**УТВЕРЖДАЮ**

Зав. кафедры программной инженерии БГТУ

к.т.н., доцент\_\_\_\_\_\_\_\_\_\_\_\_\_\_Н.В. Пацей

протокол № 5 от 03 декабря 2019 г.

**Экзаменационные вопросы**

**зимняя экзаменационная сессия 2019/20**

Кафедра: **программной инженерии**

Дисциплина: **Основы алгоритмизации и программирования**

Курс: **1**

Специальность:

**1-40 01 02-03 Информационные системы и технологии**

**1-40 01 01 Программное обеспечение информационной технологии**

**1-98 01 03 Программное обеспечение информационной безопасности**

**мобильных систем**

| **Нвп** | **Вопрос** |
| --- | --- |
| 1 | Основные принципы алгоритмизации и программирования. Алгоритмы и программы. Данные. Понятие типа данных. Логические основы алгоритмизации. |
| 2 | Основные принципы алгоритмизации и программирования. Языки программирования: эволюция, классификация. Системы программирования. Файлы данных. |
| 3 | Основные принципы алгоритмизации и программирования. Объектно-ориентированный подход к программированию. Разработка программного обеспечения (ПО). |
| 4 | Пакет компиляторов Visual C++. Рекомендуемое оборудование. Минимальные требования к аппаратному и программному обеспечению. Рекомендуемое аппаратное и программное обеспечение. |
| 5 | Пакет компиляторов Visual C++. Выбор правильных параметров установки. Какую конфигурацию выбрать?. Обычная установка под Windows. Каталоги. |
| 6 | Пакет компиляторов Visual C++. Система разработки. Новый встроенный отладчик. Новые встроенные редакторы ресурсов. Дополнительное средство TestContainer. |
| 7 | Пакет компиляторов Visual C++. Инструменты, не вошедшие в интегрированную среду. ProcessViewer (PView). WinDiff. |
| 8 | Пакет компиляторов Visual C++. Важные возможности компилятора. P-код. Предварительно откомпилированные заголовки и типы. Библиотека MicrosoftFoundationClass. Встраивание функций. |
| 9 | Пакет компиляторов Visual C++. Параметры компилятора. General. Debug. CustomBuild. |
| 10 | Пакет компиляторов Visual C++. C/С++. C++ Language. CodeGeneration. Customization. ListingFiles. Optimizations. PrecompiledHeaders. Preprocessor. |
| 11 | Пакет компиляторов Visual C++. Link. General. Customization. Debug. Input. Output. |
| 12 | Пакет компиляторов Visual C++. Resources. OLE Types. BrowseInfo. |
| 13 | Меню File (файл). New... Open... Close. Open Workspace. Close Workspace. Save. Save As... |
| 14 | Меню File (файл). Save All. Find in Files... Page Setup... Print... Список последних проектов. Exit. |
| 15 | Меню Edit. Undo. Redo. Cut. Сору. Paste. Delete. Select All. |
| 16 | Меню Edit. Find... Replace... Go To... InfoViewer Bookmarks... Bookmark. Breakpoints... Properties... |
| 17 | Меню View. ClassWizard... Resource Symbols... Resource Includes... Full Screen. Toolbars... InfoViewer Query Results. InfoViewer History List. Project Workspace. |
| 18 | Меню View. Info Viewer Topic. Output. Watch. Variables. Registers. Memory. Call Stack. Disassembly. Меню Insert. |
| 19 | Меню Build. Compile. Build. Rebuild All. Batch Build... Stop Build. Update All Dependencies. |
| 20 | Меню Build. Debug. Execute. Settings... Configurations... Subprojects... Set Default Configuration... |
| 21 | Меню Tools. Browse... Close Browse Info File. OLE Control Test Container. OLE Object View. |
| 22 | Меню Window. New Window. Split. Hide. Cascade. |
| 23 | Меню Window. Tile Horizontally, Tile Vertically. Close All... Windows. Меню Help. |
| 24 | История языка С. Взаимоотношения с другими языками. Достоинства языка С. Малый размер. Набор команд языка. Быстродействие. Язык со слабой типизацией. Структурированный язык. Поддержка модульного программирования. |
| 25 | История языка С. Простой интерфейс с ассемблерными подпрограммами. Поразрядная обработка. Переменные-указатели. Гибкие структуры. Эффективность использования памяти. Переносимость. Библиотеки специальных функций. |
| 26 | История языка С. Недостатки языка С. Слабая типизация. Отсутствие проверок на этапе исполнения. Использование языка Си. Будущее языка Си. |
| 27 | Исходные файлы и выполняемые файлы. Принципы программирования. Стандарт ANSI С. Эволюция языка C++ и объектно-ориентированное программирование. История C++. Использование объектов C++ для быстрого создания программы. |
| 28 | Исходные файлы и выполняемые файлы. Некоторые усовершенствования по сравнению с языком С. Комментарии. Имена перечисляемых типов. Имена структуры или класса. Блочные объявления. Операция уточнения области действия (scope). Описатель const. Анонимные объединения. |
| 29 | Исходные файлы и выполняемые файлы. Явное преобразование типов. Объявления функций. Перегруженные функции. Значения параметров функций по умолчанию. Функции с неуказанным числом параметров. Ссылочные параметры функции. Операторы newи delete. Указатели void и функции, возвращающие void. |
| 30 | Исходные файлы и выполняемые файлы. Основные усовершенствования по сравнению с языком С.(часто повторяется). Конструкторы классов и инкапсуляция данных. Класс struct. Конструкторы и деструкторы. Сообщения. "Дружественные" классы. |
| 31 | Исходные файлы и выполняемые файлы. Перегрузка операций. Производные классы. Полиморфизм при использовании виртуальных функций. Библиотеки потоков. Базовые элементы программы на С. Пять основных компонентов программы. |
| 32 | Написание и компиляция простых программ на C/C++. Написание вашей первой программы. Пример простой программы на С. Структура простой программы. Как сделать программу читаемой. Подготовка и компиляция простых программ на С/С++. |
| 33 | Написание и компиляция простых программ на C/C++. Редактирование текста программы. Сохранение программ. Построение программы. |
| 34 | Написание и компиляция простых программ на C/C++. Использование утилиты Project Workspace. Создание нового проекта. Добавление файлов к проекту. Запуск команд Build или Rebuild All. |
| 35 | Написание и компиляция простых программ на C/C++. Отладка программы. Понимание сообщений об ошибках и предупреждений. Распространенная ошибка при использовании нового языка. Переключение между окном вывода сообщений и окном редактирования. Использование функций замены или быстрого поиска. Выбор опций замены. |
| 36 | Написание и компиляция простых программ на C/C++. Переключение между окном вывода сообщений и окном редактирования. Быстрый способ. Значение сообщений об ошибках и предупреждений. Повторная сборка программы ERROR.С. Запуск программы. |
| 37 | Написание и компиляция простых программ на C/C++. Использование встроенного отладчика. Использование команд пошагового выполнения (Step Into и Step Over). Определение точек останова (breakpoints). Запуск программы с точками останова. Использование быстрого просмотра (QuickWatch). |
| 38 | Данные. Идентификаторы. Ключевые слова. Символы. Данные: переменные и константы. Данные: типы данных. |
| 39 | Три целочисленных типа. Описание данных целого типа. Целые константы. Инициализация переменных целого типа. Модификатор unsigned. |
| 40 | Числа с плавающей точкой. Описание переменных с плавающей точкой. Перечисляемый тип данных (enum). |
| 41 | Модификаторы доступа. Модификатор const. Определение констант через #define. Модификатор volatile. Cовместное использование const и volatile. |
| 42 | Модификаторы pascal, cdecl, near, far и huge. Модификатор pascal. Модификатор cdecl. Модификаторы near, far и huge. |
| 43 | Тип данных char Другие типы и размеры данных. Преобразование типов данных. Явные преобразования типов при помощи операции приведения типа. |
| 44 | Символьные строки. Массивы символов в C++ Функции работы со строками символов |
| 45 | Символьные строки. Определение длины строк Копирование и конкатенация строк Сравнение строк Преобразование строк |
| 46 | Символьные строки. Обращение строк Поиск символов Поиск подстрок |
| 47 | Функции преобразования типа Функции printf() и scanf(). Использование функции printf(). Модификаторы спецификации преобразования, используемые в функции printf(). Использование функции printf() для преобразования данных. Применение функции scanf(). |
| 48 | Основные операции. Операция присваивания: =. Операция сложения: +. Операция вычитания: -. Операция изменения знака: -. Операция умножения: \*. Операция деления: /. |
| 49 | Основные операции. Поразрядные операции. Поразрядное И (AND). Поразрядное ИЛИ (OR). Поразрядное исключающее ИЛИ (OR). |
| 50 | Основные операции. Поразрядный сдвиг влево и вправо. Операции отношения и логические операции. Условная операция (?:). Операция запятая (,). Порядок выполнения операций. |
| 51 | Дополнительные операции. Операция деления по модулю: %. Операции увеличения и уменьшения: ++. Операция уменьшения: --. Старшинство операций. |
| 52 | Выражения и операторы. Выражения. Операторы. Составные операторы (блоки). |
| 53 | Ввод и вывод одного символа: функции getchar() и putchar(). Буферы. Чтение данных. Чтение строки. Чтение файла. |
| 54 | Переключение и работа с файлами. Операционная система UNIX. Переключение вывода. Переключение ввода. Комбинированное переключение. |
| 55 | Выбор вариантов. Операции отношения и выражения. Понятие "истина". |
| 56 | Условные операторы. Оператор if. Оператор if-else. Вложенные операторы if-else. |
| 57 | Условные операторы. Операторы if-else-if. Условный оператор ?. Оператор switch. Совместное использование операторов if-else-if и switch. |
| 58 | Оператор цикла. Цикл while. Завершение цикла while. |
| 59 | Оператор цикла. Цикл do-while. Цикл for. |
| 60 | Оператор цикла. Операция "запятая" в цикле for. Гибкость конструкции for. Философ Зенон и цикл for. |
| 61 | Оператор цикла. Вложенные циклы. Алгоритмы и псевдокод. |
| 62 | Оператор цикла. Оператор break. Оператор continue. Совместное использование операторов break и continue. |
| 63 | Оператор goto. Использование goto. |
| 64 | Оператор exit(). Оператор atexit(). Сравнение циклов. |
| 65 | Массивы. Понятие массив. Массивы в С. Объявление массивов. Доступ к элементам массива. Размещение массивов в памяти. Проблема ввода. |
| 66 | Массивы. Инициализация массивов. Инициализация по умолчанию. Явная инициализация. Инициализация безразмерных массивов. |
| 67 | Массивы. Инициализация массивов и классы памяти. Вычисление размера массива (sizeof()). Выход индекса за пределы массива. |
| 68 | Массивы. Многомерные массивы. Инициализация двумерного массива. |
| 69 | Массивы. Массивы в качестве аргументов функций. Передача массивов функциям С. Передача массивов функциям C++. |
| 70 | Ввод и вывод строк. Строковые функции и символьные массивы. Динамическое выделение памяти. Функции gets(), puts(), fgets(), fputs() и sprintf(). Функции strcpy(), strcat(), strncmp() и strlen(). |
| 71 | Указатели. Определение переменных-указателей. Разыменование указателей. Объявление переменных-указателей. Простые операторы с указателями. Инициализация указателей. Неправильное использование операции определения адреса. |
| 72 | Указатели. Указатели на массивы. Указатели и многомерные массивы. Указатели на указатели. Указатели на строки. |
| 73 | Указатели. Арифметические операции с указателями. Арифметические операции с указателями и массивы. Операции с указателями. |
| 74 | Указатели. Применение к указателям оператора sizeof. Сложности при использовании операций ++ и --. Сравнение указателей. Переносимость указателей. Использование функции sizeof() с указателями в среде DOS. |
| 75 | Указатели. Указатели на функции. Динамическая память. Использование указателей типа void. |
| 76 | Указатели. Указатели и массивы. Функции, массивы и указатели. Использование указателей при работе с массивами. Строки (массивы типа char). |
| 77 | Указатели. Массивы указателей. Дополнительная информация об указателях на указатели. Массивы указателей на строки. |
| 78 | Указатели. Ссылочный тип в C++ (reference type). Адрес в качестве возвращаемого значения функции. Передача параметров по ссылке и по значению. Использование встроенного отладчика. Использование ссылочного типа. Использование указателей и ссылок с ключевым словом const |

Лектор Н.И. Белодед