

МИНОБРНАУКИ РОССИИ
 Глазовский инженерно-экономический институт (филиал)
 федерального государственного бюджетного образовательного
 учреждения высшего образования
 «Ижевский государственный технический университет
 имени М.Т.Калашникова»



УТВЕРЖДАЮ

Директор ГИЭТ

03 марта 2020 г.

М.А.Бабушкин

РАБОЧАЯ ПРОГРАММА

Учебной дисциплины **ОП.04 Основы алгоритмизации и программирования**

Специальность СПО **09.02.07 Информационные системы и программирование**

Цикл **профессиональный**

Форма обучения **очная**

Вид учебной работы	Объем, час.	Семестры							
		1	2	3	4	5	6	7	8
Максимальная учебная нагрузка, час.	152				152				
Обязательная аудиторная нагрузка, час.	112				112				
в том числе:									
Лекции, час.	46				46				
Лабораторные занятия, час.									
Лабораторные работы, час.	46				46				
Курсовой проект (работа), час.	20				20				
Самостоятельная работа, час.	40				40				
Виды промежуточной аттестации									
Экзамен, сем.									
Дифференцированный зачет, сем	+				+				
Зачет, сем									

Рабочая программа учебной дисциплины разработана на основе Федерального государственного образовательного стандарта по специальности среднего профессионального образования 09.02.07 "Информационные системы и программирование", утвержденного приказом Министерства образования и науки Российской Федерации 09 декабря 2016 г.

№ 1547.

Организация разработчик:

ГИЭИ (филиал) ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова»

Разработчик:

Горбушин А.Г., к.п.н., доцент кафедры «Машиностроение и информационные технологии»

Утверждено:

кафедрой «Машиностроения и информационных технологий»
Протокол № 4 от 30 января 2020 г.

Заведующий кафедрой  Беляев В.В.

Председатель учебно-методической комиссии
Глазовского инженерно-экономического института (филиала)
ФГБОУ ВО «ИжГТУ имени М.Т.Калашникова»

 Беляев В.В.

31 января 2020 г.

Согласовано:

Начальник отдела по учебно-методической работе

 И.Ф. Яковлева

31 января 2020 г.

СОДЕРЖАНИЕ

	стр.
1. ПАСПОРТ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ	4
2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ	6
3. УСЛОВИЯ РЕАЛИЗАЦИИ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ	15
4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ	17

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

Основы алгоритмизации и программирования

1.1. Область применения рабочей программы

Рабочая программа учебной дисциплины является частью основной профессиональной образовательной программы в соответствии с Федеральным государственным образовательным стандартом по специальности СПО 09.02.07 Информационные системы и программирование.

1.2. Место учебной дисциплины в структуре основной профессиональной образовательной программы: профессиональный цикл, общепрофессиональная дисциплина ОП.04

1.3. Цели и задачи учебной дисциплины – требования к результатам освоения учебной дисциплины:

В результате освоения учебной дисциплины обучающийся должен *уметь*:

- использовать языки программирования, строить логически правильные и эффективные программы;

знать:

- общие принципы построения алгоритмов, основные алгоритмические конструкции;
- понятие системы программирования;
- основные элементы процедурного языка программирования, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, кассы памяти;
- подпрограммы, составление библиотек программ;
- объектно-ориентированную модель программирования, понятие классов и объектов, их свойств и методов

1.4. Перечень формируемых компетенций

Общие компетенции (ОК):

ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.

ОК 09. Использовать информационные технологии в профессиональной деятельности.

ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке.

Профессиональные компетенции (ПК):

ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.

ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.

ПК 1.4. Выполнять тестирование программных модулей.

ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода.

ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.

ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.

1.5 Количество часов на освоение рабочей программы учебной дисциплины:

максимальной учебной нагрузки обучающегося 152 часа, в том числе:

обязательной аудиторной учебной нагрузки обучающегося 112 часов;

самостоятельной работы обучающегося 40 часа.

2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Количество во часов
Максимальная учебная нагрузка (всего)	152
Обязательная аудиторная учебная нагрузка (всего)	112
в том числе:	
лабораторные занятия	46
контрольные работы	
Самостоятельная работа обучающегося (всего)	40
в том числе:	
индивидуальное проектное задание	
тематика внеаудиторной самостоятельной работы	
<i>Итоговая аттестация в форме дифференцированного зачёта</i>	

2.2 Тематический план и содержание учебной дисциплины «Основы алгоритмизации и программирования»

Наименование разделов и тем	Содержание учебного материала, лабораторные работы, самостоятельная работа обучающихся		Объем часов	Уровень освоения
1	2		3	4
Раздел 1 Основные принципы алгоритмизации и программирования			18	
Тема 1.1 Основные понятия алгоритмизации	Содержание учебного материала		6	
	1-2	Общие принципы построения алгоритмов. Основные алгоритмические конструкции. Понятие алгоритма. Свойства алгоритмов. Формы записей алгоритмов. Общие принципы построения алгоритмов. Основные алгоритмические конструкции: линейные, разветвляющиеся, циклические. Данные: понятие и типы. Основные базовые типы данных и их характеристика. Структурированные типы данных и их характеристика. Методы сортировки данных.	2	1
	Лабораторные занятия		2	2
	1-2	Лабораторное занятие №1. Составление блок-схем алгоритмов.		
	Самостоятельная работа обучающихся: Составление словаря терминов, используемых в системе программирования Подготовить сообщение «Применение алгоритмов на практике»		2	3
Тема 1.2 Логические основы алгоритмизации	Содержание учебного материала		6	
	1-2	Логические основы алгоритмизации Основы алгебры логики. Логические операции с высказываниями: конъюнкция, дизъюнкция, инверсия. Законы логических операций. Таблицы истинности.	2	1
	Лабораторные занятия		2	2
	1-2	Лабораторное занятие №2. Составление таблиц истинности.		
	Самостоятельная работа обучающихся: Составление словаря терминов, используемых в системе программирования Подготовить сообщение: Логические основы алгоритмизации		2	3
Тема 1.3 Языки и системы программирования	Содержание учебного материала		4	
	1-2	Языки и системы программирования Эволюция языков программирования. Классификация языков программирования. Элементы языков программирования. Понятие системы программирования. Исходный, объектный и загрузочный модули. Интегрированная среда программирования.	2	2
	3-4	Основные элементы процедурного языка программирования. Структура программы.		
	5-6	Операции. Управляющие структуры. Структуры данных. Кассы памяти.		
	Самостоятельная работа обучающихся: Составление словаря терминов, используемых в системе программирования		2	3

	Создание презентации «Классификация языков программирования» Подготовить сообщение: Языки программирования			
Тема 1.4 Методы программирования	Содержание учебного материала		2	
	1-2	Методы программирования Методы программирования: структурный, модульный, объектно-ориентированный. Достоинства и недостатки методов программирования. Общие принципы разработки программного обеспечения. Жизненный цикл программного обеспечения. Типы приложений. Консольные приложения. Оконные Windows приложения. Web-приложения. Библиотеки. Web-сервисы.	1	2
	Самостоятельная работа: Составление словаря терминов, используемых в системе программирования		1	3
Раздел 2 Программирование на алгоритмическом языке			38	
Тема 2.1 Описание интегрированной среды программирования Turbo Pascal	Содержание учебного материала		2	
	1-2	Структурная схема программы на алгоритмическом языке. Лексика языка. Переменные и константы. Типы данных. Выражения и операции.	1	2
	Самостоятельная работа студентов: Составление словаря терминов, используемых в системе программирования		1	3
Тема 2.2 Операторы языка Простые и структурные операторы.	Содержание учебного материала		6	
	1-2	Операторы языка Синтаксис операторов: присваивания, ввода-вывода, безусловного и условного переходов, циклов. Составной оператор. Вложенные условные операторы. Циклические конструкции. Циклы с предусловием и постусловием.	2	1
	Лабораторные занятия		2	2
	1-2	Лабораторное занятие №3. Составление программ линейной структуры.		
	3-4	Лабораторное занятие №4. Составление программ разветвляющейся структуры.		
	5-6	Лабораторное занятие №5. Составление программ разветвляющейся усложненной структуры.		
7-8	Лабораторное занятие №6. Составление программ циклической структуры.			
9-10	Лабораторное занятие №7. Составление программ усложненной структуры.			
Самостоятельная работа студентов: Подготовить доклад: Изучение организация ввода-вывода данных; Подготовить сообщение: Одномерные массивы. Подготовить сообщение: Двумерные массивы. Составление словаря терминов, используемых в системе программирования		2	3	
Тема 2.3 Массивы. Одномерные массивы. Двумерные	Содержание учебного материала		6	
	1-2	Массивы Массивы как структурированный тип данных. Объявление массива. Ввод и вывод	2	1

массивы	одномерных массивов. Ввод и вывод двумерных массивов. Обработка массивов. Стандартные функции для массива целых и вещественных чисел.			
	Лабораторные занятия		2	2
	1	Лабораторное занятие №8. Обработка одномерных массивов.		
	2-3-4	Лабораторное занятие №9. Обработка двумерных массивов. Лабораторное занятие №10. Использование стандартных функций для работы с массивами.		
Самостоятельная работа студентов: Составление словаря терминов, используемых в системе программирования Составление блок-схемы программ по практической работе «Использование стандартных функций для работы со строками»		2	3	
Тема 2.4 Строки множества.	Содержание учебного материала		6	
	1-2	Структурированные типы данных: строки и множества. Объявление строковых типов данных. Поиск, удаление, замена и добавление символов в строке. Операции со строками. Стандартные функции и процедуры для работы со строками. Объявление множества. Операции над множествами.	2	2
	Лабораторные занятия		2	2
	1-2	Лабораторное занятие №11. Работа со строковыми переменными.		
	3-4	Лабораторное занятие №12. Использование стандартных функций и процедур для работы со строками.		
	5-6	Лабораторное занятие №13. Работа с данными типа множество.		
Самостоятельная работа студентов: Составление словаря терминов, используемых в системе программирования Подготовить сообщение: Интегрированная среда программирования; Подготовить сообщение: Операторы языка; Подготовить презентацию: Одномерные массивы. Двумерные массивы.		2	3	
Тема 2.5 Процедуры и функции	Содержание учебного материала		6	
	1-2	Подпрограммы, процедуры и функции Понятие подпрограммы. Процедуры и функции, их сущность, назначение, различие. Организация процедур, стандартные процедуры. Процедуры, определенные пользователем: синтаксис, передача аргументов. Формальные и фактические параметры. Процедуры с параметрами, описание процедур. Функции: способы организации и описание. Вызов функций, рекурсия. Программирование рекурсивных алгоритмов. Стандартные функции.	2	1
	Лабораторные занятия		2	2
	1	Лабораторное занятие №14. Организация процедур.		
2	Лабораторное занятие №15. Использование процедур.			
3	Лабораторное занятие №16. Организация функций.			
4	Лабораторное занятие №17. Использование функций.			

	Самостоятельная работа студентов: Составление словаря терминов, используемых в системе программирования Составление блок-схем по практической работе «Организация процедур» Составление блок-схем по практической работе «Организация функций»	2	3
Тема 2.6 Организация ввода-вывода данных. Работа с файлами	Содержание учебного материала	6	
	1-2 Файлы данных. Типы файлов. Организация доступа к файлам. Файлы последовательного доступа. Открытие и закрытие файла последовательного доступа. Запись в файл и чтение из файла последовательного доступа. Файлы произвольного доступа. Порядок работы с файлами произвольного доступа. Создание структуры записи. Открытие и закрытие файла произвольного доступа. Запись и считывание из файла произвольного доступа. Использование файла произвольного доступа. Стандартные процедуры и функции для файлов разного типа.	2	2
	Лабораторные занятия		2
	1-2 Лабораторное занятие №18. Работа с файлом последовательного доступа. 3-4 Лабораторное занятие №19. Работа с файлом произвольного доступа. 5-6 Лабораторное занятие №20. Использование стандартных процедур и функций для работы с файлами.	2	
	Самостоятельная работа студентов: Составление словаря терминов, используемых в системе программирования Подготовить сообщение: Организация ввода-вывода Подготовить презентацию: файлы данных	2	3
Тема 2.7 Библиотеки подпрограмм.	Содержание учебного материала	6	
	1-2 Библиотеки подпрограмм Программирование модулей. Модуль: синтаксис, заголовок, разделы. Библиотеки подпрограмм: понятие и виды. Схемы вызова библиотек. Статическое и динамическое связывание. Использование библиотек подпрограмм.	2	2
	Лабораторные занятия		2
	1-2 Лабораторное занятие №21. Программирование модуля. 3-4 Лабораторное занятие №22. Создание библиотеки подпрограмм. 5-6 Лабораторное занятие №23. Использование библиотеки подпрограмм.	2	
	Самостоятельная работа студентов: Составление словаря терминов, используемых в системе программирования Подготовить сообщение: Ознакомление со стандартными библиотеками подпрограмм; Выполнение задач по теме «Модули» Составить схему вызова библиотек	2	3
Раздел 3 Программирование в объектно-ориентированной среде		63	
Тема 3.1	Содержание учебного материала	4	

Основные принципы объектно-ориентированного программирования (ООП)	1-2	Объектно-ориентированный подход к программированию. История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс. Основные принципы ООП: инкапсуляция, наследование, полиморфизм. Событийно-управляемая модель программирования. Компонентно-ориентированный подход. Классы объектов. Компоненты и их свойства.	2	2
	Самостоятельная работа студентов: Составление словаря терминов, используемых в системе программирования		2	3
Тема 3.2 Интегрированная среда разработчика	Содержание учебного материала		10	
	1-2	Интегрированная среда разработчика Требования к аппаратным и программным средствам интегрированной среды разработчика. Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты. Форма и размещение на ней управляющих элементов. Панель компонентов и их свойства. Окно кода проекта. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.	4	2
	Лабораторные занятия			2
	1-2	Лабораторное занятие №24. Изучение интегрированной среды разработчика.	4	
	3-4	Лабораторное занятие №25. Создание простого проекта.		
Самостоятельная работа студентов: Составление словаря терминов, используемых в системе программирования Составить памятку: Этапы разработки приложения; Составить схему: Разработка оконного приложения;		2	3	
Тема 3.3 Этапы разработки приложения	Содержание учебного материала		4	
	1-2	Этапы разработки приложения Проектирование объектно-ориентированного приложения. Создание интерфейса пользователя. Программирование приложения. Тестирование, отладка приложения. Создание документации.	2	2
	Самостоятельная работа студентов: Составление словаря терминов, используемых в системе программирования		2	3
Тема 3.4 Иерархия классов.	Содержание учебного материала		13	
	1-2	Иерархия классов Классы объектно-ориентированного языка программирования: виды, назначение, свойства, методы, события. Объявление класса, свойств и методов экземпляра класса. Наследование. Перегрузка методов.	4	2
	Лабораторные занятия			2
	1-2	Лабораторное занятие №26. Объявление класса, создание экземпляров класса.	7	
	3-4	Лабораторное занятие №27. Создание наследованного класса.		
5-6	Лабораторное занятие №28 Перегрузка методов.			
Самостоятельная работа студентов: Составление словаря терминов, используемых в системе программирования		2	3	

	Составить памятку: Интегрированная среда разработчика; Составить схему: Классы объектно-ориентированного языка программирования Составить сообщение: Иерархия классов			
Тема 3.5 Визуальное событийно-управляемое программирование	Содержание учебного материала		16	
	1-2	Визуальное событийно-управляемое программирование Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение. Дополнительные элементы управления.	5	2
	3-4	Визуальное событийно-управляемое программирование События компонентов (элементов управления), их сущность и назначение. Виды свойств. Синтаксис определения свойств. Категория свойств. Назначение свойств и их влияние на результат. Управление объектом через свойства. Создание процедур на основе событий. Процедуры, определенные пользователем: синтаксис, передача аргументов. Вызов событий.		
	Лабораторные занятия		8	2
	1-2	Лабораторное занятие №29 Создание проекта с использованием кнопочных компонентов.		
	3-4	Лабораторное занятие №30 Создание проекта с использованием компонентов для работы с текстом.		
5-6	Лабораторное занятие №31 Создание проекта с использованием компонентов ввода и отображения чисел, дат и времени.			
7-8	Лабораторное занятие №32 Создание проекта с использованием компонентов стандартных диалогов и системы меню.			
Самостоятельная работа студентов: Составление словаря терминов, используемых в системе программирования Подготовить реферат: Визуальное событийно-управляемое программирование Составить памятку: Свойства основных компонентов интегрированной среды обработки Составить схему создания проекта		3	3	
Тема 3.6 Разработка оконного приложения	Содержание учебного материала		16	
	1-2-3	Разработка оконного приложения Разработка функционального интерфейса приложения. Создание интерфейса приложения. Разработка функциональной схемы работы приложения. Создание процедур обработки событий. Компиляция и запуск приложения.	5	2
	Лабораторные занятия		8	2
	1-2	Лабораторное занятие №33. Разработка оконного приложения.		
	3-4	Лабораторное занятие №34. Разработка оконного приложения с несколькими формами.		
	5-6	Лабораторное занятие №35. Разработка многооконного приложения.		
Самостоятельная работа студентов: Составление словаря терминов, используемых в системе программирования Составить памятку: Разработка оконного приложения		3	3	

	Подготовка к дифференцированному зачёту		
	Дифференцированный зачёт		
	<p>Примерные темы курсовой работы:</p> <ul style="list-style-type: none"> – Хеширование. Реализовать программу «Студенты и стипендия». Требования к программе: удобный интерфейс и защита от неправильных действий пользователя. Данные должны храниться в отдельных файлах. – Хеширование. Реализовать программу «Склад». Требования к программе: удобный интерфейс и защита от неправильных действий пользователя. Данные должны храниться в отдельных файлах. – Хеширование. Реализовать программу «Телефонный справочник». Требования к программе: удобный интерфейс и защита от неправильных действий пользователя. Данные должны храниться в отдельных файлах. – Хеширование. Реализовать программу «Библиотека». Требования к программе: удобный интерфейс и защита от неправильных действий пользователя. Данные должны храниться в отдельных файлах. – Задача о составлении магического квадрата. То есть такого квадрата, у которого суммы столбцов равны между собой, равны суммам строк, и суммам диагоналей. При этом числа в квадрате порядка n идут от 1 до $n \cdot n$. Очень интересная задача - в данном случае пользователь вводит порядок квадрата и получает этот самый квадрат. Заодно и сумма высчитывается. – Деревья. Реализовать операции с красно-черными деревьями. Требования к программе: удобный интерфейс и защита от неправильных действий пользователя. – Деревья. Реализовать операции с АВЛ – деревьями. Требования к программе: удобный интерфейс и защита от неправильных действий пользователя. – Деревья. Реализовать операции с В-деревьями. Требования к программе: удобный интерфейс и защита от неправильны действий пользователя. – Деревья. Реализовать операции с 2-3 деревьями. Требования к программе: удобный интерфейс и защита от неправильных действий пользователя. – Графы. Реализовать поиск точек сочленения в графе. – Графы. Максимальный поток минимальной стоимости. Алгоритм Клейна. – Графы. Максимальный поток минимальной стоимости. Алгоритм Басакера-Гоуэна. – Графы. Метод минимакса на двоичном дереве. Предположим, два игрока играют в игру, такую что в каждой позиции игрок, чья очередь хода, имеет только два варианта ответа. Такую игру можно представить в виде двоичного дерева. Предположим, что игрок принимающий решение о ходе способен провести анализ на N ходов вглубь и предположим, что он в состоянии оценить конечные позиции числом тем большим чем лучше эта конечная позиция. Вопрос - какой из двух имеющихся вариантов игры ему выбрать при условии, что противник играет наилучшим образом и имеет точно такие же критерии оценки позиций. – Графы. Поиск пути к бензоколонке. Некий автомобилист пытается в городе найти путь к бензоколонке. Он находится в определённой точке и он располагает картой города на которой 	20	

	<p>отмечена единственная бензоколонка. На улицах города, как и полагается установлены знаки соблюдать которые водитель в принципе обязан. Но наш водитель имеет возможность нарушить некоторое количество правил. Кроме того, его автомобиль имеет некоторое количество топлива, которого возможно более чем достаточно, но может быть и не очень много. Необходимо найти путь на котором водитель уложится в существующее количество топлива и минимальное количество раз нарушит правила дорожного движения. Экономить топливо не обязательно. Карта города представляет собой граф, для построения которого обязательно использовать указатели.</p> <ul style="list-style-type: none"> – Графы. Задача о назначениях. Венгерский алгоритм. – Графы. Реализовать приближенные методы решения задачи коммивояжера. Алгоритм Кристофидеса. Алгоритм Эйлера. Метод локальной оптимизации. – Графы. Задача китайского почтальона. – Графы. Правильной вершинной раскраской неориентированного графа называется функция $C: V \rightarrow N$, удовлетворяющая условию <ul style="list-style-type: none"> $\forall (i, j) \in E: C(i) \neq C(j)$ <p>а) То есть смежные вершины должны иметь разные цвета. б) Необходимо найти раскраску графа в минимальное количество цветов. Пример – составление расписания экзаменов.</p> <ul style="list-style-type: none"> – Графы. Задача о кратчайших путях. Реализовать алгоритм Форда-Беллмана, волновой алгоритм, алгоритм Флойда, алгоритм Флойда-Уоршалла – Графы. Задача нахождения максимального потока. Алгоритм Форда-Фалкерсона, Алгоритм Эдмондса-Карпа и один алгоритм на выбор. – Пирамида. Реализовать способы реализации и основные операции. Требования к программе: удобный интерфейс и защита от неправильных действий пользователя. 		
	Всего	152	

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 УЧЕБНОЙ ДИСЦИПЛИНЫ

3.1. Требования к минимальному материально-техническому обеспечению

Реализация учебной дисциплины осуществляется в учебном кабинете:

«Информатики, информационных технологий».

Оборудование учебного кабинета:

- персональные компьютеры (по количеству рабочих мест);
- все компьютерные классы объединены в локальную вычислительную сеть и имеют круглосуточный доступ в Интернет;
- стенды;
- методическая литература;
- комплект учебной мебели: столы (по количеству обучающихся), стулья (по количеству обучающихся), стол преподавателя, стул преподавателя,
- маркерная доска;
- наглядные пособия: демонстрационные плакаты, раздаточный материал;
- методическая литература;
- мультимедийная аппаратура: (мультимедийный портативный переносной проектор; экран);
- комплект лицензионного программного обеспечения.

3.2. Информационное обеспечение обучения

Перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительной литературы

Основная

1. Белева, Л. Ф. Программирование на языке C++ [Электронный ресурс] : учебное пособие / Л. Ф. Белева. — Электрон. текстовые данные. — Саратов : Ай Пи Эр Медиа, 2018. — 81 с. — 978-5-4486-0253-5. — Режим доступа: <http://www.iprbookshop.ru/72466.html>
2. Костюкова, Н. И. Программирование на языке Си [Электронный ресурс]: методические рекомендации и задачи по программированию / Н. И. Костюкова. — Электрон. текстовые данные. — Новосибирск : Сибирское университетское издательство, 2017. — 160 с. — 978-5-379-02016-3. — Режим доступа: <http://www.iprbookshop.ru/65289.html>
3. Устинов, В. В. Основы алгоритмизации и программирование. Часть 2 [Электронный ресурс] : конспект лекций / В. В. Устинов. — Электрон. текстовые данные. — Новосибирск : Новосибирский государственный технический университет, 2013. — 32 с. — 978-5-7782-2337-0. — Режим доступа: <http://www.iprbookshop.ru/44675.html>

Дополнительная

1. Семакин И.Г., Шестаков А.П. Основы алгоритмизации и программирования/ И.Г. Семакин ОИЦ «Академия» – М.: Издательский центр «Академия», 2016
2. Семакин И.Г., Шестаков А.П. Основы алгоритмизации и программирования. Практикум/ И.Г. Семакин ОИЦ «Академия» – М.: Издательский центр «Академия», 2016
3. Голицына, О.Л., Основы алгоритмизации и программирования: Учебное пособие / О.Л. Голицына, И.И. Попов – М.: Форум: Инфра-М, 2011.
4. Канцедал, С.А. Алгоритмизация и программирование : Учебное пособие / С.А. Канцедал.. - М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2015. - 352 с.
- Микрюков, В.Ю. Алгоритмизация и программирование: Учебное пособие / В.Ю. Микрюков. - Рн/Д: Феникс, 2015. - 304 с.
5. Незнанов, А.А. Программирование и алгоритмизация: Науч. ред. В.П. Кутепов. - М.: ИЦ Академия, 2014. - 304 с.
4. Новичков, В.С. Алгоритмизация и программирование на Турбо Паскале / В.С. Новичков, Н.И. Парфилова. - М.: ГЛТ, 2015. - 438 с.
5. Эпштейн М.С. Практикум по программированию: учебное пособие для сред. проф. образования. – М.: Издательский центр «Академия», 2009.
6. Лесневский А.С. Объектно-ориентированное программирование для начинающих. – М.: Бином. Лаборатория знаний, 2009.
7. Ахо А., Хопкрофт Дж., Ульман Дж., Построение и анализ вычислительных алгоритмов. – М.: Мир, 1989. – 369с.
8. Никлаус Вирт. Алгоритмы и структуры данных. – Санкт-Петербург: «Невский диалект», 2001.
9. Альсведе Р., Вегенер И. Задачи поиска.– М.: Мир, 1982. – 368 с.
- 10.Бауэр Ф.Л., Гооз Г., Информатика. Вводный курс, в 2-ух ч. – М., Мир,1981. – 368с.
- 11.Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. – М.: Мир, 1982. – 416 с.
- 12.Дал У., Дейкстра Э., Хоор К. Структурное программирование. – М.: Мир, 1975.
- 13.Калинин А.Г., Мацкевич И.В. Универсальные языки программирования. Семантический подход.– Радио и связь, 1991.
- 14.Кристофидес Н. Теория графов. Алгоритмический подход.–М.: Мир, 1978. – 432 с.

Интернет-источники:

1. [Электронный ресурс] <http://www.codenet.ru>
2. [Электронный ресурс] <http://www.chemisk.narod.ru/html/algoritm01.html>
3. Университетская библиотека ONLINE: <http://biblioclub.ru>
4. Научная электронная библиотека eLIBRARY.RU
5. Университетская информационная система РОССИЯ: <http://uisrussia.msu.ru/>.

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Контроль и оценка результатов освоения учебной дисциплины осуществляется преподавателем в процессе проведения практических занятий и лабораторных работ, тестирования, а также выполнения обучающимися индивидуальных заданий, проектов, исследований.

Оценка качества освоения учебной программы включает текущий контроль успеваемости, промежуточную аттестацию по итогам освоения дисциплины.

Текущий контроль проводится в форме решения подобных задач, устного опроса, тестирования, самостоятельных работ, практических работ, контрольных работ

Промежуточная аттестация проводится в форме дифференцированного зачёта

Результаты обучения: умения, знания и общие компетенции в соответствии с таблицей 2 ФГОС по УД	Показатели оценки результата	Форма контроля и оценивания
Уметь:		
<p>У1. использовать языки программирования, строить логически правильные и эффективные программы;</p> <p>ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.</p> <p>ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.</p> <p>ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.</p> <p>ОК 09. Использовать информационные технологии в профессиональной деятельности.</p> <p>ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке.</p>	<ul style="list-style-type: none"> - построение логически правильных и эффективных программы; - использование языков программирования для реализации профессиональных задач - осуществление поиска и использование информации, необходимой для эффективного выполнения профессиональных задач - использование информационно-коммуникационные технологии в профессиональной деятельности 	Выполнение и защита лабораторных работ
Знать:		
<p>З1. общие принципы построения алгоритмов, основные алгоритмические конструкции;</p> <p>ПК 1.1. Формировать алгоритмы</p>	<ul style="list-style-type: none"> - перечисление общих правил построения алгоритмов; - иллюстрирование основных алгоритмических конструкций; - нахождение путей решения 	<p>Устный опрос</p> <p>Тестирование</p> <p>Проверочная работа</p> <p>Внеаудиторная</p>

<p>разработки программных модулей в соответствии с техническим заданием.</p> <p>ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.</p> <p>ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.</p> <p>ПК 1.4. Выполнять тестирование программных модулей.</p> <p>ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода.</p> <p>ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p> <p>ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.</p>	<p>модулей в соответствии с заданием;</p> <p>- анализ произведенных изменений и документирование их в виде оформленного отчета</p>	<p>самостоятельная работа</p> <p>Дифференцированный зачёт</p>
<p>32. понятие системы программирования;</p> <p>ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.</p> <p>ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.</p> <p>ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.</p> <p>ПК 1.4. Выполнять тестирование программных модулей.</p> <p>ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода.</p> <p>ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p> <p>ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.</p>	<p>- формулирование понятия системы программирования;</p> <p>- реализация языка программирования на практических задачах;</p> <p>- анализ синтаксических конструкций;</p>	

<p>33 основные элементы процедурного языка программирования, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, кассы памяти;</p> <p>ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.</p> <p>ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.</p> <p>ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.</p> <p>ПК 1.4. Выполнять тестирование программных модулей.</p> <p>ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода.</p> <p>ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p> <p>ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.</p>	<ul style="list-style-type: none"> - систематизация элементов процедурного языка программирования; - анализ структуры программы; - разграничение понятий оператора и операций в среде программирования; - выявление управляющей структуры данных для реализации в программном коде; - описание пути к файлам, стандартные функции для работы с файлами их реализация в листинге программного продукта; - представление класса памяти в графической форме; - отладка и тестирование программного продукта;
<p>34 подпрограммы, составление библиотек программ;</p> <p>ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.</p> <p>ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.</p> <p>ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.</p> <p>ПК 1.4. Выполнять тестирование программных модулей.</p> <p>ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода.</p> <p>ПК 2.4. Осуществлять разработку тестовых наборов и тестовых</p>	<ul style="list-style-type: none"> - формулировка понятия подпрограммы; - описание библиотеки подпрограмм с дальнейшей реализацией в программном коде; - изобретение методов и средств реализующих объекты профессионального модуля

<p>сценариев для программного обеспечения.</p> <p>ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.</p>	
<p>35 объектно-ориентированную модель программирования, понятие классов и объектов, их свойств и методов</p> <p>ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.</p> <p>ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.</p> <p>ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.</p> <p>ПК 1.4. Выполнять тестирование программных модулей.</p> <p>ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода.</p> <p>ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p> <p>ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.</p>	<ul style="list-style-type: none"> - систематизация объектно-ориентированной модели программирования; - объяснение понятия класса и объектов; - представление свойств и методов в объектно-ориентированном языке программирования;

Лист утверждения рабочей программы дисциплины
на учебный год

Рабочая программа дисциплины (модуля) утверждена на ведение
учебного процесса в учебном году:

Учебный год	«СОГЛАСОВАНО»: заведующий кафедрой, ответственной за РПД (подпись и дата)
2020- 2021	
2021 – 2022	
2022 - 2023	
2023 - 2024	
2024- 2025	