C++. Опрос и установка состояния потока. Ошибки потоков. Часто применяемые функции. Двоичные файлы. Объединение программ на С и С++. Использование спецификатора extern "С". |
| 41 | Файлы в C++. Манипуляторы. Манипуляторы без параметров. Манипуляторы с одним параметром. Манипуляторы с несколькими параметрами. |
| 42 | Функции для работы с файлами. Полезные функции для работы с файлами. clearerr(). fclose(). fcloseall(). fdopen(). feof(). ferror(). fflush(). fgetc(). fgetchar(). fgetpos(). |
| 43 | Функции для работы с файлами. fgets(). filelength(). fileno(). flushall() (fflush()). fopen(). fprintf(). fputc(). fputchar(). fputs(). fread(). |
| 44 | Функции для работы с файлами. freopen(). fscanf(). fseek(). fsetpos(). fstat(). ftell(). fwrite(). getc(). getchar(). gets(). getw(). |
| 45 | Функции для работы с файлами. perror(). printf(). putc(). putchar(). puts(). putw(). remove(). rename(). rewind(). scanf(). setbuf(). |
| 46 | Функции для работы с файлами. setvbuf(). sprintf(). sscanf(). tmpfile(). tmpnam(). ungetc(). vfprintf(). vfscanf(). vprintf(). vsprintf(). vsscanf(). |
| 47 | Функции для работы с файлами. Низкоуровневый ввод и вывод. close(). lseek(). open(). read(). unlink() (\_unlink()). write(). |
| 48 | Структуры и другие типы данных. Структуры С и C++. Структуры С и C++: синтаксис и правила. Структуры C++: расширения синтаксиса и правил. Доступ к элементам структуры. Инициализация структуры. Передача структур в функции. |
| 49 | Структуры и другие типы данных. Массив структур. Описание массива структур. Определение элементов массива структур. Вложенные структуры. |
| 50 | Структуры и другие типы данных. Использование указателей на структуры. Описание и инициализация указателя на структуру. Доступ к элементу структуры при помощи указателя. |
| 51 | Структуры и другие типы данных. Структуры и функции. Передача структур по значению. Передача структур посредством указателей. Передача структур посредством ссылок. Передача элементов структуры. Передача массива структур. |
| 52 | Структуры и другие типы данных. Функция - член структуры. Особенность возврата const. Возврат структуры из функции. |
| 53 | Структуры и другие типы данных. Структуры и битовые поля. Структуры. Их дальнейшее использование. Объединения. Объединения: синтаксис и правила. Создание простого объединения. |
| 54 | Структуры и другие типы данных. Вспомогательные средства. Использование typedef. Использование enum. Сложные формы данных. Функции работы с датой и временем. |
| 55 | Динамические структуры данных C++. Линейные однонаправленные списки. Общие сведения. Однонаправленные списки без заглавного звена. Построение списка с заглавным звеном. Удаление списка из памяти. |
| 56 | Динамические структуры данных C++. Операции над списками с заглавным звеном. Поиск звена. Включение звена после звена (1-й случай). Включение звена с заданным информационным полем перед звеном (2-й случай). Удаление звена после звена (1-й случай). Удаление звена на которое указывает ссылка (2-й случай). |
| 57 | Динамические структуры данных C++. Ортогональные списки. Реализация операций над ортогональными списками. |
| 58 | Динамические структуры данных C++. Кольцевые списки. Построение и вывод кольца. Основные операции. |
| 59 | Динамические структуры данных C++. Списки магазинного типа. Списки магазинного типа. Очереди. Формирование очереди. Добавление звена к очереди. Удаление звена из очереди. |
| 60 | Динамические структуры данных C++. Стек. Формирование стека. Включение звена в стек. Удаление звена из стека. |
| 61 | Динамические структуры данных C++. Дек. |
| 62 | Динамические структуры данных C++. Линейные двунаправленные списки. Формирование линейного двунаправленного списка. Проход по линейному двунаправленному списку, начиная с его начала. Проход по линейному двунаправленному списку, начиная с его конца. Поиск звена в двунаправленном списке, начиная с начала списка. Поиск звена в двунаправленном списке, начиная с конца списка. |
| 63 | Динамические структуры данных C++. Вставка звена в двунаправленный список (1-й случай). Вставка звена в двунаправленный список (2-й случай). Удаление звена из двунаправленного списка. Указатель на удаляемое звено (1-й случай). Удаление звена из двунаправленного списка. После звена (2-й случай). |
| 64 | Динамические структуры данных C++. Двунаправленные кольцевые списки. |
| 65 | Динамические структуры данных C++. Деки на базе двунаправленных списков. Формирование дека и его просмотр. Добавление звена в начало дека. Добавление звена в конец дека. Удаление звена из дека слева. Удаление звена из дека справа. |
| 66 | Динамические структуры данных C++. Дерево. Бинарные деревья. Основная терминология. Бинарные деревья поиска. Построение бинарного дерева поиска (рекурсивный алгоритм). Анализ алгоpитма поиска с включениями. Дерево отрезков. |
| 67 | Динамические структуры данных C++. Дерево. Обход бинарного дерева. Левосторонний обход бинарного дерева поиска. Концевой обход бинарного дерева поиска. Обратный обход бинарного дерева поиска. |
| 68 | Динамические структуры данных C++. Дерево. Вывод бинарного дерева поиска. Пример программы с использованием бинарного дерева поиска. |
| 69 | Динамические структуры данных C++. Дерево. Построение бинарного дерева (нерекурсивный алгоритм). Изображение бинарного дерева (нерекурсивный алгоритм). Пример программы построения и изображения бинарного дерева (нерекурсивные алгоритмы). Поиск вершины в бинарном дереве (нерекурсивный и рекурсивный). Добавление вершины в бинарное дерево. Удаление вершины из бинарного дерева. |
| 70 | Динамические структуры данных C++. Дерево. Хэшиpование с помощью леса. Дpевовидно-кольцевая динамическая стpуктуpа данных. Деpевья Хаффмена. |
| 71 | Динамические структуры данных C++. Дерево. Деpевья-фоpмулы. Постpоение деpева-фоpмулы. Вычисление с помощью деpева-фоpмулы. Бинаpные деpевья с размеченными листьями. |
| 72 | Структуры данных: двоичная куча (binary heap). Введение. Реализация двоичной кучи (binary heap). Реализация класса кучи. Конструктор кучи. |
| 73 | Структуры данных: двоичная куча (binary heap). Добавление элемента кучи. Вывод элементов кучи. Упорядочение кучи. Удаление вершины кучи (максимального элемента). |
| 74 | Рекурсия и рекурсивные алгоритмы. Сущность рекурсии. Виды рекурсии. Прямая рекурсия. Косвенная рекурсия. Линейная рекурсия. Ветвящаяся рекурсия. Бесконечная рекурсия. Сложная рекурсия. Опережающее описание. |
| 75 | Рекурсия и рекурсивные алгоритмы. Имитация работы цикла с помощью рекурсии. Пример 1. Инкремент счетчика итераций. Пример 2. Декремент счетчика итераций. Пример 3. Двойной цикл. Пример 4: Перевод числа в двоичную систему. |
| 76 | Рекурсия и рекурсивные алгоритмы. Рекуррентные соотношения. Рекурсия и итерация. Произвольное количество вложенных циклов. Примеры рекурсивных алгоритмов. |
| 77 | Рекурсия и рекурсивные алгоритмы. Избавление от рекурсии. Явное использование стека. Запоминание последовательности рекурсивных вызовов. Определение узла дерева по его номеру. |
| 78 | Хеш-таблицы. Ключевые термины темы. Простое представление хеш-таблиц. Практическое применение хеш-таблиц. |
| 79 | Хеш-таблицы. Алгоритмы хеширования данных. Таблица прямого доступа. Метод остатков от деления. Метод функции середины квадрата. Метод свертки. Открытое хеширование. Закрытое хеширование (открытая индексация). |
| 80 | Хеш-таблицы. Основные, наиболее часто используемые функции хэширования. |
| 81 | Хеш-таблицы. Коллизи (они же столкновения). Пример коллизий. Методы разрешения коллизий. Борьба с коллизиями. Метод цепочек. Открытая индексация (или закрытое хеширование). Переполнение таблицы и рехеширование. Оценка качества хеш-функции. |
| 82 | Хеш-таблицы. Организация данных для ускорения поиска по вторичным ключам. Инвертированные индексы. Битовые карты. |
| 83 | Сортировка данных. Основные понятия и определения. Оценка эффективности алгоритмов. Нотация Big-O. Формулировка задачи. Оценка алгоритма сортировки. Нотация "большое О". Список алгоритмов сортировки. |
| 84 | Сортировка данных. Сортировка пузырьком (Bubble sort). Описание алгоритма. Реализация. Анализ алгоритма. |
| 85 | Сортировка данных. Сортировка вставками (Insertion sort). Описание алгоритма. Реализация. Анализ алгоритма. |
| 86 | Сортировка данных. Сортировка выбором (Selection sort). Описание алгоритма. Реализация. Анализ алгоритма. |
| 87 | Сортировка данных. Сортировка слиянием (Merge sort) [Рекурсивные алгоритмы]. Описание алгоритма. Реализация. Анализ алгоритма. |
| 88 | Сортировка данных. Быстрая сортировка (Quicksort) [Рекурсивные алгоритмы]. Описание алгоритма. Реализация. Анализ алгоритма. |
| 89 | Сортировка данных. Сортировка подсчётом. Простой алгоритм. Алгоритм со списком. Алгоритм с двусвязным списком. Сортировка вставками. |
| 90 | Сортировка данных. Устойчивый алгоритм. Обобщение на произвольный целочисленный диапазон. Анализ. Квадратичный алгоритм сортировки подсчётом. Анализ. |
| 91 | Сортировка данных. Пирамидальная сортировка. Суть метода. Просеивание. Структура пирамиды. Построение пирамиды. |
| 92 | Сортировка данных. Поразрядная сортировка. |



Лектор Н.И. Белодед