МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ С.А. ЕСЕНИНА»

Утверждаю:
Декан физико-математического
факультета
Н.Б. Федорова
«30» августа 2019 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

ОСНОВЫ ПРОГРАММИРОВАНИЯ

Уровень основной профессиональной образовательной программы:
бакалавриат

Направление подготовки: **02.03.03 Математическое обеспечение и администрирование информационных систем**

Направленность (профиль) подготовки: **Администрирование информационных систем**

Форма обучения: **очная**

Срок освоения ОПОП: **нормативный срок освоения 4 года**

Факультет: **физико-математический**

Кафедра: **Информатики, вычислительной техники и методики преподавания информатики**

Рязань, 2019

ВВОДНАЯ ЧАСТЬ

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью освоения учебной дисциплины «Основы программирования» является формирование у обучающихся общепрофессиональных компетенций в процессе изучения алгоритмизации и основ программирования для последующего применения в учебной и практической деятельности.

Задачи дисциплины:

- формирование основ современной культуры программирования;
- формирование теоретической базы и практических умений и навыков для решения задач на компьютере в императивных системах программирования.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВУЗА

2.1. Дисциплина Б1.О.09.01 «Основы программирования» относится к модулю Программирование I обязательной части Блока 1.

2.2. Для изучения данной дисциплины необходимы следующие предшествующие дисциплины:

– *Алгоритмизация и основы структурного программирования*.

2.3. Перечень последующих дисциплин, для которых необходимы знания, умения и навыки, формируемые данной дисциплиной:

– *Объектно-ориентированное и визуальное программирование;*
– *Структуры и алгоритмы компьютерной обработки данных;*
– *Технология разработки параллельных программ;*
– *Функциональное программирование;*
– *Кроссплатформенное программирование*.

2.4 Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения основной профессиональной образовательной программы

Изучение данной дисциплины направлено на формирование у обучающихся общепрофессиональных компетенций (ОПК):

№ п/п	Код и содержание компетенции	Код и наименование индикатора достижения компетенции	Перечень планируемых результатов обучения по дисциплине В результате изучения дисциплины обучающиеся должны:		
			Знать	Уметь	Владеть
1	*2*	*3*	*4*	*5*	*6*
1.	ОПК-3. Способен применять современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения	ОПК-3.1 Способен осуществлять выбор современные языков, утилит и сред программирования, типовых решений, компонентов библиотек программных модулей, шаблонов, классов объектов, используемых при создании программных продуктов и программных комплексов различного назначения, в том числе с учетом отечественных опыта и наработок	• основные современные алгоритмические языки программирования, среды императивного программирования, типовые решения императивного программирования, компоненты библиотек программных модулей, используемых при создании программных продуктов и программных комплексов различного назначения, в том числе с учетом отечественных опыта и наработок	• осуществлять выбор современных алгоритмических языков программирования, сред императивного программирования, типовых решений императивного программирования, компонентов библиотек программных модулей, используемых при создании программных продуктов и программных комплексов различного назначения, в том числе с учетом отечественных опыта и наработок	• навыками выбора современных алгоритмических языков программирования, сред императивного программирования, типовых решений императивного программирования, компонентов библиотек программных модулей, используемых при создании программных продуктов и программных комплексов различного назначения, в том числе с учетом отечественных опыта и наработок

2.		ОПК-3.2 Способен применять современные технологии проектирования, разработки, разметки и форматирования программного кода, в том числе отечественные, при создании программного обеспечения и баз данных, интерфейсов программных продуктов и баз данных.	• этапы решения задачи на компьютере на основе парадигмы императивного программирования; • структуру простейшей программы на языке программирования Си, синтаксис и семантику основных операторов языка Си (присваивания, ввода-вывода, ветвления, циклов); • основные синтаксические конструкции, используемые в языке программирования Си для описания и обработки данных типа массив; • основные синтаксические конструкции, используемые в языках программирования Питон, Паскаль и Си для описания и обработки строк; • синтаксис типов запись с фиксированными полями, указатель и типизированный файл в языке Паскаль; • синтаксис типов структура, указатель и файл в языке Си; • понятия статического и динамического объекта про-	• при решении учебной задачи конкретизировать и описать основные понятия, данные и их типы, организовать ввод и вывод, реализовать алгоритмические структуры средствами, по крайней мере, трех языков программирования; • разрабатывать алгоритмы и программы обработки массивов и строк; • разрабатывать простейшие алгоритмы и программы моделирования движения графического изображения; • использовать тип указатель для организации односвязного списка, очереди и стека; • решать учебные задачи по обработке данных типа запись, файл и линейных динамических структур данных средствами систем программирования Pascal и C++.	• навыками применения современных технологий проектирования, разработки, разметки и форматирования программного кода в средах, по крайней мере, трех систем программирования.

			граммы на языках Паскаль и Си; • синтаксис и семантику основных средств управления использованием динамической памяти в языках Паскаль и Си; • структуру данных файл и основные действия с файлами последовательного доступа; • синтаксис и семантику основных средств работы с файлами в системах программирования Pascal и C++ и последовательность их использования для организации ввода и вывода.		

3.		ОПК-3.3. Способен применять современные компиляторы, отладчики и оптимизаторы программного кода, осуществлять сборку модулей и компонентов программного кода, разрабатывать и реализовывать межмодульный интерфейс.	• понятие алгоритмического модуля и его свойства; • принципы выделения и правила записи алгоритмических модулей; • семантику вызова алгоритмического модуля, в том числе, рекурсивного вызова, и механизмы реализации межмодульного интерфейса; • синтаксис и семантику реализации алгоритмического модуля средствами языков программирования Питон, Паскаль и Си; • способы передачи параметров при реализации межмодульного интерфейса.	• решать учебные задачи в соответствие с принципом модульности; • разрабатывать и использовать библиотеку пользователя в системах программирования Pascal и C++.	• навыками применения современных компиляторов, отладчиков и оптимизаторов программного кода, сборки модулей и компонентов программного кода, разработки и реализации межмодульного интерфейса в соответствии с принципом модульности в средах, по крайней мере, трех систем программирования.
4.	ОПК-6. Способен использовать в педагогической деятельности научные основы знаний в сфере информационно-коммуникационных технологий	ОПК-6.4. Способен создавать, редактировать и конвертировать в основные форматы графических файлов эксплуатационно-техническую документацию программной продукции или технологии в виде схем, диаграмм, инфографики, копий экранов графического интерфейса в средах графических редакторов общего и специального назначения.	• средства записи алгоритмов в виде схем; • схемы алгоритмических структур, их основные свойства и приемы использования.	• разрабатывать и записывать алгоритмы в виде схем; • анализировать структуру эксплуатационно-технической документации программной продукции в виде схем.	• навыками разработки, записи и анализа алгоритмов в виде схем.

ОСНОВНАЯ ЧАСТЬ
1. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ

Вид учебной работы		**Всего часов**	**Семестры**	
			№ 2	**№ 3**
			часов	**часов**
1		2	4	5
Контактная работа обучающихся с преподавателем (по видам учебных занятий) (всего)		**192**	**108**	**84**
В том числе:				
Лекции (Л)		52	36	16
Лабораторные работы (ЛР)		140	72	68
Самостоятельная работа студента (всего)		**204**	**108**	**96**
В том числе:				
Изучение литературы и других источников		52	26	26
Подготовка к выполнению лабораторных работ		93	51	42
Подготовка к защите лабораторных работ		59	31	28
Контроль		**72**	**36**	**36**
Вид промежуточной аттестации	**зачет (З)**			
	экзамен (Э)		+	+
ИТОГО: общая трудоемкость	**часов**	**468**	**252**	**216**
	зач. ед.	**13**	**7**	**6**

2. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ
2.1. Содержание разделов дисциплины

№ семестра	№ раздела	Наименование раздела дисциплины	Содержание раздела в дидактических единицах
1	2	3	4
2	1	Символьный и строковый типы данных	Стандартный идентификатор строкового типа в языках Python и Pascal. Строковый тип в языке Pascal как массив символов. Операция конкатенации в языках Python и Pascal. Срез строки в языке Python. Функции и процедуры для работы со строками в языке Pascal. Примеры алгоритмов и программ обработки данных строкового типа. Написание, ввод, отладка и тестирование программ обработки строк в СП Python и Pascal
2	2	Разработка алгоритмов в соответствии с принципом модульности	Понятие алгоритмического модуля, его свойства и правила выделения. Концепция “черного ящика”. Правила записи алгоритмического модуля: спецификация и тело модуля. Входные и выходные формальные параметры. Блок вызова алгоритмического модуля. Фактические параметры. Семантика вызова алгоритмического модуля и механизм реализации межмодульного интерфейса. Понятие рекурсии. Семантика рекурсивного вызова. Примеры простейших алгоритмов, составленных в соответствии с принципом модульности. Разработка схем алгоритмов в соответствии с принципом модульности

1	2	3	4
2	3	Реализация принципа модульности средствами алгоритмических языков программирования	Способы записи алгоритмического модуля средствами алгоритмических языков программирования: процедуры и функции. Синтаксис процедуры, синтаксис и семантика оператора процедуры в языке Python. Способы передачи параметров значением и ссылкой. Синтаксис процедуры, синтаксис и семантика оператора процедуры в языке Pascal. Параметры значения и параметры переменные. Рекомендации по выбору способа передачи параметров. Примеры простейших программ, составленных в соответствии с принципом модульности. Синтаксис функции в языках Python и Pascal. Рекомендации по использованию функций в императивных СП Python и Pascal. Примеры простейших программ с использованием функций пользователя. Синтаксис процедурного типа в языке Pascal. Рекомендации по использованию процедурного типа для решения задач: параметры процедуры и параметры функции. Пример алгоритма и программы решения задачи с использованием процедурного типа. Примеры алгоритма и программ решения задачи рекурсивной природы. Понятие о библиотеке СП Pascal. Обзор стандартных модулей. Синтаксис раздела модулей программы и особенности трансляции процедур и функций модуля СП Pascal. Синтаксис модуля пользователя и рекомендации по организации библиотеки пользователя в СП Pascal. Длинная целочисленная арифметика. Примеры программ сложения и вычитания натуральных чисел с числом разрядов больше десяти. Написание, ввод, отладка и тестирование программ в соответствии с принципом модульности в СП Python и Pascal
2	4	Структурированный тип данных запись	Синтакис типа запись с фиксированными полями в языке Pascal. Рекомендации по описанию переменной типа запись, примеры. Использование переменной типа запись в операторе присваивания. Селектор записи (переменная с точкой). Синтаксис оператора присоединения. Примеры алгоритмов и программ обработки данных типа запись. Разработка схем алгоритмов обработки данных типа запись. Написание, ввод, отладка и тестирование программ обработки данных типа запись в соответствии с принципом модульности в СП Pascal
2	5	Тип данных указатель	Понятия статических и динамических объектов программы на языке Паскаль. Синтаксис типа указатель (ссылочный тип) в языке Pascal. Рекомендации по описанию переменной типа указатель, примеры. Синтаксис и семантика оператора резервирования (захвата) памяти под динамическую переменную в языке Pascal, примеры. Использование переменной типа указатель в операторе присваивания и в выражениях отношения с операциями = и <>. Динамическая переменная (переменная с крышкой). Синтаксис и семантика освобождения области памяти динамической переменной в языке Pascal. Понятие динамических структур данных. Использование типа указатель для организации динамического односвязного списка по принципам ссылки вперед и ссылки назад. Примеры алгоритмов и программ создания динамических односвязных списков. Алгоритмы вставки (добавления), удаления и поиска элемента динамического односвязного списка. Структуры данных на основе односвязных списков: очередь и стек. Рекомендации по созданию и обработке динамических структур данных очередь и стек с примерами фрагментов алгоритмов и программ. Разработка схем алгоритмов создания и обработки динамических структур данных. Написание, ввод, отладка и тестирование программ создания и обработки динамических структур данных в соответствии с принципом модульности в СП Pascal

1	2	3	4
2	6	Структурированный тип данных файл	Понятие файла, файлы данных различного доступа. Основные действия с файлами данных последовательного доступа: запись (вывод в файл) и чтение (ввод из файла). Обзор файловых типов данных в языке Pascal, синтаксис типизированного файлового типа. Рекомендации по описанию переменной типа файл, примеры. Синтаксис и семантика основных операторов для работы с типизированными файлами данных в СП Pascal: связь файловой переменной с файлом на внешнем устройстве, открытие файла для записи и чтения, вывод в файл и ввод из файла, закрытие файла. Функция конец типизированного файла. Рекомендации по корректной обработке типизированных файлов с примерами фрагментов алгоритмов и программ. Разработка схем алгоритмов создания и обработки типизированных файлов. Написание, ввод, отладка и тестирование программ создания и обработки типизированных файлов в соответствии с принципом модульности в СП Pascal
2	7	Обзор графических возможностей систем программирования (СП) Python и Pascal	Понятие о графическом режиме монитора, текущее положение графического курсора. Создание окна для графики в языке Python. Обзор стандартных средств языка Python для изображения графических примитивов: точек, отрезков, прямоугольников, окружностей и эллипсов, дуг окружностей и эллипсов. Операторы стандартных процедур модуля Graph библиотеки СП Pascal для определения подходящего графического драйвера, инициализации и прекращения графического режима и задания текущего цвета изображения. Обзор стандартных процедур модуля Graph для изображения графических примитивов: точек, отрезков, прямоугольников, окружностей и эллипсов, дуг окружностей и эллипсов. Простейший алгоритм моделирования движения графического изображения (анимация). Написание, ввод, отладка и тестирование программ моделирования движения графического изображения в соответствии с принципом модульности в СП Python и Pascal
3	8	Парадигмы программирования	Основные парадигмы программирования: императивная, событийно-управляемая (объектно-ориентированная и параллельная), декларативная (функциональная и логическая). Архитектура и возможности семейства языков высокого уровня
3	9	Общая характеристика языков программирования семейства Си	Особенности языков программирования семейства Си. Структура простейшей программы на языке Си: раздел препроцессора и раздел функций. Синтаксис разделов программы, составной оператор. Понятие о библиотеке языка Си. Трансляция программ в СП (СП) семейства Си. Основные простые типы данных в языке Си: идентификаторы типов, модификаторы типов, диапазоны значений, синтаксис и семантика оператора резервирования памяти, инициализация переменных, операции и функции, преобразование типов. Некоторые особенности операций языка Си: перегрузка операции деления, операция присваивания и возможные варианты записи выражений с использованием операции присваивания, операции инкремента и декремента в префиксной и постфиксной форме, операции взятия адреса и содержимое по указанному адресу. Знакомство с операторами потокового ввода/вывода. Общая характеристика СП C++
3	10	Разработка разветвляющихся алгоритмов и прорамм в СП C++	Интерпретация данных целого типа как данных логического типа в языке Си. Операции отношения и логические операции. Организация развилок средствами языка Си: синтаксис и семантика условных оператора и операции. Операция запятая. Примеры организации развилок средствами Си. Синтаксис и семантика оператора переключателя. Организация посредством переключателя структуры выбор, пример. Разработка схем алгоритмов разветвляющейся структуры. Написание, ввод, отладка и тестирование программ разветвляющейся структуры в СП C++

1	2	3	4
3	11	Разработка циклических алгоритмов и программ в СП С++	Синтаксис и семантика операторов циклов в языке Си: циклы с предусловием и постусловием, цикл for. Особенности системной организации цикла for в СП С++ . Рекомендации по программированию циклов. Примеры циклических алгоритмов и программ. Разработка схем алгоритмов циклической структуры. Написание, ввод, отладка и тестирование программ циклической структуры в СП С++
3	12	Ввод с клавиатуры и вывод на монитор средствами С++	Синтаксис и семантика операторов потокового ввода/вывода. Манипуляторы. Примеры организации потокового ввода/вывода. Синтаксис и семантика функций форматированного ввода/вывода. Синтаксис формата ввода, символы преобразования и модификаторы. Примеры организации форматированного ввода. Синтаксис формата вывода, управляющие символы. Примеры организации форматированного вывода. Написание, ввод, отладка и тестирование программ с форматированным выводом на монитор в СП С++
3	13	Массивы в языке Си	Синтаксис и семантика оператора резервирования памяти под массив, примеры. Идентификатор массива как указатель на область памяти, захваченной под массив. Использование переменной типа массив в выражении с операцией присваивания. Особенности распределения памяти под элементы массива и инициализация массивов в СП семейства Си. Традиционная индексная и альтернативная, с помощью операции “*”, адресация элементов массива, примеры. Примеры обработки массивов средствами языка Си. Разработка схем алгоритмов обработки массивов. Написание, ввод, отладка и тестирование программ обработки массивов в СП С++
3	14	Строки в языке Си	Строки как массивы символов. Особенности инициализации и организации ввода строк в языке Си. Стандартные функции обработки строк в языке Си. Алгоритмы копирования, удаления и замены части строки средствами языка Си, примеры. Разработка схем алгоритмов обработки строк. Написание, ввод, отладка и тестирование программ обработки строковых данных в СП С++
3	15	Реализация принципа модульности средствами языка Си	Синтаксис функции в языке Си. Формальные и фактические параметры. Способы передачи параметра простого типа. Рекомендации по использованию функций в программах на С++, примеры. Параметры массивы и параметры функции, примеры. Функции с аргументами по умолчанию, примеры. Перегрузка функций, пример. Классы памяти в языке Си. Рекомендации по организации простейшей библиотеки пользователя средствами СП С++ . Разработка схем алгоритмов в соответствии с принципом модульности. Написание, ввод, отладка и тестирование программ в соответствии с принципом модульности в СП С++
3	16	Тип данных структура	Синтаксис типа структура в языках семейства Си. Оператор определения типа пользователя, пример. Определение типа структура с помощью тэга. Рекомендации по описанию переменной типа структура, примеры. Использование переменной типа структура в выражении с операцией присваивания. Член структуры (идентификатор с точкой). Разработка схем алгоритмов обработки данных типа структура. Написание, ввод, отладка и тестирование программ обработки данных типа структура в соответствии с принципом модульности в СП С++

1	2	3	4
3	17	Динамические переменные в языках семейства Си	Синтаксис типа указатель (ссылочный тип) в языках семейства Си. Рекомендации по описанию переменной типа указатель. Синтаксис и семантика оператора захвата памяти под динамическую переменную. Идентификатор динамической переменной в языках семейства Си (идентификатор со звездочкой). Синтаксис и семантика освобождения области памяти динамической переменной. Синтаксис и семантика операторов захвата и освобождения области памяти для динамического массива. Использование типа указатель для организации односвязного списка, определение типа элемента односвязного списка с помощью тэга. Адресация члена динамической переменной типа структура с помощью литеры алфавита “–>” (идентификатор со стрелкой). Рекомендации по организации вставки (добавления), удаления и поиска элементов линейных динамических структур данных. Примеры алгоритмов создания и обработки линейных динамических структур данных и соответствующих программ на языке С++. Разработка схем алгоритмов создания и обработки линейных динамических структур данных. Написание, ввод, отладка и тестирование программ создания и обработки линейных динамических структур данных в соответствии с принципом модульности в СП С++
3	18	Тип файл в СП С++	Организация файлов данных в языках семейства Си. Синтаксис и семантика захвата памяти под указатель на буфер файла. Синтаксис и семантика функции открытия файла. Атрибуты режима по умолчанию для записи (вывода), чтения (ввода) и добавления данных. Атрибуты режимов текстового и бинарного файла данных. Синтаксис и семантика функций форматированной записи данных в файл и форматированного чтения данных из файла. Функция конец файла в СП С++ . Рекомендации для СП С++ по корректному открытию файла и чтению данных из файла, примеры. Синтаксис и семантика функции закрытия файла. Примеры алгоритма и программы обработки файла данных, в которых для хранения промежуточных результатов используется линейная динамическая структура. Разработка схем алгоритмов обработки файлов данных. Написание, ввод, отладка и тестирование программ создания и текстовых файлов в соответствии с принципом модульности в СП С++

2.2. Перечень лабораторных работ

Лабораторная работа №1. Написание, ввод, отладка и тестирование алгоритмов и программ обработки строк

Лабораторная работа №2. Написание, ввод, отладка и тестирование алгоритмов и программ обработки массивов строк

Лабораторная работа №3. Написание, ввод, отладка и тестирование алгоритмов и программ в соответствие с принципом модульности

Лабораторная работа №4. Написание, ввод, отладка и тестирование алгоритмов и программ обработки массивов в соответствие с принципом модульности

Лабораторная работа №5. Написание, ввод, отладка и тестирование алгоритмов и программ создания и обработки односвязных списков, записей и типизированных файлов в СП Pascal

Лабораторная работа №6. Написание, ввод, отладка и тестирование алгоритмов и программ моделирования движения графического изображения в соответствии с принципом модульности

Лабораторная работа №7. Знакомство с императивной системой программирования C++. Написание, ввод, отладка и тестирование алгоритмов и программ разветвляющейся структуры в СП C++

Лабораторная работа №8. Написание, ввод, отладка и тестирование разветвляющихся алгоритмов и программ с использованием структуры выбор в СП C++

Лабораторная работа №9. Написание, ввод, отладка и тестирование циклических алгоритмов и программ в СП C++

Лабораторная работа №10. Написание, ввод, отладка и тестирование циклических алгоритмов и программ целочисленной арифметики в СП C++

Лабораторная работа №11. Написание, ввод, отладка и тестирование циклических алгоритмов и программ с форматированным выводом на монитор в СП C++

Лабораторная работа №12. Написание, ввод, отладка и тестирование алгоритмов и программ обработки массивов в СП C++

Лабораторная работа №13. Написание, ввод, отладка и тестирование алгоритмов и программ обработки строковых данных в СП C++

Лабораторная работа №14. Написание, ввод, отладка и тестирование алгоритмов и программ в соответствие с принципом модульности в СП C++

Лабораторная работа №15. Написание, ввод, отладка и тестирование алгоритмов и программ создания и обработки односвязных списков, структур и файлов в СП C++

3. САМОСТОЯТЕЛЬНАЯ РАБОТА СТУДЕНТА

Самостоятельная работа осуществляется в объеме 204 часов.
Видами СРС являются:

- изучение литературы и других источников;
- подготовка к выполнению лабораторной работы;
- подготовка к защите лабораторной работы.

Формами текущего контроля успеваемости являются:

- защита лабораторной работы.

4. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

(см. фонд оценочных средств)

5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

5.1. Основная литература

№ п/п	Автор (ы), наименование, место издания и издательство, год
1	2
1.	Шелудько, В.М. Язык программирования высокого уровня Python: функции, структуры данных, дополнительные модули / В.М. Шелудько ; Министерство науки и высшего образования РФ, Южный федеральный университет, Институт компьютерных технологий и информационной безопасности. – Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2017. – 108 с. : ил. – Режим доступа: по подписке. – URL: http://biblioclub.ru/index.php?page=book&id=500060 (дата обращения: 30.08.2019)
2.	Комлев, Н.Ю. Самоучитель игры на Паскале. ABC и немного Турбо : практическое пособие / Н.Ю. Комлев. – Москва : СОЛОН-ПРЕСС, 2013. – 256 с. – Режим доступа: по подписке. – URL: http://biblioclub.ru/index.php?page=book&id=227109 (дата обращения: 30.08.2019)
3.	Каширин И.Ю., Новичков В.С. От С к С++ [Текст]: Учебное пособие для вузов.- М.: Горячая линия – Телеком, 2005

5.2. Дополнительная литература

п/п	Автор (ы), наименование, место издания и издательство, год
1	*2*
1	Северенс, Ч. Введение в программирование на Python / Ч. Северенс. – 2-е изд., испр. – Москва : Национальный Открытый Университет «ИНТУИТ», 2016. – 231 с. : схем., ил. – Режим доступа: по подписке. – URL: http://biblioclub.ru/index.php?page=book&id=429184 (дата обращения: 30.08.2019)
2	Непейвода, Н. Н. Стили и методы программирования [Электронный ресурс] / Н. Н. Непейвода. - Москва : Интернет-Университет Информационных Технологий, 2005. - 320 с. - Режим доступа: http://biblioclub.ru/index.php?page=book&id=233198 (дата обращения: 30.08.2019).
3	Информатика и программирование. Алгоритмизация и программирование [Текст]: учебник для студ. учреждений высш. проф. образования/ [Н.И.Парфилова, А.В.Пруцков, А.Н.Пылькин, Б.Г.Трусов]; под ред. Б.Г.Трусова.-М.: Издательский центр «Академия», 2012.-336 с.
4	Савич, У. Программирование на С++ [Текст] / У. Савич. – 4-е изд. – Санкт-Петербург : Питер, 2004. – 781 с.

5.3. Базы данных, информационно-справочные и поисковые системы:

1. BOOR.ru [Электронный ресурс] : электронная библиотека. – Доступ к полным текстам по паролю. – Режим доступа: http://www.book.ru (дата обращения: 30.08.2019).
2. East View [Электронный ресурс] : [база данных]. – Доступ к полным текстам статей научных журналов из сети РГУ имени С.А. Есенина. – Режим доступа: http:/dlib.eastview.com (дата обращения: 30.08.2019).
3. Moodle [Электронный ресурс] : среда дистанционного обучения / Ряз. гос. ун-т. – Рязань, [Б.г.]. – Доступ, после регистрации из сети РГУ имени С.А. Есенина, из любой точки, имеющей доступ к Интернету. – Режим доступа: http://e-learn2.rsu.edu.ru/moodle2 (дата обращения: 30.08.2019).
4. Znanium.com [Электронный ресурс] : [база данных]. – Доступ к полным текстам по паролю. – Режим доступа: http://znanium.com (дата обращения: 30.08.2019).

5. «Издательство «Лань» [Электронный ресурс] : электронная библиотека. – Доступ к полным текстам по паролю. – Режим доступа: http://e-lanbook.com (дата обращения: 30.08.2019).

6. Университетская библиотека ONLINE [Электронный ресурс] : электронная библиотека. – Доступ к полным текстам по паролю. – Режим доступа: http://www.biblioclab.ru (дата обращения: 30.08.2019).

7. Юрайт [Электронный ресурс] : электронная библиотека. – Доступ к полным текстам по паролю. – Режим доступа: http://www.biblio-online.ru (дата обращения: 30.08.2019).

8. Труды преподавателей [Электронный ресурс] : коллекция // Электронная библиотека Научной библиотеки РГУ имени С.А. Есенина. – Доступ к полным текстам по паролю. – Режим доступа: http://dspace.rsu.edu.ru/xmlui/handle/123456789/3 (дата обращения: 30.08.2019).

5.4. Перечень ресурсов информационно-телекоммуникационной сети «Интернет» (далее – сеть «Интернет»), необходимых для освоения дисциплины

1. eLIBRARY.RU [Электронный ресурс] : научная электронная библиотека. – Режим доступа: http://elibrary.ru/defaultx.asp, свободный (дата обращения: 30.08.2019).

2. Единая коллекция цифровых образовательных ресурсов [Электронный ресурс] : федеральный портал. – Режим доступа: http://school-collection.edu.ru/, свободный (дата обращения: 30.08.2019).

3. Единое окно доступа к образовательным ресурсам [Электронный ресурс] : федеральный портал. – Режим доступа: http://window.edu.ru/, свободный (дата обращения: 30.08.2019).

4. Интернет Университет Информационных технологий. [Электронный ресурс] : сайт. – Режим доступа: http://www.intuit.ru/, свободный (дата обращения 30.08.2019).

5. Петров Д.Н. Парадигмы программирования. [Электронный ресурс] : сайт. – Режим доступа: http://dnpetrov.narod.ru/, свободный (дата обращения 30.08.2019).

6. Портал естественных наук. [Электронный ресурс] : сайт. – Режим доступа: http://e-science11.ru, свободный (дата обращения 30.08.2019).

7. Портал для программистов и администраторов информационных систем. [Электронный ресурс] : сайт. – Режим доступа: http://www.coderpost.net/, свободный (дата обращения 30.08.2019).

8. Российский общеобразовательный портал [Электронный ресурс] : образовательный портал. – Режим доступа: http://www.school.edu.ru/, свободный (дата обращения: 30.08.2019).

9. Сервер Информационных Технологий [Электронный ресурс] : сайт. – Режим доступа: http://citforum.ru/, свободный (дата обращения 30.08.2019).

10. Сайт программирования. [Электронный ресурс] : сайт. – Режим доступа: http://www.cyberguru.ru/, свободный (дата обращения 30.08.2019).

11. Федеральный центр информационно-образовательных ресурсов [Электронный ресурс] : Единое окно доступа к образовательным ресурсам. – Режим доступа: http://fcior.edu.ru, свободный (дата обращения: 30.08.2019).

5.5. Периодические издания

1. Компьютерные и информационные науки. Доступ: Киберленинка [Электронный ресурс] : научная электронная библиотека. – Режим доступа: https://cyberleninka.ru/article/c/computer-and-information-sciences, свободный (дата обращения: 30.08.2019).
2. Электротехника, электронная техника, информационные технологии. Доступ: Киберленинка [Электронный ресурс] : научная электронная библиотека. – Режим доступа: https://cyberleninka.ru/article/c/electrical-electronic-information-engineering, свободный (дата обращения: 30.08.2019).
3. Архив научных статей из журнала «Программирование» [Электронный ресурс]. – Режим доступа: https://naukarus.com/j/programmirovanie, свободный (дата обращения: 30.08.2019).

6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. Требования к аудиториям (помещениям, местам) для проведения занятий: специализированные лекционные аудитории, оборудованные видеопроекционным оборудованием для презентаций.

6.2. Требования к оборудованию рабочих мест преподавателя и обучающихся: видеопроектор, ноутбук, переносной экран, для проведения демонстраций, рабочие места обучающихся оснащены ПК с доступом в Интернет.

6.3. Требования к специализированному оборудованию отсутствуют

7. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Вид учебных занятий	Организация деятельности студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; помечать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначить вопросы, термины, материал, который вызывает трудности, пометить и попытаться найти ответ в рекомендуемой литературе. Если самостоятельно не удается разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на консультации, на практическом занятии. Уделить внимание следующим понятиям: *этапы решения задачи на компьютере; основные понятия алгоритмической системы; принципы пошаговой детализации структурности и модульности; алгоритмические структуры; вложение; структурный анализ; алфавит, синтаксис и семантика языка программирования; программа для компьютера; идентификатор; переменная; данные и их типы, операция, стандартная функция, выражение; простой и структурированный оператор*.
Лабораторная работа	В соответствии с запланированным на самостоятельную работу временем составить схемы алгоритмов и программы решения соответствующего варианта учебной задачи. Согласовать заранее составленные схемы и программы с преподавателем, ведущим занятие. Тексты программ должны содержать короткие комментарии, отражающие тему и номер лабораторной работы, номер варианта, фамилию студента, связь тех или иных переменных с условием задачи, а также комментарии, отражающие этапы решения задачи. Ввод с клавиатуры и вывод на монитор числовых данных должны сопровождаться краткими текстовыми сообщениями. Ввести тексты программ в компьютер, осуществить отладку и тестирование программ, при наличии нескольких программ решения одной и той же задачи добиться с учетом возможных погрешностей одинаковых результатов и продемонстрировать работоспособность программ преподавателю. Оформить отчет о лабораторной работе с указанием фамилии студента, номера лабораторной работы и номера варианта. Оформленная работа также должна содержать полный текст задания, схему алгоритма (кроме л/р № 1), тексты отлаженных на компьютере программ с комментариями по всем переменным. После текста программы на конкретном языке программирования должны быть отражены результаты ее выполнения из окна результатов в том виде, в котором они были выведены системой программирования. В случае ветвящихся алгоритмов и программ необходимо отразить несколько вариантов выполнения программы с различными наборами исходных данных, соответствующих различным ветвям исполнения алгоритма. Защитить оформленную лабораторную работу, продемонстрировав теоретические и практические знания, умения и навыки по соответствующей теме, возможные варианты схем решения задачи, структуры алгоритмов которых отличны от структур оформленных схем.

Подготовка к экзамену (зачету)	При подготовке к экзамену (зачету) необходимо ориентироваться на конспекты лекций, рекомендуемую литературу, типовые практические задания и др.

8. ТРЕБОВАНИЯ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ УЧЕБНОГО ПРОЦЕССА

1. Операционная система Windows Pro (договор №Tr000043844 от 22.09.15г)
2. Антивирус Kaspersky Endpoint Security (договор №02-ЗК-2019 от 15.04.2019г.)
3. Офисное приложение LibreOffice (свободно распространяемое ПО)
4. Среда разработки приложений RAD Studio 10.1 Berlin Professional Concurrent ELC (договор № 11\05\2016-9774 11.05.16г.)
5. Система программирования Python (свободно распространяемое ПО)
6. Система программирования PascalABC (свободно распространяемое ПО)
7. Архиватор 7-zip (свободно распространяемое ПО);
8. Браузер изображений Fast Stone Image Viewer (свободно распространяемое ПО);
9. PDFридер Foxit Reader (свободно распространяемое ПО);
10. Медиа проигрыватель VLC mediaplayer (свободно распространяемое ПО);
11. Запись дисков ImageBurn (свободно распространяемое ПО);
12. DJVU браузер DjVu Browser Plug-in (свободно распространяемое ПО)
13. Запись дисков ImageBurn (свободно распространяемое ПО)
14. DJVU браузер DjVu Browser Plug-in (свободно распространяемое ПО