

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Выборгов Дмитрий Александрович

Должность: Ректор

Дата подписания: 08.09.2022 14:41:35

Уникальный программный ключ:

c3b3b9c625f6c113afa2a2c42ba19e05a38b76e

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СЕРВИСА»
(ФГБОУ ВО «ПВГУС»)

Кафедра «Информационный и электронный сервис»

РАБОЧАЯ УЧЕБНАЯ ПРОГРАММА

по дисциплине

«Основы алгоритмизации и программирования»

для студентов специальности

09.02.01 «Компьютерные системы и комплексы»

Тольятти 2018

Рабочая учебная программа по дисциплине «Основы алгоритмизации и программирования» включена в основную профессиональную образовательную программу специальности 09.02.01 «Компьютерные системы и комплексы» решением Президиума Ученого совета

Протокол № 4 от 28.06.2018 г.

Начальник учебно-методического отдела _____  _____ Н.М.Шемендюк
28.06.2018 г.

Рабочая учебная программа по дисциплине «Основы алгоритмизации и программирования» разработана в соответствии с Федеральным государственным образовательным стандартом специальности 09.02.01 «Компьютерные системы и комплексы», утвержденным приказом Министерства образования и науки Российской Федерации от 28 июля 2014 г. N 849.

Составил: старший преподаватель А.С. Васильева

СОГЛАСОВАНО:


Директор научной библиотеки _____  В.Н.Еремина

СОГЛАСОВАНО:

Начальник управления информатизации _____  В.В.Обухов

Рабочая программа утверждена на заседании кафедры «Информационный и электронный сервис»

Протокол № 11 от «27» июня 2018 г.

Заведующий кафедрой _____  д.т.н., профессор В.И. Воловач
(подпись)

СОГЛАСОВАНО:

Начальник учебно-методического отдела _____  Н.М.Шемендюк

1. Перечень планируемых результатов обучения по дисциплине «Основы алгоритмизации и программирования», соотнесенных с планируемыми результатами освоения образовательной программы

1.1. Цели освоения дисциплины «Основы алгоритмизации и программирования»

Целью освоения дисциплины «Основы алгоритмизации и программирования» является формирование у студентов базовых компетенций в области программирования, в том числе ознакомление студентов с понятием алгоритма, основными видами алгоритмов и способами их составления, алгоритмами некоторых стандартных процессов.

1.2. В соответствии с видами профессиональной деятельности, на которые ориентирована образовательная программа указанной специальности, содержание дисциплины «Основы алгоритмизации и программирования» позволит обучающимся решать следующие профессиональные задачи:

- применение микропроцессорных систем, установка и настройка периферийного оборудования;
- техническое обслуживание и ремонт компьютерных систем и комплексов.

1.3. Компетенции обучающегося, формируемые в результате освоения дисциплины

В результате освоения дисциплины у обучающихся формируются следующие компетенции:

Код компетенции	Наименование компетенции
ОК 1	Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.
ОК 2	Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.
ОК 3	Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.
ОК 4	Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.
ОК 5	Использовать информационно-коммуникационные технологии в профессиональной деятельности.
ОК 6	Работать в коллективе и команде, эффективно общаться с коллегами, руководством, потребителями.
ОК 7	Брать на себя ответственность за работу членов команды (подчиненных), результат выполнения заданий.
ОК 8	Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.
ОК 9	Ориентироваться в условиях частой смены технологий в профессиональной деятельности.
ПК 2.1	Создавать программы на языке ассемблера для микропроцессорных систем.
ПК 2.2	Производить тестирование, определение параметров и отладку микропроцессорных систем.
ПК 3.3	Принимать участие в отладке и технических испытаниях компьютерных систем и комплексов, инсталляции, конфигурировании программного обеспечения.

1.4. Перечень планируемых результатов обучения по дисциплине

Результаты освоения дисциплины	Технологии формирования компетенции по указанным результатам	Средства и технологии оценки по указанным результатам
<p>Знает: ОК 1-9 общие принципы построения и использования языков программирования, их классификацию; современные интегрированные среды разработки программ; процесс создания программ; стандарты языков программирования; общую характеристику языков ассемблера: назначение, принципы построения и использования. ПК 2.1 принципы построения программ на ассемблере ПК 2.2 принципы определения параметров и отладки микропроцессорных систем ПК 3.3 принципы отладки и технических испытаний компьютерных систем и комплексов</p>	Лекции	Собеседование
<p>Умеет: ОК 1-9 формализовать поставленную задачу; применять полученные знания к различным предметным областям; составлять и оформлять программы на языках программирования;</p>	Лабораторные работы	Защита лабораторных работ

<p>тестировать и отлаживать программы. ПК 2.1 создавать программы на языке ассемблера для микропроцессорных систем ПК 2.2 производить тестирование, определение параметров и отладку микропроцессорных систем ПК 3.3 принимать участие в отладке и технических испытаниях компьютерных систем и комплексов, инсталляции, конфигурировании программного обеспечения</p>		
<p><i>Имеет практический опыт:</i> ОК 1-9 составления и оформления программ на языках программирования; тестирования и отлаживания программ ПК 2.1 создания программ на языке ассемблера ПК 2.2 выполнения тестирования, определения параметров и отладки ПК 3.3 участия в отладке и технических испытаниях</p>	<p>Лекции Лабораторные работы</p>	<p>Собеседование Защита лабораторных работ</p>

2. Место дисциплины в структуре образовательной программы

Дисциплина относится общепрофессиональным дисциплинам профессионального цикла. Ее освоение осуществляется в 5* семестре (очная форма), в 6* семестре (заочная форма).

№ п/п	Наименование дисциплин, определяющих междисциплинарные связи	Код компетенции(й)
	Предшествующие дисциплины	
1	Информатика и ИКТ	ОК 1 - ОК 9
	Последующие дисциплины	
2	Технология программирования	ПК 1.3, ПК 2.1

*Здесь и далее семестры указаны для обучающихся на базе основного общего образования. Для лиц, обучающихся на базе среднего общего образования, семестры соответствуют учебному плану и нормативному сроку обучения, установленному ФГОС.

3. Объем дисциплины с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу

Распределение фонда времени по семестрам и видам занятий

Виды занятий	очная форма обучения	заочная форма обучения
Итого часов	90	90
Лекции (час)	32	8
Практические (семинарские) занятия (час)	-	-
Лабораторные работы (час)	48	4
Самостоятельная работа (час)	9	78
Консультации	5 семестр (1 ч)	-
Курсовой проект (работа) (+,-)	-	-
Контрольная работа (+,-)	+	+
Экзамен, семестр	5	6
Зачет (дифференцированный зачет), семестр	-	-
Контрольная работа, семестр	5	6

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

4.1. Содержание дисциплины

№ п/п	Раздел дисциплины	Виды учебных занятий, включая самостоятельную работу студентов и трудоемкость (в академических часах)				Средства и технологии оценки
		Лекции, час	Практические (семинарские) занятия, час	Лабораторные работы, час	Самостоятельная работа, час	
1	Основные принципы алгоритмизации и программирования	4/1	-/-	6/2	0/10	Конспект, сообщение, защита лабораторных работ
2	Основные элементы языка PASCAL	4/1	-/-	6/2	0/10	Конспект, опрос на лекции,

						защита лабораторных работ
3	Операторы управления и функции	4/1	-/-	12/0	0/10	Конспект, сообщение, защита лабораторных работ
4	Указатели. Массивы	4/1	-/-	12/0	0/10	Конспект, сообщение, защита лабораторных работ
5	Работа с файлами. Структуры и классы	4/1	-/-	12/0	0/10	Конспект, сообщение, опрос на лекции, защита лабораторных работ
6	Сортировка и поиск	4/1	-/-	-/-	0/10	Конспект, сообщение, защита лабораторных работ
7	Основные виды, этапы проектирования и жизненный цикл программных продуктов.	4/1	-/-	-/-	6/10	Конспект, сообщение
8	Стратегии конструирования ПО.	4/1	-/-	-/-	3/8	Конспект, сообщение
	Промежуточная аттестация по дисциплине	32/8	-/-	48/4	9/78	Экзамен

Примечание:

-/-, объем часов соответственно для очной, заочной форм обучения

4.2. Содержание практических (семинарских) занятий

Практические занятия планом не предусмотрены.

4.3. Содержание лабораторных работ

№	Наименование лабораторных работ	Объем часов	Наименование темы дисциплины
1	Лабораторная работа № 1. Проектирование алгоритмов. Блок-схема алгоритма.	6/2	Основные принципы алгоритмизации и программирования.
2	Лабораторная работа № 2. Знакомство с редактором Free Pascal	6/2	Основные элементы языка PASCAL.
3	Лабораторная работа № 3. Операторы циклов.	6/0	Операторы управления и функции.
4	Лабораторная работа № 4. Процедуры. Функции.	6/0	Операторы управления и функции.
5	Лабораторная работа № 5. Массивы.	12/0	Указатели. Массивы.
6	Лабораторная работа № 6. Форматирование	12/0	Работа с файлами.

	ввода-вывода. Работа с файлами. Структуры. Классы.		Структуры и классы. Сортировка и поиск.
	Итого	48/4	

Примечание:

–/–, объем часов соответственно для очной, заочной форм обучения

5. Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине

Технологическая карта самостоятельной работы студента

Код реализуемой компетенции	Вид деятельности студентов (задания на самостоятельную работу)	Итоговый продукт самостоятельной работы	Средства и технологии оценки	Объем часов
1	2	3	4	5
ОК 1 ОК 2 ОК 3 ОК 4 ОК 5 ОК 6 ОК 7 ОК 8 ОК 9 ПК 2.1 ПК 2.2 ПК 3.3	Выполнение индивидуальных заданий в виде краткого конспекта на заданную тему.	Конспект	Собеседование	4/39
	Выполнение индивидуальных заданий в виде реферата и презентации на заданную тему.	Реферат, презентация	Собеседование	5/39
Итого				9/78

Примечание:

–/–, объем часов соответственно для очной, заочной форм обучения

Рекомендуемая литература:

- Гвоздева, В. А. Информатика, автоматизированные информационные технологии и системы [Электронный ресурс] : учеб. для студентов техн. специальностей / В. А. Гвоздева. - Документ Bookread2. - М. : ФОРУМ [и др.], 2015. - 541 с. - Режим доступа: <http://znanium.com/bookread2.php?book=492670#>
- Колдаев, В. Д. Основы алгоритмизации и программирования [Электронный ресурс] : учеб. пособие для сред. проф. образования по группе специальностей "Информатика и вычисл. техника" / В. Д. Колдаев под ред. Л. Г. Гагариной. - Документ Bookread2. - М. : ФОРУМ [и др.], 2015. - 413 с. : ил., табл. - Режим доступа: <http://znanium.com/bookread2.php?book=484837#>

Содержание заданий для самостоятельной работы

Темы рефератов (письменных работ, эссе, докладов и т.п.)

- Классы и объекты.
- Сортировка и поиск.
- Методологии программирования.
- Классический жизненный цикл программного продукта.

5. Технология конструирования программного обеспечения.

**6. Методические указания для обучающихся по освоению дисциплины
Инновационные образовательные технологии**

Вид образовательных технологий, средств передачи знаний, формирования умений и практического опыта	№ темы / тема лекции	№ практического (семинарского) занятия/наименование темы	№ лабораторной работы / цель
Слайд-лекции	№ 1-8	-	-

В начале семестра студентам необходимо ознакомиться с технологической картой дисциплины, выяснить, какие результаты освоения дисциплины заявлены (знания, умения, практический опыт). Для успешного освоения дисциплины студентам необходимо выполнить задания, предусмотренные рабочей учебной программой дисциплины и пройти контрольные точки в сроки, указанные в технологической карте (раздел 11). От качества и полноты их выполнения будет зависеть уровень сформированности компетенции и оценка текущей успеваемости по дисциплине. По итогам текущей успеваемости студенту может быть выставлена оценка по промежуточной аттестации, если это предусмотрено технологической картой дисциплины. Списки учебных пособий, научных трудов, которые студентам следует прочесть и законспектировать, темы практических занятий и вопросы к ним, вопросы к экзамену и другие необходимые материалы указаны в разработанном для данной дисциплины учебно-методическом комплексе.

Основной формой освоения дисциплины является контактная работа с преподавателем - лекции, лабораторные работы (при наличии в учебном плане), консультации (в том числе индивидуальные), в том числе проводимые с применением дистанционных технологий.

По дисциплине часть тем (разделов) изучается студентами самостоятельно. Самостоятельная работа предусматривает подготовку к аудиторным занятиям, выполнение заданий (письменных работ, творческих проектов и др.) подготовку к промежуточной аттестации (экзамену).

На лекционных занятиях, лабораторных работах вырабатываются навыки и умения обучающихся по применению полученных знаний в конкретных ситуациях, связанных с будущей профессиональной деятельностью. По окончании изучения дисциплины проводится промежуточная аттестация (экзамен).

Регулярное посещение аудиторных занятий не только способствует успешному овладению знаниями, но и помогает организовать время, т.к. все виды учебных занятий распределены в семестре планомерно, с учетом необходимых временных затрат.

**6.1. Методические указания для обучающихся по освоению дисциплины на
лабораторных работах**

Лабораторные работы

№	Наименование лабораторных работ	Задание по лабораторным работам
1	Лабораторная работа № 1. Проектирование алгоритмов. Блок-схема алгоритма.	1. Изучить принципы построения алгоритма. 2. Изучить графическое представление алгоритма в виде блок-схемы. 3. Научиться создавать блок-схемы

		алгоритма любой сложности.
2	Лабораторная работа № 2. Знакомство с редактором Free Pascal	<ol style="list-style-type: none"> 1. Изучить структуру окна и меню Free Pascal. 2. Изучить стандартные процедуры ввода-вывода данных. 3. Ознакомиться с условным оператором Выполнить задания по теме.
3	Лабораторная работа № 3. Операторы циклов.	<ol style="list-style-type: none"> 1. Изучить оператора цикла с предусловием WHILE. 2. Ознакомиться с постусловием REPEAT. 3. Изучить оператор цикла с параметром FOR Выполнить задания по теме.
4	Лабораторная работа № 4. Процедуры. Функции.	<ol style="list-style-type: none"> 1. Изучить процедуру Str_Simw. 2. Изучить функцию с параметром. 3. Выполнить задания по теме
5	Лабораторная работа № 5. Массивы.	<ol style="list-style-type: none"> 1. Оформить программу с помощью процедуры ввода массива и процедуры вывода массива на экран. 2. Изучить обработку одномерных массивов.
6	Лабораторная работа № 6. Форматирование ввода-вывода. Работа с файлами. Структуры. Классы.	<ol style="list-style-type: none"> 1. Изучить особенности ввода-вывода данных. 2. Разработать программу, используя создание файлов с данными.

Лабораторные работы обеспечивают:

формирование умений и навыков обращения с приборами и другим оборудованием, демонстрацию применения теоретических знаний на практике, закрепление и углубление теоретических знаний, контроль знаний и умений в формулировании выводов, развитие интереса к изучаемой дисциплине.

Применение лабораторных работ позволяет вовлечь в активную работу всех обучающихся группы и сформировать интерес к изучению дисциплины.

Самостоятельный поиск ответов на поставленные вопросы и задачи в ходе лабораторной работы приобретают особую значимость в восприятии, понимании содержания дисциплины.

Изученный на лекциях материал лучше усваивается, лабораторные работы демонстрируют практическое их применение.

6.2. Методические указания для выполнения контрольных работ

В ходе выполнения контрольной работы приобретается опыт в разработке программ с четкой структуризацией, умение осуществлять постановку задачи, применять процедуры и функции из стандартных модулей при создании проекта, конструировать собственные подпрограммы.

Для успешного выполнения контрольной работы необходимо изучить рекомендуемую и дополнительную литературу параллельно с прослушиванием лекций для закрепления материала.

Рекомендуется следующий порядок работы:

- ознакомиться по учебной программе с содержанием темы;

- прочитать раздел учебника и другую техническую литературу, относящуюся к данной теме;
- изучить конспект лекции по темам;
- решить задачу с использованием конкретного задания.

Пояснительная записка должна состоять из введения, основной части и заключения.

В введении необходимо определить цель выполнения заданий контрольной работы, их основную идею и наметить пути достижения поставленной цели. Объем введения 1-2 страницы.

Основная часть пояснительной записки должна содержать:

- формулировку задачи;
- описание входных, выходных и промежуточных данных;
- листинг программы;
- исходные данные для тестирования (контрольный пример) с полученным результатом.

Объем основной части составляет 10-20 страниц.

В заключение пояснительной записки к контрольной работе делаются краткие выводы о полученных результатах, оценивается оптимальность решения задачи. Объем заключения - 1-2 страницы.

Пояснительная записка **оформляется** на листах бумаги формата А4. Текстовая часть оформляется на принтере на одной стороне листа с соблюдением полей: сверху и снизу 20 мм, слева 35 мм, справа 10 мм, через 1,5 межстрочных интервала. Текст пояснительной записки рамкой не обводится. Страницы пояснительной записки **нумеруются** подряд в **верхнем правом углу** страницы. Нумерация страниц начинается с титульного листа, номер на котором не проставляется. Структурная схема программы выполняется на компьютере. Заголовки разделов пишутся заглавными буквами, заголовки подразделов - строчными буквами с “красной” строки. Перенос слов в заголовках не допускается. Точки в конце заголовков разделов и подразделов не ставятся. Расстояние между заголовками и текстовой частью должно составлять 15 мм. Каждый новый раздел следует начинать с нового листа. Текст пояснительной записки должен быть кратким, содержательным и грамотным. Сокращения слов в тексте недопустимы.

В список литературы включаются все используемые в работе источники. Сведения о книгах включают фамилии и инициалы авторов, заглавие книги, место издания, издательство, год издания, количество страниц. Никаких кавычек нигде не ставится. Города Москва и Ленинград, как место издания, указываются сокращенно М., Л. Названия других городов пишутся полностью. Если на титульном листе книги не указан автор, то сведения начинают с указания названия книги, затем после наклонной черты (/) и слов “Под ред.” указываются инициалы и фамилия редактора и далее данные в той же последовательности, что и выше.

К стилю программирования предъявляются следующие требования:

- Программа должна быть удобочитаема, простая и ясная;
- Использовать в качестве идентификаторов переменных осмысленные имена;
- Записывать только один оператор в строке;
- Использовать сдвиги в строке в соответствии с уровнем вложенности;
- Использовать комментарии в тексте программы.

Для выполнения контрольной работы студенты должны изучить теоретические основы Pascal. В ходе выполнения контрольной работы приобретается опыт в разработке программ с четкой структуризацией, умение осуществлять постановку задачи, применять процедуры и функции из стандартных модулей при создании проекта, конструировать собственные подпрограммы, использовать данные в виде файлов.

Для успешного выполнения контрольной работы необходимо изучить рекомендуемую и дополнительную литературу параллельно с прослушиванием лекций для закрепления материала.

Рекомендуется следующий порядок работы:

- ознакомиться по учебной программе с содержанием темы;

- прочитать раздел учебника и другую техническую литературу, относящуюся к данной теме;
 - изучить конспект лекции по темам;
 - решить задачу с использованием конкретного задания.
- Пояснительная записка должна состоять из введения, основной части и заключения.

Варианты заданий:

ВАРИАНТ 1

1. В файле хранится информация о студентах: ФИО студента, наименование специальности, курс, средний балл за сессию. Составить программу для получения:

- Сведений о студентах заданного курса, средний балл которых равен 4;
- Информации о количестве студентов заданного курса и специальности.

2. Дан массив 10 x 10. Переписать диагональные элементы в одномерный массив тех строк, в которых первый и последний элементы равны.

ВАРИАНТ 2

1. В файле хранится информация о поступивших в ремонт бытовых приборах: ФИО клиента, тип бытового прибора, год выпуска, страховая стоимость прибора, наименование дефекта, дата приемки в ремонт. Составить программу для получения:

- Сведений о приборах, поступивших позже заданной даты;
- Сведений о средней страховой стоимости приборов, находящихся в мастерской.

2. Дан массив 7 x 9. Удалить из массива те столбцы, сумма элементов которых равна нулю.

ВАРИАНТ 3

1. В файле хранится информация о поступивших в ремонт бытовых приборах: тип бытового прибора, год выпуска, наименование дефекта, наименование необходимой детали для ремонта, стоимость ремонта. Составить программу для получения:

- Сведений о приборах, стоимость ремонта которых больше средней;
- Информацию об общей стоимости ремонта приборов заданного года выпуска.

2. Дан массив 6 x 9. Определить максимальный элемент тех строк, в которых первый и последний элементы противоположных знаков. Составить массив из этих элементов

ВАРИАНТ 4

1. В файле хранится информация о наличии на складе запасных частей для ремонта бытовых приборов: наименование детали, количество единиц на складе, цена одной детали, суммарная стоимость деталей каждого наименования, среднемесячная норма потребления. Составить программу для получения:

- Сведений о деталях, стоимость которых минимальная;
- Информации о количестве деталей имеющих равную среднемесячную норму потребления.

2. Дан массив $M \times N$. Записать в одномерный массив индексы тех столбцов, в которых максимальный элемент встречается раньше минимального.

ВАРИАНТ 5

1. В файле хранится информация о студентах: ФИО студента, группа, курс, пол, год рождения. Составить программу для получения:

- Сведений о студентах, год рождения которых меньше заданного;
- Информации о среднем возрасте студентов заданной группы.

2. Дан массив $N \times M$. Среди элементов этого массива, меньших 3, найти наибольший.

ВАРИАНТ 6

1. В файле хранится информация о студентах: ФИО студента, наименование специальности, курс, средний балл за сессию. Составить программу для получения:

- Сведений о студентах заданной специальности;

- Информации о среднем балле студентов заданного курса.
2. Дан массив $N \times M$. Получить новую матрицу путем деления всех элементов данной матрицы на ее наибольший по модулю элемент.

ВАРИАНТ 7

1. В файле хранится информация о поступивших в ремонт бытовых приборах: ФИО клиента, тип бытового прибора, год выпуска, страховая стоимость прибора, наименование дефекта, дата приемки в ремонт. Составить программу для получения:

- Сведений о приборе сданном в ремонт (ФИО);
- Информации о количестве приборов одного типа с одинаковым дефектом.

2. Дана матрица $N \times M$. Найти минимальный элемент в каждой строке, затем среди этих чисел выбрать наибольший. Указать индексы этого элемента.

ВАРИАНТ 8

1. В файле хранится информация о поступивших в ремонт бытовых приборах: тип бытового прибора, год выпуска, наименование дефекта, наименование необходимой детали для ремонта, стоимость ремонта. Составить программу для получения:

- Сведений о приборах самого раннего года выпуска;
- Информации о количестве необходимых деталей для ремонта всех приборов одного наименования.

2. Дана матрица $N \times M$. Найти сумму элементов до первого отрицательного элемента в строке K .

ВАРИАНТ 9

1. В файле хранится информация о наличии на складе запасных частей для ремонта бытовых приборов: наименование детали, количество единиц на складе, цена одной детали, суммарная стоимость деталей каждого наименования, среднемесячная норма потребления. Составить программу для получения:

- Сведений о деталях, цена которых равна заданной с клавиатуры;
- Отсортированного списка в порядке убывания цены.

2. Дана матрица 6×9 . Получить матрицу путем перестановки столбцов – первого с последним, второго с предпоследним и т.д.

ВАРИАНТ 10

1. В файле хранится информация о студентах: ФИО студента, группа, курс, пол, год рождения. Составить программу для получения:

- Информации о количестве девушек и юношей заданного курса;
- Сведений о студентах заданного года рождения.

2. Дан массив 7×5 . Определить в какой строке находится наибольший из минимальных элементов каждой строке.

7. Фонд оценочных средств для проведения текущего контроля успеваемости и промежуточной аттестации обучающихся по дисциплине (экзамен)

Код оцениваемой компетенции (или ее части)	Тип контроля	Вид контроля	Количество элементов

ОК 1 ОК 2 ОК 3 ОК 4 ОК 5	<i>текущий</i>	<i>устный опрос</i>	30
ОК 6 ОК 7 ОК 8 ОК 9 ПК 2.1 ПК 2.2 ПК 3.3			

7.1. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины

Результаты освоения дисциплины	Оценочные средства (перечень вопросов, заданий и др.)
<p>Знает: ОК 1-9 общие принципы построения и использования языков программирования, их классификацию; современные интегрированные среды разработки программ; процесс создания программ; стандарты языков программирования; общую характеристику языков ассемблера: назначение, принципы построения и использования. ПК 2.1 принципы построения программ на ассемблере ПК 2.2 принципы определения параметров и отладки микропроцессорных систем ПК 3.3 принципы отладки и технических испытаний компьютерных систем и комплексов</p>	<p>1. Для чего используется оператор «#include»? -: Подключение дополнительных библиотек -: Вывод текста на экран -: Создание процедуры -: Организация цикла с предусловием -: Организация цикла с послеусловием -: Организация ветвления программы</p> <p>2. Для чего используется оператор «cout»? -: Вывод текста на экран -: Создание процедуры -: Подключение дополнительных библиотек -: Организация цикла с предусловием -: Организация цикла с послеусловием -: Организация ветвления программы</p> <p>3. Для чего используется оператор «void»? -: Создание процедуры -: Вывод текста на экран -: Подключение дополнительных библиотек -: Организация цикла с предусловием -: Организация цикла с послеусловием -: Организация ветвления программы</p> <p>4. Для чего используется оператор «if»? -: Организация ветвления программы -: Вывод текста на экран -: Создание процедуры -: Подключение дополнительных библиотек -: Организация цикла с предусловием -: Организация цикла с послеусловием</p> <p>5. Для чего используется оператор «for»? -: Организация цикла с предусловием -: Вывод текста на экран -: Создание процедуры</p>

- : Подключение дополнительных библиотек
- : Организация цикла с послеусловием
- : Организация ветвления программы

6. Для чего используется оператор «do / while»?

- : Организация цикла с послеусловием
- : Вывод текста на экран
- : Создание процедуры
- : Подключение дополнительных библиотек
- : Организация цикла с предусловием
- : Организация ветвления программы

7. Для чего используется оператор «cin»?

- : Ввод текста с клавиатуры
- : Вывод текста на экран
- : Создание процедуры
- : Подключение дополнительных библиотек
- : Организация цикла с послеусловием
- : Организация ветвления программы

8. Какой тип данных подразумевает числа с плавающей точкой?

- : Double
- : Float
- : Integer
- : Char
- : Boolean

9. Какой тип данных подразумевает символы таблицы кодов ASCII?

- : Char
- : Integer
- : Boolean
- : Float
- : Double

10. Какой тип данных использует 8 байт для хранения данных?

- : Double
- : Integer
- : Char
- : Boolean
- : Float

11. Какой тип данных использует 1 байт для хранения данных?

- : Char
- : Boolean
- : Integer
- : Double
- : Float

12. Какой тип данных использует 4 байта для хранения данных?

- : Integer

- : Float
- : Double
- : Char
- : Boolean

13. Какой объём памяти резервируется под переменную типа boolean

- : 8 бит
- : 1 бит
- : 16 бит
- : 32 бит
- : 64 бита

14. Какой объём памяти резервируется под переменную типа integer

- : 32 бита
- : 1 бит
- : 8 бит
- : 16 бит
- : 64 бита

15. Какой объём памяти резервируется под переменную типа char

- : 8 бит
- : 1 бит
- : 16 бит
- : 32 бита
- : 64 бита

16. Какой объём памяти резервируется под переменную типа short integer

- : 16 бит
- : 1 бит
- : 8 бит
- : 32 бита
- : 64 бита

17. Какой объём памяти резервируется под переменную типа long integer

- : 32 бита
- : 1 бит
- : 8 бит
- : 16 бит
- : 64 бита

18. Какой объём памяти резервируется под переменную типа long float

- : 64 бита
- : 1 бит
- : 8 бит
- : 16 бит
- : 32 бита

19. Укажите особенность типа переменных, формируемых с приставкой long

- : Они имеют расширенный диапазон значений
- : Они принимают только положительные значения
- : Они принимают только отрицательные значения
- : Они имеют уменьшенный диапазон значений
- : Они представляют собой ряд целых чисел

20. Укажите особенность типа переменных, формируемых с приставкой `short`

- : Они имеют уменьшенный диапазон значений
- : Они принимают только положительные значения
- : Они принимают только отрицательные значения
- : Они имеют расширенный диапазон значений
- : Они представляют собой ряд целых чисел

21. Укажите особенность типа переменных, формируемых с приставкой `unsigned`

- : Они принимают только положительные значения
- : Они принимают только отрицательные значения
- : Они имеют расширенный диапазон значений
- : Они имеют уменьшенный диапазон значений
- : Они представляют собой ряд целых чисел

22. Оператор `if/else if/else` используется для организации ветвления. Что является аргументом оператора `if`?

- : логическое выражение
- : определённое значение переменной
- : оператор не имеет аргументов
- : переменная, значение которой сравнивается с эталонными значениями.

23. Оператор `if/else if/else` используется для организации ветвления. Что является аргументом оператора `else if`?

- : логическое выражение
- : определённое значение переменной
- : оператор не имеет аргументов
- : переменная, значение которой сравнивается с эталонными значениями.

24. Оператор `if/else if/else` используется для организации ветвления. Что является аргументом оператора `else`?

- : оператор не имеет аргументов
- : логическое выражение
- : определённое значение переменной
- : переменная, значение которой сравнивается с эталонными значениями.

25. Оператор `switch/case` используется для организации ветвления. Что является аргументом оператора `switch`?

- : переменная, значение которой сравнивается с эталонными значениями.
- : логическое выражение

- : определённое значение переменной
- : оператор не имеет аргументов

26. Оператор switch/case используется для организации ветвления. Что является аргументом оператора case?

- : значение сравниваемое со значением переменной, указанной выше.
- : логическое выражение.
- : оператор не имеет аргументов.
- : переменная, значение которой сравнивается с эталонными значениями.

27. Оператор switch/case используется для организации ветвления. Что является аргументом оператора case?

- : значение сравниваемое со значением переменной, указанной выше.
- : логическое выражение.
- : оператор не имеет аргументов.
- : переменная, значение которой сравнивается с эталонными значениями.

28. Оператор switch/case используется для организации ветвления. Что является аргументом оператора default?

- : оператор не имеет аргументов.
- : логическое выражение.
- : значение сравниваемое со значением переменной, указанной выше.
- : переменная, значение которой сравнивается с эталонными значениями.

29. Какой тип цикла формируется оператором «for»?

- : Организация цикла с предусловием.
- : Организация цикла с послеусловием.
- : Данный оператор не формирует цикл.

30. Какой тип цикла формируется оператором «while»?

- : Организация цикла с предусловием.
- : Организация цикла с послеусловием.
- : Данный оператор не формирует цикл.

31. Какой тип цикла формируется оператором «do/while»?

- : Организация цикла с послеусловием.
- : Организация цикла с предусловием.
- : Данный оператор не формирует цикл.

32. В C-- оператор «do/while» используется для организации цикла. Что является аргументом оператора do?

- : оператор не имеет аргументов
- : логическое выражение
- : определённое значение переменной
- : переменная, значение которой сравнивается с эталонными значениями.

33. В C++ оператор «do/while» используется для организации цикла. Что является аргументом оператора while?

- : логическое выражение
- : определённое значение переменной
- : оператор не имеет аргументов
- : переменная, значение которой сравнивается с эталонными значениями.

34. Для чего используются динамические переменные?

- : создание временных переменных, которые можно впоследствии удалить
- : создание переменных, чьи значения можно изменять
- : создание переменных с фиксированным адресом в памяти
- : создание переменных с динамически изменяющимся адресом в памяти.

35. С помощью какого оператора можно удалить динамическую переменную?

- : delete
- : clear
- : destroy
- : remove.

36. Для чего используются динамические переменные?

- : создание временных переменных, которые можно впоследствии удалить
- : создание переменных, чьи значения можно изменять
- : создание переменных с фиксированным адресом в памяти
- : создание переменных с динамически изменяющимся адресом в памяти.

37. Какое определение наиболее точно описывает массив?

- : пространство значений.
- : ряд
- : база данных
- : матрица

38. В чём сущность динамического массива?

- : массив может быть удалён в процессе работы программы
- : массив может быть перенесён из одной ячейки памяти в другую
- : массив может динамически менять свой размер
- : тип данных массива может меняться пользователем в процессе работы

39. Может ли размер динамического массива определяться переменной?

- : да, если значение переменной определяется до определения массива
- : размер динамического массива не может быть изменён, так как его размер изменяется при добавлении данных
- : нет, так как массив определяется до начала работы программы
- : нет, данная функция применима только к динамическому массиву

40. Что означает тип функции int?

- : функция возвращает значение типа int
- : функция возвращает значение типа double
- : функция возвращает значение типа void
- : функция ничего не возвращает

41. Что означает тип функции double?

- : функция возвращает значение типа double
- : функция возвращает значение типа int
- : функция возвращает значение типа void
- : функция ничего не возвращает

42. Что означает тип функции void?

- : функция ничего не возвращает
- : функция возвращает значение типа int
- : функция возвращает значение типа double
- : функция возвращает значение типа void

43. Для чего используется оператор return в конце функции типа int?

- : определяет возвращаемое функцией значение
- : определяет окончание функции
- : является синтаксической ошибкой
- : этот оператор применим только в конце функции main в основном теле программы на C--

44. Для чего используется оператор return в конце функции типа void?

- : является синтаксической ошибкой
- : определяет возвращаемое функцией значение
- : определяет окончание функции
- : этот оператор применим только в конце функции main в основном теле программы на C--

45. Что такое составная переменная?

- : переменная, представляющая собой набор переменных разных типов со своими именами
- : переменная, имя которой состоит из нескольких слов
- : многомерный массив
- : одномерный массив, каждый элемент которого имеет своё имя

46. Допустимо ли использование массивов в качестве элементов составной переменной?

- : Да
- : Нет
- : составная переменная и есть массив

47. Допустимо ли использование переменных разных типов в качестве элементов составной переменной?

- : Да
- : Нет

48. какой тип может иметь составная переменная?

- : составная переменная представляет собой тип данных определяемый пользователем вне основного тела программы
- : составная переменная представляет собой новый тип данных
- : int
- : boolean
- : char
- : double
- : float

49. Для чего используется подключаемый модуль “fstream”?

- : для использования функций чтения и записи данных в/из файлов
- : для использования функций сортировки данных
- : для использования функций математических вычислений
- : для использования функций построения графики в windows

50. Какое ключевое слово используется для записи данных в файл?

- : ofstream
- : ifstream
- : open
- : write

51. Какое ключевое слово используется для чтения данных из файла?

- : ifstream
- : ofstream

	<p>-: open -: write</p> <p>52. Допускается ли запись данных разных типов в файл?</p> <p>-: да -: нет, все данные пишутся в виде однобайтных значений char, соответствующих кодовой записи информации</p> <p>53. Можно ли записать в файл пробел?</p> <p>-: да -: нет, так как в C-- пробел не считывается с клавиатуры</p> <p>54. Можно ли с помощью оператора CIN считать с клавиатуры пробел?</p> <p>-: нет -: Да, при использовании массива типа char</p> <p>55. Можно ли с помощью оператора CIN считать с клавиатуры Enter?</p> <p>-: нет -: Да, при использовании массива типа char</p>
<p>Умеет: ОК 1-9 формализовать поставленную задачу; применять полученные знания к различным предметным областям; составлять и оформлять программы на языках программирования; тестировать и отлаживать программы. ПК 2.1 создавать программы на языке ассемблера для микропроцессорных систем ПК 2.2 производить тестирование, определение параметров и отладку микропроцессорных систем ПК 3.3 принимать участие в отладке и технических испытаниях компьютерных систем и комплексов, инсталляции, конфигурировании программного обеспечения</p>	<p>56. В C-- если присвоить переменной типа Boolean значение 0 это будет соответствовать...</p> <p>-: утверждению «Ложь» -: утверждению «Истина» -: недопустимое значение</p> <p>57. В C-- если присвоить переменной типа Boolean значение 1 это будет соответствовать...</p> <p>-: утверждению «Истина» -: утверждению «Ложь» -: недопустимое значение</p> <p>58. В C-- если присвоить переменной типа Boolean значение 122 это будет соответствовать...</p> <p>-: утверждению «Истина» -: утверждению «Ложь» -: недопустимое значение</p> <p>59. В чем заключается грубая ошибка использования оператора if в следующем фрагменте кода:</p> <pre>if (a=b a==c) { cout << "check completed!"; }</pre> <p>-: некорректное использование инструкции «\Leftarrow». -: недопустимая инструкция . -: некорректное использование инструкции «\Leftarrow» -: неправильный синтаксис при использовании оператора if.</p>

-: использование фигурных скобок недопустимо при одном выполняющемся операторе.

60. В чем заключается грубая ошибка использования оператора if в следующем фрагменте кода:

```
if (a=b&&a== c) {
cout << "check completed!";
}
```

-: некорректное использование инструкции « \Leftarrow ».

-: недопустимая инструкция « $\&\&$ ».

-: некорректное использование инструкции « \Leftarrow ».

-: неправильный синтаксис при использовании оператора if.

-: использование фигурных скобок недопустимо при одном выполняющемся операторе.

61. Укажите строку, в которой оператор "for" использован корректно:

-: for (int i=1; i<=5; i--)

-: for (int i=1; i<=5; i=i-1)

-: for (int i(0); i<=5; i-1)

-: for (int i=1; i<=5; i-1)

62. Сколько итераций будет отработывать следующий цикл:

```
for (int i = 1; i <= 5; i=i-0.5) { ...
```

-: Данный цикл будет бесконечным.

-: 6

-: 8

-: 9

-: 10

63. Для чего используется следующая конструкция?

```
int *a=new int(10)
```

-: создание динамической переменной «a» с начальным значением 10

-: создание переменной «a»

-: создание массива «a» содержащего 10 элементов

-: создание динамического массива «a» содержащего 10 элементов.

64. При работе с динамическими переменными, какая запись корректна с точки зрения синтаксиса?

-: int *a = new int(10);

-: delete a;

-: int a = new int;

-: delete *a;

65. Укажите строку с корректным описанием массива?

-: char s[10];

-: array s[10] of char;

-: s = array of char [10];
 -: char s(10);

66. Укажите строку с корректным описанием массива?

-: char s[10][10];
 -: char s[10,10]
 -: s = array of char [10;10];
 -: char s(10)(10);

67. В программе использована функция int a(f).
 Внутри функции использовалась переменная z=50. Допустимо ли использование данной переменной вне функции, после работы последней.

-: нет, это вызовет ошибку
 -: нет, так как значение данной переменной может измениться
 -: да, так как данная переменная является статической

68. Укажите корректное обращение к файлу (/ - означает вторую строку):

-: ofstream f("path");
 -: ofstream f; / f.open("path");
 -: ofstream.open f("path");
 -: f=ofstream.open("path");

69. Что означает следующая строка: ofstream f("path",ios_base::trunc)?

-: удалить содержимое файла если он существует
 -: открыть файл для чтения
 -: открыть файл для записи
 -: при открытии переместить указатель в конец файла
 -: открыть файл для записи в конец файла
 -: открыть файл в двоичном режиме

70. Что означает следующая строка: ofstream f("path",ios_base::ate)?

-: при открытии переместить указатель в конец файла
 -: открыть файл для чтения
 -: открыть файл для записи
 -: открыть файл для записи в конец файла
 -: удалить содержимое файла если он существует
 -: открыть файл в двоичном режиме

71. Что означает следующая строка: ofstream f("path",ios_base::binary)?

-: открыть файл в двоичном режиме
 -: открыть файл для чтения
 -: открыть файл для записи
 -: при открытии переместить указатель в конец файла
 -: открыть файл для записи в конец файла
 -: удалить содержимое файла если он существует

	<p>72. Что означает следующая строка: ofstream f("path",ios_base::app)?</p> <ul style="list-style-type: none"> -: открыть файл для записи в конец файла -: открыть файл для чтения -: открыть файл для записи -: при открытии переместить указатель в конец файла -: удалить содержимое файла если он существует -: открыть файл в двоичном режиме
<p>Имеет практический опыт: ОК 1-9 составления и оформления программ на языках программирования; тестирования и отлаживания программ ПК 2.1 создания программ на языке ассемблера ПК 2.2 выполнения тестирования, определения параметров и отладки ПК 3.3 участия в отладке и технических испытаниях</p>	<p>Выполнение лабораторных работ:</p> <ol style="list-style-type: none"> 1. Лабораторная работа № 1. Проектирование алгоритмов. Блок-схема алгоритма. 2. Лабораторная работа № 2. Знакомство с редактором Free Pascal 3. Лабораторная работа № 3. Операторы циклов. 4. Лабораторная работа № 4. Процедуры. Функции. 5. Лабораторная работа № 5. Массивы. 6. Лабораторная работа № 6. Форматирование ввода-вывода. Работа с файлами. Структуры. Классы.

7.2. Методические рекомендации к определению процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Рабочая учебная программа дисциплины содержит следующие структурные элементы:

- перечень компетенций, формируемых в результате изучения дисциплины в процессе освоения образовательной программы;
- типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности в процессе освоения образовательной программы (далее—задания). Задания по каждой компетенции, как правило, не должны повторяться.

Требования по формированию задания на оценку ЗНАНИЙ:

- обучающийся должен воспроизводить и объяснять учебный материал с требуемой степенью научной точности и полноты;
- применяются средства оценивания компетенций: тестирование, вопросы по основным понятиям дисциплины и т.п.

Требования по формированию задания на оценку УМЕНИЙ:

- обучающийся должен решать типовые задачи (выполнять задания) на основе воспроизведения стандартных алгоритмов решения;
- применяются следующие средства оценивания компетенций: простые ситуационные задачи (задания) с коротким ответом или простым действием, упражнения, задания на соответствие или на установление правильной последовательности, эссе и другое.

Требования по формированию задания на оценку навыков и (или) ОПЫТА ДЕЯТЕЛЬНОСТИ:

- обучающийся должен решать усложненные задачи (выполнять задания) на основе приобретенных знаний, умений и навыков, с их применением в определенных ситуациях;
- применяются средства оценивания компетенций: задания требующие многошаговых решений как в известной, так и в нестандартной ситуациях, задания,

требующие поэтапного решения и развернутого ответа, ситуационные задачи, проектная деятельность, задания расчетно-графического типа. Средства оценивания компетенций выбираются в соответствии с заявленными результатами обучения по дисциплине.

Процедура выставления оценки доводится до сведения обучающихся в течение месяца с начала изучения дисциплины путем ознакомления их с технологической картой дисциплины, которая является неотъемлемой частью рабочей учебной программы по дисциплине.

В результате оценивания компетенций по дисциплине студенту начисляются баллы по шкале, указанной в рабочей учебной программе по дисциплине.

7.3. Описание показателей и критериев оценивания компетенций, описание шкал оценивания

Успешность усвоения дисциплины характеризуется качественной оценкой на основе листа оценки сформированности компетенций, который является приложением к зачетно-экзаменационной ведомости при проведении промежуточной аттестации по дисциплине.

Критерии оценивания компетенций

Компетенция считается сформированной, если теоретическое содержание курса освоено полностью; при устных собеседованиях студент исчерпывающе, последовательно, четко и логически стройно излагает учебный материал; свободно справляется с задачами, вопросами и другими видами заданий, требующих применения знаний, использует в ответе дополнительный материал; все предусмотренные рабочей учебной программой задания выполнены в соответствии с установленными требованиями, студент способен анализировать полученные результаты; проявляет самостоятельность при выполнении заданий, качество их выполнения оценено числом баллов от 86 до 100, что соответствует *повышенному уровню* сформированности компетенции.

Компетенция считается сформированной, если теоретическое содержание курса освоено полностью; при устных собеседованиях студент последовательно, четко и логически стройно излагает учебный материал; справляется с задачами, вопросами и другими видами заданий, требующих применения знаний; все предусмотренные рабочей учебной программой задания выполнены в соответствии с установленными требованиями, студент способен анализировать полученные результаты; проявляет самостоятельность при выполнении заданий, качество их выполнения оценено числом баллов от 61 до 85,9, что соответствует *пороговому уровню* сформированности компетенции.

Компетенция считается несформированной, если студент при выполнении заданий не демонстрирует знаний учебного материала, допускает ошибки, неуверенно, с большими затруднениями выполняет практические работы, не демонстрирует необходимых умений, доля невыполненных заданий, предусмотренных рабочей учебной программой составляет 55 %, качество выполненных заданий не соответствует установленным требованиям, качество их выполнения оценено числом баллов ниже 61, что соответствует *допороговому уровню*.

Шкала оценки уровня освоения дисциплины

Качественная оценка может быть выражена: в процентном отношении качества усвоения дисциплины, которая соответствует баллам, и переводится в уровневую шкалу и оценки «отлично» / 5, «хорошо» / 4, «удовлетворительно» / 3, «неудовлетворительно» / 2, «зачтено», «не зачтено». Преподаватель ведет письменный учет текущей успеваемости студента в соответствии с технологической картой по дисциплине.

Шкала оценки результатов освоения дисциплины, сформированности компетенций

Шкалы оценки уровня сформированности компетенции (й)		Шкала оценки уровня освоения дисциплины		
<i>Уровневая шкала оценки компетенций</i>	<i>100 балльная шкала,</i>	<i>100 балльная шкала,</i>	<i>5-балльная шкала, дифференцированная оценка/балл</i>	<i>недифференцированная оценка</i>

	%	%		
допороговый	ниже 61	ниже 61	«неудовлетворительно» / 2	Не зачтено
пороговый	61-85,9	70-85,9	«хорошо» / 4	зачтено
		61-69,9	«удовлетворительно» / 3	зачтено
повышенный	86-100	86-100	«отлично» / 5	зачтено

8. Учебно-методическое и информационное обеспечение дисциплины

8.3. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Списки основной литературы

1. Гвоздева, В. А. Информатика, автоматизированные информационные технологии и системы [Электронный ресурс] : учеб. для студентов техн. специальностей / В. А. Гвоздева. - Документ Bookread2. - М. : ФОРУМ [и др.], 2015. - 541 с. - Режим доступа: <http://znanium.com/bookread2.php?book=492670#>
2. Колдаев, В. Д. Основы алгоритмизации и программирования [Электронный ресурс] : учеб. пособие для сред. проф. образования по группе специальностей "Информатика и вычисл. техника" / В. Д. Колдаев под ред. Л. Г. Гагариной. - Документ Bookread2. - М. : ФОРУМ [и др.], 2015. - 413 с. : ил., табл. - Режим доступа: <http://znanium.com/bookread2.php?book=484837#>
3. Фризен, И. Г. Основы алгоритмизации и программирования (среда PascalABC.NET) [Электронный ресурс] : учеб. пособие для сред. проф. образования по специальностям 09.02.04 "Информ. системы (по отраслям)", 09.02.03 "Программ. в компьютер. системах" / И. Г. Фризен. - Документ Bookread2. - М. : Форум [и др.], 2017. - 391 с. - Режим доступа: <http://znanium.com/bookread2.php?book=559358>

Списки дополнительной литературы

4. Голицына, О. Л. Языки программирования [Электронный ресурс] : учеб. пособие для сред. проф. образования / О. Л. Голицына, Т. Л. Партыка, И. И. Попов. - 3-е изд., перераб. и доп. - Документ Bookread2. - М. : Форум [и др.], 2015. - 398 с. - Режим доступа: <http://znanium.com/bookread2.php?book=493421>

8.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет" (далее - сеть "Интернет"), необходимых для освоения дисциплины

Интернет-ресурсы

1. ИНТУИТ. Национальный Открытый Университет [Электронный ресурс]. - Режим доступа: <http://www.intuit.ru/>. - Загл. с экрана.
2. Образовательные ресурсы Интернета. Информатика [Электронный ресурс]. - Режим доступа: <http://www.alleng.ru/edu/comp.htm>. - Загл. с экрана.
3. Электронная библиотека. Техническая литература [Электронный ресурс]. - Режим доступа: <http://techliter.ru/>. - Загл. с экрана.
4. Электронная библиотечная система Поволжского государственного университета сервиса [Электронный ресурс]. - Режим доступа: <http://elib.tolgas.ru/>. - Загл. с экрана.
5. Электронно-библиотечная система Znanium.com [Электронный ресурс]. - Режим доступа: <http://znanium.com/>. - Загл. с экрана.

9. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем

Краткая характеристика применяемого программного обеспечения

№ п/п	Программный продукт	Характеристика	Назначение при освоении дисциплины
1	Операционная система Microsoft Windows	Системное ПО: операционная система Microsoft Windows 7	Выполнение лабораторных работ
2	Пакет Microsoft Office	Офисный пакет приложений. В состав этого пакета входит программное обеспечение для работы с различными типами документов: текстами, электронными таблицами, базами данных и др.	Выполнение лабораторных работ
3	Система программирования Free Pascal	Интегрированная среда разработки программного обеспечения	Выполнение лабораторных работ
4	Moodle	Система управления курсами, также известная как система управления обучением. Представляет собой свободное веб-приложение, предоставляющее возможность создавать сайты для онлайн-обучения.	Тестирование

10. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Реализация программы дисциплины в соответствии с требованиями ФГОС СПО по специальности требует наличие учебного кабинета, укомплектованного специализированной мебелью, техническими средствами обучения, и комплексная лаборатория диагностирования и технического обслуживания, оснащенной лабораторным оборудованием различной степени сложности

