

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ОБНИНСКИЙ ИНСТИТУТ АТОМНОЙ ЭНЕРГЕТИКИ - филиал
федерального государственного автономного образовательного учреждения
высшего образования
Национальный исследовательский ядерный университет МИФИ
(ИАТЭ НИЯУ МИФИ)
ТЕХНИКУМ ИАТЭ НИЯУ МИФИ**

Утверждено
Ученый совет ИАТЭ НИЯУ МИФИ
Протокол № 25.1 от 27.01.2025 г.

**КОМПЛЕКТ КОНТРОЛЬНО – ИЗМЕРИТЕЛЬНЫХ
МАТЕРИАЛОВ
текущего и промежуточного контроля успеваемости
ПО УЧЕБНОЙ ДИСЦИПЛИНЕ
ОПЦ.04 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ**

Направление подготовки
(специальность)

09.02.07 Информационные системы и
программирование

Квалификация (степень) выпускника

специалист по информационным
системам

Форма обучения

очная

Комплект контрольно-измерительных материалов по учебной дисциплине «Основы алгоритмизации и программирования» разработан на основе Федерального государственного образовательного стандарта (далее ФГОС СПО) по специальности среднего профессионального образования 09.02.07 Информационные системы и программирование.

Организация-разработчик:

Техникум ИАТЭ НИЯУ МИФИ

Одобрено на заседании предметной цикловой комиссии информационных технологий

«21» января 2025 года, № протокола 5

Председатель предметной цикловой комиссии _____ (А.Ю. Мамонов)

I. Паспорт комплекта оценочных средств

1. Область применения комплекта оценочных средств

Комплект оценочных средств предназначен для оценки результатов освоения дисциплины ОПЦ.04 Основы алгоритмизации и программирования и является частью программы подготовки специалистов среднего звена по специальности 09.02.07 «Информационные системы и программирование».

Требования к результатам освоения учебной дисциплины:

Формирование и развитие у обучающихся умений, знаний и компетенций:

умения:

разрабатывать алгоритмы для конкретных задач.

использовать программы для графического отображения алгоритмов.

определять сложность работы алгоритмов.

работать в среде программирования.

реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.

оформлять код программы в соответствии со стандартом кодирования.

выполнять проверку, отладку кода программы.

знания:

понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции.

эволюцию языков программирования, их классификацию, понятие системы программирования, основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.

подпрограммы, составление библиотек подпрограмм

объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения

общие компетенции:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;

ОК 04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках;

профессиональные компетенции:

ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения;

ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.

2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ

2.1. В результате аттестации по учебной дисциплине осуществляется комплексная проверка следующих умений и знаний:

Результаты обучения (освоенные умения, усвоенные знания)	Показатели оценки результата	Формы и методы контроля и оценки резуль-
---	------------------------------	--

		татов обучения
уметь: разрабатывать алгоритмы для конкретных задач. использовать программы для графического отображения алгоритмов. определять сложность работы алгоритмов. работать в среде программирования. реализовывать построенные алгоритмы в виде программ на конкретном языке программирования. оформлять код программы в соответствии со стандартом кодирования. выполнять проверку, отладку кода программы. знать: понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции. эволюцию языков программирования, их классификацию, понятие системы программирования. основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти. подпрограммы, составление библиотек подпрограмм объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения	знание типов данных и базовых конструкций изучаемых языков программирования; знание интегрированных сред изучаемых языков программирования; знание основ объектно-ориентированного программирования; умение работать с разными видами алгоритмов (линейными, циклическими и разветвляющимися); уметь составлять алгоритмы и программы с использованием языков программирования, производить отладку программ, тестировать приложения и выполнять практические задания; составление простых блок-схем алгоритмов; составление программы на алгоритмическом языке высокого уровня; работа в интегрированной среде изучаемых языков программирования; знание принципов построения алгоритмов	результаты выполнения практических, лабораторных заданий и самостоятельной работы, экзамен
ОК 01. Выбирать способы решения задач профессиональной деятельности примени-	распознавать задачу и/или проблему в профессиональном и/или социальном контексте;	результаты выполнения практических, лабо-

тельно к различным контекстам; ОК 02. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; ОК 04. Эффективно взаимодействовать и работать в коллективе и команде; ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста; ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках; ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения; ПК 2.5. Производить инспектирование	анализировать задачу и/или проблему и выделять её составные части; определять этапы решения задачи; выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы; составить план действия; определить необходимые ресурсы; владеть актуальными методами работы в профессиональной и смежных сферах; реализовать составленный план; оценивать результат и последствия своих действий (самостоятельно или с помощью наставника) определять задачи для поиска информации; определять необходимые источники информации; планировать процесс поиска; структурировать получаемую информацию; выделять наиболее значимое в перечне информации; оценивать практическую значимость результатов поиска; оформлять организовывать работу коллектива и команды; взаимодействовать с коллегами, руководством, клиентами в ходе профессиональной деятельности грамотно излагать свои мысли и оформлять документы по профессиональной тематике на государственном языке, проявлять толерантность в рабочем коллективе применять средства информационных технологий для решения профессиональных задач; использовать современное программное обеспечение понимать общий смысл четко произнесенных высказываний на известные темы (профессиональные и бытовые), понимать тексты на базовые профессиональные темы; участвовать в диалогах на знакомые общие и профессиональные темы; строить простые высказывания о себе и о своей профессиональной деятельности; кратко обосновывать и объяснить свои действия (текущие и планируемые); писать простые связные сообщения на знакомые или интересующие профессиональные темы Использовать выбранную систему контроля версий. Анализировать проектную и техническую документацию.	раторных заданий и самостоятельной работы
---	---	--

	Выполнять тестирование интеграции. Организовывать постобработку данных. Использовать приемы работы в системах контроля версий. Оценивать размер минимального набора тестов. Разрабатывать тестовые пакеты и тестовые сценарии. Выполнять ручное и автоматизированное тестирование программного модуля. Выявлять ошибки в системных компонентах Использовать выбранную систему контроля версий. Использовать методы для получения кода с заданной функциональностью и степенью качества. Анализировать проектную и техническую документацию. Организовывать постобработку данных. Приемы работы в системах контроля версий. Выявлять ошибки в системных компонентах	
--	---	--

2.2 Формы текущего контроля и промежуточной аттестации по учебной дисциплине

Элемент учебной дисциплины	Текущий контроль	
	Формы и методы контроля и оценки результатов обучения	Компетенции, формированию которых способствует элемент программы
Раздел 1. Введение в программирование	Опрос, результаты выполнения практических заданий и самостоятельной работы, решения тестовых заданий	ОК 01, ОК 02, ОК 04, ОК 05, ОК 09, ПК 2.4, ПК 2.5
Раздел 2. Операторы	Решения тестовых заданий, результаты выполнения практических заданий и самостоятельной работы, опрос	ОК 01, ОК 02, ОК 04, ОК 05, ОК 09, ПК 2.4, ПК 2.5
Раздел 3. Синтаксис языка	Решения тестовых заданий, результаты выполнения практических заданий и самостоятельной работы, опрос	ОК 01, ОК 02, ОК 04, ОК 05, ОК 09, ПК 2.4, ПК 2.5
Раздел 4. Указатели	Решения тестовых заданий, результаты выполнения практических заданий и самостоятельной работы, опрос	ОК 01, ОК 02, ОК 04, ОК 05, ОК 09, ПК 2.4, ПК 2.5
Раздел 5. Программирование высокого уровня	Решения тестовых заданий, результаты выполнения практических заданий и самостоятельной работы, опрос	ОК 01, ОК 02, ОК 04, ОК 05, ОК 09, ПК 2.4, ПК 2.5

2.3 ОЦЕНКА ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Контроль и оценка результатов освоения дисциплины «Основы алгоритмизации и программирования» осуществляется преподавателем в процессе проведения:

- практических (семинарских) занятий,
- лабораторных занятий,
- тестирования,
- опроса,
- дискуссий, диспутов,
- выполнения студентами самостоятельной работы, индивидуальных заданий и т.д.

Тестирование направлено на проверку владения терминологическим аппаратом и конкретными знаниями в области по дисциплине. Тестирование занимает часть учебного занятия (10-30 минут), правильность решения разбирается на том же или следующем занятии; частота тестирования определяется преподавателем.

На семинарских занятиях осуществляется защита представленных рефератов (докладов, проектов), творческих работ или выступлений студентов.

Практические занятия проводятся в часы, выделенные учебным планом для отработки практических навыков освоения компетенциями, и предполагают аттестацию всех обучающихся за каждое занятие.

Собеседование посредством использования устного опроса на занятии позволяет выяснить объем знаний студента по определенной теме, разделу, проблеме.

Круглый стол, дискуссия, полемика, диспут, дебаты - оценочные средства, позволяющие включить обучающихся в процесс обсуждения спорного вопроса, проблемы и оценить их умение аргументировать собственную точку зрения

Доклад, сообщение является продуктом самостоятельной работы студента, представляющий собой публичное выступление по представлению полученных результатов решения определенной учебно-практической, учебно-исследовательской или научной темы.

Продуктом самостоятельной работы студента, является и реферат, представляющий собой краткое изложение в письменном виде полученных результатов теоретического анализа определенной научной (учебно-исследовательской) темы, где автор раскрывает суть исследуемой проблемы, приводит различные точки зрения, а также собственные взгляды на нее.

Подготовка студентом эссе позволяет оценить умение обучающегося письменно излагать суть поставленной проблемы, самостоятельно проводить анализ этой проблемы с использованием концепций и аналитического инструментария соответствующей дисциплины, делать выводы, обобщающие авторскую позицию по поставленной проблеме.

2. Комплект оценочных средств

2.1. ТЕКУЩИЙ КОНТРОЛЬ

(проверяемые компетенции ОК 01, ОК 02, ОК 04, ОК 05, ОК 09, ПК 2.4, ПК 2.5)

Контроль практических занятий:

Темы практических занятий	Контроль практических занятий
Раздел 1. Введение в программирование Тема 1.1. Языки программирования Развитие языков программирования. Обзор языков про-	Уровень подготовки к практическому занятию, лабораторному занятию, зна-

граммирования. Области применения языков программирования. Стандарты языков программирования. Средства проектирования. Компиляторы и интерпретаторы. Жизненный цикл программы. Программа. Программный продукт и его характеристики. Основные этапы решения задач на компьютере. Тема 1.2. Типы данных Типы данных. Простые типы данных. Производные типы данных. Структурированные типы данных. Раздел 2. Операторы Тема 2.1. Операторы языка программирования Операции и выражения. Правила формирования и вычисления выражений. Структура программы. Ввод и вывод данных. Оператор присваивания. Составной оператор. Условный оператор. Оператор выбора. Цикл с постусловием. Цикл с предусловием. Цикл с параметром. Вложенные циклы. Массивы. Двумерные массивы. Строки. Стандартные процедуры и функции для работы со строками. Структурированный тип данных – множество. Операции над множествами. Комбинированный тип данных – запись. Файлы последовательного доступа. Файлы прямого доступа Раздел 3. Синтаксис языка Тема 3.1. Процедуры и функции Общие сведения о подпрограммах. Определение и вызов подпрограмм. Область видимости и время жизни переменной. Механизм передачи параметров. Организация функций. Рекурсия. Программирование рекурсивных алгоритмов. Тема 3.2. Структуризация в программировании Основы структурного программирования. Методы структурного программирования. Программирование «сверху вниз» Программирование «снизу вверх» Тема 3.3. Модульное программирование Модульное программирование. Понятие модуля. Структура модуля. Компиляция и компоновка программы. Стандартные модули. Раздел 4. Указатели Тема 4.1 Указатели. Указатели. Описание указателей. Основные понятия и применение динамически распределяемой памяти. Создание и удаление динамических переменных. Структуры данных на основе указателей. Задача о стеке. Раздел 5. Программирование высокого уровня	ние лекционного материала, источников. Умение пользоваться основными теоремами во время ответа. Умение аргументировано отстаивать свою точку зрения при ответе. Умение структурировать и представлять математический материал в таблицах и схемах.
---	--

Тема 5.1 Основные принципы объектно-ориентированного программирования (ООП) История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс. Основные принципы ООП: инкапсуляция, наследование, полиморфизм. Классы объектов. Компоненты и их свойства. Событийно-управляемая модель программирования. Компонентно-ориентированный подход. Тема 5.2 Интегрированная среда разработчика. Требования к аппаратным и программным средствам интегрированной среды разработчика. Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты. Форма и размещение на ней управляющих элементов. Панель компонентов и их свойства. Окно кода проекта. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта. Панель компонентов и их свойства. Окно кода проекта. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта. Настройка среды и параметров проекта. Тема 5.3. Визуальное событийно-управляемое программирование. Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение. Дополнительные элементы управления. Свойства компонентов. Виды свойств. Синтаксис определения свойств. Назначения свойств и их влияние на результат. Управление объектом через свойства. События компонентов (элементов управления), их сущность и назначение. Создание процедур на основе событий. Тема 5.4 Разработка оконного приложения Разработка функционального интерфейса приложения. Создание интерфейса приложения. Разработка функциональной схемы работы приложения. Разработка игрового приложения. Тема 5.5 Этапы разработки приложений Разработка приложения. Проектирование объектно-ориентированного приложения. Создание интерфейса пользователя. Тестирование, отладка приложения. Тема 5.6 Иерархия классов. Классы ООП: виды, назначение, свойства, методы, события. Перегрузка методов. Тестирование и отладка приложения.	
--	--

Критерии оценки

Оценивается знание материала для подготовки к практическим занятиям, умение излагать его самостоятельно, умение анализировать, сравнивать, давать оценку, приводить примеры, умение пользоваться источниками.

Оценка: **зачтено/ не зачтено.**

Контроль применения активных и интерактивных методов и форм обучения

№	Название темы	Активные и интерактивные методы и формы обучения	Часы
1	Раздел 1. Введение в программирование	Дискуссия Беседа	0,5
2	Раздел 2. Операторы	Дискуссия Практическая работа Беседа	1
3	Раздел 3. Синтаксис языка	Индивидуальный проект Дискуссия Практическая работа Беседа	1
4	Раздел 4. Указатели	Дискуссия Практическая работа Беседа	1
5	Раздел 5. Программирование высокого уровня	Индивидуальный проект Дискуссия Практическая работа Беседа	1

Индивидуальный проект - особая форма организации образовательной деятельности обучающихся, структура которой приближена к формату научного исследования и содержит доказательство актуальности избранной темы, определение научной проблемы, предмета и объекта исследования, целей и задач, методов, источников, историографии, обобщение результатов, выводы.

Индивидуальный проект выполняется обучающимся самостоятельно под руководством преподавателя согласно выбранной теме в рамках данной дисциплины или в числе других изучаемых учебных предметов.

Темы проектов:

1. Место алгоритма в системе управления.
2. Системы управления с использованием компьютера
3. Использование алгоритмического языка и блок-схем для записи алгоритмов
4. Разработка линейных алгоритмов для учебного исполнителя
5. Программирование для учебного исполнителя с использованием вспомогательных алгоритмов, последовательной детализации (линейные задачи)
6. Циклические алгоритмы
7. Алгоритмы работы с величинами

8. Программирование циклов на Паскале

Контроль применения активных и интерактивных методов и форм обучения

Оценка «отлично» ставится, если студент:

- выполнил работу без ошибок и недочетов;
- допустил не более одного недочета.

Оценка «хорошо» ставится, если студент выполнил работу полностью, но допустил в ней:

- не более одной негрубой ошибки и одного недочета;
- или не более двух недочетов.

Оценка «удовлетворительно» ставится, если студент правильно выполнил не менее половины работы или допустил:

- не более двух грубых ошибок;
- или не более одной грубой и одной негрубой ошибки и одного недочета;
- или не более двух-трех негрубых ошибок;
- или одной негрубой ошибки и трех недочетов;
- или при отсутствии ошибок, но при наличии четырех-пяти недочетов.

Примечание.

- Преподаватель имеет право поставить студенту оценку выше той, которая предусмотрена нормами, если студентом оригинально выполнена работа.
- Оценки с анализом доводятся до сведения студентов, как правило, на последующем занятии, предусматривается работа над недочетами, устранение пробелов.

Оценка умений работать с источниками знаний:

Оценка «отлично» - правильный, полный отбор источников знаний, рациональное их использование в определенной последовательности; соблюдение логики в описании или характеристике объектов; самостоятельное выполнение и формулирование выводов на основе практической деятельности; аккуратное оформление результатов работы.

Оценка «хорошо» - правильный и полный отбор источников знаний, допускаются неточности в использовании источников знаний, в оформлении результатов.

Оценка «удовлетворительно» - правильное, но неполное использование основных источников знаний; допускаются неточности в формулировке выводов; неаккуратное оформление результатов.

ПРИМЕРЫ КОНТРОЛЬНОЙ РАБОТЫ

Контрольная работа по теме 1.2.

1. Составьте программу, которая формирует одномерный массив из 100 элементов, заполненный случайными неповторяющимися числами из отрезка [50,200].

2. Элемент двумерного массива называется седловым, если он является минимальным в столбце и одновременно максимальным в строке. Элемент, который является максимальным в столбце и одновременно минимальным в строке, также называется седловым. Требуется написать программу, которая определяет, имеется ли в массиве 10×10 седловой элемент. Если имеется, то вывести на экран его индексы.

3. Напишите программу, которая сортирует строки матрицы по убыванию сумм элементов строк.

Контрольная работа по теме 2.1

1. Запишите фрагмент программного кода, в котором у пользователя запрашивается величина депозита (тыс. руб.), срок депозита (лет) и годовая процентная ставка (%). Ввод данных осуществлять в виде диалога. После ввода вывести данные в формате

Сумма депозита 200000,00 тыс. руб. сроком на 5 лет; 10,50 % годовых.

Для суммы и процентной ставки использовать форматированный вывод данных.

2. Запишите фрагмент программного кода, в котором запрашивается 16 чисел в интервале от 1 до 1000. Требуется организовать форматированный вывод данных в виде

6.000	11.660	63.765	1.683
3.550	125.356	1000.000	12.005
23.664	777.200	39.800	0.001
124.311	7.000	38.656	31.989

3. У пользователя запрашивается четырехзначное число *abcd*. Необходимо вывести на экран информацию в виде:

*Количество тысяч: a
Количество сотен: b
Количество десятков: c
Количество единиц: d.*

ПРИМЕРЫ ЛАБОРАТОРНОЙ РАБОТЫ

Тема: «Разработка и отладка разветвляющихся программ».

Цель занятия: приобрести практические навыки по разработке разветвляющихся алгоритмов, программ и отладке их в интегрированной среде

Отрабатываемые вопросы:

Разработка схем алгоритмов и разветвляющихся программ.

Отладка разветвляющихся программ.

Организационно-методические указания

Один из фундаментальных операторов – оператор IF.... помогает проводить анализ выполнения заданных условий. Программы, составленные для таких вычислений, называются разветвляющимися программами.

1. Постановка задачи. Разработать схему разветвляющегося алгоритма и программу для вычисления значений функции, выбранной из перечня индивидуальных заданий.

Полная форма условного оператора в виде блока имеет вид:

IF <условие> THEN

ELSE

<операторы>

END IF

Используется при построении программ и алгоритмов также краткая форма:

IF<условие> **THEN**

<операторы>

END IF

Линейная форма записи имеет вид:

IF<условие> **THEN**<операторы> **ELSE**<операторы>

IF<условие> **THEN**<операторы>

Выполнить вычисление следующей функции, используя условный оператор:

$$y = \sin \frac{\pi}{12+x} + \begin{cases} x^2 - 5a, & x > 0 \\ x^2 + 5a, & x \leq 0 \end{cases}$$

2. Разработать схему реализации алгоритма (рис. 1.)

3. Создание текста разветвляющейся программы двумя способами.

1 способ реализации программы.

REM Линейная структура программы

Pi=3.14

INPUT «Введите а»;a

INPUT «Введите х»;x

IF x>0 THEN y=sin(pi/(12+x))+x^2-5*a ELSE y=sin(pi/(12+x))+x^2+5*a

PRINT «у=»;y

END

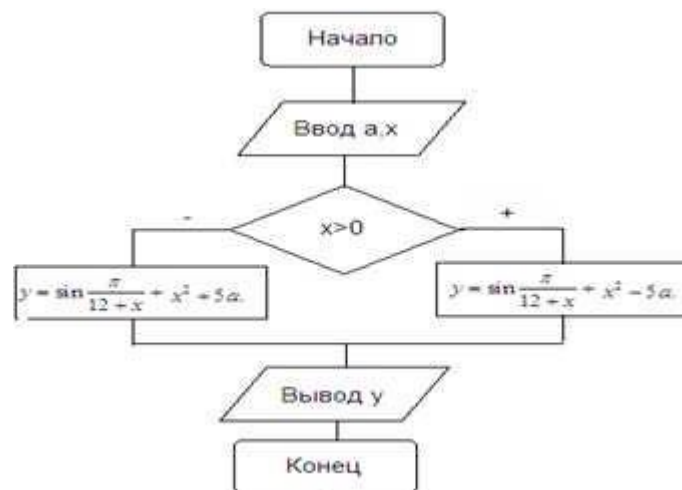


Рис.1.Схема разветвляющегося алгоритма

2 способ реализации программы.

REM Блочная структура программы

Pi=3.14

INPUT «Введите а»;a

INPUT «Введите х»;x

IF x>0 THEN

```

y=sin(Pi/(12+x))+x^2-5*a
ELSE
y=sin(Pi/(12+x))+x^2+5*a
END IF
PRINT «y=»;y
END

```

4. Оценка результатов вычислений для двух способов построения программы и значений условия «х».

2.2. ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ

(проверяемые компетенции ОК 01, ОК 02, ОК 04, ОК 05, ОК 09, ПК 2.4, ПК 2.5)

Устные задания для проведения экзамена

1. Алгоритм. Базовые алгоритмические конструкции.
2. Исполнители алгоритма. СКИ. Пошаговое выполнение алгоритма.
3. История и классификация языков программирования.
4. Алфавит. Идентификаторы, переменные, константы. Общая структура программы. Комментарии.
5. Стандартные типы данных. Арифметические и логические выражения. Стандартные функции.
6. Простые операторы. Операторы присваивания, составной и простой операторы, операторы ввода-вывода.
7. Условный оператор.
8. Оператор цикла с предусловием.
9. Оператор цикла с постусловием.
10. Оператор цикла с параметром.
11. Оператор множественного выбора.
12. Строки. Действия над строками.
13. Множества. Операции над множествами.
14. Одномерные массивы.
15. Двумерные массивы. Свойства квадратных матриц.
16. Записи. Оператор присоединения.
17. Файлы. Функции для работы с файлами.
18. Процедуры и функции.
19. Работа с графикой.
20. Анимация.
21. Основные типы данных. Стандартные функции. Математические функции. Структура программы.
22. Операторы форматного ввода и вывода. Спецификаторы и модификаторы формата.
23. Оператор присваивания. Арифметические выражения. Операторы инкремента и декремента. Сокращенный вид записи арифметических выражений.
24. Условные выражения. Оператор условия.
25. Оператор множественного выбора.
26. Циклы с предусловием.
27. Циклы с постусловием.
28. Циклы с параметром.

- 29. Массивы.
- 30. Строки.
- 31. Функции.
- 32. Структуры.
- 33. Файлы.

Примерный перечень практических работ:

- Составление программ линейной структуры.
- Составление блок-схем программ линейной структуры.
- Составление программ разветвляющейся структуры.
- Составление блок-схем программ разветвляющейся структуры.
- Составление программ циклической структуры
- Составление блок-схем программ циклической структуры
- Типы данных.
- Обработка одномерных массивов.
- Обработка двумерных массивов.
- Работа со строками.
- Работа с данными типа множество.
- Файлы последовательного доступа.
- Типизированные файлы.
- Нетипизированные файлы.
- События компонентов (элементов управления), их сущность и назначение.
- Классы ООП: виды, назначение, свойства, методы, события.
- Объявления класса.
- Перегрузка методов.

Примерный перечень лабораторных работ:

- Знакомство со средой программирования.
- Организация процедур.
- Организация функций.
- Применение рекурсивных функций.
- Создание библиотеки подпрограмм.
- Использование указателей для организации связанных списков.
- Изучение интегрированной среды разработчика.
- Создание проекта с использованием компонентов для работы с текстом.
- Создание проекта с использованием компонентов ввода и отображения чисел, дат и времени.
- События компонентов (элементов управления), их сущность и назначение.
- Создание процедур на основе событий.
- Создание проекта с использованием кнопочных компонентов.
- Создание проекта с использованием компонентов стандартных диалогов и системы меню.
- Разработка функциональной схемы работы приложения.
- Разработка оконного приложения с несколькими формами.
- Разработка игрового приложения.
- Создание процедур обработки событий. Компиляция и запуск приложения.
- Разработка интерфейса приложения.
- Тестирование, отладка приложения.
- Программирование приложений.

Темы рефератов:

1. Развитие языков программирования.
2. Области применения языков программирования.
3. Стандарты языков программирования.
4. Основные этапы решения задач на компьютере.
5. Структурированные типы данных.
6. Структура программы.
7. Структурированный тип данных – множество.
8. Область видимости и время жизни переменной.
9. Механизм передачи параметров.
10. Методы структурного программирования.
11. Модульное программирование.
12. Компиляция и компоновка программы.
13. История развития ООП.
14. Событийно-управляемая модель программирования.
15. Компонентно-ориентированный подход.
16. Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение.
17. События компонентов (элементов управления), их сущность и назначение.
18. Разработка функционального интерфейса приложения.
19. Проектирование объектно-ориентированного приложения.
20. Классы ООП: виды, назначение, свойства, методы, события.
21. Тестирование и отладка приложения.

Практические задания для проведения экзамена

1. Среда программирования — это:

- а) набор команд, воспринимаемых процессором и используемых для написания новых программ;
- б) совокупность инструментов для создания, тестирования и отладки новых программ;
- в) множество языков программирования, служащих для создания новых программ;
- г) язык программирования высокого уровня и редактор программно кода;
- д) совокупность компилятора и процессора.

2. Правильными идентификаторами являются:

- а) Summa , average , 2average ;
- б) Prod_1 , _Prod3 , Prod:6 ;
- в) Line , Sort3Line , LineSort4 ;
- г) Rare , rare , raRE ;
- д) *Temp , Temp* , temp .

3. Для программы, написанной на языке Pascal, верны следующие утверждения:

- а) заголовок программы является обязательным;
- б) в разделе описаний можно использовать операторы ввода и вывода;
- в) программа содержит как минимум одни операторные скобки;
- г) точку после завершающего слова begin можно не ставить;
- д) в программе нельзя использовать многострочные комментарии.

4. Логически законченным действиям в языках программирования высокого уровня соответствует понятие:

- а) оператора;

- б) операнда;
- в) операции;
- г) выражения;
- д) строки программного кода.

5. Операторы read и write отличаются от операторов readln и writeln тем, что:

- а) операторы read и write не могут иметь аргументов;
- б) после выполнения операторов read и write выполнение программы всегда приостанавливается;
- в) операторы read и write позволяют в качестве аргументов иметь только одну переменную;
- г) после выполнения операторов read и write не происходит перехода на новую строку в окне вывода;
- д) операторы read и write не связаны с выполнением операций ввода-вывода.

6. Укажите неверное утверждение для одномерного массива:

- а) все элементы массива должны иметь одинаковый тип;
- б) индекс элемента может быть отрицательным;
- в) наименьший индекс массива всегда равен 1;
- г) обращение к элементу массива происходит по его индексу;
- д) значениями элементов могут быть только числовые значения.

7. В результате выполнения фрагмента программного кода

```
p:=1;
s:=0;
For i:=1 to 10 do
  if a[i]>=0 then
    begin
      p:=p*a[i];
      Inc(s);
    end;
```

характеризовать переменные p и s будут величины:

- а) p — произведение неотрицательных элементов массива, s — сумма неотрицательных элементов;
- б) p — произведение всех элементов массива, s — сумма элементов;
- в) p — произведение индексов неотрицательных элементов массива, s — количество неотрицательных элементов;
- г) p — произведение индексов неотрицательных элементов массива, s — сумма неотрицательных элементов;
- д) p — произведение неотрицательных элементов массива, s — количество неотрицательных элементов.

8. Булевская переменная flag во фрагменте программного кода

```
flag:=true$
for i:=1 to n do
  for j:= 1 to n do
    if c[i;j]<>c[j;i] then flag:= false;
```

характеризует:

- а) является ли матрица нулевой;
- б) является ли матрица единичной;
- в) совпадают ли первая и последняя строки (столбцы) матрицы;

- г) является ли матрица, записанная в массиве, симметричной относительно главной диагонали;
- д) является ли матрица, записанная в массиве, симметричной относительно вторичной диагонали.

9. Результатом выполнения операции объединения множеств, если они объявлены следующим образом $x=[1,2,3]$; $y=[1..5]$ будет:

- а) $[1,1,2,2,3,3,4,5]$;
- б) $[1,2,3,4,5]$;
- в) $[1,2,3,1,2,3,4,5]$;
- г) $[1,2,3,1..5]$;
- д) $[1,2,3,5]$.

10. Оператор with...do служит:

- а) для обработки записей, полями которых являются массивы;
- б) для обработки всех полей одной записи;
- в) для обработки множества записей;
- г) для формирования значений полей записи;
- д) для перебора всех полей записи.

11. QBASIC — это

- а) алгоритмический язык, использующий команды MS-DOS
- б) алгоритмический язык программирования, работающий в режиме интерпретации
- в) алгоритмический язык, работающий только в среде Windows.

12. В QBASIC существуют следующие типы данных:

- а) числовые
- б) текстовые
- в) указатели
- г) типы данных
- д) записи.

13. Числовые данные могут быть представлены как:

- а) целые
- б) с фиксированной запятой
- в) в виде строк
- г) с плавающей запятой

14. Арифметические выражения состоят из:

- а) чисел
- б) констант
- в) команд MS-DOS
- г) машинных команд
- д) переменных
- е) функций
- ж) круглых скобок
- з) квадратных скобок.

15. Свойствами алгоритма являются:

- а) информативность
- б) массовость

- в) оперативность
- г) определенность
- д) дискретность
- е) цикличность
- ж) результативность.

16. Имя переменной — это:

- а) любая последовательность любых символов
- б) последовательность латинских букв, цифр, специальных знаков (кроме пробел)
- в) которая всегда должна начинаться с латинской буквы
- г) последовательность русских, латинских букв, начинающихся с латинской буквы и из специальных знаков, допускающая знак подчеркивания.

17. Для обозначения строковых переменных:

- а) рядом с именем слева ставится знак \$
- б) рядом с именем справа ставится знак \$
- в) имя переменной записывается в кавычках.

18. Для обозначения целочисленных переменных:

- а) рядом с именем слева ставится знак %
- б) рядом с именем слева ставится знак #
- в) рядом с именем справа ставится знак %.

19. Для обозначения действительных переменных с двойной точностью:

- а) рядом с именем слева ставится знак #
- б) рядом с именем справа ставится знак #
- в) рядом с именем справа ставятся знаки ##.

20. Верно ли утверждение? В написании имен допускаются как строчные (маленькие)

- а) так и заглавные (большие)
- б) буквы и QBASIC не делает между ними различия:
- в) верно
- г) не верно.

Критерии оценки

Оценка	Условия, при которых выставляется оценка
отлично	выполнено 20-18 тестовых заданий
хорошо	выполнено 17-14 тестовых заданий
удовлетворительно	выполнено 13-10 тестовых заданий
неудовлетворительно	выполнено менее 10 тестовых заданий

Критерии оценки промежуточной аттестации

Результаты освоения	Критерии оценки результата	Отметка о выполнении
---------------------	----------------------------	----------------------

умения: разрабатывать алгоритмы для конкретных задач. использовать программы для графического отображения алгоритмов. определять сложность работы алгоритмов. работать в среде программирования. реализовывать построенные алгоритмы в виде программ на конкретном языке программирования. оформлять код программы в соответствии со стандартом кодирования. выполнять проверку, отладку кода программы. знания: понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции. эволюцию языков программирования, их классификацию, понятие системы программирования. основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти. подпрограммы, составление библиотек подпрограмм объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; ОК 02. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; ОК 04. Эффективно взаимодействовать и работать в коллективе и команде; ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации	<i>Отлично</i>» – теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко. <i>«Хорошо»</i> – теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками. <i>«Удовлетворительно»</i> - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки. <i>«Неудовлетворительно»</i> – теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.	Выполнено
--	--	------------------

ственном языке Российской Федерации с учетом особенностей социального и культурного контекста; ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках; ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения; ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.		
---	--	--