

Рабочая программа учебной дисциплины разработана на основе Федерального государственного образовательного стандарта (далее ФГОС) по специальности среднего профессионального образования (далее СПО) 09.02.01
Компьютерные системы и комплексы

Организация-разработчик: Тамбовский железнодорожный техникум – филиал РГУПС

Разработчик: Михалина М.Л. преподаватель, первая категория

Рецензенты:

Кривенцова С. А. – преподаватель высшей категории ТаТЖТ- филиала РГУПС

Касатонов И.С. - Проректор по цифровой трансформации федерального государственного бюджетного образовательного учреждения высшего образования «Тамбовский государственный технический университет»

Рекомендована предметной (цикловой) комиссией специальности 09.02.01
Компьютерные системы и комплексы

Протокол № 09 от 16.02.2026 г

Председатель цикловой комиссии



Кривенцова С.А.

СОДЕРЖАНИЕ

1 ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ ОП.06ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ	3
1.1. Область применения программы	3
1.2. Место дисциплины в структуре образовательной программы:	3
1.3. Цель и задачи дисциплины:	3
1.4. Формируемые компетенции:	4
2. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ	5
2.1. Объем дисциплины и виды учебной работы	5
2.2. Тематический план и содержание дисциплины	6
3. УСЛОВИЯ РЕАЛИЗАЦИИ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ..	11
3.1 Требования к минимальному материально-техническому обеспечению.....	11
4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ ...	12
5. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ	13

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ ОП.06 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

1.1. Область применения программы

Рабочая программа дисциплины (далее рабочая программа) является частью основной профессиональной образовательной программы среднего профессионального образования – программы подготовки специалистов среднего звена.

Рабочая программа разработана в соответствии с ФГОС СПО по специальности 09.02.01 Компьютерные системы и комплексы и учебным планом.

1.2. Место дисциплины в структуре образовательной программы:

Дисциплина ОП.06 Основы алгоритмизации и программирования относится к профессиональному учебному циклу, является общепрофессиональной дисциплиной основной образовательной программы.

1.3. Цель и задачи дисциплины:

В результате освоения дисциплины обучающийся должен **уметь:**

- разрабатывать и анализировать алгоритмы для решения поставленных задач;
- определять сложность алгоритмов; реализовывать типовые алгоритмы в виде программ на актуальных языках программирования;
- использовать средства проектирования для создания и графического отображения алгоритмов;
- оформлять код программ в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы

В результате освоения дисциплины обучающийся должен **знать:**

— понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;

классификация языков программирования;

— понятие системы программирования; основные элементы языка, структура программы; методы реализации типовых алгоритмов;

— операторы и операции, управляющие структуры, структуры данных, классы памяти;

— понятие подпрограммы, библиотеки подпрограмм; объектно-ориентированная модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и переопределения.

1.4 Формируемые компетенции:

Код	Формируемые компетенции
ОК 1	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;
ОК 2	Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;
ПК 1.1	Анализировать требования технического задания на проектирование цифровых систем.
ПК 2.1	Проектировать, разрабатывать и отлаживать программный код модулей управляющих программ.
ПК 2.2	Владеть методами командной разработки программных продуктов.

2. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

2.1. Объем дисциплины и виды учебной работы

Вид учебной работы	Объем часов
Максимальная учебная нагрузка (всего)	127
Обязательная аудиторная учебная нагрузка (всего)	99
в том числе:	
практические занятия	55
Самостоятельная работа обучающегося (всего)	10
Консультации	6
<i>Промежуточная аттестация в форме экзамена</i>	

2.2. Тематический план и содержание дисциплины

Наименование разделов и тем	Содержание учебного материала и формы организации деятельности обучающихся	Объем, ак. ч /в том числе в форме практической подготовки, ак. ч	Коды компетенций и личностных результатов ²³ , формированию которых способствует элемент программы
<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>
Раздел 1. Основы алгоритмизации		26/16	
Тема 1.1. Понятие алгоритма и его свойства	Содержание учебного материала	8/4	ОК 01 ОК 02
	1. Понятие алгоритма. Свойства и виды алгоритмов	4	ПК 1.1 ПК 2.1 ПК 2.2
	2. Способы описания алгоритмов: псевдокоды. Блок-схема: основные элементы, правила составления. Стандарты графического оформления алгоритмов.		
	3. Базовые алгоритмические конструкции: линейная, разветвляющаяся, циклическая. Критерии «хорошего» алгоритма.		
	В том числе практических занятий	4	
	Практическое занятие №1. Составление и оформление блок-схем простых алгоритмов.	4	
	Самостоятельная работа обучающихся		
Тема 1.2. Методы	Содержание учебного материала	10/6	ОК 01 ОК 02

разработки алгоритмов	1. Основные методы и этапы проектирования алгоритмов: постановка задачи, математическое описание – математическая модель. Нисходящее, модульное и восходящее проектирование.	4	ПК 1.1 ПК 2.1 ПК 2.2
	2. Эффективность и сложность алгоритма, их практическая значимость.		
	3. Алгоритмы поиска. Алгоритмы сортировки. Вложенные циклы. Вспомогательные алгоритмы.		
	4. Различные комбинации алгоритмических конструкций. Тестовые данные. Алгоритм Евклида. Алгоритмы решения нелинейных и линейных уравнений. Декомпозиция алгоритма.		
	В том числе практических занятий	6	
	Практическое занятие № 2. Проектирование и оформление алгоритмов сортировки.		
	Самостоятельная работа обучающихся		
Раздел 2. Основы программирования		33/21	
Тема 2.1. Базовые понятия программирования	Содержание учебного материала	10/8	ОК 01 ОК 02 ПК 1.1 ПК 2.1 ПК 2.2
	1. Классификация и генеалогия актуальных языков программирования. Понятие системы программирования.		
	2. Основные элементы языка. Структура типовой программы. Особенности актуальных сред программирования	2	
	В том числе практических занятий	8	
	Практическое занятие № 3. Изучение инструментария среды программирования	4	

	Практическое занятие № 4. Подготовка структуры программы в среде программирования	4	
	Самостоятельная работа обучающихся		
Тема 2.2. Программная реализация алгоритмов	Содержание учебного материала	23/13	ОК 01
	1. Методы реализации типовых алгоритмов. Переменные: определение, правила именования. Типы данных: значимые и ссылочные. Объявление и инициализация переменных. Область действия и время существования переменных. Константы: определение, виды и правила записи в программе.	10	ОК 02 ПК 1.1 ПК 2.1 ПК 2.2
	2. Операторы и операции. Понятие выражения. Математические операторы. Старшинство операторов. Математические функции (класс Math). Ввод – вывод данных. Операторы присваивания.		
	3. Операторы отношения. Проверка простых и сложных условий. Вложенные условные операторы. Оператор выбора. Операторы перехода.		
	4. Операторы цикла. Стандартные операции при работе с циклическими алгоритмами. Принудительный выход из цикла.		
	5. Массивы: определение, виды. Объявление одномерного массива. Варианты инициализации. Ввод и вывод одномерных массивов. Стандартные операции для работы с массивами. Обработка одномерных и двумерных массивов.		
	6. Управляющие структуры. Понятие потока. Механизм буферизации. Классы памяти. Доступ к файлам.		
	7. Понятие подпрограммы, библиотеки подпрограмм. Библиотеки среды разработки.		
	В том числе практических занятий		
Практическое занятие № 7. Реализация простых циклических алгоритмов.	4		

	Практическое занятие № 10. Реализация алгоритмов обработки текстовых данных.	4	
	Практическое занятие № 11. Реализация сложных алгоритмов поиска и ввода-вывода.	5	
	Самостоятельная работа обучающихся		
Раздел 3. Основы объектно-ориентированного программирования		40/16	
Тема 3.1. Основные понятия объектно-ориентированного программирования	Содержание учебного материала	26/12	ОК 01 ОК 02 ПК 1.1 ПК 2.1 ПК 2.2
	1. Понятие класса и объекта. Характеристики объекта: поля, свойства, методы, события. Основные принципы объектно-ориентированного программирования: наследование, полиморфизм, инкапсуляция.	14	
	2. Общая форма определения класса.		
	3. Метод: понятие, правила записи. Правило триединого соответствия параметров и аргументов: по количеству, типам и по порядку следования.		
	4. Инкапсуляция как управление доступом к данным. Свойства класса: понятие, виды, правила записи. Наследование и полиморфизм.		
	5. Иерархия классов: понятие, преимущества.		
	6. Интерфейсы: назначение, правила написания.		
	В том числе практических занятий	12	
	Практическое занятие № 12. Создание простейших классов.	6	
	Практическое занятие № 13. Создание классов, иерархически связанных между собой	6	

	Самостоятельная работа обучающихся		
Тема 3.2. Реализация методов объектно- ориентированного программирования	Содержание учебного материала	22/12	ОК 01 ОК 02 ПК 1.1 ПК 2.1 ПК 2.2
	1. Модификаторы доступа к элементам класса. Переменные ссылочного типа и присваивание. Побочные эффекты множественных ссылок. 2. Методы классов. Вызов метода. Передача параметров по значению. Создание методов, возвращающих значения. Способы размещения методов. Конструкторы. 3. Синтаксис наследования. Скрытие и перекрытие методов. 4. Способы реализации интерфейсов. Работа с объектами через интерфейсы. 5. Обработка события: автоматическое создание обработчиков.	10	
	В том числе практических занятий	12	
	Практическое занятие № 14. Создание классов для обработки массива данных. Практическое занятие № 15. Создание классов для вычисления математических выражений Практическое занятие № 16. Разработка проектов с обработкой событий	12	
Промежуточная аттестация в форме экзамена			
Всего:		99	

Для характеристики уровня освоения учебного материала используются следующие обозначения:

- 1 – ознакомительный (узнавание ранее изученных объектов, свойств);
- 2 – репродуктивный (выполнение деятельности по образцу, инструкции или под руководством);
- 3 – продуктивный (планирование и самостоятельное выполнение деятельности, решение проблемных задач).

3. УСЛОВИЯ РЕАЛИЗАЦИИ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ

3.1 Требования к минимальному материально-техническому обеспечению

Учебно административный корпус, №412

Лаборатория программирование

Перечень основного оборудования Лаборатории программирование

1 Стол компьютерный СК -03 – 14 шт

Стол преподавателя -1 шт

Стул ученический 16 шт

Доска аудиторная 1 шт

Компьютер Pentium -13 шт

Плазменный телевизор Samsung – 1шт

Система программирования Visual C++2005 1 шт

Тематические плакаты 5 шт.

Учебно методический комплекс

3.1 Информационное обеспечение обучения

Основная:

1. Трофимов, В.В. Основы алгоритмизации и программирования [Электронный ресурс]: учебник для СПО /В.В. Трофимов, Т.А. Павловская; под ред. В.В. Трофимова. – 4-е изд. - М.: Издательство Юрайт, 2025. — 108 с. — (Профессиональное образование). — Режим доступа: <https://urait.ru/bcode/>

2. Кудрина, Е.В. Основы алгоритмизации и программирования на языке C# [Электронный ресурс]: учебное пособие для СПО/ Е.В. Кудрина, М.В. Огнева — М.: Издательство Юрайт, 2025. — 322 с. — (Профессиональное образование). — Режим доступа: <https://urait.ru/bcode>

Дополнительная:

1.Паронджанов, В. Д. Алгоритмические языки и программирование: ДРАКОН [Электронный ресурс]: учебное пособие для СПО /В. Д. Паронджанов. — М.: Издательство Юрайт, 2025. — 436 с. — (Профессиональное образование).

2.Огнева, М.В. Программирование на языке C++: практический курс [Электронный ресурс]: учебное пособие для СПО / М.В. Огнева, Е.В. Кудрина, А. А. Казачкова. — 2-е изд., перераб. и доп. - М.: Издательство Юрайт, 2025. - 342 с. - Режим доступа: <https://urait.ru/bcode/>доступа: <https://urait.ru/bcode/>

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Результаты обучения (освоенные умения, усвоенные знания)	Формы и методы контроля и оценки результатов обучения
<p>Знать: понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции; классификация языков программирования; понятие системы программирования; основные элементы языка, структура программы; методы реализации типовых алгоритмов; операторы и операции, управляющие структуры, структуры данных, классы памяти; понятие подпрограммы, библиотеки подпрограмм; объектно-ориентированная модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и переопределения.</p>	<p>Тестирование</p> <p>Экспертное наблюдение за ходом выполнения практической работы</p>
<p>Уметь: разрабатывать и анализировать алгоритмы для решения поставленных задач; определять сложность алгоритмов; реализовывать типовые алгоритмы в виде программ на актуальных языках программирования; использовать средства проектирования для создания и графического отображения алгоритмов; оформлять код программ в соответствии со стандартом кодирования; выполнять проверку, отладку кода программы</p>	<p>Оценка результатов выполнения практических работ.</p> <p>Экспертное наблюдение за ходом выполнения практической работы</p>

5. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Результатом освоения программы дисциплины **ОП.06 Основы алгоритмизации и программирования** является овладение обучающимися профессиональными (ПК) и общими (ОК) компетенциями:

Код	Формируемые компетенции
ОК 1	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;
ОК 2	Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;
ПК 1.1	Анализировать требования технического задания на проектирование цифровых систем.
ПК 2.1	Проектировать, разрабатывать и отлаживать программный код модулей управляющих программ.
ПК 2.2	Владеть методами командной разработки программных продуктов.