

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Тамбовский государственный университет имени Г.Р. Державина»
Институт экономики, информационных технологий и креативных индустрий
Факультет информационных технологий и экономики
Кафедра математического моделирования и информационных технологий



УТВЕРЖДАЮ:
Директор института
Кожевникова Т.М.
«03» июня 2026 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

ОП. 6 «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»

образовательной программы среднего профессионального образования – программа
подготовки специалистов среднего звена по специальности

09.02.11 «Разработка и управление программным обеспечением»

Направленность: Веб-разработка

Квалификация
Программист

Год набора 2026

Тамбов 2026

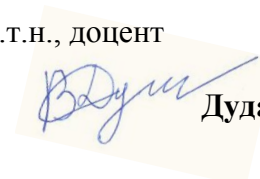
Разработчик(и) программы:

 **Скворцов А.А.**

к.п.н., доцент кафедры математического моделирования и информационных технологий ФГБОУ ВО "Тамбовский государственный университет им. Г.Р. Державина"

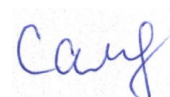
Эксперт(ы):

Системный администратор ООО «Европа-Европа 33», к.т.н., доцент

 **Дудаков В.П.**

Рабочая программа разработана на основе ФГОС СПО и утверждена на заседании кафедры математического моделирования и информационных технологий «02» февраля 2026 года протокол № 6

Зав. кафедрой математического моделирования
и информационных технологий

 **Самохвалов А.В.**

СОДЕРЖАНИЕ

1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ	4
2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ	6
3. УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДИСЦИПЛИНЫ	9
4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ	10

1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ ОП. 6 «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»

1.1. Место дисциплины в структуре основной образовательной программы:

Учебная дисциплина ОП.6 Основы алгоритмизации и программирования является обязательной частью общепрофессионального цикла основной образовательной программы в соответствии с ФГОС по специальности 09.02.11 Разработка и управление программным обеспечением.

В результате освоения учебной дисциплины обучающийся осваивает следующие компетенции.

Перечень общих компетенций

Код компетенции	Содержание 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 03	Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях
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 06	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения
ОК 07	Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях
ОК 08	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности
ОК 09	Пользоваться профессиональной документацией на государственном и иностранном языках

Перечень профессиональных компетенций

Код компетенции	Содержание компетенции
ПК 2.2	Разрабатывать модули программного обеспечения.
ПК 2.4	Выполнять тестирование и отладку программного обеспечения.

1.2. Цель и планируемые результаты освоения дисциплины:

В рамках программы учебной дисциплины обучающимися осваиваются умения и знания

<i>Код ПК, ОК</i>	<i>Умения</i>	<i>Знания</i>
ОК 01 ОК 02 ОК 03 ОК 04 ОК 05 ОК 06 ОК 07 ОК 08 ОК 09 ПК 2.2 ПК 2.4	распознавать задачу и/или проблему в профессиональном и/или социальном контексте, анализировать и выделять её составные части определять задачи для поиска информации, планировать процесс поиска, выбирать необходимые источники информации применять современную научную профессиональную терминологию взаимодействовать с коллегами, руководством, клиентами в ходе профессиональной деятельности грамотно излагать свои мысли и оформлять документы по профессиональной тематике на государственном языке демонстрировать осознанное поведение соблюдать нормы экологической безопасности пользоваться средствами профилактики перенапряжения, характерными для данной специальности понимать общий смысл четко произнесенных высказываний на известные темы (профессиональные и бытовые), понимать тексты на базовые профессиональные темы разрабатывать модули программного обеспечения с использованием различных языков программирования и технологий применять паттерны проектирования и структуры данных для создания эффективных и масштабируемых модулей анализировать требования и определять функциональность модуля создавать интерфейсы для взаимодействия с другими модулями и системами обеспечивать безопасность, производительность и масштабируемость при разработке модулей оптимизировать проектируемые модули для повышения их эффективности и качества работать с системой контроля версий	актуальный профессиональный и социальный контекст, в котором приходится работать и жить номенклатура информационных источников, применяемых в профессиональной деятельности современная научная и профессиональная терминология психологические особенности личности правила оформления документов традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений правила экологической безопасности при ведении профессиональной деятельности средства профилактики перенапряжения правила построения простых и сложных предложений на профессиональные темы язык программирования, основные конструкции, синтаксис паттерны проектирования структуры данных принципы создания интерфейсов для взаимодействия с другими модулями и системами, таких как REST API, SOAP работа с инструментальным программным обеспечением методы оптимизации кода и алгоритмов эффективные алгоритмы и структуры данных для повышения производительности многопоточность в программных модулях методы оптимизации сетевых протоколов для ускорения обмена данными кэширование данных управление памятью

- улучшать производительность модулей, выявляя и устраняя узкие места - проводить анализ и мониторинг производительности приложений - применять инструменты для рефакторинга и оптимизации программного кода - анализировать требования к программному обеспечению и составлять планы тестирования. - создавать тестовые сценарии и тест-кейсы для проверки функциональности и соответствия требованиям. - выполнять тестирование программного обеспечения вручную и автоматизировать процесс тестирования. - анализировать результаты тестирования и документировать найденные ошибки. - разрабатывать стратегии отладки и исправлять ошибки в программном обеспечении. - выполнять модульные тесты с использованием инструментов тестирования, в том числе автоматизированного тестирования - использовать системы контроля дефектов ПО - составлять отчет о выполнении тестирования ПО	- техники повышения производительности программного обеспечения - принципы и методы тестирования программного обеспечения. - основы программирования и архитектуры программного обеспечения. - основы баз данных и SQL-запросов. - инструменты для автоматизации тестирования - основы разработки и отладки программного обеспечения на разных языках программирования - понятие дефекта программного обеспечения - критерии качества ПО - виды и типы тестирования ПО - техники ручного тестирования - техники автоматизированного тестирования - жизненный цикл дефекта ПО - принципы работы в системе контроля дефектов - основные понятия о качестве ПО
---	--

2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объем часов
ОП.06 Основы алгоритмизации и программирования	130
Аудиторная учебная работа (всего)	112
в том числе:	
лекционные занятия, уроки	48
лабораторные занятия	64
Внеаудиторная (самостоятельная) работа	
иные формы самостоятельной работы (при их наличии)	
Консультации	
Промежуточная аттестация в форме экзамена	18

2.2. Тематический план и содержание учебной дисциплины

Наименование разделов и тем	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная работа обучающегося, курсовая работа (проект)	Объем в часах	Коды компетенций, формированию которых способствует элемент программы
1	2	3	4
Раздел 1. Введение в программирование			ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ПК 2.2, ПК 2.4
Тема 1.1. Основы алгоритмизации, языки и системы программирования. Наименование Основные элементы языка. Типы данных. Основы структурного программирования.	Содержание	12	
	1. Алгоритм и его свойства. Виды алгоритмов. Способы записи алгоритмов. Сложность алгоритмов. 2. Эволюция и классификация языков программирования. Среда программирования. 3. Компиляторы и интерпретаторы. Жизненный цикл программы. 4. Программа. Программный продукт и его характеристики. 5. Основные элементы языка. 6. Типы данных. Простые типы данных. Производные типы данных. Структурированные типы данных. 7. Операторы языка. Ввод/вывод данных. 8. Обработка исключений. Операторы разветвляющихся программ. 9. Циклические программы. Одномерные массивы. Двумерные массивы. 10. Работа с массивами. Сортировка массивов. 11. Строки. Коллекции. 12. Файлы. Доступ к файлам. Виды файлов. Считывание и запись в файл.		
	Практические и лабораторные занятия	16	
	1. Линейные программы 2. Составление программ разветвляющейся структуры 3. Циклические программы 4. Одномерные массивы 5. Двумерные массивы. 6. Обработка массивов 7. Сортировка массивов 8. Символы и строки. Обработка строк. 9. Использование коллекций 10. Работа с файлами 11. Работа с файлами 12. Работа с каталогами и файлами		
Раздел 2. Технологии программирования			ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ПК 2.2, ПК 2.4
Тема 2.1. Модульное программирование.	Содержание	12	
	1. Модульное программирование. 2. Локальные и глобальные переменные. Подпрограммы. 3. Модификаторы. Передача данных в подпрограммы. 4. Рекурсия. Разработка рекурсивных		

	подпрограмм.			
	Практические и лабораторные занятия	16		
	1. Использование подпрограмм.			
	2. Использование подпрограмм.			
	3. Рекурсия			
	4. Создание модулей			
Тема 2.2. Основные принципы объектно- ориентированного программирования	Содержание	12	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ПК 2.2, ПК 2.4	
	1. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс.			
	2. Основные принципы ООП: инкапсуляция, наследование, полиморфизм.			
	Практические и лабораторные занятия	16		
	1. Работа с классами. Создание конструкторов.			
	2. Применение свойств			
	3. Наследование			
	4. Полиморфизм			
Раздел 3. Разработка приложений				
Тема 3.1. Этапы разработки приложений	Содержание	12		
	1. Визуально-событийно управляемое программирование.			
	2. Разработка приложения.			
	3. Проектирование объектно-ориентированного приложения.			
	4. Создание интерфейса пользователя.			
	5. Тестирование, отладка приложения.			
	Оптимизация программы			
	Практические и лабораторные занятия	16		
	1. Создание проекта с использованием компонентов для работы с текстом			
	2. Создание проекта с использованием кнопочных компонентов			
3. Создание проекта с использованием переключателей				
4. Создание проекта с использованием компонентов для отображения таблиц				
5. Создание проекта с использованием компонентов для отображения дат и времени				
6. Разработка интерфейса приложения				
7. Разработка интерфейса приложения				
8. Тестирование приложения				
Промежуточная аттестация в форме экзамена		18		
Всего:		130		

3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ ДИСЦИПЛИНЫ

3.1. Для реализации программы учебной дисциплины должны быть предусмотрены следующие специальные помещения:

Лаборатория «Алгоритмизации и программирования».

Перечень основного оборудования:

Парты ученические 18
Стул ученический 13
Скамья ученическая 8
Стол для преподавателя 1
Стул для преподавателя 1
Проектор Epson EMP-TW620 1
Монитор 15
Компьютер (Системный блок, клавиатура, мышь) с возможностью подключения к сети "Интернет" и обеспечением доступа в ЭИОС организации 15
Компьютерный стол 7
Доска меловая 1
Перечень программного обеспечения:
Операционная система Microsoft Windows 7 Home x64
1С:Предприятие 8
Adobe Photoshop CS3
Adobe Flash CS3 Professional
Adobe Illustrator CS3
Adobe InDesign CS4
Kaspersky Endpoint Security для бизнеса - Стандартный Russian Edition. 1500-2499
Node 1 year Educational Renewal Licence
Microsoft Office Профессиональный плюс 2007
Corel DRAW Graphics Suite X3

3.2. Информационное обеспечение реализации программы

1. Семакин, И. Г. Информатика. Углубленный уровень [Текст] : учебник для 11 класса : в 2 ч. / И.Г. Семакин, Е.К. Хеннер, Л.В. Шестакова .— 4-е изд. — Москва : Издательство ЭБС Юрайт изд., стер. — Москва : БИНОМ. Лаборатория знаний, 2018- .— (ФГОС) .— ISBN 978-5-9963-3972-3. Ч. 1 .— 2018 .— 176 с. : ил. — ISBN 978-5-9963-3970-9.

3.2.1 Печатные издания

2. Клыгина, Е.В. Основы алгоритмизации и программирования для студентов-заочников специальностей "Математика", "Физика": учеб.пособие.— Тамбов : [Издат. дом ТГУ им. Г.Р. Державина], 2013 .— 152 с.
3. Гаврилов, М. В. Информатика и информационные технологии : учебник для среднего профессионального образования / М. В. Гаврилов, В. А. Климов. — 4-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2020. — 383 с. — (Профессиональное ЭБС Юрайт 9963-4064-4. образование). — ISBN 978-5-534-03051-8. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/449286> (дата обращения: 01.09.2020).
4. Вирт Н. Алгоритмы и структуры данных. Новая версия для Оберона. – М.: ДМК Пресс, 2010. – 272 с.

3.2.2 Электронные издания (электронные ресурсы)

5. Электронно-библиотечная система «Университетская библиотека онлайн» – база данных учебной, учебно-методической и научной литературы по основным изучаемым дисциплинам - <http://www.biblioclub.ru>

6. Электронно-библиотечная система «Юрайт»: коллекция «Легендарные книги» и коллекция СПО– электронные версии учебной и учебно-методической литературы по экономическим, юридическим, гуманитарным, инженерно-техническим и естественно-научным направлениям - <http://www.urait.ru>

3.2.3 Дополнительные источники

Зарубежные профессиональные базы данных:

1. WebofScience: политематическая реферативно-библиографическая и наукометрическая база данных. – URL: http://apps.webofknowledge.com/WOS_GeneralSearch_input.do?product=WOS&search_mode=GeneralSearch&SID=Q1qfWXliB25bAcrIBPM&preferencesSaved

2. Scopus: база данных. – URL: <https://www.scopus.com/>

Используемые образовательные платформы:

1. Электронная образовательная среда MOODLE <http://moodle.tsutmb.ru/>

2. Платформа для организации онлайн-конференций Zoom <https://zoom.us/>

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ.

4.1. Критерии и методы оценивания

Результаты обучения	Критерии оценки	Методы оценки
Знания: - Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции. - номенклатура информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации; - создание модулей программного обеспечения на различных языках 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,	Демонстрирует знания: - Владение профессиональной терминологией - Умение работать с информационными источниками - Использование основных алгоритмических конструкций - Разработка модулей программного обеспечения на языке программирования, используя структуры данных - Разработка модулей программного обеспечения, используя принципы объектно-ориентированного программирования - Решение ситуационных задач	Тестирование Выполнение практических заданий

инкапсуляция и полиморфизма, наследования и переопределения - отладки программного обеспечения на уровне программных модулей - тестирования программного обеспечения	- Отладка и тестирование программного обеспечения	
Умения: - распознавать задачу, анализировать и выделять её составные части; определять этапы решения задачи; выявлять и эффективно искать информацию, необходимую для решения задачи - Разрабатывать алгоритмы для конкретных задач - определять задачи для поиска информации; определять необходимые источники информации; применять средства информационных технологий для решения профессиональных задач; - Использовать программы для графического отображения алгоритмов - разрабатывать модули программного обеспечения с использованием различных языков программирования и технологий - выполнять тестирование программного обеспечения	Демонстрирует умения: - Разработка модулей программного обеспечения на языке программирования, используя структуры данных - Разработка модулей программного обеспечения, используя принципы объектно-ориентированного программирования - Решение ситуационных задач - Отладка и тестирование программного обеспечения	Тестирование Выполнение практических заданий

4.2 Типовые оценочные средства текущего контроля

Тестирование

1. Массив – это:

- 1) упорядоченная последовательность элементов одного и того же типа;
- 2) набор последовательно проиндексированных элементов одного и того же типа;
- 3) конечное упорядоченное множество элементов, которые могут быть различных стандартных типов.

2. Указать ошибку в следующих описаниях:

Const n=50;

Type

vector = array[real] of integer;

number=array[1..n] of char;

varch: char; a: vector;

- 1) Неправильно описан тип vector;
- 2) Неправильно описана переменная ch;
- 3) Неправильно описан тип number.

3. Указать описание массива, содержащего разные символы и индексами которого являются двузначные числа:

- 1) Var vector = array [10..99] of 'a' .. 'z';
- 2) Var vector = array [10..99] of char;
- 3) Var vector = array [1..10, 1..99] of char.

4. Дан массив `vector : array ['k'..'z'] of real`. Указать правильный фрагмент программы, в котором всем элементам массива присваивается нулевое значение:

- 1) `For j:='k' to 'z' do vector[j]:=0;`
- 2) `For j:=1 to 16 do vector[j]:=0;`
- 3) `For j:=k to z do vector[j]:=0.`

5. Дан массив `A`. Что означает следующая последовательность символов: `A[1]` ?

- 1) элемент массива `A` с индексом равным 1;
- 2) переменной `A` присвоить значение 1;
- 3) первое значение переменной с именем `A`;
- 4) первая переменная с именем `A`.

6. Дан массив `A`. Что означает следующая последовательность символов: `A[n]` ?

- 1) `n`-ое значение переменной с именем `A`;
- 2) переменной `A` присвоить значение `n`;
- 3) элемент массива `A` с индексом равным `n`;
- 4) `n`-ая переменная с именем `A`.

7. Дан массив `Z`. Что означает следующая последовательность символов: `Z[X]:=Y` ?

- 1) Компоненте с номером `X` массива `Z` присвоить значение, которое хранится в переменной `Y`;
- 2) Компоненте с номером `Z` массива `X` присвоить значение, которое хранится в переменной `Y`;
- 3) номеру `X` массива `Z` присвоить значение, которое хранится в переменной `Y`;
- 4) номером `Z` массива `X` присвоить значение, которое хранится в переменной `Y`.

8. Даны массивы `Z` и `Y`. Что означает следующая последовательность символов: `Z[A]:=Y[A]` ?

- 1) Элементу массива `A` с номером `Z` присвоить значение, которое хранится в элементе массива `A` с номером `Y`;
- 2) Элементу массива `Z` с номером `A` присвоить значение, которое хранится в элементе массива `Y` с номером `A`;
- 3) Номеру `A` элементу массива `Z` присвоить значение, которое хранится в номере `A` элемента массива `Y`;
- 4) Номеру `Z` элемента массива `A` присвоить значение, которое хранится в номере `Y` элемента массива `A`.

9. Даны массивы `A` и `C`. Что означает следующая последовательность символов:

`A[B]:=C[D]` ?

- 1) Элементу массива `A` с номером `B` присвоить значение, которое хранится в элементе массива `C` под номером `D`;
- 2) Элементу массива `B` с номером `A` присвоить значение, которое хранится в элементе массива `D` с номером `C`;
- 3) Номеру `A` элементу массива `B` присвоить значение, которое хранится в номере `C` элемента массива `D`;
- 4) Номеру `B` элемента массива `A` присвоить значение, которое хранится в номере `D` элемента массива `C`.

10. Укажите присвоение `K`-ому элементу массива `P` значение 123.

- 1) `K[P]:=123;`
- 2) `P[K]:=123 ;`
- 3) `P[K]:='123';`
- 4) `K[P]:='123'.`

11. Могут ли быть элементами некоторого массива следующие значения: False, True ?

- 1) Нет, так как данные значения различных типов;
- 2) Да, так как данные значения одного типа данных;
- 3) Нет, так как вещественные числа не могут быть элементами массива;
- 4) Нет, так как целые числа не могут быть элементами массива.

12. Могут ли быть индексами элементов некоторого массива следующие значения: False, True ?

- 1) Да, так как данные значения одного порядкового типа данных;
- 2) Нет, так как строки не могут быть индексами элементов массива;
- 3) Нет, так как данные значения логического типа;
- 4) Нет, так как нецелые числа не могут быть индексами элементов массива.

13. Как можно описать массив вещественных чисел с нумерацией индексов элементов массива от 1 до 10?

- 1) type Vector=Array[1..10] of real; Var A:Vector;
- 2) Var A: Array[1..10] of integer;
- 3) Var A: Array[1..10] of real;
- 4) Var A: Array[real] of 1..10.

14. Дано следующее описание:

Type Day = Array [1..365] of Integer;

VarD :Day;

Указать сколько элементов в массиве, какой тип его компонент, индекс его первого и последнего элемента.

- 1) 365; Integer; 1; 365
- 2) 364; Day; 365; 1
- 3) Day; Integer; 1; 365
- 4) 364; Integer; 1; 365

15. Дано следующее описание: Var B : Array [-2..2] ofInteger.

Указать сколько элементов в массиве, какой тип его компонент, индекс его первого и последнего элемента.

- 1) 4; Array [-2..2] ofInteger; 1; 4
- 2) 5; Integer; -2; 2
- 3) 4; Integer; -2; 2
- 4) 2; Array [-2..2] ofInteger; 1; 2

16. Дано следующее описание: Var C : Array [0..9] ofVector.

Указать сколько элементов в массиве, какой тип его компонент, индекс его первого и последнего элемента

- 1) 10; Vector; 0; 9
- 2) 9; 0..9; 1; 9
- 3) 10; 0..9; 0; 9
- 4) 9; C : Array [0..9] ofVector; 1; 9

17. Указать ошибки в следующем описании массива:

Type Vector = Array [Real] of Integer.

- 1) Тип индексов элементов массива не может быть вещественным;
- 2) Тип элементов массива не может быть вещественным;
- 3) Тип индексов элементов массива не может быть целым;
- 4) Тип элементов массива не может быть целым.

18. Дано следующее описание: Type massiv=Array[0..8] of Integer; Varaa: massiv.

Сколько элементов в массиве?

- 1) 8;
- 2) 9;
- 3) 7;
- 4) 1.

19. Дано следующее описание: `Type massiv=Array[-5..5] of Integer; Var aa: massiv.`
Сколько элементов в массиве?

- 1) 11;
- 2) 10;
- 3) 5;
- 4) 12.

20. Какие способы существуют для заполнения массива?

- 1) ввод значений с клавиатуры, описание типизированной константы-массива;
- 2) ввод значений с клавиатуры, описание типизированной константы-массива, присвоение в программе элементам массива каких-либо значений, например, случайных;
- 3) случайными и неслучайными числами;
- 4) ввод значений с клавиатуры, на экран дисплея, присвоение в программе элементам массива каких-либо значений.

21. Дано следующее описание массива: `var A: Array[1..20] of Real.`

Укажите фрагмент ввода элементов массива с клавиатуры.

- 1) `For i:=1 to 20 do A[i]:=random(15);`
- 2) `Read(A[i]); For i:=1 to 20 do;`
- 3) `For i:=1 to 20 do Read(A[i]).`

22. Дан фрагмент программы: `S:=0; For i:=1 to N do S:=S+A[i].`

Данный фрагмент программы служит для нахождения...

- 1) количества элементов одномерного массива;
- 2) суммы элементов одномерного массива;
- 3) суммы нулевых элементов одномерного массива.

23. Дан фрагмент программы:

`T:=A[1]; K:=1;`

`For i:=1 to N do If A[i,j]>T then Begin T:=A[i]; K:=i end.`

Данный фрагмент программы служит для нахождения...

- 1) максимального элемента и его индексов в одномерном массиве;
- 2) минимального элемента и его индексов в одномерном массиве;
- 3) максимального элемента в одномерном массиве;
- 4) минимального элемента в одномерном массиве.

24. Дан фрагмент программы:

`T:=A[1]; For i:=1 to N do If A[i]>T then T:=A[i].`

Данный фрагмент программы служит для нахождения...

- 1) минимального элемента в одномерном массиве;
- 2) максимального элемента в одномерном массиве;
- 3) максимального элемента и его индексов в одномерном массиве.

25. Дан фрагмент программы:

`N:=0; For j:=1 to M do If A[j]>0 then N:=N+1.`

Данный фрагмент программы служит для нахождения...

- 1) количества положительных элементов в одномерном массиве;
- 2) количества отрицательных элементов в одномерном массиве;

3) количества элементов одномерного массива.

4.3 Промежуточная аттестация по дисциплине

Промежуточная аттестация проводится в форме экзамена

Вопросы для экзамена

1. Составление программ линейной структуры.
2. Составление программ разветвляющейся структуры.
3. Составление программ циклической структуры
4. Обработка одномерных массивов.
5. Обработка двумерных массивов.
6. Работа со строками.
7. Работа с данными типа множество.
8. Файлы последовательного доступа.
9. Типизированные файлы.
10. Нетипизированные файлы.
11. Организация процедур.
12. Организация функций.
13. Применение рекурсивных функций.
14. Программирование модуля.
15. Создание библиотеки подпрограмм.
16. Использование указателей для организации связанных списков.
17. Изучение интегрированной среды разработчика.
18. Создание проекта с использованием компонентов для работы с текстом.
19. Создание проекта с использованием компонентов ввода и отображения чисел, дат и времени.
20. События компонентов (элементов управления), их сущность и назначение.
21. Создание процедур на основе событий.
22. Создание проекта с использованием кнопочных компонентов.
23. Создание проекта с использованием компонентов стандартных диалогов и системы меню.
24. Разработка функциональной схемы работы приложения.
25. Разработка оконного приложения с несколькими формами.
26. Разработка игрового приложения.
27. Создание процедур обработки событий. Компиляция и запуск приложения.
28. Разработка интерфейса приложения.
29. Тестирование, отладка приложения.
30. Классы ООП: виды, назначение, свойства, методы, события.
31. Объявления класса.
32. Создание наследованного класса.
33. Программирование приложений.
34. Перегрузка методов.