

Минобрнауки России
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»

УТВЕРЖДАЮ

И.о. проректора по учебной работе

_____ Василенко В.Н.
(подпись) (Ф.И.О.)
" _30_ " _____ 05 _____ 2024 г.

**РАБОЧАЯ ПРОГРАММА
ДИСЦИПЛИНЫ**

Алгоритмизация и программирование

Направление подготовки

15.04.03 Прикладная механика

Направленность (профиль) подготовки

**Математическое и компьютерное моделирование
механических систем и процессов**

Квалификация выпускника

Магистр

Воронеж

1. Цели и задачи дисциплины

Целями освоения дисциплины «Алгоритмизация и программирование» является формирование компетенций обучающегося в области профессиональной деятельности и сфере профессиональной деятельности:

28 Производство машин и оборудования (в сфере повышения надежности и долговечности работы деталей, узлов и механизмов);

40 Сквозные виды профессиональной деятельности в промышленности (в сферах: обеспечения необходимой динамики, прочности, устойчивости, рациональной оптимизации, долговечности, ресурса, живучести, надежности и безопасности машин, конструкций, композитных структур, сооружений, установок, агрегатов, оборудования, приборов и аппаратуры и их элементов, расчетно-экспериментальных работ с элементами научных исследований в области прикладной механики, разработки и проектирования новой техники и технологий).

Дисциплина направлена на решение задач профессиональной деятельности следующих типов:

- производственно-технологический;
- научно-исследовательский;
- проектно-конструкторский.

Программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта высшего образования по направлению подготовки 15.04.03 «Прикладная механика» (уровень образования - магистр).

2. Перечень планируемых результатов обучения, соотнесенных с планируемыми результатами освоения образовательной программы

Процесс изучения дисциплины направлен на формирование следующих компетенций (таблица).

№ п/п	Код компетенции	Наименование компетенции	Код и наименование индикатора достижения компетенции
1	ОПК-12	Способен создавать алгоритмы цифровой обработки баз данных результатов испытаний и эксплуатации сложных деталей и узлов в машиностроении, разрабатывать современные цифровые программы расчетов и проектирования деталей, узлов, конструкций, машин и материалов с учетом требований надежности, долговечности и безопасности их эксплуатации.	ИД1 _{опк-12} – Создает алгоритмы цифровой обработки баз данных результатов испытаний и эксплуатации сложных деталей и узлов в машиностроении,
			ИД2 _{опк-12} – Разрабатывает цифровые программы расчетов и проектирования деталей, узлов, конструкций, машин и материалов с учетом требований надежности, долговечности и безопасности их эксплуатации

Код и наименование индикатора достижения компетенции	Результаты обучения (показатели оценивания)
ИД1 _{опк-12} – Создает алгоритмы	Знает: этапы решения задачи на компьютере на основе парадигмы императивного программирования; понятие алгоритма и его основ-

<p>цифровой обработки баз данных результатов испытаний и эксплуатации сложных деталей и узлов в машиностроении,</p>	<p>ные свойства; основные понятия алгоритмической системы: представление информации в виде данных, система команд исполнителя, алгоритмический язык исполнителя; основные средства записи и типы алгоритмов; алгоритмические структуры, их основные свойства и приемы использования; 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</p> <p>Умеет: при решении стандартных задач профессиональной деятельности на основе парадигмы императивного программирования конкретизировать и описать основные понятия, результаты других научных дисциплин, данные и их типы средствами записи алгоритмов и программ; разрабатывать и записывать алгоритмы и программы на языках высокого уровня в соответствии с технологией нисходящего структурного проектирования; записывать алгоритмы на языке схем; анализировать структуру алгоритмов; описывать алфавит и основные синтаксические конструкции языков программирования.</p> <p>Владеет: основными методами, способами и средствами переработки информации на основе парадигмы императивного программирования в соответствии с технологией нисходящего структурного проектирования.</p>
<p>ИД2ОПК-12 – Разрабатывает цифровые программы расчетов и проектирования деталей, узлов, конструкций, машин и материалов с учетом требований надежности, долговечности и безопасности их эксплуатации</p>	<p>Знает: составные части алгоритмического языка программирования (алфавит, синтаксис, семантика) и способы их описания; концепцию типов данных; структуры программ на языках программирования; основные понятия алгоритмических языков программирования и их классификацию: идентификаторы, данные и их типы, операции, стандартные функции, выражения, операторы; синтаксис и семантику основных операторов языков программирования (присваивания, ввода-вывода, ветвления, циклов), способы реализации алгоритмических структур средствами языков программирования; основные синтаксические конструкции, используемые в языках программирования для описания и обработки данных типа массив; способы передачи параметров при реализации межмодульного интерфейса; структуру данных файл и основные действия с файлами последовательного доступа.</p> <p>Умеет: <input type="checkbox"/> разрабатывать и записывать алгоритмы и программы на языках высокого уровня в соответствии с принципом структурности; при решении учебной задачи конкретизировать и описать основные понятия, данные и их типы, организовать ввод и вывод, реализовать алгоритмические структуры средствами, по крайней мере, двух императивных языков программирования; разрабатывать алгоритмы и программы обработки массивов; разрабатывать алгоритмы и программы обработки строк; решать учебные задачи в соответствии с принципом модульности; 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.</p> <p>Владеет: навыками проектирования, ввода, отладки и тестирования программ в средах, по крайней мере, двух императивных систем программирования.</p>

3. Место дисциплины в структуре образовательной программы ВО

Дисциплина «Алгоритмизация и программирование» относится к блоку один ОП и ее обязательной части.

Курс вариативной части цикла обязательных дисциплин **«Алгоритмизация и программирование»** базируется на знаниях, умениях и компетенциях, полученных при изучении дисциплин «Физика», «Математика», «Информатика».

Дисциплина **«Алгоритмизация и программирование»** является предшествующей для освоения дисциплин: «Вычислительная механика и компьютерный инжиниринг», «Теория планирования и методы экспериментальных исследований в механике», *Теории упругости и пластичности, Производственная практика, преддипломная практика.*

4. Объем дисциплины и виды учебной работы

Общая трудоемкость дисциплины (модуля) составляет 7 зачетных единиц.

Виды учебной работы	Всего часов	Семестр	
		2	3
	акад.	акад.	акад.
Общая трудоемкость дисциплины	252	108	144
Контактная работа, в т.ч. аудиторные занятия:	96,85	58,05	38,8
Лекции	31	19	12
<i>в том числе в форме практической подготовки</i>			
Лабораторные занятия (ЛЗ)	62	38	24
<i>в том числе в форме практической подготовки</i>			
Консультации текущие	1,55	0,95	0,6
Консультация перед экзаменом	2	-	2
Виды аттестации (зачет/экзамен)	0,3	0,1	0,2
Самостоятельная работа:	121,35	49,95	71,4
Подготовка к защите по практическим занятиям (собеседование)	40,9	26,9	14
Изучение материалов по учебникам (собеседование, тестирование, подготовка к аудиторным контрольным работам)	30,8	11,4	19,4
Изучение материалов, изложенных в лекциях (собеседование, тестирование, подготовка к аудиторным контрольным работам)	48,05	10,05	38
Выполнение домашних контрольных работ	1,6	1,6	-
Контроль	33,8	-	33,8

5 Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

5.1 Содержание разделов дисциплины

№ п/п	Наименование раздела дисциплины	Содержание раздела	Трудоемкость раздела, часы
2 семестр			
1.	Алгоритмическая система и ее составные части	Алгоритмическая система как совокупность средств и понятий. Понятие алгоритма и исполнителя алгоритма. Представление информации в виде данных: понятие о вводе/выводе; входные, выходные и промежуточные данные. Система команд исполнителя. Константы, переменные, операции, функции и выражения. Понятие типа данных. Алгоритмический язык исполнителя. Принципы фон Неймана как основа структуры	6

		персонального компьютера. Представление в компьютере целых чисел. Представление в компьютере действительных чисел. Средства записи алгоритмов. Схемы алгоритмов	
2.	Основные понятия и простейшие средства алгоритмических языков программирования	Программирование как этап решения задачи на компьютере. Понятие языка программирования высокого уровня. Понятие о исполнителе программы на языке высокого уровня. Составные части языка высокого уровня: алфавит, синтаксис, семантика. Метаязыки описания конструкций языков программирования. Расширенные Бэкуса-Наура формы, примеры метаформул. Структуры программ и концепции типов данных в алгоритмических языках. Оператор как логически завершенная конструкция алгоритмического языка программирования. Различные классификации операторов: исполняемые и неисполняемые, простые и составные. Различные классификации типов данных: простые и структурированные, стандартные и определяемые пользователем. Основные простые типы данных в языках программирования: идентификаторы стандартных типов, диапазоны, операции и функции, синтаксис типа диапазон. Особенности целочисленной и действительной арифметики в системах программирования. Синтаксис и семантика оператора присваивания, знакомство с операторами ввода/вывода. Написание, ввод, отладка и тестирование программ линейной структуры.	12
3.	Технология нисходящего структурного проектирования. Разработка разветвляющихся алгоритмов и программ в соответствии с принципом структурности	Цели и составные части технологии нисходящего структурного проектирования. Взаимосвязь принципов нисходящего проектирования, модульности и структурности. Алгоритмические структуры как основа структурного кодирования. Базовые алгоритмические структуры: следование и развилка. Вложение (суперпозиция) алгоритмических структур как основной прием принципа структурности. Присоединение алгоритмических структур. Правила пунктуации в программах на алгоритмическом языке. Составной оператор в языке программирования. Логический тип данных в алгоритмических языках. Операции отношения и логические операции. Алгоритмические структуры, реализующие ветвления. Организация ветвлений средствами алгоритмических языков: синтаксис и семантика строчного и блочного условных операторов. Понятие о приеме программирования "флаг". Примеры алгоритмов и программ разветвляющейся структуры. Разработка схем алгоритмов разветвляющейся структуры. Написание, ввод, отладка и тестирование программ разветвляющейся структуры.	18
4.	Разработка циклических алгоритмов и программ в соответствии с принципом структурности	Базовая алгоритмическая структура цикл с предусловием. Цикл с постусловием. Проблема закливания. Цикл с параметром. Синтаксис и семантика операторов циклов. Особенности системной организации циклов с параметром. Рекомендации по программированию циклов. Примеры циклических алгоритмов и программ. Разработка схем алгоритмов циклической структуры. Написание, ввод, отладка и тестирование программ циклической структуры.	14
5	Ввод с клавиатуры	Понятие о текстовом режиме монитора, текущее	10

	и вывод на монитор средствами алгоритмических языков	положение курсора. Синтаксис и семантика операторов ввода с клавиатуры. Особенности ввода с клавиатуры символьной и строковой информации. Общая характеристика операторов вывода на монитор. Синтаксис оператора PRINT. Организация слитного и зонального вывода средствами оператора PRINT, примеры. Особенности вывода значений числового и строкового типа. Организация форматированного вывода средствами оператора PRINT USING: синтаксис, символы форматирования, основные форматы для вывода значений строкового и числового типа, семантика. Синтаксис и семантика оператора вывода на монитор. Форматированный вывод текстовой и числовой информации на монитор. Диалоговые программы. Дружественность интерфейса с пользователем. Примеры организации форматированного вывода на монитор. Разработка схем алгоритмов, написание, ввод, отладка и тестирование программ с форматированным выводом на монитор.	
6	Структурированный тип данных массив	Синтаксис и семантика оператора резервирования памяти, примеры. Особенности распределения памяти. Синтаксис определения типа массив. Рекомендации по описанию переменной типа массив и понятие полной совместимости типов, примеры. Использование переменной типа массив в операторе присваивания. Элемент массива (переменная с индексами). Основные алгоритмы и программы обработки числовых массивов: поиск максимума (минимума), поиск элементов с заданными свойствами, сортировка методом пузырька, модифицированный алгоритм сортировки методом пузырька. Примеры алгоритмов и программ обработки массивов, в том числе с использованием приема "флаг". Разработка схем алгоритмов обработки массивов. Написание, ввод, отладка и тестирование программ обработки массивов.	8
7	Символьный и строковый типы данных	Стандартный идентификатор строкового типа. Строковый тип в языке программирования как массив символов. Операция конкатенации. Функции и процедуры для работы со строками. Примеры алгоритмов и программ обработки данных строкового типа. Написание, ввод, отладка и тестирование программ обработки строк.	4
8	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	
3 семестр			
9	Реализация принципа модульности средствами алгоритмических	Способы записи алгоритмического модуля средствами алгоритмических языков программирования: процедуры и функции. Синтаксис процедуры,	16

	языков программирования	синтаксис и семантика оператора процедуры. Способы передачи параметров значением и ссылкой. Синтаксис процедуры, синтаксис и семантика оператора процедуры. Параметры значения и параметры переменные. Рекомендации по выбору способа передачи параметров. Примеры простейших программ, составленных в соответствии с принципом модульности. Рекомендации по использованию функций. Примеры простейших программ с использованием функций пользователя. 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. Обзор стандартных модулей. Синтаксис раздела модулей программы и особенности трансляции процедур и функций. Синтаксис модуля пользователя и рекомендации по организации библиотеки пользователя. 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	
10	Структурированный тип данных запись	Синтаксис типа запись с фиксированными полями. 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.	84
11	Тип данных указатель	Понятия статических и динамических объектов программы на языке высокого уровня. Синтаксис типа указатель (ссылочный тип) в языке программирования. Рекомендации по описанию переменной типа указатель, примеры. Синтаксис и семантика оператора резервирования (захвата) памяти под динамическую переменную в языке программирования, 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 программирования. 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 обработ-	28

		ки динамических структур данных. Написание, ввод, отладка и тестирование программ создания и обработки динамических структур данных в соответствии с принципом модульности.	
12	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 программирования, синтаксис типизированного файлового типа. Рекомендации по описанию переменной типа файл, примеры. Синтаксис и семантика основных операторов для работы с типизированными файлами данных: 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.	14
13	Обзор графических возможностей систем программирования на языках высокого уровня	Понятие о графическом режиме монитора, текущее положение графического курсора. Операторы установки параметров окна результатов и задания текущего цвета изображения в языках программирования. Обзор стандартных средств языка программирования для изображения графических примитивов: точек, отрезков, прямоугольников, окружностей и эллипсов, дуг окружностей и эллипсов. Операторы стандартных процедур модуля библиотеки 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.	8

5.2 Разделы дисциплины и виды занятий

№ п/п	Наименование раздела дисциплины	Лекции, час	Лабораторные работы (ЛР), час	СРО, час
2 семестр				
1.	Алгоритмическая система и ее составные части	2	4	6
2.	Основные понятия и простейшие средства алгоритмических языков программирования	2	4	6
3.	Технология нисходящего структурного проектирования. Разработка разветвляющихся алгоритмов и программ в соответствии с принципом структурности	4	6	6,5
4.	Разработка циклических алгоритмов и программ в соответствии с принципом структурности	4	6	7,45
5.	Ввод с клавиатуры и вывод на монитор средствами алгоритмических языков	1	4	6
6.	Структурированный тип данных массив	2	4	6
7.	Символьный и строковый типы данных	2	6	6
8.	Разработка алгоритмов в соответствии с принципом модульности	2	4	6
3 семестр				
9.	Реализация принципа модульности средствами алгоритмических языков программирования	4	4	20
10.	Структурированный тип данных запись	2	6	15
11.	Тип данных указатель	2	4	15
12.	Структурированный тип данных файл	2	4	10
13.	Обзор графических возможностей систем программирования на языках высокого уровня	2	6	11,4

5.2.1 Лекции

№ п/п	Наименование раздела дисциплины	Тематика лекционных занятий	Трудоемкость раздела, часы
2 семестр			
1.	Алгоритмическая система и ее составные части	Алгоритмическая система как совокупность средств и понятий. Понятие алгоритма и исполнителя алгоритма. Представление информации в виде данных: понятие о вводе/выводе; входные, выходные и промежуточные данные. Система команд исполнителя. Константы, переменные, операции, функции и выражения. Понятие типа данных. Алгоритмический язык исполнителя. Принципы фон Неймана как основа структуры персонального компьютера. Представление в компьютере целых чисел. Представление в компьютере действительных чисел. Средства записи алгоритмов. Схемы алгоритмов	2
2.	Основные понятия и простейшие средства алгоритмических языков программирования	Программирование как этап решения задачи на компьютере. Понятие языка программирования высокого уровня. Понятие о исполнителе программы на языке высокого уровня. Составные части языка высокого уровня: алфавит, синтаксис, семантика. Метаязыки описания конструкций языков программирования. Расширенные Бэкуса-Наура формы, примеры метаформул. Структуры программ и концепции типов данных в алгоритмических языках. Оператор как логически завершенная конструкция алгоритмического языка про-	2

		граммирования. Различные классификации операторов: исполняемые и неисполняемые, простые и составные. Различные классификации типов данных: простые и структурированные, стандартные и определяемые пользователем. Основные простые типы данных в языках программирования: идентификаторы стандартных типов, диапазоны, операции и функции, синтаксис типа диапазон. Особенности целочисленной и действительной арифметики в системах программирования. Синтаксис и семантика оператора присваивания, знакомство с операторами ввода/вывода. Написание, ввод, отладка и тестирование программ линейной структуры.	
3.	Технология нисходящего структурного проектирования. Разработка разветвляющихся алгоритмов и программ в соответствии с принципом структурности	Цели и составные части технологии нисходящего структурного проектирования. Взаимосвязь принципов нисходящего проектирования, модульности и структурности. Алгоритмические структуры как основа структурного кодирования. Базовые алгоритмические структуры: следование и развилка. Вложение (суперпозиция) алгоритмических структур как основной прием принципа структурности. Присоединение алгоритмических структур. Правила пунктуации в программах на алгоритмическом языке. Составной оператор в языке программирования. Логический тип данных в алгоритмических языках. Операции отношения и логические операции. Алгоритмические структуры, реализующие ветвления. Организация ветвлений средствами алгоритмических языков: синтаксис и семантика строчного и блочного условных операторов. Понятие о приеме программирования "флаг". Примеры алгоритмов и программ разветвляющейся структуры. Разработка схем алгоритмов разветвляющейся структуры. Написание, ввод, отладка и тестирование программ разветвляющейся структуры.	4
4.	Разработка циклических алгоритмов и программ в соответствии с принципом структурности	Базовая алгоритмическая структура цикла с предусловием. Цикл с постусловием. Проблема закливания. Цикл с параметром. Синтаксис и семантика операторов циклов. Особенности системной организации циклов с параметром. Рекомендации по программированию циклов. Примеры циклических алгоритмов и программ. Разработка схем алгоритмов циклической структуры. Написание, ввод, отладка и тестирование программ циклической структуры.	4
5	Ввод с клавиатуры и вывод на монитор средствами алгоритмических языков	Понятие о текстовом режиме монитора, текущее положение курсора. Синтаксис и семантика операторов ввода с клавиатуры. Особенности ввода с клавиатуры символьной и строковой информации. Общая характеристика операторов вывода на монитор. Синтаксис оператора PRINT. Организация слитного и зонального вывода средствами оператора PRINT, примеры. Особенности вывода значений числового и строкового типа. Организация форматированного вывода средствами оператора PRINT USING: синтаксис, символы форматирования, основные форматы для вывода значений строкового и числового типа, семантика. Синтаксис и семантика оператора вывода на монитор. Форматированный вывод текстовой и числовой информации на монитор. Диалоговые	1

		программы. Дружественность интерфейса с пользователем. Примеры организации форматированного вывода на монитор. Разработка схем алгоритмов, написание, ввод, отладка и тестирование программ с форматированным выводом на монитор.	
6	Структурированный тип данных массив	Синтаксис и семантика оператора резервирования памяти, примеры. Особенности распределения памяти. Синтаксис определения типа массив. Рекомендации по описанию переменной типа массив и понятие полной совместимости типов, примеры. Использование переменной типа массив в операторе присваивания. Элемент массива (переменная с индексами). Основные алгоритмы и программы обработки числовых массивов: поиск максимума (минимума), поиск элементов с заданными свойствами, сортировка методом пузырька, модифицированный алгоритм сортировки методом пузырька. Примеры алгоритмов и программ обработки массивов, в том числе с использованием приема "флаг". Разработка схем алгоритмов обработки массивов. Написание, ввод, отладка и тестирование программ обработки массивов.	2
7	Символьный и строковый типы данных	Стандартный идентификатор строкового типа. Строковый тип в языке программирования как массив символов. Операция конкатенации. Функции и процедуры для работы со строками. Примеры алгоритмов и программ обработки данных строкового типа. Написание, ввод, отладка и тестирование программ обработки строк.	2
8	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	2
3 семестр			
9	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. Способы передачи параметров значением и ссылкой. Синтаксис процедуры, синтаксис и семантика оператора процедуры. Параметры значения и параметры переменные. Рекомендации по выбору способа передачи параметров. Примеры простейших программ, составленных в соответствии с принципом модульности. Рекомендации по использованию функций. Примеры простейших программ с использованием функций пользователя. Рекомендации по использованию процедурного типа для решения задач: параметры процедуры и параметры функции. Пример алгоритма и программы решения задачи с использованием процедурного типа. Примеры алго-	4

		ритма и программ решения задачи рекурсивной природы. Понятие о библиотеке. Обзор стандартных модулей. Синтаксис раздела модулей программы и особенности трансляции процедур и функций. Синтаксис модуля пользователя и рекомендации по организации библиотеки пользователя. 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	
10	Структурированный тип данных запись	Синтаксис типа запись с фиксированными полями. 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.	2
11	Тип данных указатель	Понятия статических и динамических объектов программы на языке высокого уровня. Синтаксис типа указатель (ссылочный тип) в языке программирования. Рекомендации по описанию переменной типа указатель, примеры. Синтаксис и семантика оператора резервирования (захвата) памяти под динамическую переменную в языке программирования, 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 программирования. 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.	2
12	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 программирования, синтаксис типизированного файлового типа. Рекомендации по описанию переменной типа файл, примеры. Синтаксис и семантика основных операторов для работы с типизированными файлами данных: связь файловой переменной с файлом на внешнем устройстве, открытие файла для записи и чтения,	2

		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.	
13	Обзор графических возможностей систем программирования на языках высокого уровня	Понятие о графическом режиме монитора, текущее положение графического курсора. Операторы установки параметров окна результатов и задания текущего цвета изображения в языках программирования. Обзор стандартных средств языка программирования для изображения графических примитивов: точек, отрезков, прямоугольников, окружностей и эллипсов, дуг окружностей и эллипсов. Операторы стандартных процедур модуля библиотеки 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.	2

5.2.2 Практические занятия (планом не предусмотрено)

5.2.3 Лабораторный практикум

№ п/п	Наименование раздела дисциплины	Тематика лабораторных работ	Трудоемкость раздела, часы
2 семестр			
1.	Алгоритмическая система и ее составные части	ЛР вводная. <i>Знакомство с императивными системами программирования. Ввод, отладка и тестирование простейших программ</i>	4
2.	Основные понятия и простейшие средства алгоритмических языков программирования	ЛР №1. <i>Написание, ввод, отладка и тестирование программ линейной структуры</i>	4
3.	Технология нисходящего структурного проектирования. Разработка разветвляющихся алгоритмов и программ в соответствии с принципом структурности	ЛР №2. <i>Написание, ввод, отладка и тестирование алгоритмов и программ разветвляющейся структуры</i>	3
		ЛР №3. <i>Написание, ввод, отладка и тестирование разветвляющихся алгоритмов и программ с использованием структуры выбор</i>	3
4.	Разработка циклических алгоритмов и программ в соответствии с принципом структурности	ЛР №4. <i>Написание, ввод, отладка и тестирование циклических алгоритмов и программ</i>	3
		ЛР №5. <i>Написание, ввод, отладка и тестирование циклических алгоритмов и программ целочисленной арифметики</i>	3
5	Ввод с клавиатуры и вывод на монитор средствами алгоритми-	ЛР №6. <i>Написание, ввод, отладка и тестирование циклических алгоритмов и программ с форматированным выводом на монитор</i>	4

	чешских языков		
6	Структурированный тип данных массив	ЛР №7. <i>Написание, ввод, отладка и тестирование алгоритмов и программ обработки числовых массивов</i>	4
7	Символьный и строковый типы данных	ЛР №8. <i>Написание, ввод, отладка и тестирование алгоритмов и программ обработки строк</i>	2
		ЛР №9. <i>Написание, ввод, отладка и тестирование алгоритмов и программ обработки массивов строк</i>	4
8	Разработка алгоритмов в соответствии с принципом модульности	ЛР №10. <i>Написание, ввод, отладка и тестирование алгоритмов и программ в соответствии с принципом модульности</i>	4
3 семестр			
9	Реализация принципа модульности средствами алгоритмических языков программирования	ЛР №11. <i>Написание, ввод, отладка и тестирование алгоритмов и программ обработки массивов в соответствии с принципом модульности</i>	4
10	Структурированный тип данных запись	ЛР №12. <i>Написание, ввод, отладка и тестирование алгоритмов и программ создания и обработки односвязных списков, записей и типизированных файлов</i>	10
11	Тип данных указатель		
12	Структурированный тип данных файл	ЛР №13. <i>Написание, ввод, отладка и тестирование алгоритмов и программ длинной арифметики в соответствии с принципом модульности</i>	4
13	Обзор графических возможностей систем программирования на языках высокого уровня	ЛР №14. <i>Написание, ввод, отладка и тестирование алгоритмов и программ моделирования движения графического изображения в соответствии с принципом модульности</i>	6

5.2.4 Самостоятельная работа обучающихся (СРО)

№ п/п	Наименование раздела дисциплины	Вид СРО	Трудоемкость , час
2 семестр			
1.	Алгоритмическая система и ее составные части	Проработка материалов по конспекту лекций и учебникам. Подготовка к выполнению и защите лабораторной работы	6
2.	Основные понятия и простейшие средства алгоритмических языков программирования	Проработка материалов по конспекту лекций и учебникам. Подготовка к выполнению и защите лабораторной работы	6
3.	Технология нисходящего структурного проектирования. Разработка Разветвляющихся алгоритмов и программ в соответствии с принципом структурности	Проработка материалов по конспекту лекций и учебникам. Подготовка к выполнению и защите лабораторной работы	6,5
4.	Разработка циклических алгоритмов и программ в соответствии с принципом структурности	Проработка материалов по конспекту лекций и учебникам. Подготовка к выполнению и защите лабораторной работы	7,45
5	Ввод с клавиатуры и вывод на монитор средствами алгоритмических языков	Проработка материалов по конспекту лекций и учебникам. Подготовка к выполнению и защите лабораторной работы	6
6	Структурированный тип данных массив	Проработка материалов по конспекту лекций и учебникам. Подготовка к выполнению и защите лабораторной работы	6
7	Символьный и строковый типы дан-	Проработка материалов по конспекту	6

	ных	лекций и учебникам. Подготовка к выполнению и защите лабораторной работы	
8	Разработка алгоритмов в соответствии с принципом модульности	Проработка материалов по конспекту лекций и учебникам. Подготовка к выполнению и защите лабораторной работы	6
3 семестр			
9	Реализация принципа модульности средствами алгоритмических языков программирования	Проработка материалов по конспекту лекций и учебникам. Подготовка к выполнению и защите лабораторной работы	20
10	Структурированный тип данных запись	Проработка материалов по конспекту лекций и учебникам. Подготовка к выполнению и защите лабораторной работы	15
11	Тип данных указатель	Проработка материалов по конспекту лекций и учебникам. Подготовка к выполнению и защите лабораторной работы	15
12	Структурированный тип данных файл	Проработка материалов по конспекту лекций и учебникам. Подготовка к выполнению и защите лабораторной работы	10
13	Обзор графических возможностей систем программирования на языках высокого уровня	Проработка материалов по конспекту лекций и учебникам. Подготовка к выполнению и защите лабораторной работы	11,4

6 Учебно-методическое и информационное обеспечение дисциплины

6.1 Основная литература

1. Ульянова, Н. Д. Основные принципы алгоритмизации : учебно-методическое пособие / Н. Д. Ульянова. — Брянск : Брянский ГАУ, 2020. — 56 с. <https://e.lanbook.com/book/172114>
2. Юрина, Т. А. Программирование и алгоритмизация : учебно-методическое пособие / Т. А. Юрина. — Омск : СибАДИ, 2021. — 88 с. <https://e.lanbook.com/book/179228>
3. Кривцов, А. Н. Алгоритмизация и программирование. Основы программирования на C/C++ : учебное пособие / А. Н. Кривцов, С. В. Хорошенко. — Санкт-Петербург : СПбГУТ им. М.А. Бонч-Бруевича, 2020. — 202 с. <https://e.lanbook.com/book/180057>

6.2 Дополнительная литература

1. Рачишкин, А. А. Основы алгоритмизации и программирование на языках высокого уровня : учебное пособие / А. А. Рачишкин. — Тверь : ТвГТУ, 2018. — 132 с. <https://e.lanbook.com/book/171310>

6.3 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся

1. Ахмедханлы, Д. М. Основы алгоритмизации и программирования : учебно-методическое пособие / Д. М. Ахмедханлы, Н. В. Ушмаева. — Тольятти : ТГУ, 2016. — 123 с. <https://e.lanbook.com/book/139958>

6.4 Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля)

Наименование ресурса сети «Интернет»	Электронный адрес ресурса
Научная электронная библиотека	http://www.elibrary.ru/defaulttx.asp
Образовательная платформа «Юрайт»	https://urait.ru/

ЭБС «Лань»	https://e.lanbook.com/
АИБС «МегаПро»	https://biblos.vsu.ru/MegaPro/Web
Сайт Министерства науки и высшего образования РФ	http://minobrnauki.gov.ru
Электронная информационно-образовательная среда ФГБОУ ВО «ВГУИТ»	http://education.vsu.ru

6.5 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем

При изучении дисциплины используется программное обеспечение, современные профессиональные базы данных и информационные справочные системы: ЭИОС университета, в том числе на базе программной платформы «Среда электронного обучения 3KL».

При освоении дисциплины используется лицензионное и открытое программное обеспечение

Программы	Лицензии, реквизиты подтверждающего документа
Adobe Reader XI	(бесплатное ПО) https://acrobat.adobe.com/ru/ru/acrobat/pdf-reader/volume-distribution.html
Альт Образование	Лицензия № AAA.0217.00 с 21.12.2017 г. по «Бессрочно»
Microsoft Windows 8	Microsoft Open License
Microsoft Windows 8.1	Microsoft Windows Professional 8 Russian Upgrade Academic OPEN 1 License No Level#61280574 от 06.12.2012 г. https://www.microsoft.com/ru-ru/licensing/licensing-programs/open-license
Microsoft Office Professional Plus 2010	Microsoft Open License Microsoft Office Professional Plus 2010 Russian Academic OPEN 1 License No Level #48516271 от 17.05.2011 г. https://www.microsoft.com/ru-ru/licensing/licensing-programs/open-license Microsoft Open License Microsoft Office Professional Plus 2010 Russian Academic OPEN 1 License No Level #61181017 от 20.11.2012 г. https://www.microsoft.com/ru-ru/licensing/licensing-programs/open-license
Microsoft Office 2007 Standart	Microsoft Open License Microsoft Office 2007 Russian Academic OPEN No Level #44822753 от 17.11.2008 https://www.microsoft.com/ru-ru/licensing/licensing-programs/open-license
Libre Office 6.1	Лицензия № AAA.0217.00 с 21.12.2017 г. по «Бессрочно» (Включен в установочный пакет операционной системы Альт Образование 8.2)

Справочно-правовые системы

Программы	Лицензии, реквизиты подтверждающего документа
Справочные правовая система «Консультант Плюс»	Договор о сотрудничестве с «Информсвязь-черноземье», Региональный информационный центр общероссийской сети распространения правовой информации Консультант Плюс № 8-99/RD от 12.02.1999 г.

7 Материально-техническое обеспечение дисциплины

Материально-техническая база приведена в лицензионных формах и расположена по ад-

ресу <https://vsuet.ru>

Для проведения учебных занятий используются учебные аудитории:

Ауд. № 127. Учебная аудитория для проведения учебных занятий	Машина испытания на растяжение МР-0,5, машина испытания на кручение КМ-50, машина универсальная разрывная УММ-5, машина испытания пружин МИП-100, машина разрывная УГ 20/2, машина испытания на усталость МУИ6000, копер маятниковый
Ауд. № 227. Учебная аудитория для проведения учебных занятий	Интерактивная доска SMART Board SB660 64, комплект лабораторного оборудования для проведения дисциплины "Детали машин и основы конструирования": машина тарировочная, прибор ТММ105-1, стенды методические
Ауд. № 127а. Компьютерный класс	Моноблок Гравитон - 12 шт.
Ауд. № 133. Учебная аудитория для проведения учебных занятий	Переносное мультимедийное оборудование: проектор View Sonic PJD 5232, экран на штативе Di-gisKonturCDSKS-1101

Самостоятельная работа обучающихся может осуществляться при использовании:
Зал научной литературы ресурсного центра ВГУИТ: компьютеры Regard - 12 шт.
Студенческий читальный зал ресурсного центра ВГУИТ: моноблоки - 16 шт.

8 Оценочные материалы для промежуточной аттестации обучающихся по дисциплине

8.1 Оценочные материалы (ОМ) для дисциплины включают в себя:

- перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы;
- описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания;
- типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы;
- методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

8.2 Для каждого результата обучения по дисциплине определяются показатели и критерии оценивания сформированности компетенций на различных этапах их формирования, шкалы и процедуры оценивания.

ОМ представляются отдельным комплектом и входят в состав рабочей программы дисциплины.

Оценочные материалы формируются в соответствии с П ВГУИТ «Положение об оценочных материалах».

ПРИЛОЖЕНИЕ 1
к рабочей программе

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ
ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

по дисциплине
«Алгоритмизация и программирование»

1 Перечень компетенций с указанием этапов их формирования

Процесс изучения дисциплины направлен на формирование следующих компетенций (таблица).

№ п/п	Код компетенции	Наименование компетенции	Код и наименование индикатора достижения компетенции
1	ОПК-12	Способен создавать алгоритмы цифровой обработки баз данных результатов испытаний и эксплуатации сложных деталей и узлов в машиностроении, разрабатывать современные цифровые программы расчетов и проектирования деталей, узлов, конструкций, машин и материалов с учетом требований надежности, долговечности и безопасности их эксплуатации.	ИД1 _{опк-12} – Создает алгоритмы цифровой обработки баз данных результатов испытаний и эксплуатации сложных деталей и узлов в машиностроении.
			ИД2 _{опк-12} – Разрабатывает цифровые программы расчетов и проектирования деталей, узлов, конструкций, машин и материалов с учетом требований надежности, долговечности и безопасности их эксплуатации

Код и наименование индикатора достижения компетенции	Результаты обучения (показатели оценивания)
ИД1 _{опк-12} – Создает алгоритмы цифровой обработки баз данных результатов испытаний и эксплуатации сложных деталей и узлов в машиностроении,	Знает: этапы решения задачи на компьютере на основе парадигмы императивного программирования; понятие алгоритма и его основные свойства; основные понятия алгоритмической системы: представление информации в виде данных, система команд исполнителя, алгоритмический язык исполнителя; основные средства записи и типы алгоритмов; алгоритмические структуры, их основные свойства и приемы использования; 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
	Умеет: при решении стандартных задач профессиональной деятельности на основе парадигмы императивного программирования конкретизировать и описать основные понятия, результаты других научных дисциплин, данные и их типы средствами записи алгоритмов и программ; разрабатывать и записывать алгоритмы и программы на языках высокого уровня в соответствие с технологией нисходящего структурного проектирования; записывать алгоритмы на языке схем; анализировать структуру алгоритмов; описывать алфавит и основные синтаксические конструкции языков программирования.
	Владеет: основными методами, способами и средствами переработки информации на основе парадигмы императивного программирования в соответствии с технологией нисходящего структурного проектирования.

ИД2 _{ОПК-12} – Разрабатывает цифровые программы расчетов и проектирования деталей, узлов, конструкций, машин и материалов с учетом требований надежности, долговечности и безопасности их эксплуатации	Знает: составные части алгоритмического языка программирования (алфавит, синтаксис, семантика) и способы их описания; концепцию типов данных; структуры программ на языках программирования; основные понятия алгоритмических языков программирования и их классификацию: идентификаторы, данные и их типы, операции, стандартные функции, выражения, операторы; синтаксис и семантику основных операторов языков программирования (присваивания, ввода-вывода, ветвления, циклов), способы реализации алгоритмических структур средствами языков программирования; основные синтаксические конструкции, используемые в языках программирования для описания и обработки данных типа массив; способы передачи параметров при реализации межмодульного интерфейса; структуру данных файл и основные действия с файлами последовательного доступа.
	Умеет: разрабатывать и записывать алгоритмы и программы на языках высокого уровня в соответствии с принципом структурности; при решении учебной задачи конкретизировать и описать основные понятия, данные и их типы, организовать ввод и вывод, реализовать алгоритмические структуры средствами, по крайней мере, двух императивных языков программирования; разрабатывать алгоритмы и программы обработки массивов; разрабатывать алгоритмы и программы обработки строк; решать учебные задачи в соответствии с принципом модульности; 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.
	Владеет: навыками проектирования, ввода, отладки и тестирования программ в средах, по крайней мере, двух императивных систем программирования.

2. Паспорт фонда оценочных средств по дисциплине

№ п/п	Контролируемые модули/разделы/темы дисциплины	Индекс контролируемой компетенции (или ее части)	Оценочные средства		Технология оценки (способ контроля)
			наименование	№№ заданий	
1.	Алгоритмическая система и ее составные части	ОПК-12	Банк тестовых заданий Устный опрос	1-2 30-31	Компьютерное тестирование Процентная шкала. 0-100 %; 0-59,99% - неудовлетворительно; 60-74,99% - удовлетворительно; 75- 84,99% -хорошо; 85-100% - отлично.
2.	Основные понятия и простейшие средства алгоритмических языков программирования	ОПК-12	Банк тестовых заданий Устный опрос	3-4 32-33	Компьютерное тестирование Процентная шкала. 0-100 %; 0-59,99% - неудовлетворительно; 60-74,99% - удовлетворительно; 75- 84,99% -хорошо;

					85-100% - отлично.
3	Технология нисходящего структурного проектирования. Разработка разветвляющихся алгоритмов и программ в соответствии с принципом структурности	ОПК-12	Банк тестовых заданий Устный опрос	5-6 34-35	Проверка преподавателем Компьютерное тестирование Процентная шкала. 0-100 %; 0-59,99% - неудовлетворительно; 60-74,99% - удовлетворительно; 75- 84,99% -хорошо; 85-100% - отлично.
4	Разработка циклических алгоритмов и программ в соответствии с принципом структурности	ОПК-12	Банк тестовых заданий Устный опрос	7-10 36-37	Компьютерное тестирование Процентная шкала. 0-100 %; 0-59,99% - неудовлетворительно; 60-74,99% - удовлетворительно; 75- 84,99% -хорошо; 85-100% - отлично.
5	Ввод с клавиатуры и вывод на монитор средствами алгоритмических языков	ОПК-12	Банк тестовых заданий Устный опрос	11-13 38-39	Компьютерное тестирование Процентная шкала. 0-100 %; 0-59,99% - неудовлетворительно; 60-74,99% - удовлетворительно; 75- 84,99% -хорошо; 85-100% - отлично.
6	Структурированный тип данных массив	ОПК-12	Банк тестовых заданий Устный опрос	14-15 40-41	Компьютерное тестирование Процентная шкала. 0-100 %; 0-59,99% - неудовлетворительно; 60-74,99% - удовлетворительно; 75- 84,99% -хорошо; 85-100% - отлично.
7	Символьный и строковый типы данных	ОПК-12	Банк тестовых заданий Устный опрос	16-18 42-43	Компьютерное тестирование Процентная шкала. 0-100 %; 0-59,99% - неудовлетворительно; 60-74,99% - удовлетворительно; 75- 84,99% -хорошо; 85-100% - отлично.
8	Разработка алгоритмов в соответствии с принципом модульности	ОПК-12	Банк тестовых заданий Устный опрос	19-20 44-45	Компьютерное тестирование Процентная шкала. 0-100 %; 0-59,99% - неудовлетворительно;

					60-74,99% - удовлетворительно; 75- 84,99% -хорошо; 85-100% - отлично.
9	Реализация принципа модульности средствами алгоритмических языков программирования	ОПК-12	Банк тестовых заданий Устный опрос	21-22 46-47	Компьютерное тестирование Процентная шкала. 0-100 %; 0-59,99% - неудовлетворительно; 60-74,99% - удовлетворительно; 75- 84,99% -хорошо; 85-100% - отлично.
10	Структурированный тип данных запись	ОПК-12	Банк тестовых заданий Устный опрос	23-24 48-49	Компьютерное тестирование Процентная шкала. 0-100 %; 0-59,99% - неудовлетворительно; 60-74,99% - удовлетворительно; 75- 84,99% -хорошо; 85-100% - отлично.
11	Тип данных указатель	ОПК-12	Банк тестовых заданий Устный опрос	25-26 50-52	Компьютерное тестирование Процентная шкала. 0-100 %; 0-59,99% - неудовлетворительно; 60-74,99% - удовлетворительно; 75- 84,99% -хорошо; 85-100% - отлично.
12	Структурированный тип данных файл	ОПК-12	Банк тестовых заданий Устный опрос	27-28 53-55	Компьютерное тестирование Процентная шкала. 0-100 %; 0-59,99% - неудовлетворительно; 60-74,99% - удовлетворительно; 75- 84,99% -хорошо; 85-100% - отлично.
13	Обзор графических возможностей систем программирования на языках высокого уровня	ОПК-12	Банк тестовых заданий Устный опрос	29 56-57	Компьютерное тестирование Процентная шкала. 0-100 %; 0-59,99% - неудовлетворительно; 60-74,99% - удовлетворительно; 75- 84,99% -хорошо; 85-100% - отлично.

3. Оценочные средства для промежуточной аттестации (зачёт, экзамен)

Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Для оценки знаний, умений, навыков студентов по дисциплине применяется бально-рейтинговая система оценки сформированности компетенций студента.

Бально-рейтинговая система оценки осуществляется в течение всего семестра при проведении аудиторных занятий и контроля самостоятельной работы. Показателями ОМ являются: текущий опрос в виде собеседования на лабораторных работах, практических занятиях, тестовые задания в виде решения контрольных работ на практических работах и самостоятельно (домашняя контрольная работа) и сдачи курсовой работы по предложенной преподавателем теме. Оценки выставляются в соответствии с графиком контроля текущей успеваемости студентов в автоматизированную систему баз данных (АСУБД) «Рейтинг студентов».

Обучающийся, набравший в семестре более 60 % от максимально возможной бально-рейтинговой оценки работы в семестре получает экзамен автоматически.

Студент, набравший за текущую работу в семестре менее 60 %, т.к. не выполнил всю работу в семестре по объективным причинам (болезнь, официальное освобождение и т.п.) допускается до экзамена, однако ему дополнительно задаются вопросы на собеседовании по разделам, выносимым на экзамен.

Аттестация обучающегося по дисциплине проводится в форме тестирования и предусматривает возможность последующего собеседования (экзамена). Зачет проводится в виде тестового задания.

3. 1 Тесты (тестовые задания)

3.1.1 ОПК-12 - Способен создавать алгоритмы цифровой обработки баз данных результатов испытаний и эксплуатации сложных деталей и узлов в машиностроении, разрабатывать современные цифровые программы расчетов и проектирования деталей, узлов, конструкций, машин и материалов с учетом требований надежности, долговечности и безопасности их эксплуатации.

(ИД₁опк-12 – Создает алгоритмы цифровой обработки баз данных результатов испытаний и эксплуатации сложных деталей и узлов в машиностроении. ИД₂опк-12 – Разрабатывает цифровые программы расчетов и проектирования деталей, узлов, конструкций, машин и материалов с учетом требований надежности, долговечности и безопасности их эксплуатации.)

Номер задания	Тестовое задание
1	Какой тип из перечисленных занимает в памяти ровно 1 байт? 1) Integer 2) Word 3) Char 4) Longint
2	Какой из разделов указывает подключение библиотеки подпрограмм? 1) Const 2) Label 3) Uses 4) Type
3	В каком из разделов определяется тип пользователя? 1) Const 2) Label 3) Uses 4) Type
4	В каком из перечисленных разделов определяется размер массива? 1) Var 2) Label 3) Uses 4) Type
5	В какой из строк задается перечисляемый тип? 1) Type s = (a,b,d,f);

	<p>2) Type s = 1..56; 3) Type s : real; 4) Var s : char;</p>
6	<p>В какой из строк задается интервальный тип? 1) Type s = (a,b,d,f); 2) Type s = 1..56; 3) Type s : real; 4) Var s : char;</p>
7	<p>В какой строке допущена ошибка? 1) Type s = (a,b,d,f); 2) Type s = 1..56; 3) Type s : real; 4) Var s : char;</p>
8	<p>В какой из строк определяется переменная? 1) Type s = (a,b,d,f); 2) Type s = 1..56; 3) Type s : real; 4) Var s : char;</p>
9	<p>Укажите неверное утверждение. 1) Процедура Read используется для ввода данных с клавиатуры. 2) После процедуры Readln происходит переход к новой строке. 3) Write(f:9:6) — число выводится с 6 знаками после запятой. 4) Writeln(a); writeln(b); — числа выводятся в одной строке.</p>
10	<p>Укажите неверное утверждение. 1) Процедура Read используется для ввода данных с клавиатуры. 2) После процедуры Readln происходит переход к новой строке. 3) Write(f:9:6) — число выводится с 9 знаками после запятой. 4) Writeln(a); writeln(b); — числа выводятся в разных строках.</p>
11	<p>Укажите неверное утверждение. 1) Процедура Read используется для ввода данных с клавиатуры. 2) После процедуры Readln происходит выход из программы. 3) Write(f:9:6) — число выводится с 6 знаками после запятой. 4) Writeln^); writeln(b); — числа выводятся в разных строках.</p>
12	<p>Укажите неверное утверждение. 1) Процедура Read используется для вывода данных на экран. 2) После процедуры Readln происходит переход к новой строке. 3) Write(f:9:6) — число выводится с 6 знаками после запятой. 4) Writeln^); writeln(b); — числа выводятся в разных строках.</p>
13	<p>Какая связка операторов обозначает цикл с выходом по условию? 1) For...to...do 2) if then else 3) Case. of. end 4) While...do</p>
14	<p>Какая связка обозначает цикл с известным числом повторений? 1) For...to...do 2) if.then.else 3) Case. of. end 4) While. do</p>
15	<p>Какая связка операторов дает выбор из 3-х и более вариантов? 1) For.to.do 2) if.then.else 3) Case.of. end 4) While. do</p>
16	<p>Какая связка операторов дает выбор из 2-х вариантов? 1) For.to.do 2) if.then.else 3) Case. of. end</p>

	4) Begin. end
17	Какая из операций является логической? 1) div 2) mod 3) in 4) or
18	Сообщение, записанное буквами из 32-символьного алфавита, содержит 30 символов. Какой объем информации оно несет? 1) 960 байт 2) 150 бит 3) 150 байт 4) 1,5 Кбайт
19	Перевести число 110111 из двоичной в 16-ричную систему счисления: 1) 23 2) 45 3) 37 4) 54
20	Операция дизъюнкция называется иначе: 1) логическое умножение 2) логическое сложение 3) логическое следование 4) логическое равенство 5) логическое отрицание
21	Когда $2 * 2 = 11$? 1) в двоичной системе счисления; 2) в троичной системе счисления; 3) в четверичной системе счисления.
22	Кодирование информации - это... 1) Информация представленная различными способами 2) Представление информации посредством какого-либо алфавита 3) Преобразование одного набора знаков в другой 4) Набор кодировочной комбинации символов 5) Нет правильного ответа
23	Что является результатом этапа "формализации" решения задачи на компьютере? 1) словесная информационная модель 2) математическая модель 3) алгоритм 4) программа
24	Что такое управление? Выберите полное определение. 1) перевод объекта из одного состояния в другое 2) удержание объекта в существующем состоянии 3) процесс целенаправленного воздействия одних объектов на другие объекты 4) регулирование движения автомашин на перекрестке
25	Подпрограмма, имеющая произвольное количество входных и выходных данных, называется: 1) процедурой 2) функцией 3) вспомогательным алгоритмом
26	Алгоритм, целиком используемый в составе другого алгоритма, называется: 1) рекурсивным 2) вспомогательным 3) основным 4) дополнительным
27	Алгоритм: а) последовательность действий, которая приводит к решению за-

	дачи б) набор команд для компьютера в) ориентированный граф, указывающий порядок выполнения команд
28	Геометрическая фигура ромб используется в блок-схемах для обозначения принятия решения выполнения действия
29	Геометрическая фигура прямоугольник используется в блок-схемах для обозначения выполнения действия принятия решения

3.2 Собеседование (вопросы для устного опроса)

3.2.1 ОПК-12 - Способен создавать алгоритмы цифровой обработки баз данных результатов испытаний и эксплуатации сложных деталей и узлов в машиностроении, разрабатывать современные цифровые программы расчетов и проектирования деталей, узлов, конструкций, машин и материалов с учетом требований надежности, долговечности и безопасности их эксплуатации. (ИД1опк-12 –

Создает алгоритмы цифровой обработки баз данных результатов испытаний и эксплуатации сложных деталей и узлов в машиностроении. ИД2опк-12 – Разрабатывает цифровые программы расчетов и проектирования деталей, узлов, конструкций, машин и материалов с учетом требований надежности, долговечности и безопасности их эксплуатации.)

Номер задания	Формулировка вопроса
30	Понятие архитектуры вычислительных систем (ВС).
31	Классификация ВС.
32	Принципы организации CISC и RISC архитектур.
33	Многопроцессорные системы.
34	Симметричная и асимметричная многопроцессорность.
35	Методы организации памяти и обработки информации
36	Методы организации сетей ЭВМ.
37	Классификация сетей по масштабу и топологии.
38	Понятие сетевого протокола.
39	Семиуровневая модель OSI/ISO.
40	Способы маршрутизации сообщений в сетях ЭВМ.
41	Сетевая архитектура TCP/IP: основные принципы организации и функционирования.
42	Основные средства аппаратной поддержки функций ОС: система прерываний, защита памяти, механизм преобразования адресов в системах виртуальной памяти, управление периферийными устройствами.
43	Стратегии управления оперативной памятью.
44	Виртуальная память.
45	Статическая и динамическая сборка.
46	Основные подходы и алгоритмы планирования.
47	Системы реального и разделенного времени.
48	Управление доступом к данным.
49	Языки программирования.
50	Концепции процедурно-ориентированного, объектно-ориентированного, логического и функционального программирования.
51	Понятие о методах трансляции.
52	Лексический, синтаксический, семантический анализ.

53	Основные алгоритмы генерации объектного кода.
54	Системы программирования, типовые компоненты СП: языки, трансляторы, редакторы связей, отладчики, текстовые редакторы.
55	Принципы модульного, компонентного, объектно-ориентированного проектирования, шаблоны проектирования.
56	Моделирование программных систем.
57	Современные подходы к автоматическому синтезу программ.

4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

Процедуры оценивания в ходе изучения дисциплины знаний, умений и навыков, характеризующих этапы формирования компетенций, регламентируются положениями:

- П ВГУИТ 2.4.03 Положение о курсовых экзаменах и зачетах;
- П ВГУИТ 4.1.02 Положение о рейтинговой оценке текущей успеваемости.

Для оценки знаний, умений, навыков обучающихся по дисциплине применяется рейтинговая система. Итоговая оценка по дисциплине определяется на основании определения среднеарифметического значения баллов по каждому заданию.

5. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания для каждого результата обучения по дисциплине

Результаты обучения (на основе обобщённых компетенций)	Предмет оценки (продукт или процесс)	Показатель оценки	Критерии оценки	Шкала оценки	
				Академическая оценка (зачтено/незачтено)	Уровень освоения компетенции
<p>ОПК-12- Способен создавать алгоритмы цифровой обработки баз данных результатов испытаний и эксплуатации сложных деталей и узлов в машиностроении, разрабатывать современные цифровые программы расчетов и проектирования деталей, узлов, конструкций, машин и материалов с учетом требований надежности, долговечности и без-опасности их эксплуатации.</p>					
<p>Знать</p> <p>этапы решения задачи на компьютере на основе парадигмы императивного программирования; понятие алгоритма и его основные свойства; основные понятия алгоритмической системы: представление информации в виде данных, система команд исполнителя, алгоритмический язык исполнителя; основные средства записи и типы алгоритмов; алгоритмические структуры, их основные свойства и приемы использования; 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.</p> <p>- составные части алгоритмического языка программирования (алфавит, синтаксис, семантика) и способы их описания; концепцию типов данных; структуры программ</p>	Собеседование	Знание методов и алгоритмов конструирования элементов различных механических систем	если он ответил на все вопросы, допустил не более 1 ошибки в ответе;	Отлично	Освоена (повышенный)
			- оценка “хорошо”, если студент ответил на все вопросы, допустил более 1, но менее 3 ошибок;	Хорошо	Освоена (повышенный)
			- оценка “удовлетворительно”, если студент ответил не на все вопросы, но в тех, на которые дал ответ не допустил ошибки;	Удовлетворительно	Освоена (базовый)
			- оценка “неудовлетворительно”, если студент ответил не на все вопросы, допустил более 5 ошибок.	Не удовлетворительно	Не освоена (не достаточный)

<p>на языках программирования; основные понятия алгоритмических языков программирования и их классификацию: идентификаторы, данные и их типы, операции, стандартные функции, выражения, операторы; синтаксис и семантику основных операторов языков программирования (присваивания, ввода-вывода, ветвления, циклов), способы реализации алгоритмических структур средствами языков программирования; основные синтаксические конструкции, используемые в языках программирования для описания и обработки данных типа массив; способы передачи параметров при реализации межмодульного интерфейса; структуру данных файл и основные действия с файлами последовательного доступа.</p>					
<p>Уметь при решении стандартных задач профессиональной деятельности на основе парадигмы императивного программирования конкретизировать и описать основные понятия, результаты других научных дисциплин, данные и их типы средствами записи алгоритмов и программ; разрабатывать и записывать алгоритмы и программы на языках высокого уровня в соответствии с технологией нисходящего структурного проектирования; записывать алгоритмы на языке схем; анализировать структуру алгоритмов; описывать алфавит и основные синтаксические конструкции языков программирова-</p>	Тест	Результат тестирования	75% и более правильных ответов	Отлично	Освоена (повышенный)
			60-75% правильных ответов	Хорошо	Освоена (повышенный)
			50-60% правильных ответов	удовлетворительно	Освоена (базовый)
			Менее 50% правильных ответов	Не удовлетворительно	Не освоена

<p>ния.</p> <p>- разрабатывать и записывать алгоритмы и программы на языках высокого уровня в соответствии с принципом структурности; при решении учебной задачи конкретизировать и описать основные понятия, данные и их типы, организовать ввод и вывод, реализовать алгоритмические структуры средствами, по крайней мере, двух императивных языков программирования; разрабатывать алгоритмы и программы обработки массивов; разрабатывать алгоритмы и программы обработки строк; решать учебные задачи в соответствии с принципом модульности; 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.</p>					
<p>Владеть основными методами, способами и средствами переработки информации на основе парадигмы императивного программирования в соответствии с технологией нисходящего структурного проектирования.</p> <p>Навыками проектирования, ввода, отладки и тестирования программ в средах, по крайней мере, двух императивных систем программирования.</p>	<p>Расчётно-графические работы</p>	<p>Материалы расчётно-графической работы</p>	<p>- оценка «отлично» выставляется студенту, если работа выполнена верно, представлена расчётная часть на листах формата А4 в объёме не менее 10стр. и графическая часть на листах формата А4 в объёме не менее 2стр. (для РГР №2) и не содержит вычислительных ошибок, ответил на все вопросы, допустил не более 1 ошибки в ответе;</p>	<p>отлично</p>	<p>освоена (повышенный)</p>

			<p>- оценка «хорошо» выставляется студенту, если работа выполнена верно и не содержит существенных вычислительных ошибок, представлена расчётная часть на листах формата А4 в объёме не менее 10стр. и графическая часть на листах формата А4 в объёме не менее 2стр. (для РГР №2) ответил на все вопросы, имеются незначительные замечания по тексту и оформлению работы, допустил не более 3 ошибок в ответе;</p>	<p>Хорошо</p>	<p>Освоена (повышенный)</p>
			<p>- оценка «удовлетворительно» выставляется студенту, если работа выполнена верно и содержит существенные вычислительные ошибки, представлена расчётная часть на листах формата А4 в объёме не менее 10стр. и графическая часть на листах формата А4 в объёме не менее 2стр. (для РГР №2), имеются значительные замечания по тексту и оформлению работы, допустил не более 5 ошибок в ответе;</p>	<p>Удовлетворительно</p>	<p>Освоена (базовый)</p>
			<p>- оценка «не удовлетворительно» выставляется студенту, если работа выполнена не верно, не ответил на большинство вопросов, имеются значительные замечания по тексту и оформлению работы, допустил более 5 ошибок в ответе.</p>	<p>Не удовлетворительно</p>	<p>Не освоена (не достаточный)</p>

**АННОТАЦИЯ
К РАБОЧЕЙ ПРОГРАММЕ
ДИСЦИПЛИНЫ**

«Алгоритмизация и программирование»

(наименование дисциплины)

Процесс изучения дисциплины направлен на формирование следующих компетенций:

Код и наименование индикатора достижения компетенции	Результаты обучения (показатели оценивания)
ИД1 _{опк-12} – Создает алгоритмы цифровой обработки баз данных результатов испытаний и эксплуатации сложных деталей и узлов в машиностроении,	Знает: этапы решения задачи на компьютере на основе парадигмы императивного программирования; понятие алгоритма и его основные свойства; основные понятия алгоритмической системы: представление информации в виде данных, система команд исполнителя, алгоритмический язык исполнителя; основные средства записи и типы алгоритмов; алгоритмические структуры, их основные свойства и приемы использования; 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
	Умеет: при решении стандартных задач профессиональной деятельности на основе парадигмы императивного программирования конкретизировать и описать основные понятия, результаты других научных дисциплин, данные и их типы средствами записи алгоритмов и программ; разрабатывать и записывать алгоритмы и программы на языках высокого уровня в соответствие с технологией нисходящего структурного проектирования; записывать алгоритмы на языке схем; анализировать структуру алгоритмов; описывать алфавит и основные синтаксические конструкции языков программирования.
	Владеет: основными методами, способами и средствами переработки информации на основе парадигмы императивного программирования в соответствии с технологией нисходящего структурного проектирования.
ИД2 _{опк-12} – Разрабатывает цифровые программы расчетов и проектирования деталей, узлов, конструкций, машин и материалов с учетом требований надежности, долговечности и безопасности их эксплуатации	Знает: составные части алгоритмического языка программирования (алфавит, синтаксис, семантика) и способы их описания; концепцию типов данных; структуры программ на языках программирования; основные понятия алгоритмических языков программирования и их классификацию: идентификаторы, данные и их типы, операции, стандартные функции, выражения, операторы; синтаксис и семантику основных операторов языков программирования (присваивания, ввода-вывода, ветвления, циклов), способы реализации алгоритмических структур средствами языков программирования; основные синтаксические конструкции, используемые в языках программирования для описания и обработки данных типа массив; способы передачи параметров при реализации межмодульного интерфейса; структуру данных файл и основные действия с файлами последовательного доступа.

	<p>Умеет: разрабатывать и записывать алгоритмы и программы на языках высокого уровня в соответствии с принципом структурности; при решении учебной задачи конкретизировать и описать основные понятия, данные и их типы, организовать ввод и вывод, реализовать алгоритмические структуры средствами, по крайней мере, двух императивных языков программирования; разрабатывать алгоритмы и программы обработки массивов; разрабатывать алгоритмы и программы обработки строк; решать учебные задачи в соответствии с принципом модульности; 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.</p>
	<p>Владеет: навыками проектирования, ввода, отладки и тестирования программ в средах, по крайней мере, двух императивных систем программирования.</p>

Содержание разделов дисциплины.

Алгоритмическая система как совокупность средств и понятий. Понятие алгоритма и исполнителя алгоритма. Представление информации в виде данных: понятие о вводе/выводе; входные, выходные и промежуточные данные. Программирование как этап решения задачи на компьютере. Понятие языка программирования высокого уровня. Цели и составные части технологии нисходящего структурного проектирования. Взаимосвязь принципов нисходящего проектирования, модульности и структурности. Алгоритмические структуры как основа структурного кодирования. Базовые алгоритмические структуры: следование и развилка.

Базовая алгоритмическая структура цикл с предусловием. Цикл с постусловием. Примеры циклических алгоритмов и программ. Разработка схем алгоритмов циклической структуры. Написание, ввод, отладка и тестирование программ циклической структуры. Понятие о текстовом режиме монитора, текущее положение курсора. Синтаксис и семантика операторов ввода с клавиатуры. Особенности ввода с клавиатуры символьной и строковой информации. Общая характеристика операторов вывода на монитор.

Синтаксис и семантика оператора резервирования памяти, примеры. Особенности распределения памяти. Синтаксис определения типа массив. Рекомендации по описанию переменной типа массив и понятие полной совместимости типов, примеры. Использование переменной типа массив в операторе присваивания.

Стандартный идентификатор строкового типа. Строковый тип в языке программирования как массив символов. Операция конкатенации. Функции и процедуры для работы со строками. Примеры алгоритмов и программ обработки данных строкового типа. Написание, ввод, отладка и тестирование программ обработки строк.

Способы записи алгоритмического модуля средствами алгоритмических языков программирования: процедуры и функции. Синтаксис процедуры, синтаксис и семантика оператора процедуры. Способы передачи параметров значением и ссылкой.

Понятия статических и динамических объектов программы на языке высокого уровня. Синтаксис типа указатель (ссылочный тип) в языке программирования. Рекомендации по описанию переменной типа указатель, примеры. Синтаксис и семантика оператора резервирования (захвата) памяти под динамическую переменную в языке программирования, примеры.

Понятие файла, файлы данных различного доступа.

Понятие о графическом режиме монитора, текущее положение графического курсора. Операторы установки параметров окна результатов и задания текущего цвета изображения в языках программирования.