

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

ОП.04 Основы алгоритмизации и программирования

СОДЕРЖАНИЕ

1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ	3
2. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ	4
3. УСЛОВИЯ РЕАЛИЗАЦИИ ДИСЦИПЛИНЫ	10
4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ	11

1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ «ОП.04 Основы алгоритмизации и программирования»

1.1. Место дисциплины в структуре основной образовательной программы:

Дисциплина «Основы алгоритмизации и программирования» является обязательной частью общепрофессионального цикла ПОП-П в соответствии с ФГОС СПО по специальности 09.02.07 Информационные системы и программирование.

Особое значение дисциплина имеет при формировании и развитии ОК.01, ОК.02, ОК.04, ОК.05, ОК.09.

1.2. Цель и планируемые результаты освоения дисциплины:

В рамках программы дисциплины обучающимися осваиваются умения и знания

Код ПК, ОК	Умения	Знания
ОК.01 ОК.02 ОК.04 ОК.05 ОК.09 ПК 1.1. ПК 1.2. ПК 1.3. ПК 1.4. ПК 1.5. 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 ДИСЦИПЛИНЫ

2.1. Объем дисциплины и виды учебной работы

Вид учебной работы	Объем в часах
Объем образовательной программы учебной дисциплины	184
в т.ч. в форме практической подготовки	80
в т. ч.:	
теоретическое обучение	74
лабораторные работы <i>(если предусмотрено)</i>	
практические занятия <i>(если предусмотрено)</i>	80
курсовая работа (проект) <i>(если предусмотрено для специальностей)</i>	
<i>Самостоятельная работа</i>	6
Промежуточная аттестация	24

2.2. Тематический план и содержание дисциплины

Наименование разделов и тем	Содержание учебного материала и формы организации деятельности обучающихся	Объем, акад. ч / в том числе в форме практической подготовки, акад. ч		Коды компетенций, формированию которых способствует элемент программы
1	2	3		4
		184/80		ОК.01 ОК.02 ОК.04 ОК.05 ОК.09 ПК 1.1. ПК 1.2. ПК 1.3. ПК 1.4. ПК 1.5. ПК 2.4. ПК 2.5.
Раздел 1. Введение в программирование		32	16	
Тема 1.1. Языки программирования	Содержание учебного материала	16		
	1. Развитие языков программирования.			
	2. Обзор языков программирования. Области применения языков программирования. Стандарты языков программирования. Среда проектирования. Компиляторы и интерпретаторы.			
	3. Жизненный цикл программы. Программа. Программный продукт и его характеристики.			
	4. Основные этапы решения задач на компьютере.			
	В том числе практических занятий и лабораторных работ			
	Самостоятельная работа обучающихся			
Тема 1.2. Типы данных	Содержание учебного материала	16		
	1. Типы данных. Простые типы данных. Производные типы данных. Структурированные типы данных.			
	В том числе практических занятий и лабораторных работ			
	Самостоятельная работа обучающихся			
Раздел 2. Содержание учебного материала		32	16	
Тема 2.1 Операторы языка программирования	1. Операции и выражения. Правила формирования и вычисления выражений. Структура программы. Ввод и вывод данных. Оператор присваивания. Составной оператор.	32		
	2. Условный оператор. Оператор выбора.			
	3. Цикл с постусловием. Цикл с предусловием. Цикл с параметром. Вложенные циклы.			
	4. Массивы. Двумерные массивы. Строки. Стандартные процедуры и функции для работы со строками.			
	5. Структурированный тип данных – множество. Операции над множествами.			
	6. Комбинированный тип данных – запись. Файлы последовательного доступа. Файлы прямого доступа			
	В том числе практических занятий и лабораторных работ			

	Самостоятельная работа обучающихся			
Раздел 3.	Содержание учебного материала	32	16	
Тема 3.1. Процедуры и функции	1. Общие сведения о подпрограммах. Определение и вызов подпрограмм. Область видимости и время жизни переменной. Механизм передачи параметров. Организация функций.	10		
	2. Рекурсия. Программирование рекурсивных алгоритмов.			
	В том числе практических занятий и лабораторных работ			
	Самостоятельная работа обучающихся			
Тема 3.2. Структуризация в программировании	Содержание учебного материала	10		
	1. Основы структурного программирования. Методы структурного программирования.			
	В том числе практических занятий и лабораторных работ			
	Самостоятельная работа обучающихся			
Тема 3.3. Модульное программирование	Содержание учебного материала	12		
	1. Модульное программирование. Понятие модуля. Структура модуля. Компиляция и компоновка программы.			
	2. Стандартные модули.			
	В том числе практических занятий и лабораторных работ			
	Самостоятельная работа обучающихся			
Раздел 4. Основные конструкции языков программирования		32	16	
Тема 4.1. Указатели.	Содержание учебного материала	32		
	1. Указатели. Описание указателей. Основные понятия и применение динамически распределяемой памяти. Создание и удаление динамических переменных.			
	2. Структуры данных на основе указателей.			
	3. Задача о стеке.			
	В том числе практических занятий и лабораторных работ			
	Самостоятельная работа обучающихся			
Раздел 5	Содержание учебного материала	32	16	
Тема 5.1. Основные принципы объектно-ориентированного программирования (ООП)	1. История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс.	6		
	2. Основные принципы ООП: инкапсуляция, наследование, полиморфизм.			
	3. Классы объектов. Компоненты и их свойства.			
	4. Событийно-управляемая модель программирования. Компонентно-ориентированный подход.			
	В том числе практических занятий и лабораторных работ			
	Самостоятельная работа обучающихся			

Тема 5.2. Интегрированная среда разработчика.	Содержание учебного материала			
	1. Требования к аппаратным и программным средствам интегрированной среды разработчика.	6		
	2. Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты. Форма и размещение на ней управляющих элементов.			
	3. Панель компонентов и их свойства. Окно кода проекта.			
	4. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.			
	5. Панель компонентов и их свойства. Окно кода проекта. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.			
	6. Настройка среды и параметров проекта.			
	В том числе практических занятий и лабораторных работ			
	Самостоятельная работа обучающихся			
Тема 5.3. Визуальное событийно- управляемое программирование	Содержание учебного материала	6		
	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.			
	В том числе практических занятий и лабораторных работ			
	Самостоятельная работа обучающихся			
Тема 5.4. Разработка оконного приложения	Содержание учебного материала	6		
	1. Разработка функционального интерфейса приложения. Создание интерфейса приложения.			
	2. Разработка функциональной схемы работы приложения.			
	3. Разработка игрового приложения.			
	В том числе практических занятий и лабораторных работ			
	Самостоятельная работа обучающихся			
Тема 5.5. Этапы разработки приложений	Содержание учебного материала	4		
	1. Разработка приложения.			
	2. Проектирование объектно-ориентированного приложения.			
	3. Создание интерфейса пользователя.			
	4. Тестирование, отладка приложения.			
	В том числе практических занятий и лабораторных работ			
	Самостоятельная работа обучающихся			

Тема 5.6. Иерархия классов.	Содержание учебного материала	4		
	1. Классы ООП: виды, назначение, свойства, методы, события.			
	2. Перегрузка методов.			
	3. Тестирование и отладка приложения.			
	4. Решение задач			
	В том числе практических занятий и лабораторных работ			
	Самостоятельная работа обучающихся			
Примерная тематика практических занятий и лабораторных работ: Знакомство со средой программирования. Составление программ линейной структуры. Составление программ разветвляющейся структуры. Составление программ циклической структуры Обработка одномерных массивов. Обработка двумерных массивов. Работа со строками. Работа с данными типа множество. Файлы последовательного доступа. Типизированные файлы. Нетипизированные файлы. Организация процедур. Организация функций. Применение рекурсивных функций. Программирование модуля. Создание библиотеки подпрограмм. Использование указателей для организации связанных списков. 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. 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. Разработка оконного приложения с несколькими формами. Разработка игрового приложения. Создание процедур обработки событий. Компиляция и запуск приложения. Разработка интерфейса приложения. Тестирование, отладка приложения. Классы ООП: виды, назначение, свойства, методы, события.				

Объявления класса. Создание наследованного класса. Программирование приложений. Перегрузка методов.			
Промежуточная аттестация	24		
Всего:	<i>184/80</i>		

3. УСЛОВИЯ РЕАЛИЗАЦИИ ДИСЦИПЛИНЫ

3.1 Для реализации программы дисциплины должны быть предусмотрены следующие специальные помещения:

Лаборатория «Программирования баз данных», оснащенный в соответствии с п. 6.1.2.1 образовательной программы по специальности 09.02.07 Информационные системы и программирование.

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. Семакин И.Г., Шестаков А.П Основы алгоритмизации и программирования. – М.: Издательский центр «Академия», 2021. – 304 с.

3.2.2. Основные электронные издания

1. Трофимов В. В. Основы алгоритмизации и программирования: учебник для среднего профессионального образования / В. В. Трофимов, Т. А. Павловская; под редакцией В. В. Трофимова. – М.: Издательство Юрайт, 2021. – 137 с. – (Профессиональное образование). – ISBN 978-5-534-07321-8. – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/473347> (дата обращения: 13.12.2022).

3.2.3. Дополнительные источники

1. Кудрина Е.В. Основы алгоритмизации и программирования на языке C#: учебное пособие для среднего профессионального образования / Е. В. Кудрина, М. В. Огнева. – М.: Издательство Юрайт, 2021. – 322 с. – (Профессиональное образование). – ISBN 978-5-534-10772-2. – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/475228> (дата обращения: 13.12.2022).

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Результаты обучения	Критерии оценки	Формы и методы оценки
<p>Перечень умений, осваиваемых в рамках дисциплины:</p> <p>Разрабатывать алгоритмы для конкретных задач.</p> <p>Использовать программы для графического отображения алгоритмов.</p> <p>Определять сложность работы алгоритмов.</p> <p>Работать в среде программирования.</p> <p>Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.</p> <p>Оформлять код программы в соответствии со стандартом кодирования.</p> <p>Выполнять проверку, отладку кода программы.</p>	<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>«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.</p> <p>«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.</p>	<p>Примеры форм и методов контроля и оценки</p> <p>Компьютерное тестирование на знание терминологии по теме;</p> <p>Тестирование</p> <p>Контрольная работа</p> <p>Самостоятельная работа.</p> <p>Защита реферата</p> <p>Семинар</p> <p>Защита курсовой работы (проекта)</p> <p>Выполнение проекта;</p> <p>Наблюдение за выполнением практического задания. (деятельностью студента)</p> <p>Оценка выполнения практического задания(работы)</p> <p>Подготовка и выступление с докладом, сообщением, презентацией</p> <p>Решение ситуационной задачи</p>
<p>Перечень знаний, осваиваемых в рамках дисциплины:</p> <p>Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции.</p> <p>Эволюцию языков программирования, их классификацию, понятие системы программирования.</p> <p>Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.</p> <p>Подпрограммы, составление библиотек подпрограмм</p> <p>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.</p>		