

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
**Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Нижегородский государственный университет
им. Н.И. Лобачевского»**

Павловский филиал ННГУ

УТВЕРЖДЕНО
решением Ученого совета ННГУ
протокол от 16.06.2021 г. № 8

Рабочая программа дисциплины

АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ

(наименование дисциплины (модуля))

Уровень высшего образования

БАКАЛАВРИАТ

Направление подготовки / специальность

09.03.03 ПРИКЛАДНАЯ ИНФОРМАТИКА

Направленность образовательной программы

ПРИКЛАДНАЯ ИНФОРМАТИКА В ЭКОНОМИКЕ И УПРАВЛЕНИИ

Форма обучения

ОЧНАЯ

Павлово
2021 год

Лист актуализации

Визирование РПД для исполнения в очередном учебном году

Председатель ОМК
__ 2019 г.

Рабочая программа пересмотрена, обсуждена и одобрена для
исполнения в 2019-2020 учебном году на заседании кафедры

Протокол от __ 20__ г. № __
Зав. кафедрой _____

Визирование РПД для исполнения в очередном учебном году

Председатель ОМК
__ 20__ г.

Рабочая программа пересмотрена, обсуждена и одобрена для
исполнения в 2020-2021 учебном году на заседании кафедры

Протокол от __ 20__ г. № __
Зав. кафедрой _____

Визирование РПД для исполнения в очередном учебном году

Председатель ОМК
__ 20__ г.

Рабочая программа пересмотрена, обсуждена и одобрена для
исполнения в 2021-2022 учебном году на заседании кафедры

Протокол от __ 20__ г. № __
Зав. кафедрой _____

Визирование РПД для исполнения в очередном учебном году

Председатель ОМК
__ 20__ г.

Рабочая программа пересмотрена, обсуждена и одобрена для
исполнения в 2022-2023 учебном году на заседании кафедры

Протокол от __ 20__ г. № __
Зав. кафедрой _____

1. Место и цели дисциплины в структуре ОПОП

Дисциплина Б1.О.14 «Алгоритмизация и программирование» относится к обязательной части Блока 1. «Дисциплины (модули)» учебного плана ООП по направлению подготовки 09.03.03 «Прикладная информатика» (квалификация (степень) «бакалавр»).

2. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине (модулю), в соответствии с индикатором достижения компетенции		Наименование оценочного средства
	Индикатор достижения компетенции (код, содержание индикатора)	Результаты обучения по дисциплине	
ОПК-2. Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности	ОПК-2.1. Способен продемонстрировать знание современных информационных технологий и программных средств, в том числе отечественного производства, решения задач профессиональной деятельности.	Знать методы и технологии программирования, абстракции основных структур данных и методы их обработки и реализации, базовые алгоритмы обработки данных, иметь представление о структуре вычислительных систем и способах сетевого взаимодействия.	Тестирование, практические задачи, коллоквиум
	ОПК-2.2. Способен применять современные информационные технологии и программные средства, в том числе отечественного производства, для решения задач профессиональной деятельности.	Уметь разрабатывать алгоритмы, реализовывать алгоритмы на языке программирования высокого уровня, описывать основные структуры данных, реализовывать методы анализа и обработки данных, работать в средах программирования.	Тестирование, практические задачи, коллоквиум
	ОПК-2.3. Способен решать задачи профессиональной деятельности с применением современных информационных технологий и программных средств, в том числе отечественного производства.	Владеть методами и технологиями разработки алгоритмов, описания структур данных и других базовых представлений данных, программирования на языке высокого уровня, работы в различных средах программирования.	Тестирование, практические задачи, коллоквиум
ОПК-3. Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	ОПК-3.1. Способен использовать принципы, методы и средства решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности.	Знать этапы и стадии подготовки и решения задач на компьютере.	Тестирование, практические задачи, коллоквиум
	ОПК-3.2. Способен применять информационно-коммуникационные технологии	Уметь выполнять формализацию содержательной постановки задачи.	Тестирование, практические задачи, колло-

	решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с учетом основных требований информационной безопасности.		квиум
	ОПК-3.2. Способен решать стандартные задачи профессиональной деятельности с соблюдением требований информационной безопасности.	Владеть навыками принятия оптимальных решений, основанных на использовании экономико-математических методах.	Тестирование, практические задачи, коллоквиум
ОПК-4. Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью	ОПК-4.1. Способен продемонстрировать знание основных стандартов, норм и правил оформления технической документации на различных стадиях проектирования и поддержки жизненного цикла информационной системы.	Знать методы принятия оптимальных решений в задачах анализа и управления экономическими системами.	Тестирование, практические задачи, коллоквиум
	ОПК-4.2. Способен применять стандарты, нормы и правила (в том числе установленные самостоятельно) при оформлении технической документации на различных стадиях проектирования и поддержки жизненного цикла информационной системы.	Уметь анализировать альтернативные варианты решений для достижения намеченных результатов; определять цели и этапы выполнения работ.	Тестирование, практические задачи, коллоквиум
	ОПК-4.3. Способен составлять техническую документацию на различных этапах жизненного цикла информационной системы.	Владеть методиками разработки целей и задач проекта; методами принятия оптимальных решений с учетом имеющихся ресурсов и ограничений.	Тестирование, практические задачи, коллоквиум
ОПК-5. Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем	ОПК-5.1. Способен использовать знания основ системного администрирования, администрирования СУБД, современные стандарты информационного взаимодействия систем.	Знать методы принятия оптимальных решений в задачах анализа и управления экономическими системами	Тестирование, практические задачи, коллоквиум
	ОПК-5.2. Способен выполнять параметрическую настройку ИС.	Уметь анализировать альтернативные варианты решений для достижения намеченных результатов; определять цели и этапы выполнения работ.	Тестирование, практические задачи, коллоквиум
	ОПК-5.3. Способен устанавливать программное и аппаратное обеспечение информационных и автоматизированных систем.	Владеть навыками адаптации программного обеспечения на операционной системе Linux.	Тестирование, практические задачи, коллоквиум
ОПК-7. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-7.1. Способен использовать основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий.	Знать принципы проектирования и разработки программ	Тестирование, практические задачи, коллоквиум
	ОПК-7.2. Способен применять языки программирования и работы с базами данных, современные программные среды разработки	Уметь работать с базами данных, использовать современные программные среды разработки информа-	Тестирование, практические задачи, коллоквиум

	информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ.	ционных систем и технологий для автоматизации бизнес-процессов	
	ОПК-7.3. Способен осуществлять программирование, отладку и тестирование прототипов программно-технических комплексов задач.	Владеть навыками работы с персональным компьютером с использованием интегрированной среды MS Visual Studio и Qt Creator.	Тестирование, практические задачи, коллоквиум
ПК-3. Способен вводить в эксплуатацию и осуществлять сопровождение ИС на всех этапах ее жизненного цикла, включая ее презентацию и начальное обучение пользователей	ПК-3.1. Способен использовать знания методологических и технических основ ввода ИС в эксплуатацию.	Знать методологические и технические основы ввода ИС в эксплуатацию Уметь разрабатывать инструменты для ввода ИС в эксплуатацию Владеть навыками ввода ИС в эксплуатацию	Тестирование, практические задачи, коллоквиум
	ПК-3.2. Способен организовать репозиторий хранения данных о создании ИС, вводе ее в эксплуатацию и модификации в процессе жизненного цикла.	Знать основные этапы жизненного цикла ИС Уметь организовать репозиторий хранения данных о создании ИС, вводе ее в эксплуатацию и модификации Владеть навыками управления ИС на всех стадиях ЖЦ	Тестирование, практические задачи, коллоквиум
	ПК-3.3. Способен осуществлять установку программного обеспечения ИС, его тестирование и начальное обучение пользователей.	Знать основы установки программного обеспечения ИС, его тестирование и начальное обучение пользователей Уметь устанавливать и тестировать разработанное ПО Владеть навыками начального обучения пользователей	Тестирование, практические задачи, коллоквиум

3. Структура и содержание дисциплины

3.1 Трудоемкость дисциплины

Общая трудоемкость	7 ЗЕТ
Часов по учебному плану	252
в том числе	
аудиторные занятия (контактная работа):	116
- занятия лекционного типа	32
- занятия семинарского типа	16
- занятия лабораторного типа	64
самостоятельная работа	100
Промежуточная аттестация - зачет, экзамен, курсовая работа	36

3.2. Содержание дисциплины

Наименование и краткое содержание раз-	Всего (часы)	В том числе		
		Контактная работа (работа во взаимодействии с преподавателем), часы из них	цельная работа обучающегося,	гося,

делов и тем дисциплины				Занятия лекционного типа			Занятия лабораторного типа			Занятия семинарского типа			Всего					
	Очное	Очно-заочное	Заочное	Очное	Очно-заочное	Заочное	Очное	Очно-заочное	Заочное	Очное	Очно-заочное	Заочное	Очное	Очно-заочное	Заочное	Очное	Очно-заочное	Заочное
Тема 1. Введение в информатику	26			4			8			2			14			12		
Тема 2. Понятие типа данных	27			4			8			2			14			13		
Тема 3. Алгоритмизация процессов обработки данных.	26			4			8			2			14			12		
Тема 4 Функции	27			4			8			2			14			13		
Тема 5. Структуры	26			4			8			2			14			12		
Тема 6. Файловый ввод - вывод	27			4			8			2			14			13		
Тема 7. Введение в объектно-ориентированное программирование. Классы	26			4			8			2			14			12		
Тема 8. Объектно-событийное и визуальное программирование в оконной операционной среде.	27			4			8			2			14			13		
КСР	4												4					
Контроль	36																	
ИТОГО	252			32			64			16			116			100		

Содержание дисциплины по темам

1. Введение в информатику.

Основы архитектуры и организации ЭВМ

Базовые представления об архитектуре ЭВМ. История развития вычислительных средств. Принципы фон Неймана. Классификация компьютеров. Архитектура современного компьютера

Архитектура микропроцессора

Классы процессоров. Конвейерная обработка команд. Векторная обработка. Регистры процессоров. Системы команд x86. Макроассемблер.

Запоминающие устройства. Иерархия запоминающих устройств. Устройства ввода и вывода

Организация оперативной памяти компьютера.

Логическая память. Сегменты. Связывание адресов. Простейшие схемы управления памятью. Схемы с фиксированными и переменными разделами. Страничная память. Сегментно-страничная организация памяти.

Арифметические основы ЭВМ

Системы счисления. Представление чисел в позиционных системах счисления. Обоснование экономичности систем счисления.

Представление целых и вещественных чисел в позиционных системах счисления.

Арифметические операции над числами, представленными в двоичной системе счисления.

Способы представления чисел в ЭВМ: с фиксированной запятой, с плавающей запятой.

Машинные методы выполнения арифметических операций над числами, представленными в двоичной системе счисления: преобразование кодов (прямой, обратный, дополнительный).

Умножение с младших разрядов и старших разрядов в прямом коде. Деление с восстановлением остатка и без восстановления остатка.

Проблема переполнения. Ошибка усечения. Десятичные двоично-кодированные системы.

Логические основы ЭВМ

Алгебра высказываний. Операции над высказываниями. Логические связи: инверсия, дизъюнкция, конъюнкция, импликация, эквиваленция, сложение по mod 2. Ранги логических связей.

Таблицы истинности. Правила составления таблиц истинности.

Формулы алгебры высказываний и равносильные преобразования. Основы равносильности. Равносильности, выражающие одни логические операции через другие.

Функции алгебры логики (ФАЛ). ФАЛ одной и двух переменных.

Дизъюнктивные и конъюнктивные нормальные формы алгебры высказываний. СДНФ. СКНФ.

Минимизация ФАЛ: метод непосредственных преобразований, метод графической минимизации Карно.

Типовые логические элементы и узлы ЭВМ: «И», «ИЛИ», «НЕ», «И-НЕ», «ИЛИ-НЕ», «триггер», «полусумматор».

Системное программное обеспечение

1. Базовое программное обеспечение. Служебные программы.

2. Операционные системы.

2.1. Назначение операционной системы. Виды операционных систем. Базовые понятия операционных систем. Процессы и потоки. Управление памятью. Драйверы устройств. Файловые системы.

2.2. Архитектура Windows.

2.3. Архитектура Linux.

Прикладное программное обеспечение.

Текстовый процессор Word

Ввод и редактирование текста. Ввод текста с использованием “горячих” клавиш. Списки. Таблицы. Создание рисунков с помощью инструментов Word. Формулы и вставка символов. Вставка рисунков в текст. Шаблоны и формы. Построение диаграмм. Документы слияния. Главный документ. Исправления. Автозамена. Примечания. Сноски.

Табличный процессор Excel

Создание и редактирование таблиц. Диаграммы. Формы и сортировка таблиц. Консолидация данных. Сводная таблица. Использование финансовых функций в формулах таблицы. Подбор параметра. Поиск решения. Шаблоны и элементы управления на рабочих листах. Функции пользователя. Элементы управления на листах – формы.

Основы информационных систем. Базы данных

Основные понятия Классификация БД Модели данных. Проектирование баз данных. CASE- системы для разработки информационных систем. Реляционная модель данных. Получение реляционной схемы из инфологической модели.

База данных Microsoft Access. Потенциальные и внешние ключи. Связывание таблиц. Нормализация таблиц. Ограничения целостности.

Компьютерные сети.

1. Сетевые архитектуры

1.1. Назначение и классификация компьютерных сетей. Типы сетей. Топология сетей. Сетевые компоненты. Сети Ethernet. Сети Token Ring. Сетевые протоколы. Среда клиент-сервер. Internet как иерархия сетей.

2. Сетевые информационные технологии

2.1. История создания Internet. Административное устройство. Основные понятия (гипертекст, Web, http, URL, IP – адрес, доменное имя, браузер).

2.2. Язык разметки гипертекста HTML.

2.2.1. История развития. Принцип гипертекстовой разметки. Структура документа HTML. Элементы структурной организации текста. Элементы логического форматирования символов. Элементы физического форматирования символов. Ссылки. Списки. Таблицы. Вставка в документ объектов. Элементы , <MAP>, <AREA> Элемент <BODY>. Фреймы.

2.3. Введение в CSS. Слои. Введение в HTML5.

Основные понятия языков программирования. Сравнительная характеристика языков программирования. Синтаксис, семантика языков программирования. Сравнение развития языков в представлении данных и способах реализации алгоритмов. Общая характеристика языка программирования. Основные элементы языка программирования: алфавит, идентификаторы, ключевые слова, константы, переменные, операторы, выражения, функции. Общая структура программы. Исполняемые инструкции и определения. Пример простой программы.

Особенности программирования в современных оконных операционных средах. Среда разработки. Порядок создания консольного приложения.

Элементы модульного программирования. Многофайловая организация программ. Состав файлов проекта. Подключение библиотек.

Процесс обработки программы на языке высокого уровня: компиляция, сборка, выполнение и отладка.

2. Понятие типа данных. Целочисленные и вещественные типы данных Представление в ЭВМ знаковых и беззнаковых целочисленных типов. Характеристики целочисленных типов данных: ключевые слова, размер, диапазон представления. Операции над данными целого типа. Представление в ЭВМ вещественных типов данных. Характеристики вещественных типов данных. Операции над данными вещественных типов. Преобразование типа. Вычисление значений выражений.

3. Алгоритмизация процессов обработки данных.

Введение в алгоритмы обработки данных. Понятие алгоритма и его свойства. Основные конструкции для записи алгоритмов. Базисные алгоритмы: следование, выбор, повторение. Структурное программирование

Представление базисных алгоритмов на языке высокого уровня Управляющие инструкции. Условный оператор. Логические выражения. Переключатель. Операторы цикла с предусловием и постусловием. Вложенные циклы

Простейшие алгоритмы обработки данных. Вычисление значений выражений. Использование логических выражений в задачах на выделение геометрических областей. Итерационные алгоритмы. Циклы с известным и неизвестным числом повторений. Сочетание цикла и разветвлений. Рекуррентные вычисления. Алгоритмы поиска корней функции. Применение циклов в задачах на темы: суммирование рядов, позиционная запись числа, делители целого числа.

Одномерные массивы данных. Построение и преобразование одномерных массивов. Алгоритмы обработки массивов данных. Последовательный поиск. Поиск макси-

мального и минимального элементов. Построение массива без повторений. Бинарный поиск в упорядоченном массиве. Алгоритмы сортировки. Операции с многочленами, заданными массивами своих коэффициентов.

Построение и преобразование матриц. Матричная алгебра.

Форматный ввод вывод. Функции форматного ввода вывода стандартной библиотеки.

Символьный тип данных. Код символа. Кодовая таблица. Символьная константа. Ввод и вывод символов

4. Функции.

Описание функции. Возврат значений. Передача параметров. Массивы как аргументы функций.

Строковый тип данных. Строковые константы. Ввод и вывод строк. Функции для работы со строками.

Классы памяти. Внешние, статические, автоматические, регистровые переменные. Правила областей действия.

Структуры данных стек, очередь. Непрерывная реализация структур данных на базе линейной памяти ЭВМ. Пример использования стека в программе-калькулятор.

Препроцессор. Включение файлов. Макроподстановка. Условная компиляция.

Модульное программирование. Многофайловая организация программы-калькулятор.

Рекурсивные алгоритмы обработки данных. Условия, обеспечивающие завершение рекурсивных вызовов. Реализация рекурсивных вызовов. Сравнение рекурсивных и итерационных алгоритмов обработки данных.

Указатели. Указатели и адреса. Указатели и аргументы функций. Указатели и массивы. Адресная арифметика. Указатели и строки. Динамическое выделение памяти.

Массивы указателей. Сортировка текстовых строк входного потока. Многомерные массивы. Массивы и функции. Двумерные массивы и указатели. Функция, суммирующая матрицы произвольной размерности.

Указатели на функции. Массивы указателей на функции. Сложные типы данных. Разработка полиморфных программ.

5. Структуры.

Расположение в памяти. Инициализация. Операции. Приоритет операций. Вложенные структуры. Объединения и поля битов. Структуры и функции. Массивы структур. Указатели на структуры

6. Файловый ввод - вывод

Современные средства языка программирования. Оператор разрешения области видимости. Использование функции в качестве элементов структур. Объявления в операторах. Константные объекты. Встроенные функции. Параметры по умолчанию. Передача аргументов функциям и возврат значений по адресу. Перегрузка имен функций

7. Введение в объектно-ориентированное программирование.

Стили программирования. Принципы объектно-ориентированного программирования. Абстракция данных. Инкапсуляция. Наследование. Полиморфизм. Повторное использование кода.

Классы.

Инкапсуляция и члены-функции. Управление доступом. Конструкторы и деструкторы. Отличительные свойства. Взаимодействие с компилятором.

Копирование объектов классов. Массивы объектов классов. Объекты классов как элементы

Примеры разработки классов: строка, вектор, матрица.

8. Объектно-событийное программирование в операционной среде. Событие и

сообщение. Кодирование сообщений и механизмы реализации обмена сообщениями в операционной среде. Программирование, управляемое событиями. Природа событий. Виды событий. События от мыши. События от клавиатуры. Передача сообщений.

Визуальное программирование в оконной операционной среде. Обзор современных инструментальных систем визуального программирования. Основные характеристики среды. Настройка среды. Основные элементы интегрированной среды разработки. Основы визуального программирования. Методы визуальной разработки оконных приложений. Выбор шаблона приложения. Работа с формой. Стандартные визуальные компоненты. Перенос компонент на форму. Установка свойств. События, на которые реагируют компоненты. Задание обработчиков событий. Примеры разработки оконных приложений. Организация ввода вывода данных. Разработка программы по обработке простых типов данных. Приложение с кнопками, редактируемым полем и списком. (С помощью кнопок можно добавлять элементы в список, удалять из списка) Приложение - простейший калькулятор

Практические занятия

Содержание практических занятий: решение базовых задач по курсу лекций, закрепляющих теоретический материал и иллюстрирующих его применение к решению задач.

Перечень тем практических занятий

№п/п	№ раздела дисциплины	Наименование лабораторных работ
1	4	Решение задач на линейные, разветвляющие и циклические алгоритмы
2	5	Функции для работы со строками.
3	6	Разработка структур данных, описывающих объекты из различных предметных областей
4	7	Ввод и вывод структурных типов данных

Занятия лабораторного и семинарского типа организуются в том числе в форме практической подготовки, которая предусматривает участие обучающихся в выполнении отдельных элементов работ, связанных с будущей профессиональной деятельностью.

Практическая подготовка предусматривает решение прикладных задач. На проведение занятий лабораторного и семинарского типа в форме практической подготовки отводится 8 часов.

Практическая подготовка направлена на формирование и развитие:

- ✓ практических навыков в соответствии с профилем ОП:
 - формирование и анализ требований к информатизации и автоматизации прикладных процессов, формализация предметной области проекта.
 - моделирование прикладных и информационных процессов;
 - программирование приложений, создание прототипа информационной системы;
 - тестирование компонентов ИС по заданным сценариям;
- ✓ компетенций - ОПК-2, ОПК-3, ОПК-4, ОПК-5, ОПК-7, ПК-3.

Текущий контроль успеваемости реализуется в рамках практических занятий.

Промежуточная аттестация проходит в традиционных формах (зачёт, курсовая работа, экзамен).

4. Учебно-методическое обеспечение самостоятельной работы обучающихся

Цель самостоятельной работы - формирование навыков непрерывного самообразования и профессионального совершенствования.

Самостоятельная работа способствует формированию аналитического и творческого мышления, совершенствует способы организации исследовательской деятельности, воспитывает целеустремленность, системность и последовательность в работе студентов, развивает у них навык завершать начатую работу.

Основные виды самостоятельной работы студентов:

- работа с основной и дополнительной литературой;
- изучение категориального аппарата дисциплины;
- самостоятельное изучение тем дисциплины;
- подготовка к зачёту, экзамену;
- работа в библиотеке;
- изучение сайтов по темам дисциплины в сети Интернет.

Работа с основной и дополнительной литературой

Изучение рекомендованной литературы следует начинать с учебников и учебных пособий, затем переходить к научным монографиям и материалам периодических изданий. Работа с литературой предусматривает конспектирование наиболее актуальных и познавательных материалов. Это не только мобилизует внимание, но и способствует более глубокому осмыслению материала, его лучшему запоминанию, а также позволяет студентам проводить систематизацию и сравнительный анализ изучаемой информации. Таким образом, конспектирование – одна из основных форм самостоятельного труда, которая требует от студента активно работать с учебной литературой и не ограничиваться конспектом лекций.

Студент должен уметь самостоятельно подбирать необходимую литературу для учебной и научной работы, уметь обращаться с предметными каталогами и библиографическим справочником библиотеки.

Изучение категориального аппарата дисциплины

Изучение и осмысление экономических категорий требует проработки лекционного материала, выполнения практических заданий, изучение словарей, энциклопедий, справочников.

Индивидуальная самостоятельная работа студента направлена на овладение и грамотное применение экономической терминологии в области компьютерного моделирования.

Самостоятельное изучение тем дисциплины

Особое место отводится самостоятельной проработке студентами отдельных разделов и тем изучаемой дисциплины. Такой подход вырабатывает у студентов инициативу, стремление к увеличению объема знаний, умений и навыков, всестороннего овладения способами и приемами профессиональной деятельности.

Изучение вопросов определенной темы направлено на более глубокое усвоение основных категорий экономической теории, понимание экономических процессов, происходящих в обществе, совершенствование навыка анализа теоретического и эмпирического материала.

Подготовка к зачёту, экзамену

Промежуточная аттестация студентов по дисциплине проходит в виде зачёта и экзамена, предусматривающего оценку. Условием успешного прохождения промежуточной аттестации является систематическая работа студента в течение семестра. В этом случае подготовка к зачёту, экзамену является систематизацией всех полученных знаний по данной дисциплине.

Рекомендуется внимательно изучить перечень вопросов к зачёту, экзамену, а также использовать в процессе обучения программу, учебно-методический комплекс, другие методические материалы.

Желательно спланировать троекратный просмотр материала перед зачётом, экзаменом. Во-первых, внимательное чтение с осмыслением, подчеркиванием и составлением краткого плана ответа. Во-вторых, повторная проработка наиболее сложных вопросов. В-третьих, быстрый просмотр материала или планов ответов для его систематизации в памяти.

Самостоятельная работа в библиотеке

Важным аспектом самостоятельной подготовки студентов является работа с библиотечным фондом.

Это работа предполагает различные варианты повышения профессионального уровня студентов:

- а) получение книг для подробного изучения в течение семестра на научном абонементе;
- б) изучение книг, журналов, газет - в читальном зале;
- в) возможность поиска необходимого материала посредством электронного каталога;
- г) получение необходимых сведений об источниках информации у сотрудников библиотеки.

Изучение сайтов по темам дисциплины в сети Интернет

Ресурсы Интернет являются одним из альтернативных источников быстрого поиска требуемой информации. Их использование возможно для получения основных и дополнительных сведений по изучаемым материалам. Необходимо помнить об оформлении ссылок на Интернет-источники.

Для повышения эффективности самостоятельной работы студентов преподавателю целесообразно использовать следующие виды деятельности:

- консультации,
- выдача заданий на самостоятельную работу,
- информационное обеспечение обучения,
- контроль качества самостоятельной работы студентов.

Контрольные вопросы и задания для проведения текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведены в п. 5.2.

5. Фонд оценочных средств для промежуточной аттестации по дисциплине (модулю), включающий

5.1. Описание шкал оценивания результатов обучения по дисциплине

Уровень сформированности компетенций (индикаторы достижения компетенций)	Шкала оценивания сформированности компетенций						
	плохо	неудовлетворительно	удовлетворительно	хорошо	очень хорошо	отлично	превосходно
	Не зачтено		Зачтено				
Знания	Отсутствие знаний теоретического материала. Невозможность оценить полноту знаний вследствие отказа обучающегося от ответа	Уровень знаний ниже минимальных требований. Имели место грубые ошибки.	Минимально допустимый уровень знаний. Допущено много негрубых ошибок.	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько незначительных ошибок	Уровень знаний в объеме, соответствующем программе подготовки, без ошибок.	Уровень знаний в объеме, превышающем программу подготовки.

Умения	Отсутствие минимальных умений. Невозможность оценить наличие умений вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы основные умения. Имели место грубые ошибки.	Продемонстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном объеме.	Продемонстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания, в полном объеме, но некоторые с недочетами.	Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме, но некоторые с недочетами.	Продемонстрированы все основные умения, решены все основные задачи с отдельными несущественными недочетами, выполнены все задания в полном объеме.	Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме без недочетов
Навыки	Отсутствие владения материалом. Невозможность оценить наличие навыков вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы базовые навыки. Имели место грубые ошибки.	Имеется минимальный набор навыков для решения стандартных задач с некоторыми недочетами	Продемонстрированы базовые навыки при решении стандартных задач с некоторыми недочетами	Продемонстрированы базовые навыки при решении стандартных задач без ошибок и недочетов.	Продемонстрированы навыки при решении нестандартных задач без ошибок и недочетов.	Продемонстрирован творческий подход к решению нестандартных задач

Шкала оценки при промежуточной аттестации

Оценка		Уровень подготовки
зачтено	Превосходно	Вся компетенция (части компетенции), на формирование которой направлена дисциплина, сформированы на уровне не ниже «превосходно»
	Отлично	Вся компетенция (части компетенции), на формирование которой направлена дисциплина, сформированы на уровне не ниже «отлично», при этом хотя бы одна компетенция сформирована на уровне «отлично»
	Очень хорошо	Вся компетенция (части компетенции), на формирование которой направлена дисциплина, сформированы на уровне не ниже «очень хорошо», при этом хотя бы одна компетенция сформирована на уровне «очень хорошо»
	Хорошо	Вся компетенция (части компетенции), на формирование которой направлена дисциплина, сформированы на уровне не ниже «хорошо», при этом хотя бы одна компетенция сформирована на уровне «хорошо»
	Удовлетворительно	Вся компетенция (части компетенции), на формирование которой направлена дисциплина, сформированы на уровне не ниже «удовлетворительно», при этом хотя бы одна компетенция сформирована на уровне «удовлетворительно»
Не зачтено	Неудовлетворительно	Хотя бы одна часть компетенции сформирована на уровне «неудовлетворительно», ни одна из компетенций не сформирована на уровне «плохо»
	Плохо	Хотя бы одна часть компетенции сформирована на уровне «плохо»

5.2. Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения

5.2.1. Контрольные вопросы

Вопросы для итогового контроля по дисциплине

1. Характеристика языка программирования Си. Операторы ввода и вывода. Форматный ввод, вывод библиотеки `stdio`. Спецификаторы формата. Эскейп-последовательности.

2. Понятие типа данных. Целые типы данных в Си. Вещественные типы данных в Си. Преобразование типа.
3. Понятие алгоритма. Базисные алгоритмы: следование, выбор, повторение. Принцип структурного программирования. Логические операции. Логические выражения. Истина и ложь в Си, условный оператор, переключатель. Циклы в Си Примеры.
4. Массивы Линейный поиск в массиве. Построение массива без повторений.
5. Максимальный и минимальный элементы массива. Сортировка.
6. Двумерные массивы Преобразование и построение матриц
7. Символьный тип данных. Программа подсчета строк, слов и литер. Массивы и строки в Си. Программа подсчета цифр, пробельных и прочих литер.
8. Функции в Си. Программа, определяющая принадлежность точки заданному треугольнику.
9. Рекурсия.
10. Функции `strlen`, `copy`, `getline`. Функции перевода числа в строку и строки в число.
11. Операции увеличения и уменьшения. Операции присваивания. Приоритет операций. Побитовые операции. Функция `getbits(x,p,n)`, формирующая поле из n бит, вырезанное из x , начиная с p позиции, прижимая поле к правому краю. Выражения и операторы. Условное выражение.
12. Классы памяти. Автоматические, внешние, статические и регистровые переменные. Многофайловая организация программы. Модуль. Реализация модуля `Stack`.
13. Указатели и адреса. Операции взятия адреса и раскрытия ссылки, их приоритет. Указатели и аргументы функций.
14. Указатели и массивы. Операции над указателями, ссылающимися на элементы одного массива.
15. Указатели и строки. Функции `strcpy`, `strcmp`.
16. Указатели и динамическая память. Простейший распределитель памяти на основе функции `alloc`, `afree`. Динамический массив. Операторы `new`, `delete`. Функция, создающая копию строки в динамической памяти.
17. Массивы указателей и задача сортировки текстовых строк входного потока. Реализация функций ввода-вывода и функции `main`.
18. Массивы указателей и задача сортировки текстовых строк входного потока. Реализация функции быстрой сортировки
19. Реализация модуля функций для работы со строками `item.h`
20. Реализация модуля таблицы строк `Tabl.h`
21. Двумерные массивы и указатели. Функция, суммирующая матрицы произвольной размерности (два варианта с использованием различных подходов).
22. Файловый ввод - вывод.
23. Структуры. Расположение в памяти. Инициализация. Операции. Приоритет операций. Вложенные структуры. Структуры и функции.
24. Реализация таблицы строк в виде структуры.
25. Файловый ввод - вывод
26. Современные средства языка программирования. Оператор разрешения области видимости. Использование функции в качестве элементов структур. Объявления в операторах. Константные объекты. Встроенные функции. Параметры по умолчанию. Передача аргументов функциям и возврат значений по адресу. Перегрузка имен функций
27. Введение в объектно-ориентированное программирование. Стили программирования. Принципы объектно-ориентированного программирования. Абстракция данных. Инкапсуляция. Наследование. Полиморфизм. Повторное использование кода.
28. Классы. Инкапсуляция и члены-функции. Управление доступом. Конструкторы и деструкторы. Отличительные свойства. Взаимодействие с компилятором.

29. Копирование объектов классов. Массивы объектов классов. Объекты классов как элементы
30. Примеры разработки классов: строка, вектор, матрица
31. Объектно-событийное программирование в операционной среде. Событие и сообщение. Кодирование сообщений и механизмы реализации обмена сообщениями в операционной среде. Программирование, управляемое событиями. Природа событий. Виды событий. События от мыши. События от клавиатуры. Передача сообщений.
32. Визуальное программирование в оконной операционной среде. Обзор современных инструментальных систем визуального программирования. Основные характеристики среды. Настройка среды. Основные элементы интегрированной среды разработки. Основы визуального программирования
33. Методы визуальной разработки оконных приложений. Выбор шаблона приложения. Работа с формой. Стандартные визуальные компоненты. Перенос компонент на форму. Установка свойств. События, на которые реагируют компоненты. Задание обработчиков событий
34. Примеры разработки оконных приложений. Организация ввода вывода данных. Разработка программы по обработке простых типов данных. Приложение с кнопками, редактируемым полем и списком. (С помощью кнопок можно добавлять элементы в список, удалять из списка) Приложение - простейший калькулятор
35. Базовые представления об архитектуре ЭВМ. История развития вычислительных средств. Принципы фон Неймана.
36. Классификация компьютеров. Архитектура современного компьютера
37. Классы процессоров. Конвейерная обработка команд. Векторная обработка.
38. Регистры процессоров. Системы команд x86. Макроассемблер.
39. Запоминающие устройства. Иерархия запоминающих устройств. Устройства ввода и вывода
40. Организация оперативной памяти компьютера. Логическая память. Сегменты. Связывание адресов.
41. Простейшие схемы управления памятью. Схемы с фиксированными и переменными разделами. Страничная память. Сегментно-страничная организация памяти.
42. Системы счисления. Представление чисел в позиционных системах счисления. Обоснование экономичности систем счисления.
43. Представление целых и вещественных чисел в позиционных системах счисления.
44. Арифметические операции над числами, представленными в двоичной системе счисления.
45. Способы представления чисел в ЭВМ: с фиксированной запятой, с плавающей запятой.
46. Алгебра высказываний. Операции над высказываниями. Логические связки: инверсия, дизъюнкция, конъюнкция, импликация, эквиваленция, сложение по mod 2. Ранги логических связей.
47. Таблицы истинности. Правила составления таблиц истинности.
48. Формулы алгебры высказываний и равносильные преобразования. Основы равносильности. Равносильности, выражающие одни логические операции через другие.
49. Базовое программное обеспечение. Служебные программы.
50. Назначение операционной системы. Виды операционных систем. Базовые понятия операционных систем.
51. Процессы и потоки.
52. Управление памятью.
53. Драйверы устройств. Файловые системы.
54. Архитектура Windows.

55. Архитектура Linux.
56. Базовые представления об архитектуре ЭВМ. История развития вычислительных средств. Принципы фон Неймана.
57. Классификация компьютеров. Архитектура современного компьютера
58. Классы процессоров. Конвейерная обработка команд. Векторная обработка.
59. Классификация БД Модели данных.
60. Проектирование баз данных. CASE- системы для разработки информационных систем. Реляционная модель данных. Получение реляционной схемы из инфологической модели.
61. База данных Microsoft Access. Потенциальные и внешние ключи. Связывание таблиц.
62. Нормализация таблиц. Ограничения целостности.
63. Назначение и классификация компьютерных сетей. Типы сетей. Топология сетей. Сетевые компоненты.
64. Сети Ethernet. Сети Token Ring.
65. Сетевые протоколы.

5.2.2. Типовые тестовые задания для оценки сформированности компетенций ОПК-2, ОПК-3, ОПК-4, ОПК-5, ОПК-7, ПК-3

Вариант 1.

1. Невытесняющая многозадачность
 - a. обладает высокой степенью централизации механизма планирования процессов
 - b. обладает низкой степенью централизации механизма планирования процессов
 - c. планирование процессов сосредоточено в операционной системе
 - d. планирование процессов осуществляется в прикладных программах
 - e. планирование процессов распределено между операционной системой и прикладными программами
 - f. решение о переключении процессора с одного процесса на другой осуществляется операционной системой
 - g. решение о переключении процессора с одного процесса на другой осуществляется активным процессом
 - h. решение о переключении процессора с одного процесса на другой осуществляется совместно операционной системой и активным процессом
2. Вытесняющая многозадачность
 - a. обладает высокой степенью централизации механизма планирования процессов
 - b. обладает низкой степенью централизации механизма планирования процессов
 - c. планирование процессов сосредоточено в операционной системе
 - d. планирование процессов осуществляется в прикладных программах
 - e. планирование процессов распределено между операционной системой и прикладными программами
 - f. решение о переключении процессора с одного процесса на другой осуществляется операционной системой
 - g. решение о переключении процессора с одного процесса на другой осуществляется активным процессом
 - h. решение о переключении процессора с одного процесса на другой осуществляется совместно операционной системой и активным процессом
3. Системы пакетной обработки

- a. предназначены для управления техническими объектами
 - b. предназначены для решения вычислительных задач, не требующих быстрого получения результата
 - c. предназначены для обеспечения диалога пользователя с компьютером
 - d. наличие предельно допустимого времени для решения задачи
 - e. наличие дополнительных расходов на переключение между задачами
 - f. наличие полной изоляции пользователя от процесса выполнения задачи
 - g. критерием эффективности является удобство работы пользователя
 - h. критерием эффективности является максимальная пропускная способность
 - i. критерием эффективности является реактивность системы
4. Системы разделения времени
- a. предназначены для управления техническими объектами
 - b. предназначены для решения вычислительных задач, не требующих быстрого получения результата
 - c. предназначены для обеспечения диалога пользователя с компьютером
 - d. наличие предельно допустимого времени для решения задачи
 - e. наличие дополнительных расходов на переключение между задачами
 - f. наличие полной изоляции пользователя от процесса выполнения задачи
 - g. критерием эффективности является удобство работы пользователя
 - h. критерием эффективности является максимальная пропускная способность
 - i. критерием эффективности является реактивность системы
5. Системы реального времени
- a. предназначены для управления техническими объектами
 - b. предназначены для решения вычислительных задач, не требующих быстрого получения результата
 - c. предназначены для обеспечения диалога пользователя с компьютером
 - d. наличие предельно допустимого времени для решения задачи
 - e. наличие дополнительных расходов на переключение между задачами
 - f. наличие полной изоляции пользователя от процесса выполнения задачи
 - g. критерием эффективности является удобство работы пользователя
 - h. критерием эффективности является максимальная пропускная способность
 - i. критерием эффективности является реактивность системы
6. Операционная система на базе монолитного ядра
- a. компонуется как одна программа
 - b. содержит компоненты, работающие в пользовательском режиме
 - c. не требует переключения из привилегированного режима в пользовательский и наоборот
 - d. выполняются переходы из привилегированного режима в пользовательский и наоборот
 - e. основные функции управления аппаратурой выполняются в привилегированном режиме
 - f. позволяет наращивать число выполняемых функций
7. Операционная система на базе микроядерного подхода
- a. компонуется как одна программа
 - b. содержит компоненты, работающие в пользовательском режиме
 - c. не требует переключения из привилегированного режима в пользовательский и наоборот
 - d. выполняются переходы из привилегированного режима в пользовательский и наоборот
 - e. основные функции управления аппаратурой выполняются в привилегированном режиме

- f. позволяет наращивать число выполняемых функций
- 8. Рыночными требованиями к операционным системам являются
 - a. Расширяемость
 - b. компактность
 - c. переносимость
 - d. модульность
 - e. надежность
 - f. отказоустойчивость
 - g. помехозащищенность
 - h. совместимость
 - i. безопасность
 - j. производительность
 - k. экономичность
- 9. MS-DOS
 - a. однозадачная
 - b. многозадачная
 - c. однопользовательская
 - d. многопользовательская
 - e. с вытесняющей многозадачностью
 - f. с невытесняющей многозадачностью
 - g. с поддержкой многоплатформенности
 - h. с многопроцессорной обработкой
 - i. без многопроцессорной обработки
 - j. с многопроцессорной асимметричной аппаратурой
 - k. с многопроцессорной симметричной аппаратурой
 - l. сетевая
 - m. с монолитным ядром
 - n. на базе микроядра
 - o. с одновременной поддержкой многих прикладных сред
 - p. с поддержкой графического интерфейса
- 10. Windows 3.x
 - a. однозадачная
 - b. многозадачная
 - c. однопользовательская
 - d. многопользовательская
 - e. с вытесняющей многозадачностью
 - f. с невытесняющей многозадачностью
 - g. с поддержкой многоплатформенности
 - h. с многопроцессорной обработкой
 - i. без многопроцессорной обработки
 - j. с многопроцессорной асимметричной аппаратурой
 - k. с многопроцессорной симметричной аппаратурой
 - l. сетевая
 - m. с монолитным ядром
 - n. на базе микроядра
 - o. с одновременной поддержкой многих прикладных сред
 - p. с поддержкой графического интерфейса
- 11. Windows 95
 - a. однозадачная
 - b. многозадачная

- c. однопользовательская
- d. многопользовательская
- e. с вытесняющей многозадачностью
- f. с невытесняющей многозадачностью
- g. с поддержкой многопитевности
- h. с многопроцессорной обработкой
- i. без многопроцессорной обработки
- j. с многопроцессорной асимметричной аппаратурой
- k. с многопроцессорной симметричной аппаратурой
- l. сетевая
- m. с монолитным ядром
- n. на базе микроядра
- o. с одновременной поддержкой многих прикладных сред
- p. с поддержкой графического интерфейса

12. Windows XP

- a. однозадачная
- b. многозадачная
- c. однопользовательская
- d. многопользовательская
- e. с вытесняющей многозадачностью
- f. с невытесняющей многозадачностью
- g. с поддержкой многопитевности
- h. с многопроцессорной обработкой
- i. без многопроцессорной обработки
- j. с многопроцессорной асимметричной аппаратурой
- k. с многопроцессорной симметричной аппаратурой
- l. сетевая
- m. с монолитным ядром
- n. на базе микроядра
- o. с одновременной поддержкой многих прикладных сред
- p. с поддержкой графического интерфейса

13. Linux

- a. однозадачная
- b. многозадачная
- c. однопользовательская
- d. многопользовательская
- e. с вытесняющей многозадачностью
- f. с невытесняющей многозадачностью
- g. с поддержкой многопитевности
- h. с многопроцессорной обработкой
- i. без многопроцессорной обработки
- j. с многопроцессорной асимметричной аппаратурой
- k. с многопроцессорной симметричной аппаратурой
- l. сетевая
- m. с монолитным ядром
- n. на базе микроядра
- o. с одновременной поддержкой многих прикладных сред
- p. с поддержкой графического интерфейса

Вариант 2.

1. Тип данных определяется
 - a. множеством значений
 - b. внутренней структурой
 - c. отношением между составляющими элементами
 - d. свойствами составляющих элементов
 - e. набором операций над объектами данного типа
2. Поле характеризуется
 - a. уникальным именем
 - b. внутренней структурой
 - c. отношением между составляющими элементами
 - d. свойствами составляющих элементов
 - e. типом данных, которые хранятся в поле
 - f. цветом ячейки
 - g. шрифтом заголовка
3. Поля таблицы Access содержат следующие типы данных
 - a. байт
 - b. MEMO
 - c. форматированный текст
 - d. счетчик
 - e. логический тип
 - f. дата-время
 - g. гиперссылка
 - h. объект OLE
 - i. объект GRAPHICS
4. Запись - это
 - a. множество логически связанных полей
 - b. множество атрибутов некоторого семейства объектов
 - c. множество свойств некоторого семейства объектов
 - d. строка таблицы
 - e. столбец таблицы
 - f. заголовок таблицы
5. База данных Access - это
 - a. папка с файлами таблиц
 - b. папка с файлами таблиц, запросов, форм, отчетов
 - c. файл, содержащий таблицы
 - d. файл, содержащий таблицы, запросы, формы, отчеты
 - e. файл с расширением .db
 - f. файл с расширением .bd
 - g. файл с расширением .mdb
 - h. файл с расширением .dbm
 - i. файл с расширением .mbd
6. Связь один - к - одному означает
 - a. каждый экземпляр первой сущности связан с одним экземпляром второй сущности
 - b. каждый экземпляр первой сущности связан с несколькими экземплярами второй сущности
 - c. каждый экземпляр второй сущности связан с одним экземпляром первой сущности

- d. каждый экземпляр второй сущности связан с несколькими экземплярами первой сущности
 - e. сущность неправильно разделенную на две
 - f. необходимость введения вспомогательной сущности
 - g. наиболее часто используемый тип связи
 - h. временный тип связи
7. Связь один - ко – многим означает
- a. каждый экземпляр первой сущности связан с одним экземпляром второй сущности
 - b. каждый экземпляр первой сущности связан с несколькими экземплярами второй сущности
 - c. каждый экземпляр второй сущности связан с одним экземпляром первой сущности
 - d. каждый экземпляр второй сущности связан с несколькими экземплярами первой сущности
 - e. сущность неправильно разделенную на две
 - f. необходимость введения вспомогательной сущности
 - g. наиболее часто используемый тип связи
 - h. временный тип связи
8. Связь много - ко – многим означает
- a. каждый экземпляр первой сущности связан с одним экземпляром второй сущности
 - b. каждый экземпляр первой сущности связан с несколькими экземплярами второй сущности
 - c. каждый экземпляр второй сущности связан с одним экземпляром первой сущности
 - d. каждый экземпляр второй сущности связан с несколькими экземплярами первой сущности
 - e. сущность неправильно разделенную на две
 - f. необходимость введения вспомогательной сущности
 - g. наиболее часто используемый тип связи
 - h. временный тип связи
9. Ключ сущности является
- a. средством ограничения доступа к экземпляру сущности
 - b. средством определения прав доступа
 - c. средством задания пароля доступа
 - d. средством идентификации сущности
10. Ключ сущности представляет собой
- a. набор ограничений на выполнение определенных операций с экземпляром сущности
 - b. набор прав на выполнение определенных операций с экземпляром сущности
 - c. пароль доступа к экземпляру сущности
 - d. минимальный набор атрибутов, по значениям которых можно однозначно найти требуемый экземпляр сущности
 - e. набор свойств, уникальных для каждого экземпляра сущности
 - f. набор всех связей с другими сущностями
 - g. набор связей типа один – ко -многим с другими сущностями
11. Ключ может быть
- a. открытым
 - b. закрытым
 - c. защищенным паролем

- d. с высокой степенью защиты
- e. простым
- f. смешанным
- g. составным
- h. комбинированным
- i. естественным
- j. абстрактным
- k. суррогатным
- l. формальным
- m. первичным
- n. вторичным
- o. потенциальным
- p. активным

12. Внешний ключ

- a. является идентификатором объектов предметной области
- b. является набором прав на выполнение определенных операций с экземпляром сущности
- c. отражает связи один

Вариант 3.

1. Клиент - это

- a. Компьютер, который находится в режиме ожидания запросов
- b. Компьютер, который обслуживает запросы на доступ к ресурсам
- c. Компьютер, который вырабатывает запросы на доступ к удаленным ресурсам
- d. Компьютер, который посылает запросы по сети
- e. Компьютер, который содержит набор программных модулей сетевой операционной системы
- f. Компьютер, который содержит набор программных модулей способных отличить запрос на доступ к локальным ресурсам от запроса на доступ к удаленным ресурсам.

2. Сервер - это

- a. Компьютер, который находится в режиме ожидания запросов
- b. Компьютер, который обслуживает запросы на доступ к ресурсам
- c. Компьютер, который вырабатывает запросы на доступ к удаленным ресурсам
- d. Компьютер, который посылает запросы по сети
- e. Компьютер, который содержит набор программных модулей сетевой операционной системы
- f. Компьютер, который содержит набор программных модулей способных отличить запрос на доступ к локальным ресурсам от запроса на доступ к удаленным ресурсам

3. Повторитель

- a. соединяют устройства сети
- b. используется для расширения сети
- c. используется для объединения сетей с разными типами оборудования и программного обеспечения
- d. работает в сетях с замкнутыми контурами
- e. разбивает сеть на сегменты
- f. повторяют всю поступающую информацию
- g. имеет несколько портов

- h. обрабатывает кадры в параллельном режиме
- i. локализует трафик в пределах сегмента
- j. увеличивает необязательный поток данных
- k. минимизирует необязательный поток данных
- l. анализирует адрес назначения данных
- m. выбирает наиболее рациональный маршрут из нескольких возможных
- n. использует аппаратные адреса компьютеров
- o. использует числовые составные адреса компьютеров
- p. изменяет физическую топологию сети
- q. не изменяет физическую топологию сети
- r. изменяет логическую топологию сети
- s. не изменяет логическую топологию сети

4. Концентратор

- a. соединяют устройства сети
- b. используется для расширения сети
- c. используется для объединения сетей с разными типами оборудования и программного обеспечения
- d. работает в сетях с замкнутыми контурами
- e. разбивает сеть на сегменты
- f. повторяют всю поступающую информацию
- g. имеет несколько портов
- h. обрабатывает кадры в параллельном режиме
- i. локализует трафик в пределах сегмента
- j. увеличивает необязательный поток данных
- k. минимизирует необязательный поток данных
- l. анализирует адрес назначения данных
- m. выбирает наиболее рациональный маршрут из нескольких возможных
- n. использует аппаратные адреса компьютеров
- o. использует числовые составные адреса компьютеров
- p. изменяет физическую топологию сети
- q. не изменяет физическую топологию сети
- r. изменяет логическую топологию сети
- s. не изменяет логическую топологию сети

5. Мост

- a. соединяют устройства сети
- b. используется для расширения сети
- c. используется для объединения сетей с разными типами оборудования и программного обеспечения
- d. работает в сетях с замкнутыми контурами
- e. разбивает сеть на сегменты
- f. повторяют всю поступающую информацию
- g. имеет несколько портов
- h. обрабатывает кадры в параллельном режиме
- i. локализует трафик в пределах сегмента
- j. увеличивает необязательный поток данных
- k. минимизирует необязательный поток данных
- l. анализирует адрес назначения данных
- m. выбирает наиболее рациональный маршрут из нескольких возможных
- n. использует аппаратные адреса компьютеров
- o. использует числовые составные адреса компьютеров
- p. изменяет физическую топологию сети

- q. не изменяет физическую топологию сети
- r. изменяет логическую топологию сети
- s. не изменяет логическую топологию сети

6. Коммутатор

- a. соединяют устройства сети
- b. используется для расширения сети
- c. используется для объединения сетей с разными типами оборудования и программного обеспечения
- d. работает в сетях с замкнутыми контурами
- e. разбивает сеть на сегменты
- f. повторяют всю поступающую информацию
- g. имеет несколько портов
- h. обрабатывает кадры в параллельном режиме
- i. локализует трафик в пределах сегмента
- j. увеличивает необязательный поток данных
- k. минимизирует необязательный поток данных
- l. анализирует адрес назначения данных
- m. выбирает наиболее рациональный маршрут из нескольких возможных
- n. использует аппаратные адреса компьютеров
- o. использует числовые составные адреса компьютеров
- p. изменяет физическую топологию сети
- q. не изменяет физическую топологию сети
- r. изменяет логическую топологию сети
- s. не изменяет логическую топологию сети

7. Шлюз

- a. соединяют устройства сети
- b. используется для расширения сети
- c. используется для объединения сетей с разными типами оборудования и программного обеспечения
- d. работает в сетях с замкнутыми контурами
- e. разбивает сеть на сегменты
- f. повторяют всю поступающую информацию
- g. имеет несколько портов
- h. обрабатывает кадры в параллельном режиме
- i. локализует трафик в пределах сегмента
- j. увеличивает необязательный поток данных
- k. минимизирует необязательный поток данных
- l. анализирует адрес назначения данных
- m. выбирает наиболее рациональный маршрут из нескольких возможных
- n. использует аппаратные адреса компьютеров
- o. использует числовые составные адреса компьютеров
- p. изменяет физическую топологию сети
- q. не изменяет физическую топологию сети
- r. изменяет логическую топологию сети
- s. не изменяет логическую топологию сети

8. Полносвязная топология

- a. удобна для организации обратной связи
- b. каждый компьютер связан со всеми остальными

- с. удобна для поиска некорректно работающего узла
- d. для каждой пары компьютеров выделена отдельная электрическая линия
- e. данные передаются в одном направлении
- f. данные передаются в двух направлениях
- g. используется в сетях с малым (меньше 10) количеством компьютеров
- h. используется в сетях с большим (много больше 10) количеством компьютеров
- i. используется в сетях со средним (порядка 10) количеством компьютеров
- j. имеет низкую надежность
- k. имеет относительно высокую надежность
- l. имеет небольшую стоимость реализации
- m. имеет относительно высокую стоимость оборудования
- n. имеет невысокую производительность
- o. имеет строгое ограничение по числу узлов в сети
- p. не имеет ограничений на число узлов в сети

9. Ячеистая топология

- a. удобна для организации обратной связи
- b. каждый компьютер связан со всеми остальными
- с. удобна для поиска некорректно работающего узла
- d. для каждой пары компьютеров выделена отдельная электрическая линия
- e. данные передаются в одном направлении
- f. данные передаются в двух направлениях
- g. используется в сетях с малым (меньше 10) количеством компьютеров
- h. используется в сетях с большим (много больше 10) количеством компьютеров
- i. используется в сетях с средним (порядка 10) количеством компьютеров
- j. имеет низкую надежность
- k. имеет относительно высокую надежность
- l. имеет небольшую стоимость реализации
- m. имеет относительно высокую стоимость оборудования
- n. имеет невысокую производительность
- o. имеет строгое ограничение по числу узлов в сети
- p. не имеет ограничений на число узлов в сети

10. Общая шина

- a. удобна для организации обратной связи
- b. каждый компьютер связан со всеми остальными
- с. удобна для поиска некорректно работающего узла
- d. для каждой пары компьютеров выделена отдельная электрическая линия
- e. данные передаются в одном направлении
- f. данные передаются в двух направлениях
- g. используется в сетях с малым (меньше 10) количеством компьютеров
- h. используется в сетях с большим (много больше 10) количеством компьютеров
- i. используется в сетях с средним (порядка 10) количеством компьютеров
- j. имеет низкую надежность
- k. имеет относительно высокую надежность
- l. имеет небольшую стоимость реализации
- m. имеет относительно высокую стоимость оборудования
- n. имеет невысокую производительность
- o. имеет строгое ограничение по числу узлов в сети

р. не имеет ограничений на число узлов в сети

Вариант 4.

1. Согласно процедурному стилю программирования программа представляет собой
 - a. последовательность исполняемых инструкций
 - b. последовательность описаний данных и операций над ними
 - c. набор взаимосвязанных процедур
 - d. набор иерархически организованных процедур
 - e. набор файлов
 - f. набор модулей
 - g. набор классов
 - h. набор иерархически организованных классов
2. Согласно объектно-ориентированному стилю программирования программа представляет собой
 - a. последовательность исполняемых инструкций
 - b. последовательность описаний данных и операций над ними
 - c. набор взаимосвязанных процедур
 - d. набор иерархически организованных процедур
 - e. набор файлов
 - f. набор модулей
 - g. набор классов
 - h. набор иерархически организованных классов
3. Абстрактный тип данных (АТД)- это тип данных, у которого
 - a. отсутствует внутренняя структура
 - b. скрыта внутренняя структура
 - c. внутренняя структура представлена в абстрактной форме
 - d. полностью закрыт доступ внутренней структуре данных
 - e. доступ к данным разрешен только функциям, определенным в интерфейсе типа
 - f. инкапсулированы данные и обрабатывающие их функции
 - g. свойства и операции выражаются в абстрактном виде
 - h. свойства и операции не зависят от его внутреннего устройства
 - i. свойства определяются набором операций над ним
4. Модуль в языке C++
 - a. представляет собой набор данных и соответствующих им функций, размещенный в отдельном файле
 - b. представляет собой набор данных и соответствующих им функций, размещенный в двух файлах
 - c. представляет собой набор данных и соответствующих им функций, размещенный в нескольких файлах
 - d. обеспечивает сокрытие данных
 - e. имеет свое собственное пространство имен
 - f. имеет отдельно пространство имен данных и пространство имен функций
5. Типы, создаваемые как модули
 - a. обрабатываются компилятором
 - b. обрабатываются препроцессором
 - c. является элементом языка
 - d. являются абстрактными типами данных
6. Класс – это
 - a. встроенный тип данных языка Си
 - b. встроенный тип данных языка C++

- c. обобщение понятия структуры
 - d. тип, который содержит данные и ссылки на функции
 - e. тип, который содержит данные и функции
 - f. определяемый пользователем тип данных
7. Класс отличается от структуры языка Си
- a. ключевым словом в описании типа
 - b. использованием структур в качестве элементов
 - c. использованием указателей на функции в качестве элементов
 - d. использованием функций в качестве элементов
 - e. наличием специальных функций-элементов класса
 - f. наличием внутреннего пространства имен
 - g. наличием отдельного пространства имен данных и пространство имен функций
 - h. ограничением доступа к данным
 - i. ограничением доступа к данным и функциям-элементам
8. Для работы с классом в язык C++ введены следующие ключевые слова:
- a. class
 - b. private
 - c. protected
 - d. arrived
 - e. abstract
 - f. free
9. Для конструктора справедливы следующие утверждения:
- a. имя конструктора начинается с большой буквы
 - b. имя конструктора совпадает с именем класса.
 - c. имя конструктора – Init
 - d. имя конструктора определяется разработчиком
 - e. класс может содержать несколько конструкторов с различными именами
 - f. конструктор имеет тип возвращаемого значения void
 - g. конструктор имеет тип возвращаемого значения int
 - h. конструктор не имеет возвращаемого значения.
 - i. тип возвращаемого значения определяется разработчиком
 - j. нельзя в явном виде вызвать конструктор
 - k. вызов конструктора осуществляется так же, как и любой другой функции элемента класса
 - l. конструктор имеет свою семантику вызова
 - m. вызов конструктора осуществляется автоматически компилятором при создании объектов класса
 - n. без определения в классе конструктора нельзя создать объект класса
 - o. без определения в классе конструктора нельзя инициализировать закрытые данные класса
 - p. конструктор не содержит параметров
 - q. в классе всегда должен быть определен конструктор без параметров
10. Конструктор копирования в качестве аргумента может иметь:
- a. Тип void
 - b. Объект некоторого класса
 - c. Объект такого же класса
 - d. Указатель на объект такого же класса
 - e. Ссылку на объект такого же класса
 - f. Константную ссылку на объект такого же класса
11. Для деструктора справедливы следующие утверждения:
- a. Деструктор используется для удаления объекта класса

- b. Деструктор используется для освобождения памяти, выделенной под элементы объекта класса, до удаления самого объекта
- c. имя деструктора начинается с большой буквы
- d. имя деструктора совпадает с именем класса.
- e. Имя деструктора совпадает с именем класса с предшествующим символом ~ (тильда).
- f. имя деструктора – Dispose
- g. имя деструктора определяется разработчиком
- h. класс может содержать несколько деструкторов с различными именами
- i. деструктор имеет тип возвращаемого значения void
- j. деструктор имеет тип возвращаемого значения int
- k. деструктор не имеет возвращаемого значения.
- l. тип возвращаемого значения определяется разработчиком
- m. деструктор не имеет параметров
- n. в классе всегда должен быть определен деструктор
- o. нельзя в явном виде вызвать деструктор
- p. вызов деструктора осуществляется так же, как и любой другой функции элемента класса
- q. Вызов деструктора выполняется компилятором автоматически при выходе переменной из области действия

5.2.3. Типовые практические задачи для оценки сформированности компетенций ОПК-2, ОПК-3, ОПК-4, ОПК-5, ОПК-7, ПК-3

1. ВЫЧИСЛЕНИЕ ЗНАЧЕНИЙ ФУНКЦИИ

Вычислить значение функции y от x . Значение переменной x и параметров a и b вводить с клавиатуры (Параметр a вещественного типа, параметр b целочисленный, справа от функции даны значения для теста программы):

$$1. y = \ln(5 + ax)^b; \quad \text{при } x=3,5; \quad a=0,2; -1,5 \quad b=2; 3$$

$$2. y = \ln(\cos ax)^{\frac{b}{3}}; \quad \text{при } x=0,3; \quad a=2,1; 9,8 \quad b=1; 6;$$

$$3. y = \sqrt[b]{\sin(b + ax)}; \quad \text{при } x=1,03; \quad a=2,5; 0,3; \quad b=2; 3$$

$$4. y = \log_{ax}(3 + bx); \quad \text{при } x=0,3; \quad a=0,2; -1,2 \quad b=6; -1$$

$$5. y = (x^2 \sin ax)^b; \quad \text{при } x=1,3; \quad a=0,2; 3,1; \quad b=2; 3;$$

$$6. y = |(ax)^{b/2} - 3|; \quad \text{при } x=2,3; \quad a=1,2; -1,5; \quad b=2; 3;$$

$$7. y = \ln^{\frac{b}{4}}(ax - b); \quad \text{при } x=1,03; \quad a=1,2; 4,5; \quad b=4; 1;$$

$$8. y = \log_3^{b/3}(5 + ax^3); \quad \text{при } x=1,03; \quad a=-4; -5,2; \quad b=6; 1$$

$$9. y = \left| (b + ax^2)^{\frac{b}{2}} \right|; \quad \text{при } x=2,3; \quad a=-1,2; 0,3 \quad b=2; 3$$

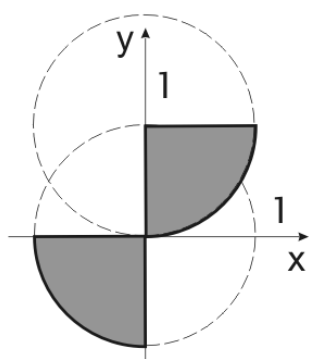
$$10. y = \sqrt[b]{x^{\frac{5}{3}} + ax}; \quad \text{при } x=-1,3; \quad a=-2,2; 1,1 \quad b=2; 3$$

2. ВЫДЕЛЕНИЕ ОБЛАСТЕЙ

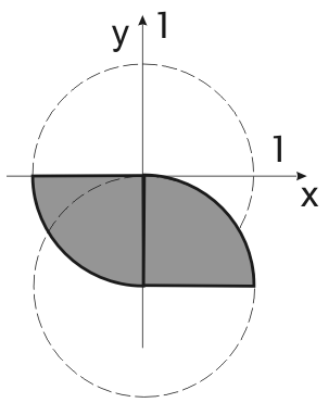
Определить принадлежность точки с координатами x, y заштрихованной части плоскости:

1.

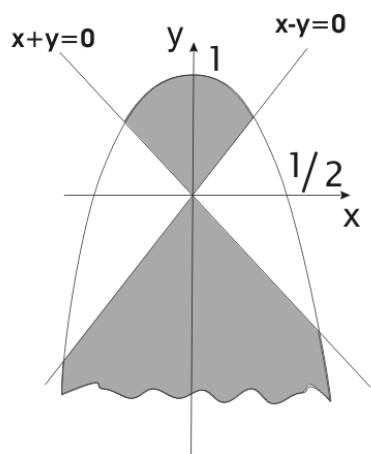
2.



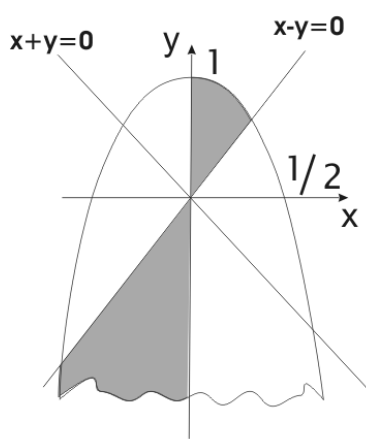
3.



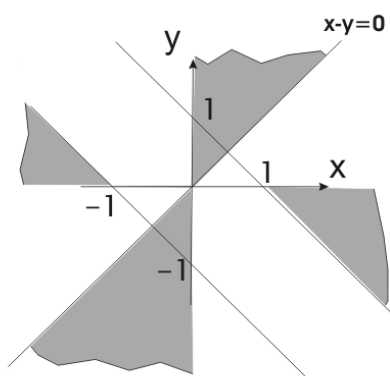
4.



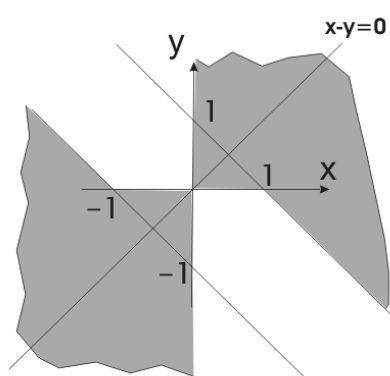
5.



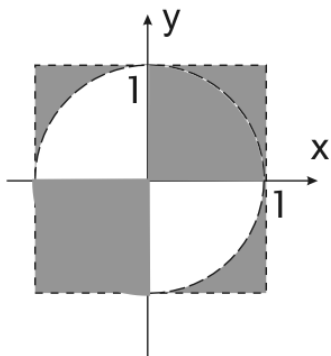
6.



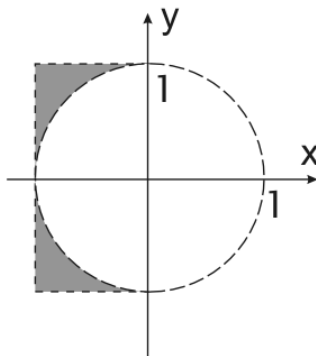
7.



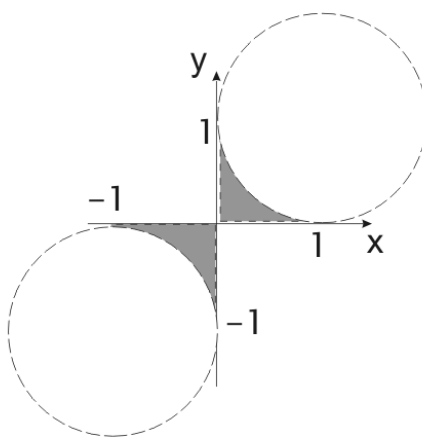
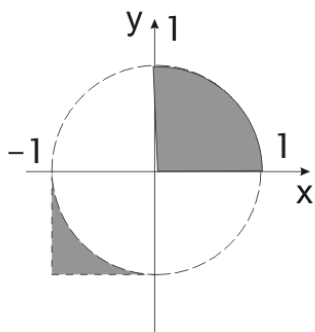
8.



9.



10.



2. СУММИРОВАНИЕ РЯДОВ

Выполнить следующие вычисления:

а) задавая x и n (число членов суммы ряда), найти сумму S и

$$\delta = |s - f(x)|;$$

б) задавая x и n (число членов суммы ряда), найти сумму S в порядке убывания индекса до нуля;

в) задавая x и δ_0 , найти сумму S и число членов суммы n . Суммирование выполнять до тех пор, пока не выполнится условие:

$$|s - f(x)| < \delta_0$$

1. $\sum_{n=0}^{\infty} \frac{1}{n(n+1)}$, $f(x) = 1$.

2. $\sum_{n=0}^{\infty} \frac{1}{(2n-1)(2n+1)}$, $f(x) = 1/2$.

3. $\sum_{n=0}^{\infty} \frac{1}{2^n}$, $f(x) = 2$.

4. $\sum_{n=0}^{\infty} (-1)^n \frac{1}{2^n}$, $f(x) = 2/3$.

5. $\sum_{n=0}^{\infty} \frac{1}{n!}$, $f(x) = e$.

6. $\sum_{n=0}^{\infty} (-1)^n \frac{1}{n!}$, $f(x) = 1/e$.
7. $\sum_{n=0}^{\infty} (-1)^{n+1} \frac{1}{2n-1}$, $f(x) = \pi/4$.
8. $\sum_{n=0}^{\infty} (-1)^{n+1} \frac{1}{n^2}$, $f(x) = \pi^2/12$.
9. $\sum_{n=0}^{\infty} (-1)^{n+1} \frac{1}{n^4}$, $f(x) = 7\pi^4/720$.
10. $\sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!}$, $f(x) = \cos(x)$.

3. ПОЗИЦИОННАЯ ЗАПИСЬ ЧИСЛА

1. Определить количество натуральных трехзначных чисел, сумма цифр которых равна заданному числу N.
2. Среди четырехзначных чисел выбрать те, у которых все четыре цифры различны.
3. Найти количество четных цифр заданного натурального числа N.
4. Найти все симметричные натуральные числа (палиндромы) из интервала от K до L.
5. Для данного натурального числа N определить: содержит ли число одинаковые цифры и их количество.
6. Найти все натуральные числа, не превосходящие заданного N, и делящиеся на каждую из своих цифр
7. Найти все натуральные числа, не превосходящие заданного N, десятичная запись которых есть строго возрастающая или строго убывающая последовательность цифр.
8. Найти все пары двухзначных натуральных чисел M и N таких, что значение произведения M*N не изменится, если поменять местами цифры каждого из сомножителей (такой парой будет , например, 38 и 83).
9. Определить сколько раз данная цифра K встречается в целом числе N.
10. Среди двузначных чисел найти те, сумма квадратов цифр которых делится на 13.

4. ДЕЛИТЕЛИ ЦЕЛОГО ЧИСЛА

1. Дано натуральное число N. Получить все его натуральные делители.
2. Найти количество делителей натурального числа N. Сколько из них четных?
3. Найти сумму нечетных делителей натурального числа N.
4. Найти все натуральные числа из интервала от 1 до 200, у которых количество делителей равно N.
5. Найти сумму целых чисел из промежутка от 1 до 200, у которых ровно 5 делителей
6. Найти все целые числа из промежутка от 100 до 300, у которых сумма делителей равна K.
7. Найти натуральное число, лежащее в диапазоне от 1 до 10000 с максимальной суммой делителей.
8. Найти все натуральные числа из промежутка от K до L, у которых количество делителей превышает заданное число M.
9. Найти сумму четных делителей натурального числа N.
10. Найти все натуральные числа из промежутка от 1 до 200, у которых сумма четных делителей равна N.

5. ПРЕОБРАЗОВАНИЕ И ПОСТРОЕНИЕ ОДНОМЕРНЫХ МАССИВОВ

Даны два массива целых чисел a и b , каждый из которых не содержит повторяющихся элементов

1. Построить пересечение массивов a и b
2. Построить объединение массивов a и b
3. Получить все элементы массивов a , которые не входят в массив b .
4. Получить все элементы массивов a и b , которые не входят в них одновременно.
5. Получить все элементы, содержащиеся хотя бы в одном из массивов a и b (если число есть в обоих массивах, в новом оно должно появиться только один раз)
6. Верно ли, что все элементы массива a входят в массив b
7. Верно ли, что все элементы массива a входят в массив b и при этом $a[1]$ встречается в массиве b не позднее, чем $a[2]$, $a[2]$ встречается в массиве b не позднее, чем $a[3]$ и т.д.

8. Верно ли, что два массива, которые могут содержать повторяющиеся элементы, отличаются не более чем порядком их следования

Дан массив целых чисел a , который может содержать повторяющиеся элементы

9. Получить все числа, которые входят в массив по одному разу
10. Получить все числа, взятые по одному из каждой группы равных элементов
11. Найти число различных элементов массива
12. Выяснить, сколько чисел входят в массив по одному разу
13. Выяснить, сколько чисел входят в массив более чем по одному разу
14. Выяснить, имеется ли массиве хотя бы одна пара совпадающих чисел
15. Найти число повторяющееся в массиве максимальное количество раз
16. Найти число вхождений каждого элемента массива
17. Удалить из массива a все отрицательные элементы
18. Удалить из массива a все элементы, значения которых совпадают с максимальным
19. Переставить все элементы массива так, чтобы сначала расположились все ее неотрицательные элементы, а затем все отрицательные. Порядок как среди неотрицательных элементов, так и среди отрицательных должен быть сохранен прежним
20. Циклически сдвинуть на k позиций влево все элементы массива
21. Расположить все элементы массива в обратном порядке
22. Преобразовать массив по следующему правилу: $a[k] = \max a[j]$ при $0 \leq j \leq k$
23. Найти упорядоченную по возрастанию последовательность подряд расположенных элементов массива максимальной длины
24. Найти симметричную последовательность подряд расположенных элементов массива максимальной длины
25. Найти последовательность подряд расположенных неповторяющихся элементов массива максимальной длины

5. МАКСИМАЛЬНЫЙ И МИНИМАЛЬНЫЙ ЭЛЕМЕНТЫ МАССИВА. СОРТИРОВКА

1. Даны пять различных целых чисел. Упорядочить их по возрастанию, используя не более семи сравнений.
2. Определить упорядочены ли элементы массива
3. Заданы координаты n точек на плоскости. Найти прямоугольник, объемлющий все эти точки.
4. Поменять местами максимальный отрицательный и минимальный положительный элементы массива
5. Вывести на экран элементы массива целых чисел, имеющих максимальное количество делителей.

6. Вывести на экран элементы массива целых чисел, имеющих максимальную сумму цифр.
7. Вывести на экран в порядке возрастания четные элементы массива
8. Вывести на экран различные элементы массива целых чисел в порядке возрастания их числа повторения.
9. Вывести на экран элементы массива целых чисел в порядке возрастания их числа делителей.
10. Вывести на экран элементы массива целых чисел в порядке возрастания их суммы цифр.
11. Найти методом бинарного поиска в упорядоченном массиве местонахождения всех чисел от 0 до 9.
12. Заданы два одномерных упорядоченных массива a и b . Найти методом бинарного поиска все элементы массива a , которые не входят в массив b .
13. Заданы два одномерных упорядоченных массива a и b . Вывести на экран различные элементы массива a в порядке появления их в массиве b .
14. Заданы два одномерных упорядоченных массива a и b . Вывести на экран различные элементы массива a в порядке обратном появлению их в массиве b .
15. Заданы два одномерных упорядоченных массива a и b . Получить новый массив, состоящий из чисел массивов a и b без повторений, упорядоченный по возрастанию.
16. Заданы два одномерных упорядоченных массива размерностью m и n соответственно. Образовать из этих элементов упорядоченный массив размерностью $m+n$
17. Упорядочить массив, используя алгоритм сортировки слиянием упорядоченных групп элементов массива. Вначале весь массив рассматривается как совокупность упорядоченных групп по одному элементу в каждом. Слиянием соседних групп получаем упорядоченные группы, каждая из которых содержит два элемента. Далее упорядоченные группы укрупняются тем же способом и т.д. Алгоритм предполагает использование вспомогательного массива.
18. Упорядочить массив, используя алгоритм сортировки выбором: отыскивается максимальный элемент и переносится в конец массива; затем этот метод применяется ко всем элементам, кроме последнего (он уже находится на своем окончательном месте), и т.д.
19. Упорядочить массив, используя алгоритм сортировки вставками: пусть первые n элементов уже упорядочены; берется $(n+1)$ -й элемент и с помощью последовательного просмотра размещается среди первых n элементов так, чтобы упорядоченными оказались уже $(n+1)$ первых элементов, и т.д.
20. Упорядочить массив, используя алгоритм сортировки бинарными вставками, в котором место размещения элемента в упорядоченном отрезке массива определяется методом бинарного поиска.
21. Дана ведомость зарплаты сотрудников, в которой указаны табельный номер сотрудников и зарплата каждого. Вывести на экран список табельных номеров сотрудников в порядке увеличения их зарплаты.
22. В налоговой инспекции составлен реестр налогоплательщиков, в котором для каждого из них указаны фамилия и сумма уплаченного налога. Упорядочить налогоплательщиков по убыванию налоговой суммы.
23. В деканате составлена ведомость, в которой указаны фамилия студентов, название предметов и количество прогулов по каждому предмету. Вывести на экран фамилии студентов, имеющих максимальное суммарное число прогулов по всем предметам.
24. В деканате составлена ведомость, в которой указаны фамилия студентов, название предметов и количество прогулов по каждому предмету. Вывести на экран фамилии студентов в порядке увеличения их суммарного числа прогулов по всем предметам.

25. Дана таблица стран-участниц олимпийских игр с указанием для каждой из них количества завоеванных золотых серебряных и бронзовых медалей. Упорядочить все страны по убыванию количества золотых медалей. Из двух стран с одинаковым числом золотых медалей выше должна оказаться страна, у которой больше серебряных медалей. Если и здесь равенство, то преимущество должна иметь страна с большим числом бронзовых медалей.

6. ОБРАБОТКА ПОСЛЕДОВАТЕЛЬНОСТЕЙ СИМВОЛОВ

1. удалить символы, стоящие перед символом *. Рассмотреть случай наличия в последовательности нескольких символов *;
2. подсчитать сколько раз среди символов встречается буква х;
3. исключить все символы между круглыми скобками, включая сами скобки. Рассмотреть случай вложенных скобок;
4. подсчитать наибольшее число подряд идущих пробелов;
5. заменить все восклицательные знаки точками;
6. заменить каждую группу символов NN группой символов Nizhny Novgorod;
7. удалить все символы, не являющимися буквами;
8. распечатать все символы, расположенные между двумя запятыми;
9. подсчитать число вхождений в строку группы символов abc;
10. заменить все прописные латинские буквы одноименными строчными;
11. удалить из каждой группы идущих подряд цифр нули;
12. удалить из каждой группы идущих подряд цифр, все цифры, начиная с третьей;
13. удалить из строки все запятые, предшествующие первой точке;
14. преобразовать строку, удалив из нее каждый символ * и повторив каждый символ, отличный от *;
15. выяснить, входят ли в строку все символы, входящие в слово STRING;
16. проверить, является ли строка палиндромом (выражение, читающееся слева направо и справа налево одинаково);
17. преобразовать строку так, чтобы сначала в ней шли все цифры, а потом все буквы исходной строки;
18. выбрать из строки все буквы и отсортировать их в алфавитном порядке;
19. заметить все русские буквы в строке их порядковыми номерами в русском алфавите;
20. подсчитать частоту встречаемости каждой цифры в строке;
21. подсчитать наибольшее число букв А, идущий подряд;
22. реверсировать (расположить в обратном порядке) все символы строки, расположенные до первой точки;
23. разделить исходную строку на несколько строк длины к.
24. определить содержит ли строка дату в виде xx.xx.xx;
25. ввести строку, содержащую дату в формате xx-месяц-xx и преобразовать ее в формат xx.xx.xx

5.2.4. Темы курсовых работ

Курсовая работа состоит из набора лабораторных работ, выполняемых студентами в течение семестра, охватывающих основные темы курса. Каждая лабораторная работа содержит необходимый теоретический материал по теме работы, разработанный студентами алгоритм решения задачи, его блок-схему, практическую реализацию на языке программирования Си, а также результаты тестирования программного кода. Разработанные в ходе выполнения лабораторных работ программные коды, будут использоваться студентами во время выполнения практических экзаменационных заданий по курсу.

№	Тема лабораторной работы	Список реализуемых задач
1	Обработка одномерных (линейных) массивов	<ul style="list-style-type: none"> • вставка элементов одного линейного массива внутрь другого; • удаление элементов линейного массива, совпадающих по значению с элементами другого линейного массива; • сжатие по индексам • нахождение скалярного произведения двух n-мерных векторов; • нахождение в линейном массиве суммы его элементов и наиболее близкого и наиболее далекого по значениям элементов к ней;
2	Обработка двумерных (плоских) массивов	<ul style="list-style-type: none"> • формирование по плоскому прямоугольному массиву линейного массива с заданным правилом обхода; • транспонирование квадратной матрицы; • произведение двух матриц; • нахождение в квадратной матрице разности сумм элементов главной и побочной диагоналей;
3	Обработка текстовых данных	<ul style="list-style-type: none"> • определение символьной длины строки; • конкатенация (объединение строк); • нахождение вхождения одной строки в другую; • удаление из строки слов, размещенных в другой строке; • формирование гистограммы распределения символов в строке с размещением в 2 массивах: символьном и числовом;
4	Реализация множеств посредством линейных массивов	<ul style="list-style-type: none"> • добавление элемента в множество; • удаление элементов из множества; • объединение, пересечение, разность и симметричная разность множеств;
5	Элементы кластерного анализа	Формирование кластера методом ближайшего соседа
6	Динамическая структура данных – Линейный список	<ul style="list-style-type: none"> • реализация базовых операций работы со списком: создание списка, вставка, удаление, нахождение элемента по значению; • формирование гистограммы символов в строке с размещением в динамическом списке;
7	Динамическая структура данных – Бинарное дерево	<ul style="list-style-type: none"> • реализация базовых операций работы с бинарным деревом: создание, вставка, удаление, нахождение элемента по значению, реализация базовых обходов бинарного дерева;

Примерная тема курсовой работы «Обработка массивов и реализация динамических структур данных в интегрированной среде разработки Dev C++».

5.2.5. Вопросы для самоконтроля

1. Характеристика языка программирования Си. Операторы ввода >> и вывода <<. Форматный ввод, вывод библиотеки `stdio`. Спецификаторы формата. Эскейп-последовательности.

2. Понятие типа данных. Целые типы данных в Си. Вещественные типы данных в Си. Преобразование типа.

3. Понятие алгоритма. Базисные алгоритмы: следование, выбор, повторение. Принцип структурного программирования. Логические операции. Логические выражения. Истина и ложь в Си, условный оператор, переключатель. Циклы в Си Примеры.

4. Массивы Линейный поиск в массиве. Построение массива без повторов.

5. Максимальный и минимальный элементы массива. Сортировка.

6. Двумерные массивы Преобразование и построение матриц

7. Символьный тип данных. Программа подсчета строк, слов и литер. Массивы и строки в Си. Программа подсчета цифр, пробельных и прочих литер.

8. Функции в Си. Программа, определяющая принадлежность точки заданному треугольнику.

9. Рекурсия.

10. Функции `strlen`, `copy`, `getline`. Функции перевода числа в строку и строки в число.

11. Операции увеличения и уменьшения. Операции присваивания. Приоритет операций. Побитовые операции. Функция `getbits(x,p,n)`, формирующая поле из n бит, вырезанное из x , начиная с p позиции, прижимая поле к правому краю. Выражения и операторы. Условное выражение.

12. Классы памяти. Автоматические, внешние, статические и регистровые переменные. Многофайловая организация программы. Модуль. Реализация модуля `Stack`.

13. Указатели и адреса. Операции взятия адреса и раскрытия ссылки, их приоритет. Указатели и аргументы функций.

14. Указатели и массивы. Операции над указателями, ссылающимися на элементы одного массива.

15. Указатели и строки. Функции `strcpy`, `strcmp`.

16. Указатели и динамическая память. Простейший распределитель памяти на основе функции `alloc`, `afree`. Динамический массив. Операторы `new`, `delete`. Функция, создающая копию строки в динамической памяти.

17. Массивы указателей и задача сортировки текстовых строк входного потока. Реализация функций ввода-вывода и функции `main`.

18. Массивы указателей и задача сортировки текстовых строк входного потока. Реализация функции быстрой сортировки

19. Реализация модуля функций для работы со строками `item.h`

20. Реализация модуля таблицы строк `Tabl.h`

21. Двумерные массивы и указатели. Функция, суммирующая матрицы произвольной размерности (два варианта с использованием различных подходов).

22. Файловый ввод - вывод.

23. Структуры. Расположение в памяти. Инициализация. Операции. Приоритет операций. Вложенные структуры. Структуры и функции.

24. Реализация таблицы строк в виде структуры.

7. Учебно-методическое и информационное обеспечение дисциплины «Информатика и программирование»

а) основная литература:

1. Гаврилов, М. В. Информатика и информационные технологии : учебник для вузов / М. В. Гаврилов, В. А. Климов. — 4-е изд., перераб. и доп. — Москва : Издательство

Юрайт, 2020. — 383 с. — (Высшее образование). — ISBN 978-5-534-00814-2. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/449779>

2. Дорогов, В. Г. Основы программирования на языке С : учебное пособие / В. Г. Дорогов, Е. Г. Дорогова ; под ред. проф. Л. Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2019. — 224 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-8199-0882-2. - Текст : электронный. - URL(доступно в ЭБС «Знаниум», режим доступа: : <https://znanium.com/catalog/product/1016471>

3. Информатика для гуманитариев : учебник и практикум для вузов / Г. Е. Кедрова [и др.] ; под редакцией Г. Е. Кедровой. — Москва : Издательство Юрайт, 2020. — 439 с. — (Высшее образование). — ISBN 978-5-534-01031-2. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/450494>)

б) дополнительная литература

1. Гуриков, С. Р. Программирование в среде Lazarus для школьников и студентов : учебное пособие / С. Р. Гуриков. — Москва : ФОРУМ : ИНФРА-М, 2019. — 336 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-00091-137-2. - Текст : электронный. - URL: (доступно в ЭБС «Знаниум», режим доступа: <https://znanium.com/catalog/product/1010784>)

2. Зыков, С. В. Программирование: учебник и практикум для академического бакалавриата / С. В. Зыков. — М.: Издательство Юрайт, 2017. — 320 с. — (Серия: Бакалавр. Академический курс). — ISBN 978-5-534-02444-9. (доступно в ЭБС «Юрайт», режим доступа: <https://urait.ru/bcode/401291>)

3. Немцова Т.И. Практикум по информатике: учебное пособие. Ч. 1. / Т.И. Немцова, Ю.В. Назарова; Под ред. Л.Г. Гагариной. - М.: ИД ФОРУМ: ИНФРА-М, 2017. - 288 с. (доступно в ЭБС «Знаниум», режим доступа: <http://znanium.com/bookread2.php?book=899497>).

4. Эйдлина, Г. М. Delphi: программирование в примерах и задачах. Практикум : учеб.пособие / Г.М. Эйдлина, К.А. Милорадов. — Москва : РИОР : ИНФРА-М, 2017. — 116 с. — (Высшее образование: Бакалавриат). — <https://doi.org/10.12737/13667>. - ISBN 978-5-369-01084-6. - Текст : электронный. - URL: (доступно в ЭБС «Знаниум», режим доступа: , <https://znanium.com/catalog/product/858775>)

5. Черпаков, И. В. Основы программирования : учебник и практикум для прикладного бакалавриата / И. В. Черпаков. — Москва : Издательство Юрайт, 2018. — 219 с. — (Высшее образование). — ISBN 978-5-9916-9983-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: (доступно в ЭБС «Юрайт», режим доступа: <https://urait.ru/bcode/413593>)

в) программное обеспечение и Интернет-ресурсы

1. Операционная система Microsoft Windows
2. Пакет прикладных программ Microsoft Office
3. Правовая система «Консультант плюс»
4. Правовая система «Гарант».
5. Интернет браузеры (Mozilla Firefox, Google Chrome)
6. Свободно распространяемая среда разработки Pascal ABC.
7. Среда разработки программного обеспечения Lazarus.
8. Свободная интегрированная среда разработки приложений Dev-C++.

7. Материально-техническое обеспечение дисциплины

Помещения представляют собой учебные аудитории для проведения учебных занятий, предусмотренных программой, оснащенные оборудованием и техническими средствами обучения: компьютерная техника с подключением к сети «Интернет», экран, проектор для вывода мультимедиа материалов на экран, динамики для воспроизведения звука, доска.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечены доступом в электронную информационно-образовательную среду.

Специальные условия организации обучения по дисциплине для инвалидов и лиц с ограниченными возможностями здоровья

Организация обучения по дисциплине инвалидов и лиц с ограниченными возможностями здоровья осуществляется с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья при наличии таких обучающихся путем создания специальных условий для получения образования.

Профессорско-преподавательский состав знакомится с психолого-физиологическими особенностями обучающихся инвалидов и лиц с ограниченными возможностями здоровья, индивидуальными программами реабилитации инвалидов (при наличии).

В соответствии с Методическими рекомендациями по организации образовательного процесса для обучения инвалидов и лиц с ограниченными возможностями здоровья в образовательных организациях высшего образования, в том числе оснащенности образовательного процесса, утв. Минобрнауки РФ 08.04.2014 АК-44/05вн при изучении дисциплины предполагается использование социально-активных и рефлексивных методов обучения, технологий социокультурной реабилитации с целью оказания помощи в установлении полноценных межличностных отношений с другими студентами, создании комфортного психологического климата в студенческой группе.

При освоении дисциплины используются различные сочетания видов учебной работы с методами и формами активизации познавательной деятельности обучающихся для достижения запланированных результатов обучения и формирования компетенций. Форма проведения промежуточной аттестации для обучающихся-инвалидов и лиц с ограниченными возможностями здоровья устанавливается с учетом индивидуальных психофизиологических особенностей. По личной просьбе обучающегося с ограниченными возможностями здоровья, изложенной в форме письменного заявления, по дисциплине предусматриваются:

- замена устного ответа на письменный ответ при сдаче зачёта, экзамена;
- увеличение продолжительности времени на подготовку к ответу на зачёте, экзамене;
- при подведении результатов промежуточной аттестации студентов выставляется максимальное количество баллов за посещаемость аудиторных занятий.

Программа составлена в соответствии с требованиями ОС ННГУ по направлению подготовки 09.03.03 «Прикладная информатика», профиль «Прикладная информатика в экономике и управлении».

Программа одобрена на заседании учебно-методической комиссии Павловского филиала ННГУ протокол № 3 от 22.03.2021.