

Министерство промышленности и торговли Тверской области
Государственное бюджетное профессиональное образовательное учреждение
«Тверской химико-технологический колледж»

РАССМОТРЕНО
цикловой комиссией
информационных технологий
и технических дисциплин
Протокол № ____
от «____» _____ 2025 г.
Председатель ЦК
_____ К.В. Мальцев

ПРИНЯТО
Методическим советом

Протокол № ____
от «____» _____ 2025 г.

Зам. директора по УР
_____ Е.А. Гусева

**РАБОЧАЯ ПРОГРАММА
УЧЕБНОЙ ДИСЦИПЛИНЫ**

ОП.06. Основы алгоритмизации и программирования

Для специальности:
09.02.01

Разработчик:
Пирогова А.А.,
преподаватель

Тверь
2025

Содержание

1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ	3
2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ	4
3. УСЛОВИЯ РЕАЛИЗАЦИИ РАБОЧЕЙ ПРОГРАММЫ	8
4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ	10

1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ

1.1 Место учебной дисциплины в структуре ППССЗ

Рабочая программа учебной дисциплины (далее – программа) предназначена для реализации программы подготовки специалистов среднего звена (далее – ППССЗ) по специальности 09.02.01 Компьютерные системы и комплексы.

Учебная дисциплина (далее – УД) является частью общепрофессионального цикла ППССЗ.

Нормативная основа разработки программы:

- Федеральный государственный образовательный стандарт среднего профессионального образования (далее – ФГОС СПО) (утв. приказом Министерства просвещения Российской Федерации № 362 от 25.05.2022);

- Примерная рабочая программа учебной дисциплины, включенная в состав ПОП по специальности 09.02.01 Компьютерные системы и комплексы.

1.2 Цели и планируемые результаты освоения учебной дисциплины

Код ПК, ОК	Умения	Знания
ОК 01 ОК 02 ПК 1.1 ПК 2.1 ПК 2.2 ЛР 4 ЛР 10 ЛР 13 ЛР 14 ЛР 15	- 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.	- 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.

2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1 Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объем в часах
Объем образовательной программы УД	114
в т.ч.:	
теоретическое обучение	24
практические занятия и лабораторные работы	78
курсовая работа (проект)	0
самостоятельная работа	12
промежуточная аттестация (экзамен)	

2.2 Тематический план и содержание учебной дисциплины

Наименование разделов и тем	Содержание учебного материала и формы организации деятельности обучающихся	Объем часов	Формируемые компетенции
Раздел 1. Программирование и языки программирования.		2	
	Алгоритм и программа. Типы алгоритмов. Языки программирования. Язык программирования Python. Понятие алгоритма. Свойства и виды алгоритмов. Способы описания алгоритмов: псевдокоды. Блок-схема: основные элементы, правила составления. Стандарты графического оформления алгоритмов. Базовые алгоритмические конструкции: линейная, разветвляющаяся, циклическая. Критерии «хорошего» алгоритма. Составление и оформление блок-схем простых алгоритмов. Основные методы и этапы проектирования алгоритмов: постановка задачи, математическое описание – математическая модель. Нисходящее, модульное и восходящее проектирование. Эффективность и сложность алгоритма, их практическая значимость. Алгоритмы поиска. Алгоритмы сортировки. Вложенные циклы. Вспомогательные алгоритмы. Различные комбинации алгоритмических конструкций. Тестовые данные. Алгоритм Евклида. Алгоритмы решения нелинейных и линейных уравнений. Декомпозиция алгоритма. Проектирование и оформление различных алгоритмов. Классификация и генеалогия актуальных языков программирования. Понятие системы программирования. Основные элементы языка. Структура типовой программы. Особенности актуальных сред программирования. Изучение инструментария среды программирования. Подготовка структуры программы в среде программирования.	2	ОК 01-02 ПК 1.1, 2.1, 2.2 ЛР 4, 10, 13-15
Раздел 2. Вычисления.		10	
	Переменные и их типы. Основные конструкции языка. Математические вычисления. Обработка исключительных ситуаций. 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. Операторы и операции. Понятие выражения. Математические операторы. Старшинство операторов. Математические функции (класс Math). Ввод – вывод данных. Операторы присваивания. Операторы отношения. Проверка простых и сложных условий. Вложенные условные операторы. Оператор выбора. Операторы перехода. Операторы цикла. Стандартные операции при работе с циклическими алгоритмами. Принудительный выход из цикла.	4	ОК 01-02 ПК 1.1, 2.1, 2.2 ЛР 4, 10, 13-15
	Функции. Форматный вывод. ПР № 1: Расчет треугольника.	6	
Раздел 3. Обработка табличных данных и их визуализация.		26	
	Списки. Построение графиков и диаграмм. Реализация простых циклических алгоритмов. Рисование плоских фигур.	4	ОК 01-02 ПК 1.1, 2.1, 2.2

	К теме «Списки»: ПР № 2: Табличный расчет функции (6 ч). К теме «Построение графиков и диаграмм»: ПР № 3: Построение графиков функций (8 ч). К теме «Рисование плоских фигур»: ПР № 4: Рисование изображений (8 ч).	22	ЛР 4, 10, 13-15
Раздел 4. Матрицы и векторы.		24	
	Массивы. Массивы: определение, виды. Объявление одномерного массива. Варианты инициализации. Ввод и вывод одномерных массивов. Стандартные операции для работы с массивами. Обработка одномерных массивов. Многомерные массивы. Тренды. Обработка двумерных массивов.	4	ОК 01-02 ПК 1.1, 2.1, 2.2 ЛР 4, 10, 13-15
	К темам «Массивы», «Многомерные массивы»: ПР № 5: Матричные вычисления (6 ч). Реализация алгоритмов обработки одномерных массивов. ПР № 6: Решение системы линейных уравнений (6 ч). Реализация алгоритмов обработки двумерных массивов. К теме «Тренды»: ПР № 7: Решение прикладных задач (8 ч).	20	
Раздел 5. Работа с файлами.		8	
	Работа с текстовыми файлами. Чтение содержимого файла. Запись в файл. Управляющие структуры. Понятие потока. Механизм буферизации. Классы памяти. Доступ к файлам.	2	ОК 01-02 ПК 1.1, 2.1, 2.2 ЛР 4, 10, 13-15
	ПР № 8: Работа с текстовыми файлами. Реализация алгоритмов обработки текстовых данных. Реализация сложных алгоритмов поиска и ввода-вывода.	6	
Раздел 6. Объектно-ориентированное программирование.		10	
	Классы. Понятие класса и объекта. Характеристики объекта: поля, свойства, методы, события. Основные принципы объектно-ориентированного программирования: наследование, полиморфизм, инкапсуляция. Общая форма определения класса. Иерархия классов: понятие, преимущества. Модификаторы доступа к элементам класса. Переменные ссылочного типа и присваивание. Побочные эффекты множественных ссылок. Методы. Наследование. Метод: понятие, правила записи. Правило триединого соответствия параметров и аргументов: по количеству, типам и по порядку следования. Методы классов. Вызов метода. Передача параметров по значению. Создание методов, возвращающих значения. Способы размещения методов. Конструкторы. Инкапсуляция как управление доступом к данным. Свойства класса: понятие, виды, правила записи. Наследование и полиморфизм. Синтаксис наследования. Скрытие и перекрытие методов.	4	ОК 01-02 ПК 1.1, 2.1, 2.2 ЛР 4, 10, 13-15

	ПР № 9: Работа с классами, объектами и методами. Создание простейших классов. Создание классов, иерархически связанных между собой. Создание классов для обработки массива данных. Создание классов для вычисления математических выражений.	6	
Раздел 7. Приложения с графическим интерфейсом.		22	
	Графическая библиотека. Окно приложения. Понятие подпрограммы, библиотеки подпрограмм. Библиотеки среды разработки. Интерфейсы: назначение, правила написания. Способы реализации интерфейсов. Работа с объектами через интерфейсы. Обработка события: автоматическое создание обработчиков. Виджеты.	4	ОК 01-02 ПК 1.1, 2.1, 2.2 ЛР 4, 10, 13-15
	ПР № 10: Создание кросс-платформенного приложения (8 ч). Разработка проектов с обработкой событий. ПР № 11: Создание приложения с базой данных (10 ч).	18	
Самостоятельная работа.	Подготовка к практическим занятиям и промежуточной аттестации.	12	ОК 01-02 ПК 1.1, 2.1, 2.2 ЛР 4, 10, 13-15
Промежуточная аттестация (экзамен).			
ВСЕГО:		114	

3. УСЛОВИЯ РЕАЛИЗАЦИИ РАБОЧЕЙ ПРОГРАММЫ

3.1 Требования к материально-техническому обеспечению

Реализация программы требует наличия лаборатории прикладного программирования.

Оборудование лаборатории:

- многофункциональный комплекс преподавателя;
- рабочие места обучающихся, оборудованные персональными компьютерами;
- программное обеспечение общего и профессионального назначения;
- наглядные пособия;
- информационно-коммуникативные средства;
- экранно-звуковые пособия;
- комплект технической документации, в том числе паспорта на средства обучения, инструкции по их использованию и технике безопасности;
- библиотечный фонд;
- доступ к электронным учебным материалам по учебной дисциплине, имеющимся в свободном доступе в Интернете (электронным книгам, практикумам, тестам и др.).

3.2 Информационное обеспечение обучения

1. Колдаев В.Д. Основы алгоритмизации и программирования: учебное пособие / В.Д. Колдаев; под ред. проф. Л.Г. Гагариной. – М.: ИНФРА-М, 2025. – 414 с.

2. Голицына О.Л. Основы алгоритмизации и программирования: учебное пособие / О.Л. Голицына, И.И. Попов. – 4-е изд., испр. и доп. – М.: ИНФРА-М, 2025. – 431 с.
3. Гуриков С.Р. Основы алгоритмизации и программирования на Python: учебное пособие / С.Р. Гуриков. – М.: ИНФРА-М, 2025. – 343 с.

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Результаты обучения	Критерии оценки	Тип оценочных мероприятий
Умения: - 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.	«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко. «Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками. «Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки. «Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.	Оценка результатов выполнения практических заданий Экзамен
Знания: - 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.		Электронное тестирование Оценка результатов выполнения практических заданий Экзамен