

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Шарифуллин Федеральное государственное бюджетное образовательное учреждение высшего образования  
Должность: Директор «Института»  
Дата подписания: 14.10.2024 10:18:54  
Уникальный программный ключ:  
65fd6cbdf7eae29c01b701aabc1fbc13d72d7bd0b08b122e44091c482448eba9

**РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ПРАВОСУДИЯ»**

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)**

**«ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»**

(наименование дисциплины в соответствии с учебным планом)

**Специальность:**

09.02.07 Информационные системы и программирование

(код и наименование)

Рабочая программа разработана в соответствии с требованиями ФГОС.

Разработчик (-): Скотченко А.С., к.т.н., доцент

(ФИО, ученая степень, ученое звание)

Зав. кафедрой Ловцов Д.А., д.т.н., профессор

(ФИО, ученая степень, ученое звание)

(подпись)

ПРОТОКОЛ ИЗМЕНЕНИЙ  
рабочей программы дисциплины (модуля)

«ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»  
(наименование дисциплины в соответствии с учебным планом)

Краткое содержание изменения	Дата и номер протокола заседания кафедры

Актуализация выполнена: Скотченко А.С., к.т.н., доцент  
(ФИО, ученая степень, ученое звание) (подпись)

Зав. кафедрой Ловцов Д.А., д.т.н., профессор  
(ФИО, ученая степень, ученое звание) (подпись)

**ОГЛАВЛЕНИЕ**

АННОТАЦИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ.....	4
1. ЦЕЛИ И ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ .....	5
2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП .....	6
3. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ .....	6
4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ.....	6
4.1. ТЕКСТ РАБОЧЕЙ ПРОГРАММЫ ПО ТЕМАМ .....	6
4.2. РАЗДЕЛЫ И ТЕМЫ ДИСЦИПЛИНЫ, ВИДЫ ЗАНЯТИЙ (ТЕМАТИЧЕСКИЙ ПЛАН) .....	8
4.3. САМОСТОЯТЕЛЬНОЕ ИЗУЧЕНИЕ ОБУЧАЮЩИМИСЯ РАЗДЕЛОВ ДИСЦИПЛИНЫ.....	9
5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ .....	9
5.1. УЧЕБНО-МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ИЗУЧЕНИЮ ДИСЦИПЛИНЫ.....	9
5.2. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ.....	11
6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ.....	12
7. КАРТА ОБЕСПЕЧЕННОСТИ ЛИТЕРАТУРОЙ.....	13
8. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ.....	14
8.1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ.....	14
8.2. ОЦЕНОЧНЫЕ СРЕДСТВА .....	14
8.3. СВОДНАЯ ТАБЛИЦА ПО ПРИМЕНЯЕМЫМ ФОРМАМ И МЕТОДАМ ТЕКУЩЕГО КОНТРОЛЯ И ОЦЕНКИ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ .....	21
8.4. ФОРМА ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ.....	22
8.5. СИСТЕМА ОЦЕНИВАНИЯ КОМПЛЕКТА ФОС ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ.....	22

## АННОТАЦИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ

### «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»

(наименование дисциплины в соответствии с учебным планом)

Разработчик (-и): Скотченко А.С., к.т.н., доцент  
(ФИО, ученая степень, ученое звание)

<b>Цель изучения дисциплины (модуля)</b>	Целью освоения учебной дисциплины «Основы алгоритмизации и программирования» является: изучение и освоение базовых понятий и приемов программирования, применяемых на всех основных этапах разработки программ; изучение методов программирования для овладения знаниями в области технологии программирования; подготовка к осознанному использованию как языков программирования, так и методов программирования.
<b>Место дисциплины в структуре ООП</b>	Учебная дисциплина «Основы алгоритмизации и программирования» относится к базовой части общепрофессионального цикла ОП.Б.4.
<b>Компетенции, формируемые в результате освоения дисциплины (модуля)</b>	<p>ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам.</p> <p>ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности.</p> <p>ОК 04. Эффективно взаимодействовать и работать в коллективе и команде.</p> <p>ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.</p> <p>ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.</p> <p>ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.</p> <p>ПК 1.4. Выполнять тестирование программных модулей.</p> <p>ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода.</p> <p>ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p> <p>ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.</p> <p>ПК 7.3. Формировать требования к конфигурации локальных компьютерных сетей и серверного оборудования, необходимые для работы баз данных и серверов.</p> <p>ПК 7.5. Проводить аудит систем безопасности баз данных и серверов с использованием регламентов по защите информации.</p>
<b>Содержание дисциплины (модуля)</b>	<p><b>Раздел 1. Введение в программирование</b></p> <p><b>Тема 1.1.</b> Языки программирования</p> <p><b>Тема 1.2.</b> Типы данных</p> <p><b>Раздел 2. Операторы и выражения</b></p> <p><b>Тема 2.1.</b> Операторы языка программирования</p>

	<b>Раздел 3. Подпрограммы и функции</b> <b>Тема 3.1.</b> Процедуры и функции <b>Тема 3.2.</b> Структуризация в программировании <b>Тема 3.3.</b> Модульное программирование <b>Раздел 4. Основные конструкции языков программирования</b> <b>Тема 4.1.</b> Указатели <b>Раздел 5. Интегрированные среды разработки</b> <b>Тема 5.1</b> Основные принципы объектно- ориентированного программирования (ООП) <b>Тема 5.2</b> Интегрированная среда разработчика. <b>Тема 5.3.</b> Визуальное событийно- управляемое программирование <b>Тема 5.4.</b> Разработка оконного приложения <b>Тема 5.5.</b> Этапы разработки приложений <b>Тема 5.6.</b> Иерархия классов.
<b>Общая трудоемкость дисциплины (модуля)</b>	Общая трудоемкость дисциплины составляет <b>137</b> часов.
<b>Форма промежуточной аттестации</b>	3 семестр – контрольное задание; 4 семестр – экзамен.

## 1. ЦЕЛИ И ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ

Целью изучения дисциплины (модуля) является освоение компетенций (индикаторов достижения компетенций), предусмотренных рабочей программой.

В совокупности с другими дисциплинами ООП дисциплина обеспечивает формирование следующих компетенций:

Таблица 1

№ п/п	Код компетенции	Название
1	<b>ОК 01.</b>	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам.
2	<b>ОК 02.</b>	Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности.
3	<b>ОК 04.</b>	Эффективно взаимодействовать и работать в коллективе и команде.
4	<b>ПК 1.1.</b>	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.
5	<b>ПК 1.2.</b>	Разрабатывать программные модули в соответствии с техническим заданием.
6	<b>ПК 1.3.</b>	Выполнять отладку программных модулей с использованием специализированных программных средств.
7	<b>ПК 1.4.</b>	Выполнять тестирование программных модулей.
8	<b>ПК 1.5.</b>	Осуществлять рефакторинг и оптимизацию программного кода.
9	<b>ПК 2.4.</b>	Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
10	<b>ПК 2.5.</b>	Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.
11	<b>ПК 7.3.</b>	Формировать требования к конфигурации локальных компьютерных сетей и серверного оборудования, необходимые для работы баз данных и серверов.

12	<b>ПК 7.5.</b>	Проводить аудит систем безопасности баз данных и серверов с использованием регламентов по защите информации.
----	----------------	--

Планируемые результаты освоения дисциплины в части каждой компетенции указаны в картах компетенций по ООП.

В рамках дисциплины осуществляется воспитательная работа, предусмотренная рабочей программой воспитания, календарным планом воспитательной работы.

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП

Учебная дисциплина «Основы алгоритмизации и программирования» относится к базовой части общепрофессионального цикла ОП.Б.4.

## 3. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ

Таблица 2  
очная форма обучения

Вид учебной работы	Трудоемкость			
	зач. ед.	час.	по семестрам	
			3	4
<b>Общая трудоемкость</b> дисциплины по учебному плану		137	46	83
Контактная работа		126	44	82
Занятия лекционного типа		58	16	42
Занятия семинарского типа				
Занятия семинарского типа с практической подготовкой		68	28	40
Самостоятельная работа		11		
<b>Форма промежуточной аттестации</b>			<i>КЗ</i>	<i>Экз.</i>

## 4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

### 4.1. ТЕКСТ РАБОЧЕЙ ПРОГРАММЫ ПО ТЕМАМ

#### **Раздел 1. Введение в программирование**

##### **Тема 1.1.** Языки программирования.

Развитие языков программирования. Обзор языков программирования. Области применения языков программирования. Стандарты языков программирования. Компиляторы и интерпретаторы. Жизненный цикл программы. 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 и выражения**

##### **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. Знакомство со средой программирования. Составление

программ линейной структуры. Составление программ разветвляющейся структуры. Составление программ циклической структуры. Обработка одномерных массивов. Обработка двумерных массивов. Обработка символьных строк. Работа с текстовыми файлами. Работа с двоичными файлами. Составление программ на типизированные файлы. Составление программ на нетипизированные файлы.

### **Раздел 3. Подпрограммы и функции**

#### **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. Организация процедур и функций.

#### **Тема 3.2. Структуризация в программировании.**

Основы структурного программирования.  
Методы структурного программирования.

#### **Тема 3.3. Модульное программирование.**

Модульное программирование. Понятие модуля. Структура модуля.  
Компиляция и компоновка программы. Стандартные модули.  
Программирование модуля.

### **Раздел 4. Основные конструкции языков программирования**

#### **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. Программирование модуля. Использование указателей для организации связанных списков. Создание и удаление динамических переменных.

### **Раздел 5. Интегрированные среды разработки**

#### **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. 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.

#### **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. 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.

**Тема 5.4.** Разработка оконного приложения.

Разработка функционального интерфейса приложения. Создание интерфейса приложения. Разработка функциональной схемы работы приложения. Разработка оконного приложения с несколькими формами.

Разработка игрового приложения.

**Тема 5.5.** Этапы разработки приложений.

Разработка приложения. Проектирование объектно-ориентированного приложения. Создание интерфейса пользователя. Тестирование, отладка приложения. Создание процедур обработки событий. Компиляция и запуск приложения.

**Тема 5.6.** Иерархия классов.

Классы ООП: виды, назначение, свойства, методы, события.

Перегрузка методов. Тестирование и отладка приложения. Решение задач

Создание наследованного класса. Классы и объекты.

Составление начальной иерархии и структуры классов.

## 4.2. РАЗДЕЛЫ И ТЕМЫ ДИСЦИПЛИНЫ, ВИДЫ ЗАНЯТИЙ (ТЕМАТИЧЕСКИЙ ПЛАН)

Таблица 3

## Тематический план

очная форма обучения

№	Раздел дисциплины, тема	Код компетенции	Общая трудоёмкость дисциплины	в том числе			Наименование оценочного средства
				Контактная работа	Занятия лекционного типа	Занятия семинарского типа с практической подготовкой	
				час.	час.	час.	
1	<i>Введение в программирование</i>	ОК 01, ОК 02, ОК 04, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5, ПК 7.3, ПК 7.5.	10	10	8	2	Вопросы по темам, тестирование, защита практических работ
2	<i>Операторы и выражения</i>		34	34	8	26	
3	<i>Подпрограммы и функции</i>		16	16	8	8	
4	<i>Основные конструкции языков программирования</i>		12	12	4	8	
5	<i>Интегрированные среды разработки</i>		65	54	30	24	
<b>ВСЕГО</b>			137	126	58	68	

### 4.3. САМОСТОЯТЕЛЬНОЕ ИЗУЧЕНИЕ ОБУЧАЮЩИМИСЯ РАЗДЕЛОВ ДИСЦИПЛИНЫ

Таблица 4

№ раздела (темы) дисциплины (модуля)	Вопросы, выносимые на самостоятельное изучение	Кол-во часов
Раздел 5	Разработка игрового приложения.	11

## 5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### 5.1. УЧЕБНО-МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ИЗУЧЕНИЮ ДИСЦИПЛИНЫ

*Структура и отчётность* – дисциплина включает последовательное изучение пяти разделов, обязательное самостоятельное конспектирование основных понятий и определений и заканчивается экзаменом.

#### *Список основных понятий дисциплины:*

1. Алгоритмы их основные свойства.
2. Понятие трудоемкости.
3. Теория сложности вычислений. Класс P, NP.
4. Алгоритм Эвклида.
5. Решето Эратосфена
6. Эвристические алгоритмы решения NP-трудных задач.
7. Жадные алгоритмы.
8. Построение минимального остовного дерева (алгоритм Краскала).
9. Генетический алгоритм.
10. Рекурсивные алгоритмы
11. Теория автоматов.
12. Машина Тьюринга.
13. Фракталы. Виды фракталов. Размерность фрактала.
14. Алгоритмы сжатия данных.
15. Информационная энтропия.
16. Динамическое программирование (идея, классические задачи ).
17. Сортировки. Устойчивость сортировок. Оценка алгоритма сортировки.
18. Сортировка и группировка данных в SQL.

#### **Методические указания студентам по подготовке к практическим занятиям**

Подготовка к практическим занятиям является одной из важнейших форм самостоятельной работы студентов. Целью практических занятий является закрепление знаний полученных на лекционных занятиях и в ходе самостоятельной работы, а также выработка навыков работы с учебной и научной литературой. Приступая к решению задачи, студент должен, прежде всего, уяснить содержание задачи. Подготовку к занятиям следует начинать с внимательного изучения соответствующих разделов учебных пособий и учебников. По завершению изучения рекомендованной литературы, студенты могут проверить свои знания с помощью вопросов для самоконтроля, входящих в состав вопросов к зачету (экзамену) по соответствующему разделу (темы) изучаемой дисциплины.

### **Методические рекомендации преподавателям при организации практических занятий**

Работа по подготовке к практическим занятиям и активное в них участие - одна из форм изучения программного материала курса. Практика показывает, что основные формы занятий следующие:

- беседа на основе составленного преподавателем плана,
- подготовка письменного доклада студентом,
- устный доклад и обсуждение его на практическом занятии.

Учебные и воспитательные цели практических занятий:

- Актуализировать знания студентов из курса по соответствующему разделу (теме) учебного плана по дисциплине.
- Создать условия для самостоятельного изучения образовательных ресурсов.
- Создать условия для развития у студентов творческой активности, самостоятельности и критичности мышления.
- Содействовать развитию у студентов общенаучных компетенций (аналитико-синтетической, прогностической, проектировочной).
- Создать условия для развития коммуникативной, адаптивной и информационной компетенций.

Выполнению практических занятий предшествует проверка знаний студентов - их теоретической готовности к выполнению задания.

Практические занятия могут носить:

- репродуктивный характер;
- частично-поисковый характер;
- поисковый характер.

Работы, носящие репродуктивный характер, отличаются тем, что при их проведении студенты пользуются подробными инструкциями, в которых указаны: цель работы, пояснения, контрольные вопросы, учебная и специальная литература.

Работы, носящие частично-поисковый характер, отличаются тем, что при их проведении студенты не пользуются подробными инструкциями, им не дан порядок выполнения необходимых действий, и требуют от студентов самостоятельного подбора оборудования, выбора способов выполнения работы в инструктивной и справочной литературе и др.

Работы, носящие поисковый характер, характеризуются тем, что студенты должны решить новую для них проблему, опираясь на имеющиеся у них теоретические знания. При планировании практических занятий необходимо находить оптимальное соотношение репродуктивных, частично-поисковых и поисковых работ, чтобы обеспечить высокий уровень интеллектуальной деятельности.

Формы организации работы студентов на занятиях:

- фронтальная;
- групповая;
- индивидуальная.

При фронтальной форме организации занятий все студенты выполняют одновременно одну и ту же работу.

При групповой форме организации занятий одна и та же работа выполняется бригадами по 2 - 5 человек.

При индивидуальной форме организации занятий каждый студент выполняет индивидуальное задание.

Для повышения эффективности проведения практических занятий рекомендуется:

- разработка сборников задач, заданий и упражнений, сопровождающихся методическими указаниями, применительно к конкретным специальностям;

- разработка заданий для автоматизированного тестового контроля за подготовленностью студентов к практическим занятиям;
- подчинение методики проведения практических занятий ведущим дидактическим целям с соответствующими установками для студентов;
- использование в практике работ, построенных на проблемной основе применения коллективных и групповых форм работы, максимальное использование индивидуальных форм с целью повышения ответственности каждого студента за самостоятельное выполнение полного объема работ;
- проведение практических занятий на повышенном уровне трудности с включением в них заданий, связанных с выбором студентами условий выполнения работы, конкретизацией целей, самостоятельным отбором необходимого оборудования;
- подбор дополнительных задач и заданий для студентов, работающих в более быстром темпе, для эффективного использования времени, отводимого на лабораторные работы и практические занятия.

## 5.2. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ

1. Трофимов, В. В. Основы алгоритмизации и программирования: учебник для среднего профессионального образования / В. В. Трофимов, Т. А. Павловская; под редакцией В. В. Трофимова. – Москва: Издательство Юрайт, 2024. – 119 с. – (Профессиональное образование). – ISBN 978-5-534-17498-4. – URL : <https://urait.ru/bcode/539994>

2. Кудрина, Е. В. Основы алгоритмизации и программирования на языке C# : учебное пособие для среднего профессионального образования / Е. В. Кудрина, М. В. Огнева. – Москва: Издательство Юрайт, 2024. – 322 с. – (Профессиональное образование). – ISBN 978-5-534-10772-2. – URL: <https://urait.ru/bcode/541725>

3. Макарова, Н.В., Основы программирования : учебник и практикум / Н.В. Макарова, Ю.Н. Нилова, С.Б. Зеленина, Е.В. Лебедева. — Москва : КноРус, 2023. — 452 с. — ISBN 978-5-406-03394-4. — URL:<https://book.ru/book/936582>.

4. Дорогов, В. Г. Основы программирования на языке C : учебное пособие / В.Г. Дорогов, Е.Г. Дорогова ; под ред. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2023. — 224 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0809-9. - Текст : электронный. - URL: <https://znanium.com/catalog/product/2010597>.

### Интернет-источники

1. Сайт о программировании. [Электронный ресурс]. URL: <https://metanit.com/web/php/3.4.php>

2. Шестаков А.П. Учителям информатики и математики и их любознательным ученикам (дидактические материалы по информатике и математике). [Электронный ресурс]. URL: <http://comp-science.narod.ru/>

**Основная и дополнительная литература** указана в Карте обеспеченности литературой.

## 6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Для материально-технического обеспечения дисциплины (модуля) используются специальные помещения. Специальные помещения представляют собой учебные аудитории для проведения занятий лекционного типа, занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы и помещения для хранения и профилактического обслуживания учебного оборудования.

Для проведения занятий лекционного типа предлагаются наборы демонстрационного оборудования и учебно-наглядных пособий, обеспечивающие тематические иллюстрации, соответствующие рабочим программам дисциплин. Демонстрационное оборудование представлено в виде мультимедийных средств. Учебно-наглядные пособия представлены в виде экранно-звуковых средств, печатных пособий, слайд-презентаций, видеофильмов, макетов и т.д., которые применяются по необходимости в соответствии с темами (разделами) дисциплины (модуля).

Для самостоятельной работы обучающихся помещения оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду университета.

Предусмотрены помещения для хранения и профилактического обслуживания учебного оборудования.

Перечень специальных помещений ежегодно обновляется и отражается в справке о материально-техническом обеспечении основной образовательной программы.

Состав необходимого комплекта лицензионного программного обеспечения ежегодно обновляется, утверждается и отражается в справке о материально-техническом обеспечении основной образовательной программы.

№ п\п	Наименование дисциплины в соответствии с учебным планом	Наименование специальных помещений и помещений для самостоятельной работы
1	ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ	Студия разработки дизайна веб-приложений. (ИЛК-1 (помещение 1001)-9 (330))

## 7. КАРТА ОБЕСПЕЧЕННОСТИ ЛИТЕРАТУРОЙ

Кафедра Информационного права, информатики и математики

Специальность: 09.02.07 Информационные системы и программирование

Дисциплина «Основы алгоритмизации и программирования»

Курс 2.

№ п/п	Полное библиографическое описание
<b>Основная литература</b>	
1	Трофимов, В. В. Основы алгоритмизации и программирования: учебник для среднего профессионального образования / В. В. Трофимов, Т. А. Павловская; под редакцией В. В. Трофимова. – Москва: Издательство Юрайт, 2024. – 119 с. – (Профессиональное образование). – ISBN 978-5-534-17498-4. – URL : <a href="https://urait.ru/bcode/539994">https://urait.ru/bcode/539994</a>
<b>Дополнительная литература</b>	
1	Кудрина, Е. В. Основы алгоритмизации и программирования на языке C# : учебное пособие для среднего профессионального образования / Е. В. Кудрина, М. В. Огнева. – Москва: Издательство Юрайт, 2024. – 322 с. – (Профессиональное образование). – ISBN 978-5-534-10772-2. – URL: <a href="https://urait.ru/bcode/541725">https://urait.ru/bcode/541725</a>
2	Макарова, Н.В., Основы программирования : учебник и практикум / Н.В. Макарова, Ю.Н. Нилова, С.Б. Зеленина, Е.В. Лебедева. — Москва : КноРус, 2023. — 452 с. — ISBN 978-5-406-03394-4. — URL: <a href="https://book.ru/book/936582">https://book.ru/book/936582</a> .
<b>Дополнительная литература для углубленного изучения дисциплины</b>	
1	Дорогов, В. Г. Основы программирования на языке C : учебное пособие / В.Г. Дорогов, Е.Г. Дорогова ; под ред. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2023. — 224 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0809-9. - Текст : электронный. - URL: <a href="https://znanium.com/catalog/product/2010597">https://znanium.com/catalog/product/2010597</a> .

## 8. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

### 8.1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

В результате освоения дисциплины обучающийся *должен 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.

В результате освоения дисциплины обучающийся *должен 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.

№ п/п	Раздел дисциплины (модуля), тема	Код компетенции	Наименование оценочного средства
1.	Введение в программирование	ОК 01, ОК 02, ОК 04, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5, ПК 7.3, ПК 7.5.	Вопросы по темам, тестирование, защита практических работ
2.	Операторы и выражения		
3.	Подпрограммы и функции		
4.	Основные конструкции языков программирования		
5.	Интегрированные среды разработки		

### 8.2. ОЦЕНОЧНЫЕ СРЕДСТВА

#### ПРИМЕРЫ КОНТРОЛЬНЫХ ЗАДАНИЙ

1. Перечень компетенций (части компетенции), проверяемых оценочным средством (наименование, код):

№ п/п	Код компетенции	Название
1	<b>ОК 01.</b>	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам.

2	<b>ОК 02.</b>	Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности.
3	<b>ОК 04.</b>	Эффективно взаимодействовать и работать в коллективе и команде.
4	<b>ПК 1.1.</b>	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.
5	<b>ПК 1.2.</b>	Разрабатывать программные модули в соответствии с техническим заданием.
6	<b>ПК 1.3.</b>	Выполнять отладку программных модулей с использованием специализированных программных средств.
7	<b>ПК 1.4.</b>	Выполнять тестирование программных модулей.
8	<b>ПК 1.5.</b>	Осуществлять рефакторинг и оптимизацию программного кода.
9	<b>ПК 2.4.</b>	Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
10	<b>ПК 2.5.</b>	Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.
11	<b>ПК 7.3.</b>	Формировать требования к конфигурации локальных компьютерных сетей и серверного оборудования, необходимые для работы баз данных и серверов.
12	<b>ПК 7.5.</b>	Проводить аудит систем безопасности баз данных и серверов с использованием регламентов по защите информации.

#### **Формы контроля и оценки результатов освоения УД.**

Контроль и оценка результатов освоения – это выявление, измерение и оценивание знаний, умений и формирующихся общих и профессиональных компетенций в рамках освоения УД.

В соответствии с учебным планом специальности 09.02.07 Информационные системы (по отраслям), рабочей программой дисциплины «Основы проектирования баз данных предусматривается текущий и промежуточный контроль результатов освоения.

#### **Формы текущего контроля УД.**

Текущий контроль успеваемости представляет собой проверку усвоения учебного материала, регулярно осуществляемую на протяжении курса обучения.

Текущий контроль результатов освоения УД (МДК) в соответствии с рабочей программой и календарно-тематическим планом происходит при использовании следующих обязательных форм контроля:

- выполнение и защита практических работ,
- проверка выполнения самостоятельной работы студентов,

Во время проведения учебных занятий дополнительно используются следующие формы текущего контроля – устный опрос, тестирование по темам отдельных занятий.

#### **Выполнение и защита практических работ.**

Практические работы проводятся с целью усвоения и закрепления практических умений и знаний, овладения профессиональными компетенциями. В ходе практической работы студенты приобретают умения, предусмотренные рабочей программой УД, учатся использовать операторы, анализировать полученные результаты и делать выводы, опираясь на теоретические знания.

#### **Список практических работ:**

##### **Тема 1.1. Языки программирования**

*Содержание учебного материала:*

Развитие языков программирования.

Обзор языков программирования. Области применения языков программирования. Стандарты языков программирования. Среда проектирования. Компиляторы и интерпретаторы. Жизненный цикл программы. Программа. Программный продукт и его характеристики. Основные этапы решения задач на компьютере.

*Самостоятельная работа:* проработка конспекта лекции, подготовка к опросу.

*Формы текущего контроля по теме:* устный опрос.

*Вопросы для подготовки к текущей аттестации по теме:*

1. Развитие языков программирования.
2. Обзор языков программирования.
3. Области применения языков программирования.
4. Стандарты языков программирования.
5. Среда проектирования.
6. Компиляторы и интерпретаторы.
7. Жизненный цикл программы.
8. Программа.
9. Программный продукт и его характеристики.
10. Основные этапы решения задач на компьютере.

## **Тема 1.2. Типы данных**

*Содержание учебного материала:*

Типы данных. Простые типы данных. Производные типы данных. Структурированные типы данных.

*Формы текущего контроля по теме:* письменный опрос.

*Вопросы для подготовки к текущей аттестации по теме:*

1. Этапы развития ЭВМ.
2. Классификация ЭВМ по принципу действия.
3. Поколения ЭВМ.
4. Классификация ЭВМ по назначению.
5. Классификация ЭВМ по функциональным возможностям.

## **Тема 2.1. Операторы языка программирования**

*Содержание учебного материала*

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.

*Практические работы (практические занятия):*

1. Знакомство со средой программирования.
2. Алгоритмические структуры.

*Лабораторные работы (лабораторные занятия):*

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.

*Самостоятельная работа:* проработка конспекта лекции.  
*Формы текущего контроля по теме:* практическая работа.  
*Вопросы для подготовки к текущей аттестации по теме:*

1. Операции и выражения.
2. Правила формирования и вычисления выражений.
3. Структура программы.
4. Ввод и вывод данных.
5. Оператор присваивания.
6. Составной оператор.
7. Условный оператор.
8. Оператор выбора.
9. Цикл с постусловием.
10. Цикл с предусловием.
11. Цикл с параметром.
12. Вложенные циклы.
13. Массивы.
14. Двумерные массивы.
15. Строки.
16. Стандартные процедуры и функции для работы со строками.
17. Структурированный тип данных – множество.
18. Операции над множествами.
19. Комбинированный тип данных – запись.
20. Файлы последовательного доступа.
21. Файлы прямого доступа.

### **Тема 3.1. Процедуры и функции**

*Содержание учебного материала:*

Общие сведения о подпрограммах. Определение и вызов подпрограмм. Область видимости и время жизни переменной. Механизм передачи параметров. Организация функций. Рекурсия. Программирование рекурсивных алгоритмов.

*Практические работы:*

1. Организация процедур.
2. Организация функций.
3. Применение рекурсивных функций.

*Формы текущего контроля по теме: тест, практическая работа.*

*Вопросы для подготовки к текущей аттестации по теме:*

1. Общие сведения о подпрограммах.
2. Определение и вызов подпрограмм.
3. Область видимости и время жизни переменной.
4. Механизм передачи параметров.
5. Организация функций.
6. Рекурсия.
7. Программирование рекурсивных алгоритмов.

### **Тема 3.2. Структуризация в программировании**

*Содержание учебного материала:*

Основы структурного программирования. Методы структурного программирования.

*Формы текущего контроля по теме: устный опрос.*

*Вопросы для подготовки к текущей аттестации по теме:*

1. Основы структурного программирования.
2. Методы структурного программирования.

### **Тема 3.3. Модульное программирование**

*Содержание учебного материала:*

Модульное программирование. Понятие модуля. Структура модуля. Компиляция и компоновка программы. Стандартные модули.

*Практические работы:*

1. Программирование модуля.
2. Создание библиотеки подпрограмм.

*Формы текущего контроля по теме:* разработка ментальной карты, лабораторная работа.

### **Тема 4.1. Указатели**

*Содержание учебного материала:*

Указатели. Описание указателей. Основные понятия и применение динамически распределяемой памяти. Создание и удаление динамических переменных. Структуры данных на основе указателей.

*Практические работы:*

1. Задача о стеке.
2. Использование указателей для организации связанных списков.

*Самостоятельная работа:* проработка конспекта лекции.

*Формы текущего контроля по теме:* практическая работа.

*Вопросы для подготовки к текущей аттестации по теме:*

1. Указатели.
2. Описание указателей.
3. Основные понятия и применение динамически распределяемой памяти.
4. Создание и удаление динамических переменных.
5. Структуры данных на основе указателей.
6. Задача о стеке.

### **Тема 5.1. Основные принципы объектно-ориентированного программирования (ООП)**

*Содержание учебного материала:*

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.

*Практические работы:*

1. Классы ООП: виды, назначение, свойства, методы, события.
2. Объявления класса.
3. Создание наследованного класса.

*Формы текущего контроля по теме:* устный опрос, лабораторная работа.

*Вопросы для подготовки к текущей аттестации по теме:*

1. История развития ООП.
2. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс.
3. Основные принципы ООП: инкапсуляция, наследование, полиморфизм.
4. Классы объектов.
5. Компоненты и их свойства.
6. Событийно-управляемая модель программирования.
7. Компонентно-ориентированный подход.

### **Тема 5.2. Интегрированная среда разработчика**

*Содержание учебного материала:*

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.

*Практические работы (практические занятия):*

1. Изучение интегрированной среды разработчика. Настройка среды и параметров проекта.

*Самостоятельная работа: проработка конспекта лекции.*

Формы текущего контроля по теме: практическая работа.

*Вопросы для подготовки к текущей аттестации по теме:*

1. Требования к аппаратным и программным средствам интегрированной среды разработчика.
2. Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты.
3. Форма и размещение на ней управляющих элементов.
4. Панель компонентов и их свойства.
5. Окно кода проекта.
6. Состав и характеристика проекта.
7. Выполнение проекта.
8. Настройка среды и параметров проекта.
9. Панель компонентов и их свойства. Окно кода проекта. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.
10. Настройка среды и параметров проекта.

### **Тема 5.3. Визуальное событийно-управляемое программирование**

*Содержание учебного материала:*

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.

*Практические работы (практические занятия):*

1. Создание процедур на основе событий.
2. Создание проекта с использованием кнопочных компонентов.
3. Создание проекта с использованием компонентов стандартных диалогов и системы меню.

Формы текущего контроля по теме: практическая работа.

*Вопросы для подготовки к текущей аттестации по теме:*

1. Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение.
2. Дополнительные элементы управления. Свойства компонентов. Виды свойств. Синтаксис определения свойств. Назначения свойств и их влияние на результат. Управление объектом через свойства.
3. События компонентов (элементов управления), их сущность и назначение. Создание процедур на основе событий.

### **Тема 5.4. Разработка оконного приложения**

*Содержание учебного материала:*

Разработка функционального интерфейса приложения. Создание интерфейса приложения. Разработка функциональной схемы работы приложения. Разработка игрового приложения.

*Практические работы (практические занятия):*

1. Создание процедур на основе событий.

2. Создание проекта с использованием кнопочных компонентов.
3. Создание проекта с использованием компонентов стандартных диалогов и системы меню.

Формы текущего контроля по теме: практическая работа.

Самостоятельная работа: проработка конспекта лекции.

Вопросы для подготовки к текущей аттестации по теме:

1. Разработка интерфейса приложения.
2. Создание проекта с использованием кнопочных компонентов.
3. Создание проекта с использованием компонентов стандартных диалогов и системы меню.

### **Тема 5.5. Этапы разработки приложений**

*Содержание учебного материала:*

Разработка приложения. Проектирование объектно-ориентированного приложения.

Создание интерфейса пользователя. Тестирование, отладка приложения.

*Практические работы (практические занятия):*

1. Разработка функциональной схемы работы приложения.
2. Разработка оконного приложения с несколькими формами.
3. Разработка игрового приложения.
4. Тестирование, отладка приложения.

*Самостоятельная работа:* проработка конспекта лекции.

Вопросы для подготовки к текущей аттестации по теме:

1. Разработка приложения.
2. Проектирование объектно-ориентированного приложения.
3. Создание интерфейса пользователя.
4. Тестирование, отладка приложения.

### **Тема 5.6. Иерархия классов**

*Содержание учебного материала:*

Классы ООП: виды, назначение, свойства, методы, события. Перегрузка методов.

Тестирование и отладка приложения. Решение задач.

*Практические работы (практические занятия):*

1. Программирование приложений.
2. Перегрузка методов.

Формы текущего контроля по теме: практическая работа.

*Самостоятельная работа:* проработка конспекта лекции.

*Вопросы для подготовки к текущей аттестации по теме:*

1. Классы ООП: виды, назначение, свойства, методы, события.
2. Перегрузка методов.
3. Тестирование и отладка приложения.
4. Решение задач.

### **Проверка выполнения самостоятельной работы.**

Самостоятельная работа направлена на самостоятельное освоение и закрепление студентами практических умений и знаний, овладение профессиональными компетенциями.

Самостоятельная подготовка студентов по УД(МДК) предполагает следующие виды и формы работы:

- Систематическая проработка конспектов занятий, учебной и специальной технической литературы.
- Самостоятельное изучение материала и конспектирование лекций по учебной и специальной технической литературе.

- Написание и защита доклада; подготовка к сообщению или беседе на занятии по заданной преподавателем теме.
- Оформление отчетов и практическим работам, и подготовка к их защите.

### 8.3. СВОДНАЯ ТАБЛИЦА ПО ПРИМЕНЯЕМЫМ ФОРМАМ И МЕТОДАМ ТЕКУЩЕГО КОНТРОЛЯ И ОЦЕНКИ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ

Результаты обучения (освоенные умения, усвоенные знания)	Формы и методы контроля и оценки результатов обучения
<b>Уметь</b>	
разрабатывать алгоритмы для конкретных задач;	Индивидуальная: практические занятия.
использовать программы для графического отображения алгоритмов;	Индивидуальная: практические занятия.
определять сложность работы алгоритмов;	Индивидуальная: практические занятия.
работать в среде программирования;	Индивидуальная: практические занятия.
реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;	Индивидуальная: практические занятия.
оформлять код программы в соответствии со стандартом кодирования;	Индивидуальная: практические занятия.
выполнять проверку, отладку кода программы.	Индивидуальная: практические занятия.
<b>Знать</b>	
понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;	Индивидуальная: тестирование, индивидуальные задания, ответ на контрольные вопросы. Групповая: устный опрос.
эволюцию языков программирования, их классификацию, понятие системы программирования;	Индивидуальная: тестирование, индивидуальные задания, ответ на контрольные вопросы
основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;	Индивидуальная: тестирование, индивидуальные задания, ответ на контрольные вопросы
подпрограммы, составление библиотек подпрограмм;	Индивидуальная: тестирование, индивидуальные задания, ответ на контрольные вопросы
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.	Индивидуальная: тестирование, индивидуальные задания, ответ на контрольные вопросы

--	--

#### **8.4. ФОРМА ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ**

Студенты допускаются к сдаче экзамена при выполнении всех видов самостоятельной работы, практических работ, предусмотренных рабочей программой и календарно-тематическим планом УД.

#### **8.5. СИСТЕМА ОЦЕНИВАНИЯ КОМПЛЕКТА ФОС ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ**

При оценивании практической и самостоятельной работы студента учитывается следующее:

- качество выполнения практической части работы;
- качество оформления отчета по работе;
- качество устных ответов на контрольные вопросы при защите работы. Каждый вид работы оценивается по пяти бальной шкале.

«5» (отлично) – за глубокое и полное овладение содержанием учебного материала, в котором студент свободно и уверенно ориентируется; за умение практически применять теоретические знания, высказывать и обосновывать свои суждения. Оценка «5» (отлично) предполагает грамотное и логичное изложение ответа.

«4» (хорошо) – если студент полно освоил учебный материал, владеет научно-понятийным аппаратом, ориентируется в изученном материале, осознанно применяет теоретические знания на практике, грамотно излагает ответ, но содержание и форма ответа имеют отдельные неточности.

«3» (удовлетворительно) – если студент обнаруживает знание и понимание основных положений учебного материала, но излагает его неполно, непоследовательно, допускает неточности, в применении теоретических знаний при ответе на практико-ориентированные вопросы; не умеет доказательно обосновать собственные суждения.

«2» (неудовлетворительно) – если студент имеет разрозненные, бессистемные знания, допускает ошибки в определении базовых понятий, искажает их смысл; не может практически применять теоретические знания.

Тест оценивается по пяти бальной шкале следующим образом: стоимость каждого вопроса 1 балл. За правильный ответ студент получает 1 балл. За неверный ответ или его отсутствие баллы не начисляются.

Оценка «5» соответствует 86% – 100% правильных ответов.

Оценка «4» соответствует 73% – 85% правильных ответов.

Оценка «3» соответствует 53% – 72% правильных ответов.

Оценка «2» соответствует 0% – 52% правильных ответов.

#### **ТЕСТОВЫЕ ЗАДАНИЯ**

#### **СОДЕРЖАНИЕ БАНКА ТЕСТОВЫХ ЗАДАНИЙ**

*Алгоритмические структуры одинакового типа, находящиеся одна внутри другой, называют ...*

- А) рекурсивными
- Б) вложенными
- В) итерационными
- Г) прогрессивными

**Принцип модульности при разработке программного обеспечения предполагает ...**

- А) объединение всех элементов программы в единый модуль
- Б) вычисление абсолютной величины всех переменных программы
- В) разделение программы на отдельные элементы
- Г) вычисление модулей пользовательских функций

**Алгоритмическая структура, предписывающая выполнение последовательности действий одно за другим без пропусков и повторений**

- А) ветвление
- Б) следование
- В) цикл
- Г) рекурсия

**Операции отношения в традиционных алгоритмических языках обозначаются: ...**

- А) -, +, \*, /, ^
- Б) and, or, not
- В) >, <, >=, <=, =, <>
- Г) ||, \$, %, @

**Отладка программы заключается ...**

- А) в ее модификации
- Б) в устранении программных ошибок
- В) в ее рациональном использовании
- Г) в сокращении занимаемой ею оперативной памяти

**Алгоритмическая структура, обеспечивающая многократное выполнение некоторой совокупности действий.**

- А) ветвление
- Б) следование
- В) цикл
- Г) останов

**Укажите НЕ существующую разновидность базовой алгоритмической структуры «цикл».**

- А) цикл с предусловием
- Б) цикл с постусловием
- В) цикл с константой
- Г) цикл с параметром

**Математическая модель реального объекта ...**

- А) всегда должна быть абсолютно точной
- Б) допускает наличие погрешностей расчетов
- В) всегда обеспечивает устранение погрешностей расчетов
- Г) не допускает накопление погрешностей расчетов

**Алгоритмическая структура, предписывающая выполнять набор инструкций в циклическом режиме до тех пор, пока не выполнится некоторое заданное условие.**

- А) регрессионная
- Б) прогрессивная
- В) рекурсивная
- Г) итерационная

**Математическая модель, доступная для реализации в двоичных кодах ЭВМ называется ...**

- А) модульной
- Б) объектной
- В) дискретной
- Г) двоичной

**Алгоритмическая структура, определяющая набор повторяющихся действий, отличающаяся тем, что каждая стадия повторений реализуется как подзадача предыдущей стадии.**

- А) регрессионная
- Б) прогрессивная
- В) рекурсивная
- Г) итерационная

**Основные типы выражений в традиционных языках программирования: ...**

- А) механические, электронные, печатные
- Б) арифметические, логические, строковые
- В) арифметические, алгебраические, тригонометрические
- Г) функциональные, декларативные, объектные

**Неустойчивый вычислительный метод характеризуется ...**

- А) обязательным наличием ошибки переполнения
- Б) последовательным уточнением решения задачи
- В) не выполнением условия истинности логических высказываний
- Г) накоплением в процессе вычислений погрешностей округления

**Языки программирования первого поколения являются ...**

- А) машинно-независимыми
- Б) машинно-зависимыми
- В) объектно-ориентированными
- Г) декларативными

**Фраза традиционного алгоритмического языка, определяющая законченный этап обработки данных -**

- А) массив
- Б) выражение
- В) операция
- Г) оператор

**Ассемблер – это ...**

- А) язык программирования низкого уровня
- Б) язык программирования высокого уровня
- В) объектно-ориентированный язык программирования
- Г) декларативный язык программирования

**Программная процедура в алгоритмическом языке определяет обращение ...**

- А) к соответствующей последовательности операторов
- Б) к ячейке оперативной памяти
- В) к серверу по выделенной линии связи
- Г) к пользовательскому интерфейсу операционной системы

**Упорядоченный набор недвусмысленных и выполнимых этапов, определяющих некоторый конечный процесс -**

- А) команда
- Б) алгоритм
- В) оператор
- Г) кластер

**Алгоритмические структуры одинакового типа, находящиеся одна внутри другой, называют ...**

- А) рекурсивными
- Б) вложенными
- В) итерационными
- Г) прогрессивными

**Языки программирования высокого уровня являются ...**

- А) машинно-независимыми

- Б) машинно-зависимыми
- В) машинными
- Г) естественными

**Жизненный цикл программного обеспечения НЕ включает стадию ...**

- А) разработки
- Б) использования
- В) интерактивности
- Г) модификации

**Транслятор – это ...**

- А) устройство-передатчик файлов между компьютерами;
- Б) основной модуль операционной системы;
- В) программа-переводчик с языка программирования высокого уровня в машинные коды;
- Г) текстовый редактор.

**Степень совпадения полученного результата с идеальным характеризует ...**

- А) точность алгоритма (программы)
- Б) универсальность алгоритма (программы)
- В) адекватность алгоритма (программы)
- Г) экономичность алгоритма (программы)

**Псевдокод - это**

- А) язык программирования низкого уровня
- Б) язык программирования высокого уровня
- В) система записи алгоритма с помощью текстовых структур
- Г) система записи алгоритма с помощью графических структур

**Компилятор и интерпретатор – это разновидности ...**

- А) программ-переводчиков языка программирования высокого уровня в машинные коды
- Б) пользователей ЭВМ по уровню их профессионализма
- В) периферийных устройств ЭВМ
- Г) языков программирования

**Применимость к решению некоторого круга задач характеризует ...**

- А) точность алгоритма (программы)
- Б) универсальность алгоритма (программы)
- В) адекватность алгоритма (программы)
- Г) экономичность алгоритма (программы)

**Способность приводить к результату с погрешностью не выше заданной характеризует ...**

- А) точность алгоритма (программы)
- Б) универсальность алгоритма (программы)
- В) адекватность алгоритма (программы)
- Г) экономичность алгоритма (программы)

**Если результат компьютерного вычисления превосходит по абсолютной величине наибольшее представимое в данной кодировке значение происходит ...**

- А) ошибка округления
- Б) ошибка переполнения
- В) ошибка исходных данных
- Г) ошибка измерения информации

**Словесный способ записи алгоритма представляет собой описание последовательности этапов обработки данных ...**

- А) на машинном языке
- Б) на языке программирования
- В) на естественном языке

Г) графической схемой

**Применимость к решению некоторого круга задач характеризует ...**

- А) точность алгоритма (программы)
- Б) универсальность алгоритма (программы)
- В) адекватность алгоритма (программы)
- Г) экономичность алгоритма (программы)

**Основными компонентами традиционного алгоритмического языка являются ...**

- А) константа, переменная, функция
- Б) орфография, грамматика, пунктуация
- В) алфавит, синтаксис, семантика
- Г) символ, число, операция

**Затратами вычислительных ресурсов характеризуется ...**

- А) точность алгоритма (программы)
- Б) универсальность алгоритма (программы)
- В) адекватность алгоритма (программы)
- Г) экономичность алгоритма (программы)

**В языке программирования имена служат для ... объектов программы**

- А) идентификации
- Б) структурирования
- В) конкатенации
- Г) итерации

**Совокупность математических объектов и отношений между ними, отражающая некоторые свойства реального объекта -**

- А) формула
- Б) функция
- В) программная процедура
- Г) математическая модель

**Константы – это данные, которые в процессе выполнения программы ...**

- А) изменяют свои значения
- Б) не изменяют свои значения
- В) не имеют значений
- Г) теряют свое значение

**Данные в виде последовательности однотипных элементов, число которых фиксировано и которым присвоено одно имя ...**

- А) константа
- Б) массив
- В) переменная
- Г) алгоритм

**Математические прообразы массивов -**

- А) переменные и константы
- Б) векторы и матрицы
- В) числа и функции
- Г) операции и выражения

## **ВОПРОСЫ, ВЫНОСИМЫЕ НА ЗАЧЕТ (ЭКЗАМЕН), ПО ДИСЦИПЛИНЕ**

1. Понятие алгоритма. Приведите примеры. Этапы решения задачи на ЭВМ.
2. Охарактеризуйте этап постановка задачи. Приведите примеры.
3. Охарактеризуйте этап построения модели. Приведите примеры.
4. Охарактеризуйте этап разработка алгоритма. Приведите примеры.
5. Охарактеризуйте этап проверки правильности алгоритма. Приведите примеры.
6. Охарактеризуйте этап реализации алгоритма. Приведите примеры.

7. Охарактеризуйте этап анализа алгоритма и его сложности. Приведите примеры.
8. Охарактеризуйте этап проверки программы. Приведите примеры.
9. Охарактеризуйте этап документирования программы.
10. Охарактеризуйте методику структурного программирования сверху - вниз.
11. Понятие блок-схемы. Элементы структурной блок-схемы.
12. Приведение неструктурированной блок-схемы к эквивалентной структурированной блок-схеме. Приведите примеры.
13. Программирование сверху вниз и правильность программ.
14. Постановка задачи линейного программирования.
15. Основные этапы симплексного метода решения задачи линейного программирования.
16. Алгоритм выбора исходного базисного решения при использовании симплексного алгоритма.
17. Алгоритм перехода к новому базису при решении задачи линейного программирования.
18. Двойственная задача линейного программирования.
19. Алгоритм проверки окончания итераций при решении задачи линейного программирования.
20. Программная реализация симплексного алгоритма решения задачи линейного программирования.
21. Постановка классической транспортной задачи.
22. Шаги симплексного метода решения транспортной задачи.
23. Использование двойственной задачи для решения транспортной задачи.
24. Алгоритм перехода к новому базису в транспортной задаче.
25. Алгоритм построения исходного базисного решения в транспортной задаче.
26. Алгоритм окончания итераций при решении транспортной задачи.
27. Программная реализация алгоритма решения транспортной задачи.
28. Охарактеризуйте метод ветвей и границ.
29. Операция редукции. Приведите примеры.
30. Постановка задачи о гамильтоновом пути. Приближенные методы решения.
31. Алгоритм ветвления в задаче о коммивояжере.
32. Основные понятия и определения метода ветвей и границ. Понятие нижней граничной оценки.
33. Алгоритм вычисления нижних граничных оценок в задаче о коммивояжере.
34. Понятие вторичного штрафа в задаче о коммивояжере. Приведите примеры.
35. Графический метод решения задачи линейного программирования.
36. Построение области допустимых решений в задаче линейного программирования (графический метод).
37. Основные соотношения в постановке транспортной задачи.
38. Задание графа матрицами смежности и инцидентности.
39. Зачем нужна организация упорядоченного представления информации в виде структур данных?
40. Расскажите о базовых структурах данных, их достоинствах и недостатках.
41. Поясните алгоритм поиска в ширину.
42. Поясните алгоритм поиска в глубину.
43. Понятие точного алгоритма. Погрешность в худшем и в лучшем.
44. Алгоритм Прима поиска решений.
45. Алгоритм Дейкстры поиска решений.
46. Поясните основные методы оптимизации.
47. Сведение матричной игры к задаче линейного программирования.
48. Постановка задачи о назначениях. Венгерский алгоритм её решения.
49. Метод отсечений Гомори решения задачи целочисленного программирования.

50. Постановка задачи квадратичного программирования и методы её решения.

51. Охарактеризуйте основные положения динамического программирования.

### **Итоговый тест.**

**Алгоритм – это:**

1. правила выполнения определенных действий
2. набор команд для компьютера
3. протокол вычислительной сети
4. описание последовательности действий, строгое исполнение которых приводит к решению поставленной задачи за конечное число шагов
5. ориентированный граф, указывающий порядок выполнения некоторого набора команд

**Алгоритм называется линейным, если:**

1. он представим в табличной форме
2. он включает в себя вспомогательный алгоритм
3. его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий
4. он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий
5. ход его выполнения зависит от истинности тех или иных условий

**Алгоритм называется циклическим, если:**

1. он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий
2. он представим в табличной форме
3. ход его выполнения зависит от истинности тех или иных условий
4. он включает в себя вспомогательный алгоритм
5. его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий

**Алгоритм включает в себя ветвление, если:**

1. его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий
2. он представим в табличной форме
3. он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий
4. ход его выполнения зависит от истинности тех или иных условий
5. он включает в себя вспомогательный алгоритм

**Свойством алгоритма является:**

1. цикличность
2. простота записи на языках программирования
3. возможность выполнения алгоритма в обратном порядке
4. возможность изменения последовательности выполнения команд
5. результативность

**Свойство алгоритма, заключающиеся в том, что каждое действие и алгоритм в целом должны иметь возможность завершения, называется:**

1. результативность
2. детерминированность
3. массовость
4. дискретность
5. конечность

**Свойство алгоритма, заключающееся в том, что алгоритм должен состоять из конкретных действий, следующих в определенном порядке, называется:**

1. дискретность

2. конечность
3. массовость
4. детерминированность
5. результативность

**Свойство алгоритма, заключающееся в отсутствие ошибок (алгоритм должен приводить к правильному результату при всех допустимых входных значениях), называется:**

1. детерминированность
2. дискретность
3. результативность
4. массовость
5. конечность

**Свойство алгоритма, заключающееся в том, что один и тот же алгоритм можно использовать с разными исходными данными, называется:**

1. результативность
2. детерминированность
3. конечность
4. массовость
5. дискретность

**Свойство алгоритма, заключающееся в том, что любое действие должно быть строго и недвусмысленно определено в каждом случае, называется:**

1. конечность
2. детерминированность
3. результативность
4. массовость
5. дискретность

**Выберите верное представление арифметического выражения  $\frac{x+3y}{5xy}$  на алгоритмическом языке:**

1.  $x + 3y / 5xy$
2.  $(x + 3y) / 5xy$
3.  $x + 3*y / (5*x*y)$
4.  $x + 3*y / 5*x*y$
5.  $(x + 3*y) / (5*x*y)$

**Алгоритм, записанный на «понятном» компьютеру языке программирования, называется:**

1. протоколом алгоритма
2. исполнителем алгоритмов
3. листингом
4. текстовой
5. программой

### Критерии оценки контрольного задания

Критерии	Баллы
Высокий уровень подготовки, безупречное владение материалом, студент демонстрирует творческий подход к решению нестандартных ситуаций. Студент дал полный и развернутый ответ на все теоретические вопросы, подтверждая теоретический материал практическими примерами. Студент активно работал на практических занятиях.	23-25

В целом хорошая подготовка с заметными ошибками или недочетами. Студент дает полный ответ на все вопросы, но имеются неточности в определениях понятий, процессов и т.п. Допускаются ошибки при ответах на вопросы контрольного задания. Студент работал на практических занятиях.	18-22
Минимально достаточный уровень подготовки. Студент показывает минимальный уровень знаний, делает существенные ошибки, но при ответах на вопросы контрольного задания, может правильно сориентироваться и в общих чертах дать правильный ответ. Студент посещал практические занятия.	11-17
Подготовка недостаточная и требует дополнительного изучения материала. Студент дает ошибочные ответы, в том числе на вопросы контрольного задания. Студент пропустил большую часть практических занятий.	10 и менее

Зачтено	11-17 баллов – удовлетворительно, 18-22 баллов – хорошо, 23-25 баллов – отлично
Не зачтено	10 баллов – неудовлетворительно

**Ответ обучающегося на экзамене оценивается по следующей шкале:**

Критерии	Баллы
На теоретические вопросы даны практически полные ответы. Задание выполнено полностью без ошибок, могут присутствовать отдельные недочеты.	от 16 до 60 баллов - зачтено
Получены неполные ответы. Задание не выполнено полностью.	15 и менее баллов - не зачтено

**Ориентировочное распределение максимальных баллов по видам работы**

№ п/п	Вид отчетности	Баллы
1	Оценка качества работы студента в семестре:	<b>0 - 26</b>
	▪ конспект теоретического материала	0 - 4
	▪ решение практических заданий	0 - 12
	▪ прохождение теста по теме	0 - 10
2	Оценка за посещаемость учебных занятий	<b>0 - 14</b>
3	Контрольное задание	<b>0 - 60</b>
<b>ИТОГО:</b>		<b>0 - 100</b>