

Государственное бюджетное профессиональное образовательное учреждение
«Кунгурский автотранспортный колледж»

Рабочая программа учебной дисциплины
ОП.04 Основы алгоритмизации и программирования

09.02.07 Информационные системы и программирование


Одобрено на заседании
комиссии информационно-математических
дисциплин
Протокол № 1 от «27» августа 2019г.

Председатель комиссии:
Наговицын /И. Г. Наговицын

УТВЕРЖДАЮ:

Заместитель директора

Целишева М. Г. Целишева



Составитель: О.В.Архипова, преподаватель ГБПОУ КАТК

СОДЕРЖАНИЕ

1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ	4
2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ	5
3. УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДИСЦИПЛИНЫ	11
4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ	12

1.ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

1.1 Место учебной дисциплины в структуре основной профессиональной образовательной программы

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

1.2Цели и задачи учебной дисциплины – требования к результатам освоения учебной дисциплины:

Обязательная часть

Код ОК	ПК,	Умения	Знания
ОК 1 ОК 2 ОК 4 ОК 5 ОК 9 ОК 10 ПК 1.1- ПК 1.5 ПК 2.4, 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

Вариативная часть вводится с целью расширения и углубления подготовки определяемой содержанием обязательной части по ОК 4 , ОК 5 , ОК 9 , ОК 10 , ПК 1.1- ПК 1.5 , 2.5

2 СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1 Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объем часов
Суммарная учебная нагрузка во взаимодействии с преподавателем	196
<i>Самостоятельная работа</i>	-
Объем образовательной программы <i>(вариативная часть 44 часа)</i>	184
в том числе:	
теоретическое обучение	86
практические занятия	98
<i>Самостоятельная работа</i>	-
<i>Консультации</i>	6
<i>Промежуточная аттестация в IV семестре в форме экзамена</i>	6

2.2 Тематический план и содержание учебной дисциплины: ОП. 04 Основы алгоритмизации и программирования

Наименование разделов и тем	Содержание учебного материала и формы организации деятельности обучающихся	Объем часов	Коды компетенций, формированию которых способствует элемент программы
1	2	3	4
Введение		12	
	Содержание учебного материала		
	1 Роль и место знаний по дисциплине «Основы алгоритмизации и программирования» в сфере профессиональной деятельности.	4	
	2 Эволюция языков программирования, их классификация.		
Раздел 1. Понятие алгоритмизации		18	
Тема 1.1 Структурное программирование. Общие принципы построения алгоритмов	Содержание учебного материала		
	1 Структурное программирование: основные принципы.		
	2 Основные алгоритмические конструкции: последовательность, ветвление, циклы.	4	ОК 1 ОК 2 ОК 4 ОК 5 ОК 9 ОК 10 ПК 1.1- ПК 1.5 ПК 2.4, 2.5
Тема 1.2 Методы разработки алгоритмов	Содержание учебного материала		
	1 Методы разработки алгоритма. Постановка задачи. Построение модели. Разработка алгоритма, проверка его правильности. Реализация алгоритма. Анализ алгоритма и его сложности.	4	
	2 Составление алгоритмов на разветвляющиеся и циклические процессы счета.		
	Практические занятия Разработать блок-схему алгоритма решения задач.	4	
Раздел 2. Программирование на алгоритмическом языке		180	
Тема 2.1 Элементы языка	Содержание учебного материала		
	1 Основные понятия языка: алфавит, константы и идентификаторы.	4	
	2 Структура программ на языке Pascal.		

1	2	3	4
Тема 2.2. Типы данных. Операции и выражения	Содержание учебного материала	8	ОК 1 ОК 2 ОК 4 ОК 5 ОК 9 ОК 10 ПК 1.1- ПК 1.5 ПК 2.4, 2.5
	1 Типы данных. Описание типов, переменных, констант.		
	2 Операции.		
	3 Выражения. Совместимость типов в выражениях, совместимость по присваиванию.		
	Практические занятия Составление программ линейной структуры	4	
Тема 2.3. Операторы языка. Управляющие структуры	Содержание учебного материала	6	
	1 Операторы присваивания, составной оператор.		
	2 Управляющие структуры: условные операторы, операторы цикла, перехода.		
	3 Процедуры ввода с клавиатуры и вывода на дисплей.		
	Практические занятия Составление программ разветвляющейся структуры. Составление программ разветвляющейся усложненной структуры. Составление программ циклической структуры. Составление программ усложненной структуры. Составление программ с процедурами ввода с клавиатуры. Составление программ с процедурами вывода на дисплей.	16	
Тема 2.4. Подпрограммы: процедуры и функции	Содержание учебного материала	8	
	1 Процедуры, их назначение и описание.		
	2 Списки формальных и фактических параметров, их назначение и взаимосвязь. Особенности применения процедур и функций. Параметры процедурного типа.		
	3 Функции, их назначение и описание. Особенности применения процедур и функций.		
	4 Параметры процедурного типа.		
	Практические занятия Ввод и отладка программы с использованием процедур. Ввод и отладка программы с использованием функций.	4	
Тема 2.5. Графика и анимация	Содержание учебного материала	8	
	1 Основные понятия и средства компьютерной графики. Модуль Graph.		
	2 Построение примитивных фигур: точка, линия, прямоугольник, круг и т.д. Работа с цветом и палитрой. Закраска примитивных фигур.		
	3 Основы и принципы анимации. Способ перерисовки движущихся объектов на экране.		
	4 Спрайты. Анимация с помощью спрайтов.		

	Практические занятия Процедуры работы с геометрическими объектами. Построение сложных геометрических объектов. Работа со спрайтами. Анимация движущихся объектов.	6	
	Самостоятельная работа №7. Реферат	6	
Тема 2.6. Сложные структуры: массивы и записи	Содержание учебного материала	8	
	1 Регулярные типы данных: одномерные и многомерные массивы.		
	2 Объявление массивов, доступ к элементам массива, организация работы с массивами.		
	3 Структурированные типы - записи. Объявление, доступ к элементам записи.		
	4 Организация работы с записями.		
	Практические занятия Ввод и отладка программы по обработке одномерного массива. Ввод и отладка программы по обработке двумерного массива. Ввод и отладка программы по работе с записями.	10	
Тема 2.7. Строки и множества	Содержание учебного материала		
	1 Объявление строковых типов данных. Ввод строк, присваивание, получение длины строки, удаление строки, вставка строки.	4	
	2 Объявление множества. Операции над множествами.		
	Практические занятия Ввод и отладка программы по обработке строк. Ввод и отладка программы с использованием множеств.	6	
Тема 2.8. Файлы	Содержание учебного материала		
	1 Типы файлов.	4	
	2 Организация последовательного доступа к файлам, стандартные процедуры для работы с файлами, текстовые файлы.		
	Практические занятия Работа с файлом последовательного доступа. Работа с файлом произвольного доступа.	8	
Тема 2.9. Модули	Содержание учебного материала		
	1 Программирование модулей. Модуль: синтаксис, заголовок, разделы.		
	2 Библиотеки подпрограмм: понятие и виды. Схемы вызова библиотек. Статическое и динамическое связывание.		
	Практические занятия Программирование модуля.	4	
		2	

	Создание библиотеки подпрограмм. Использование библиотеки подпрограмм.		
Раздел 3. Программирование в объектно-ориентированной среде		54	
Тема 3.1. . Основные принципы объектно- ориентированного программирования (ООП)	Содержание учебного материала	8	ОК 1 ОК 2 ОК 4 ОК 5 ОК 9 ОК 10 ПК 1.1- ПК 1.5 ПК 2.4, 2.5
	1 История развития ООП.		
	2 Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс. Основные принципы ООП: инкапсуляция, наследование, полиморфизм.		
	3 Событийно-управляемая модель программирования. Компонентно-ориентированный подход.		
	4 Классы объектов. Компоненты и их свойства.		
	Практические занятия Ввод и отладка программы с использованием классов. Объявление класса, создание экземпляров класса. Ввод и отладка программы с использованием наследованного класса. Ввод и отладка программы с использованием перегрузки методов.	10	
Тема 3.2. Интегрированная среда разработчика	Содержание учебного материала	6	
	1 Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты.		
	2 Форма и размещение на ней управляющих элементов. Панель компонентов и их свойства.		
	3 Окно кода проекта. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.		
	Практические занятия Изучение интегрированной среды разработчика.	4	
Тема 3.3. Визуальное событийно- управляемое программирование	Содержание учебного материала	8	
	1 Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение. Дополнительные элементы управления.		
	2 Свойства компонентов (элементов управления). Виды свойств. Синтаксис определения свойств. Категория свойств. Назначение свойств и их влияние на результат. Управление объектом через свойства.		
	3 События компонентов (элементов управления), их сущность и назначение. Создание процедур на основе событий. Процедуры, определенные пользователем: синтаксис, передача аргументов. Вызов событий.		
	Практические занятия Создание проекта с использованием различных компонентов.	10	

	Разработка оконного приложения. Разработка многооконного приложения.		
<i>Консультации</i>		6	
<i>Промежуточная аттестация в форме экзамена</i>		6	
Всего:		196	

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 УЧЕБНОЙ ДИСЦИПЛИНЫ

3.1. Программа дисциплины реализуется в учебном кабинете «Программирования баз данных», оснащенная необходимым для реализации программы учебной дисциплины оборудованием:

Компьютерный класс, с установленным программным обеспечением (интегрированной средой разработки).

Технические средства обучения:

- интерактивная доска
- проектор
- принтер лазерный (принтер лазерный сетевой);
- источник бесперебойного питания;
- сканер, цифровой фотоаппарат, Web-камера;
- аудиторная доска для письма фломастером с магнитной поверхностью;
- шкафы для хранения оборудования;
- демонстрационные печатные пособия и демонстрационные ресурсы в электронном

представлении.

3.2. Информационное обеспечение обучения

Для реализации программы библиотечный фонд образовательной организации имеет печатные образовательные и информационные ресурсы, рекомендуемые для использования в образовательном процессе

Основные источники:

1. И.Г. Семакин, А.П.Шестаков. Основы алгоритмизации и программирования: учебник для сред.проф. образования –М.:Издательский центр «Академия», 2016.
2. И.Г. Семакин, А.П. Шестаков. Основы программирования: учебник для сред.проф. образования – М.:Издательский центр «Академия», 2016.
3. О.Л. Голицына, И.И Попов. Основы алгоритмизации и программирования: учеб. пособие. – 3-е издание, испр. и доп. – М: ФОРУМ, 2015.

Дополнительные источники:

1. М.С. Эпштейн. Практикум по программированию на языке С: учебн. Пособие для сред.проф. Образования – М.:Издательский центр «Академия», 2014.
2. Ю. Сальников. Программирование. Базовый курс. – М.: МаркетДС, 2015.

4 КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

<i>Результаты обучения</i>	<i>Критерии оценки</i>	<i>Формы и методы оценки</i>
<p>Перечень умений, осваиваемых в рамках дисциплины:</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>«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.</p> <p>«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.</p>	<p>Формы и методы контроля и оценки:</p> <ul style="list-style-type: none"> • Тестирование • Контрольная работа • Оценка выполнения практического задания • Решение ситуационной задачи
<p>Перечень знаний, осваиваемых в рамках дисциплины:</p> <ul style="list-style-type: none"> • 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- 	<p>«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.</p> <p>«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки</p>	

ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.		
---	--	--