

государственное автономное профессиональное образовательное учреждение
Самарской области
«Новокуйбышевский гуманитарно-технологический колледж»

УТВЕРЖДЕНО
Директор ГАПОУ «НГТК»
В.М. Земалиндинова
Приказ от 06.02.2025 г. № 26-К

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

ОП.04 Основы алгоритмизации и программирования

для специальности:

09.02.07 Информационные системы и программирование

профиль подготовки: технический

на базе основного общего образования

РАССМОТРЕНО НА ЗАСЕДАНИИ

Предметно-цикловой комиссии
математики, информационных
технологий и программирования
И.Г. Фролова

ОДОБРЕНО

Заместитель директора
Н.П. Свириденко

Рабочая программа учебной дисциплины ОП.04 Основы алгоритмизации и программирования составлена в соответствии с Федеральным государственным образовательным стандартом по специальности среднего профессионального образования 09.02.07 Информационные системы и программирование, утвержденного приказом Министерства образования и науки Российской Федерации № 1547 от 09.12.2016 г., зарегистрированного Министерством юстиции России № 44936 от 26 декабря 2016 г.

Разработчик:

ГАПОУ «НГТК»
(место работы)

преподаватель
(занимаемая должность)

Нагорный А.В.
(инициалы, фамилия)

СОДЕРЖАНИЕ

1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ.....	4
2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ.....	5
3. УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДИСЦИПЛИНЫ.....	12
4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ.....	14

1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

ОП.04 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

1.1. Место дисциплины в структуре основной образовательной программы

Учебная дисциплина «Основы алгоритмизации и программирования» принадлежит к общепрофессиональному циклу.

1.2. Цель и планируемые результаты освоения дисциплины

В рамках программы учебной дисциплины обучающимися осваиваются умения и знания:

Код ПК, ОК	Умения	Знания
ОК 01 ОК 02 ОК 03 ОК 04 ОК 05 ОК 09 ПК 2.4 ПК 2.5	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.	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.

2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объем в часах
Объем образовательной программы учебной дисциплины	156
в т.ч. в форме практической подготовки	156
в т.ч.:	
теоретическое обучение	56
практические занятия	100
Дифференцированный зачет	2

2.2. Тематический план и содержание учебной дисциплины

Наименование тем учебной дисциплины	Содержание учебного материала, практические занятия, самостоятельная работа обучающихся, курсовая работа (проект)	Объем часов	Код образовательного результата ФГОС СПО
1	2	3	4
Тема 1. Введение в алгоритмизацию и программирование на языке C#	Содержание	8	
	1. История развития языков программирования		
	2. Основные понятия и термины		
	3. Языки программирования и их классификация		
	4. Преимущества и особенности языка C#		
Тема 2. Изучение синтаксиса и основных конструкций языка C#	Содержание	28	ПК 2.4 ПК 2.5
	1. Переменные, типы данных и операции		
	2. Условные операторы и циклы		
	3. Массивы и коллекции		
	Практическое занятие 1,2,3,4,5,6,7,8,9,10,11	22	ОК 1 ОК 2 ОК 4 ОК 5 ОК 9
	1. Простые вычисления: сложение, вычитание, умножение и деление чисел		
	2. Решение квадратного уравнения		
	3. Обработка одномерного массива: поиск максимального и минимального элемента, сортировка массива		
	4. Обработка двумерного массива: заполнение случайными числами, нахождение суммы элементов		
	5. Создание простого калькулятора с функциями сложения, вычитания, умножения и деления		
	6. Создание простой программы для решения линейного уравнения с одной переменной		

	7.	Программирование простого текстового редактора с функциями создания, открытия, сохранения и редактирования текста		
	8.	Создание простого графического приложения, выводящего на экран геометрические фигуры (например, окружность, прямоугольник, линию)		
	9.	Программирование калькулятора для решения квадратных уравнений		
	10.	Создание программы для решения системы линейных уравнений методом Гаусса		
	11.	Написание программы для работы с матрицами: сложение, умножение на число, транспонирование		
Тема 3. Основы алгоритмики	Содержание		20	
	1.	Алгоритмы и их свойства		
	2.	Классификация алгоритмов		
	Практические занятия 12,13,14,15,16,17,18,19		16	
	1.	Разработка алгоритма нахождения суммы элементов массива		
	2.	Разработка алгоритма сортировки массива методом пузырька		
	3.	Разработка алгоритма поиска максимального элемента в массиве		
	4.	Разработка алгоритма решения квадратного уравнения		
	5.	Разработка алгоритма бинарного поиска элемента в отсортированном массиве		
	6.	Разработка алгоритма быстрой сортировки массива		
7.	Разработка алгоритма подсчета количества элементов в массиве, удовлетворяющих некоторому условию			
8.	Разработка алгоритма, который находит все простые числа в заданном диапазоне			

Тема 4. Объектно-ориентированное программирование в C#	Содержание		36	
	1.	Классы и объекты в C#		
	2.	Инкапсуляция: основные принципы и применение на практике		
	3.	Свойства и методы классов: определение, использование и модификаторы доступа		
	4.	Конструкторы и деструкторы классов: назначение, использование и параметры		
	5.	Статические элементы классов: свойства, методы и поля		
	6.	Наследование: определение и принципы работы		
	7.	Полиморфизм: основные понятия и использование на примерах		
	8.	Абстрактные классы и интерфейсы: назначение и использование		
	9.	Вложенные и внутренние классы: использование и особенности		
	Практические занятия 20,21,22,23,24,25,26,27,28		18	
	1.	Создание класса «Автомобиль» с определенными методами		
	2.	Создание класса «Велосипед» с определенными методами		
	3.	Создание класса «Кошка» с определенными методами		
	4.	Создание абстрактного класса «Животное» с определенными методами		
5.	Создание классов «Книга» и «Журнал» с определенными методами			
6.	Создание интерфейса «ICalculator» и классов, с различными реализациями методов			
7.	Создание базового класса «Shape» с определенными методами и производными от него классами			

	8.	Реализация паттерна «Factory Method» с абстрактными и конкретными классами, каждый из которых имеет свой метод		
	9.	Реализация паттернов «Strategy» и «State» с абстрактными и конкретными классами и состояниями		
Тема 5. Работа с файлами и потоками ввода-вывода	Содержание		20	
	1.	Чтение и запись данных		
	2.	Работа с ошибками ввода-вывода		
	Практические занятия 29,30,31,32,33,34,35,36		16	
	1.	Открытие и закрытие файла. Чтение и запись данных в файл. Создание и удаление файлов		
	2.	Простейшие операции с каталогами (создание, удаление, переименование)		
	3.	Использование потоков ввода-вывода для чтения и записи данных		
	4.	Работа с ошибками ввода-вывода, обработка исключений		
	5.	Использование буферизованного и небуферизованного ввода-вывода		
	6.	Использование асинхронного ввода-вывода и многопоточности		
	7.	Сериализация и десериализация данных с использованием XML, JSON, бинарных форматов		
	8.	Работа с ZIP-архивами		
Тема 6. Исключения и обработка ошибок	Содержание		14	
	1.	Исключения в C#		
	2.	Обработка исключений		
	Практические занятия 37,38,39,40,41		10	
	1.	Использование ключевых слов throw и try-catch-finally для обработки исключений		

	2.	Использование блоков try-catch для обработки ошибок времени выполнения		
	3.	Создание пользовательских исключений и их обработка		
	4.	Использование исключений для управления потоком выполнения программы		
	5.	Отладка исключений с помощью инструментов Visual Studio		
Тема 7. Разработка программ и проектов	Содержание		28	
	1.	Создание проектов в Visual Studio		
	2.	Использование NuGet для установки и управления пакетами и библиотеками		
	3.	Использование основных стандартных библиотек .NET		
	4.	Использование сторонних библиотек и пакетов		
	5.	Создание многофайловых проектов с использованием папок и подпапок		
	Практические занятия 42,43,44,45,46,47,48,49,50		18	
	1.	Разработка приложения Windows Forms на C# и реализация работы с формами, элементами управления и событиями		
	2.	Разработка программы, которая использует классы, структуры, интерфейсы и перечисления в C#		
	3.	Разработка программы на C#, которая демонстрирует использование обобщений		
	4.	Разработка программы на C# для работы со строками с использованием регулярных выражений		
	5.	Разработка консольного приложения на C# и использовать в нем многопоточность с использованием классов Thread и Tasks		
6.	Разработка приложения Windows Forms на C# и реализовать в нем работу с формами, элементами управления и событиями			

	7.	Разработка веб-приложения ASP.NET Core и использовать в нем MVC, Razor Pages, Entity Framework Core и SQL Server		
	8.	Разработка индивидуального проекта на языке программирования C#		
	9.	Разработка индивидуального проекта на языке программирования C#		
Дифференцированный зачет			2	
Всего			156	

3. УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДИСЦИПЛИНЫ

3.1. Требования к минимальному материально-техническому обеспечению

Реализация учебной дисциплины требует наличия учебного кабинета и лаборатории.

Оборудование учебного кабинета: стандартное оборудование рабочих мест преподавателя и студента.

Технические средства обучения: персональный компьютер, мультимедиа проектор, экран.

Оборудование лаборатории и рабочих мест лаборатории:

Персональный компьютер и периферийные устройства для проведения практических работ, телекоммуникационные средства.

3.2. Информационное обеспечение реализации программы

Для реализации программы библиотечный фонд образовательной организации должен иметь печатные и/или электронные образовательные и информационные ресурсы для использования в образовательном процессе. При формировании библиотечного фонда образовательной организацией выбирается не менее одного издания из перечисленных ниже печатных изданий и (или) электронных изданий в качестве основного, при этом список, может быть дополнен новыми изданиями.

3.2.1. Основные печатные издания

1. Зыков, С. В. Программирование: учебник и практикум для академического бакалавриата / С. В. Зыков. — 2-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2023.

2. Казанский, А. А. Программирование на Visual C#: учебное пособие для среднего профессионального образования / А. А. Казанский. — 2-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2023.

3. 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, 2023.

4. Подбельский, В. В. Программирование. Базовый курс C#: учебник для среднего профессионального образования / В. В. Подбельский. — Москва: Издательство Юрайт, 2023.

3.2.2. Дополнительные источники

1. Программирование: математическая логика: учебное пособие для вузов / М. В. Швецкий, М. В. Демидов, А. В. Голанова, И. А. Кудрявцева. — 2-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2023.
2. Трофимов, В. В. Алгоритмизация и программирование : учебник для вузов / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. — 4-е изд. — Москва: Издательство Юрайт, 2023.
3. Черпаков, И. В. Основы программирования: учебник и практикум для среднего профессионального образования / И. В. Черпаков. — Москва: Издательство Юрайт, 2023.
4. Черпаков, И. В. Основы программирования: учебник и практикум для вузов / И. В. Черпаков. — Москва: Издательство Юрайт, 2023.

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Результаты обучения (освоенные умения, усвоенные знания)	Критерии оценки	Методы оценки
<p>Уметь</p> <ul style="list-style-type: none"> – 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. 	<p>Оценка «отлично» выставляется обучающемуся, если он глубоко и прочно усвоил программный материал курса, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами и вопросами, не затрудняется с ответами при видоизменении заданий, правильно обосновывает принятые решения, владеет разносторонними дискуссионными навыками и приемами, активно проявляет себя в групповой работе;</p> <p>Оценка «хорошо» выставляется обучающемуся, если он твердо знает материал курса, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении дискуссионных вопросов и задач, владеет необходимыми навыками</p>	<p>Текущий контроль: - опрос устный; - тестирование; - выполнение письменной работы; - выполнение практической работы</p> <p>Оценка результатов выполнения самостоятельной работы.</p> <p>Дифференцированный зачет</p>
<p>Знать</p> <ul style="list-style-type: none"> – понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические 		

<p>конструкции;</p> <ul style="list-style-type: none"> – 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. 	<p>и приемами их выполнения, способен проявлять себя в групповой работе;</p> <p>Оценка «удовлетворительно» выставляется обучающемуся, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических задач, не активен в групповой работе;</p> <p>Оценка «неудовлетворительно» выставляется обучающемуся, который не знает значительной части программного материала, допускает существенные ошибки, неуверенно, с большими затруднениями решает практические задачи или не справляется с ними самостоятельно, не принимает участие в групповой работе.</p>	
---	--	--