



1920

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Филиал федерального государственного бюджетного образовательного
учреждения высшего образования
«Кубанский государственный университет» в г. Геленджике



УТВЕРЖДАЮ

Проректор по работе с филиалами

 А.А. Евдокимов

2020 г.

**Рабочая программа дисциплины
ОП.08 ТЕОРИЯ АЛГОРИТМОВ**

специальность 09.02.03 Программирование в компьютерных системах

Рабочая программа учебной дисциплины **ОП.08 Теория алгоритмов** разработана на основе Федерального государственного образовательного стандарта (далее – ФГОС) по специальности среднего профессионального образования (далее СПО) 09.02.03 Программирование в компьютерных системах, утвержденного приказом Минобрнауки РФ от 28.07.2014 №804 (зарегистрирован в Минюсте России 21.08.2014 № 33733)

Дисциплина	ОП.08 Теория алгоритмов	
Форма обучения	очная	
Учебный год	2020-2021	
3 курс		5 семестр
лекции		56 час.
практические занятия		56 час.
самостоятельные занятия		56 час.
форма итогового контроля		экзамен

Составитель: преподаватель



Т.П. Кривошеенко

подпись

Утверждена на заседании предметной (цикловой) комиссии профессиональных дисциплин специальности Программирование в компьютерных системах протокол № 10 от «27» мая 2020 г.

Председатель предметной (цикловой) комиссии профессиональных дисциплин специальности Программирование в компьютерных системах



Л.А. Благова

« 27 » мая 2020 г.

Рецензент (-ы):

Директор ООО «Современные
информационные технологии»

А.В.Сметанин

Программист ГБУЗ
"Геленджикский психоневрологический диспансер"
министерства здравоохранения Краснодарского края

Е.В.Мельников

ЛИСТ

согласования рабочей учебной программы по дисциплине
ОП.08 Теория алгоритмов

Специальность среднего профессионального образования:
09.02.03 Программирование в компьютерных системах

СОГЛАСОВАНО:

Зам. директора по УР филиала



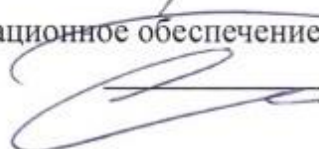
Т. А. Резуненко
«27» мая 2020 г.

Заведующая сектором библиотеки



Л. Г. Соколова
«27» мая 2020 г.

Инженер-электроник (программно-информационное обеспечение
образовательной программы)



А. В. Сметанин
«27» мая 2020 г.

СОДЕРЖАНИЕ

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ.....	5
1.1 Область применения программы	5
1.2. Место учебной дисциплины в структуре программы подготовки специалистов среднего звена:	5
1.3. Цели и задачи учебной дисциплины – требования к результатам освоения дисциплины:	5
1.4. Перечень планируемых результатов обучения по дисциплине (перечень формируемых компетенций)	6
2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ	8
2.1. Объем учебной дисциплины и виды учебной работы	8
2.2. Структура дисциплины:	8
2.3. Тематический план и содержание учебной дисциплины	9
2.4. Содержание разделов дисциплины	12
2.4.1. Занятия лекционного типа	12
2.4.2. Занятия семинарского типа	13
2.4.3. Практические занятия (Лабораторные занятия).....	13
2.4.4. Содержание самостоятельной работы (Примерная тематика рефератов).....	13
2.4.5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине	14
3. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ	16
3.1. Образовательные технологии при проведении лекций	16
3.2. Образовательные технологии при проведении практических занятий	16
4. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ ДИСЦИПЛИНЫ.....	17
4.1. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине	17
4.2. Перечень необходимого программного обеспечения	17
5. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ	19
5.1. Основная литература	19
5.2. Дополнительная литература	19
5.3. Периодические издания	20
5.4. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	20
6. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ	22
7. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ КОНТРОЛЯ УСПЕВАЕМОСТИ.....	23
7.1. Паспорт фонда оценочных средств.....	23
7.2. Критерии оценки знаний.....	23
7.3. Оценочные средств для проведения для текущей аттестации	24
7.4. Оценочные средств для проведения промежуточной аттестации	26
7.4.1. Примерные вопросы для проведения промежуточной аттестации	26
7.4.2. Примерные задачи для проведения промежуточной аттестации.....	27
8. ДОПОЛНИТЕЛЬНОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ.....	30

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

ОП.08 Теория алгоритмов

1.1. Область применения программы

Рабочая программа учебной дисциплины **ОП.08 Теория алгоритмов** является частью программы подготовки специалистов среднего звена в соответствии с ФГОС СПО по специальности 09.02.03 Программирование в компьютерных системах.

1.2. Место учебной дисциплины в структуре программы подготовки специалистов среднего звена:

Дисциплина **ОП.08 Теория алгоритмов** входит в основную часть ППССЗ.

Для освоения дисциплины студенты используют знания, умения и навыки, сформированные при изучении предметов: Основы программирования, Прикладное программирование и др.

Изучение дисциплины «**ОП.08 Теория алгоритмов**» предваряет Технология разработки ПО, Технология разработки и защиты баз данных.

1.3. Цели и задачи учебной дисциплины ОП.08 Теория алгоритмов – требования к результатам освоения

Целью является получение студентами представления об алгоритмизации и о содержании программирования как научных дисциплин, ознакомление их с основными понятиями, принципами, методологией, методиками анализа алгоритмов и программирования.

Задачи предусматривают систематизацию, закрепление и расширение теоретических и практических знаний по проблемам алгоритмизации и программирования.

В результате изучения профессионального модуля обучающийся должен **уметь:**

- разрабатывать алгоритмы для конкретных задач;
- определять сложность работы алгоритмов.

знать:

- основные модели алгоритмов;
- методы построения алгоритмов;
- методы вычисления сложности работы алгоритмов;

Максимальная учебная нагрузка обучающегося 168 часов, в том числе:

- обязательная аудиторная учебная нагрузка обучающегося 112 часа;
- самостоятельная работа обучающегося 56 часа.

1.4. Перечень планируемых результатов обучения по дисциплине ОП.08 Теория алгоритмов

Учащийся должен обладать **общими и профессиональными компетенциями**, включающими в себя способности:

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 в команде, эффективно общаться с коллегами, руководством, потребителями.

ОК 7. Брать на себя ответственность за работу членов команды (подчиненных), за 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 1.1. Выполнять разработку спецификаций отдельных компонент.

ПК 1.2. Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля.

Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
		знать	уметь	иметь практический опыт
ОК 1.	Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.	сущность и социальную значимость будущей профессии.	проявлять к будущей профессии устойчивый интерес	-повышение успеваемости по МДК, положительный отзыв руководителя практики. -систематического посещения учебных занятий и практики, консультаций.
ОК 2.	Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.	типовые методы и способы выполнения профессиональных задач.	организовывать собственную деятельность, оценивать эффективность и качество профессиональных задач.	-мотивированного обоснования выбора и применения методов и способов решения профессиональных задач. -точного, правильного и полного выполнения профессиональных задач. -разработки пользовательского интерфейса
ОК 3.	Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.	основы нормативной в области разработки и эксплуатации программных продуктов.	принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.	-демонстрации способности 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.	перечень профессиональных задач и способы их эффективного решения.	осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.	-обоснования выбора информационных источников для решения профессиональных задач. -оперативности поиска и использования необходимой информации для качественного выполнения профессиональных задач и личностного развития. -использования различных источников информации, включая электронные.
ОК 5.	Использовать	современное ПО для	использовать	-осуществления операций с

	информационно-коммуникационные технологии в профессиональной деятельности.	поддержки информационно-коммуникационных технологий	информационно-коммуникационные технологии в профессиональной деятельности.	использованием общего и специализированного программного обеспечения. -создания отдельных компонент.
ОК 6.	Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями.	методы коллективной разработки ПО	проводить удаленное решение проблем заказчика.	-выполнения спецификаций компонент
ОК 7.	Брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий.	методы декомпозиции задач	вести график выполняемых работ	тестирования компонент и проекта в целом
ОК 8.	Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.	задачи профессионального и личностного развития.	заниматься самообразованием, осознанно планировать повышение квалификации.	-качественного, своевременного и полного выполнения заданий внеаудиторной самостоятельной работы. -обоснования постановки целей и задач самообразования. -планирования создания кода программного продукта на уровне модуля.
ОК 9.	Ориентироваться в условиях частой смены технологий в профессиональной деятельности.	направления и перспективы развития технологий в области разработки и эксплуатации ПО.	ориентироваться в условиях частой смены технологий в профессиональной деятельности.	-анализа инноваций в области профессиональной деятельности; -отслеживания динамики развития языков программирования и средств его автоматизации.
ПК 1.1.	Выполнять разработку спецификаций отдельных компонент.	*основные модели алгоритмов; * методы построения алгоритмов;	* разрабатывать алгоритмы для конкретных задач;	-разрабатывать алгоритмы типовых задач; -разрабатывать по заданному алгоритму блок-схему программы.
ПК 1.2.	Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля.	* методы вычисления сложности работы алгоритмов;	* определять сложность работы алгоритмов.	разрабатывать программу модуля по имеющейся блок-схеме.

2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Всего
	часов
Учебная нагрузка (всего)	168
Аудиторные занятия (всего)	112
В том числе:	
занятия лекционного типа	56
практические занятия (практикумы)	56
лабораторные занятия	
Самостоятельная работа (всего)	56
в том числе:	
<i>Самостоятельная внеаудиторная работа в виде домашних практических заданий, индивидуальных заданий, самостоятельного подбора и изучения дополнительного теоретического материала</i>	56
Вид промежуточной аттестации (экзамен)	Экзамен
Общая трудоемкость час	168

2.2. Структура дисциплины:

Наименование разделов и тем	Всего	Количество аудиторных часов		Самостоятельная работа студента (час)
		Теоретическое обучение	Практические и лабораторные занятия	
Тема 1. Этапы решения задач на ЭВМ	24	8	8	8
Тема 2. Алгоритмы и исполнители	18	6	6	6
Тема 3. Алгоритм. Свойства алгоритмов	18	6	6	6
Тема 4. Алгоритмическая конструкция ветвления	18	6	6	6
Тема 5. Алгоритмическая конструкция цикла	18	6	6	6
Тема 6. Процедуры и функции	18	6	6	6
Тема 7. Алгоритмы для файлов	18	6	6	6
Тема 8. Объектно - ориентированное программирование	18	6	6	6
Тема 9. Использование функций в приближенных вычислениях	18	6	6	6
Всего по дисциплине	168	56	56	56

2.3. Тематический план и содержание учебной дисциплины ОП.08 Теория алгоритмов

Наименование разделов и тем	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная работа обучающихся	Объем часов	Уровень освоения
1	2	3	4
Тема 1. Этапы решения задач на ЭВМ	Содержание учебного материала	24	
	Лекции. Постановка задачи. Формализация. Выбор (или разработка) метода решения. Разработка алгоритма, блок-схемы. Составление программы. Отладка программы. Вычисление и обработка результатов.	8	2
	Практические занятия. Тема: Алгоритм в виде блок-схемы	8	2
	Самостоятельная работа обучающихся: Выяснение конечной цели и выработка общего подхода к решению задачи. Выясняется, сколько решений имеет задача и имеет ли их вообще. Изучаются общие свойства рассматриваемого физического явления или объекта.	8	2
Тема 2. Алгоритмы и исполнители	Содержание учебного материала	18	
	Лекции. Основные алгоритмы обработки информации, алгоритмы над числами; алгоритмы последовательного и бинарного поиска; алгоритмы сортировки; хеш-функции и методы исключения коллизий	6	3
	Практические занятия. Тема: Программная система «Исполнители»	6	2
	Самостоятельная работа обучающихся: «Черепашка», Робот, мозаика.	6	2
Тема 3. Алгоритм. Свойства алгоритмов.	Содержание учебного материала	18	
	Лекции. Алгоритм как определенным образом организованная последовательность действий, за конечное число шагов приводящая к решению задачи. Свойства: Определенность. Дискретность. Целенаправленность. Конечность. Массовость	6	2
	Практические занятия. Тема: Графический вид (язык блок-схем). ГОСТ 19.003-80, ИСО 1028-73. Регламент использования графических примитивов.	6	2
	Самостоятельная работа обучающихся: Языки алгоритмов: Естественный (словесная запись), Формулы, Псевдокод, Структурограммы, Синтаксические диаграммы	6	2
Тема 4. Алгоритмическая конструкция ветвления	Содержание учебного материала	18	
	Лекции. Ветвление - управляющая структура, организующая выполнение лишь одного из двух указанных действий в зависимости от справедливости некоторого условия.	6	2
	Практические занятия. Программирование ветвлений. Пример: найти наименьшее из трех чисел.	6	2
	Самостоятельная работа обучающихся: Условие - вопрос, имеющий два варианта ответа: да или нет. Запись ветвления выполняется в двух формах: полной и неполной.	6	2

Тема 5. Алгоритмическая конструкция цикла	Содержание учебного материала	18	
	Лекции. Цикл "пока". Цикл "до". Параметр цикла, его начальное и конечное значения и шаг должны быть одного типа. Запрещено изменять в теле цикла значения начальное, текущее и конечное для параметра	6	2
	Практические занятия. Тема: Использование циклов с параметром для обработки массивов	6	2
	Самостоятельная работа обучающихся: Массив задается именем, типом данных и размерностью - максимально возможное количество элементов в массиве.	6	2
Тема 6. Процедуры и функции	Содержание учебного материала	18	
	Лекции. Использование: Алгоритм или программа содержат одинаковые действия, различающиеся, возможно исходными данными; Решаемая задача состоит из нескольких задач, меньших по объему и сложности; Решением задачи занимается коллектив программистов.	6	2
	Практические занятия. Тема: Объявление функции и обращение к функции.	6	2
	Самостоятельная работа обучающихся: Задача: вычислить факториалы первых 10 натуральных чисел:	6	2
Тема 7. Алгоритмы для файлов	Содержание учебного материала	18	
	Лекции. Файл - это последовательность однотипных компонентов. Компонентом файла может быть значение простого типа или структура. Различают текстовые, типизированные и нетипизированные файлы.	6	2
	Практические занятия. Тема: Файловые операции, описание, открытие, чтение, запись, управление указателем.	6	2
	Самостоятельная работа обучающихся: Сортировка файлов. Через рабочий массив или файл.	6	2
Тема 8. Объектно - ориентированное программирование	Содержание учебного материала	18	
	Лекции. Объект как усовершенствование типа запись, в которой описание свойств и параметры моделируемой сущности дополняются методами - описаниями действий с объектом.	6	2
	Практические занятия. Тема: Управление объектами на форме и из программы.	6	2
	Самостоятельная работа обучающихся: Управление программой с помощью событий.	6	2
Тема 9. Использование функций в приближенных вычислениях.	Содержание учебного материала	18	
	Лекции. Алгоритмы вычисления интегралов. Правила написания процедур, параметры входные и выходные.	6	2
	Практические занятия. Тема: Использование процедур и функций для повышения надежности программ.	6	2
	Самостоятельная работа обучающихся: Локализация переменных внутри процедуры.	6	2

Итого	Лекции	56	
	Практические занятия.	56	
	Самостоятельная работа обучающихся	56	

Для характеристики уровня освоения учебного материала используются следующие обозначения: 1. – ознакомительный (узнавание ранее изученных объектов, свойств); 2. – репродуктивный (выполнение деятельности по образцу, инструкции или под руководством) 3. – продуктивный (планирование и самостоятельное выполнение деятельности, решение проблемных задач)

2.4. Содержание разделов дисциплины

2.4.1. Занятия лекционного типа

№ тем ы	Наименование темы	Содержание темы	Форма текущего контроля
1	2	3	4
1	Этапы решения задач на ЭВМ	Постановка задачи. Формализация. Выбор (или разработка) метода решения. Разработка алгоритма, блок-схемы. Составление программы Отладка программы. Вычисление и обработка результатов.	Т, У
2	Алгоритмы и исполнители	Основные алгоритмы обработки информации, алгоритмы над числами; алгоритмы последовательного и бинарного поиска; алгоритмы сортировки; хеш-функции и методы исключения коллизий	Т, У
3	Алгоритм. Свойства алгоритмов	Алгоритм как определенным образом организованная последовательность действий, за конечное число шагов приводящая к решению задачи. Свойства: Определенность. Дискретность. Целенаправленность. Конечность. Массовость	Т, У
4	Алгоритмическая конструкция ветвления	Ветвление - управляющая структура, организующая выполнение лишь одного из двух указанных действий в зависимости от справедливости некоторого условия.	Т, У
5	Алгоритмическая конструкция цикла	Цикл "пока". Цикл "до". Параметр цикла, его начальное и конечное значения и шаг должны быть одного типа. Запрещено изменять в теле цикла значения начальное, текущее и конечное для параметра	Т, У
6	Процедуры и функции	Использование: Алгоритм или программа содержат одинаковые действия, различающиеся, возможно исходными данными; Решаемая задача состоит из нескольких задач, меньших по объему и сложности; Решением задачи занимается коллектив программистов.	Т, У
7	Алгоритмы для файлов	Файл - это последовательность однотипных компонентов. Компонентом файла может быть значение простого типа или структура. Различают текстовые, типизированные и нетипизированные файлы.	Т, У
8	Объектно - ориентированное программирование	Объект как усовершенствование типа запись, в которой описание свойств и параметры моделируемой сущности дополняются методами - описаниями действий с объектом	Т, У
9	Использование функций в приближенных вычислениях	Алгоритмы вычисления интегралов. Правила написания процедур, параметры входные и выходные.	Т, У
Примечание: Т – тестирование, У – устный опрос			

2.4.2. Занятия семинарского типа

не предусмотрены

2.4.3. Практические занятия

№ тем ы	Наименование темы	Содержание темы	Форма текущего контроля
1	2	3	4
1	Этапы решения задач на ЭВМ	Алгоритм в виде блок-схемы	Т, У
2	Алгоритмы и исполнители	Программная система «Исполнители»	Т, У
3	Алгоритм. Свойства алгоритмов	Графический вид (язык блок-схем). ГОСТ 19.003-80, ИСО 1028-73. Регламент использования графических примитивов.	Т, У

№ тем ы	Наименование темы	Содержание темы	Форма текущего контроля
1	2	3	4
4	Алгоритмическая конструкция ветвления	Программирование ветвлений. Пример: найти наименьшее из трех чисел.	Т, У
5	Алгоритмическая конструкция цикла	Использование циклов с параметром для обработки массивов	Т, У
6	Процедуры и функции	Объявление функции и обращение к функции.	Т, У
7	Алгоритмы для файлов	Файловые операции, описание, открытие, чтение, запись, управление указателем.	Т, У
8	Объектно - ориентированное программирование	Управление объектами на форме и из программы.	Т, У
9	Использование функций в приближенных вычислениях	Использование процедур и функций для повышения надежности программ.	Т, У
Примечание: Т – тестирование, У – устный опрос			

2.4.4. Содержание самостоятельной работы

На самостоятельную работу студентов отводится 20 часов учебного времени.

№	Наименование темы, вида СРС	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
1	Выяснение конечной цели и выработка общего подхода к решению задачи. Выясняется, сколько решений имеет задача и имеет ли их вообще. Изучаются общие свойства рассматриваемого физического явления или объекта.	1. Игошин, В.И. Теория алгоритмов: учебное пособие для СПО. - М.: Академия, 2013.-316с. 2. Гринченков, Д.В. Математическая логика и теория алгоритмов для программистов: учебное пособие для вузов/Д.В. Гринченков, С.И. Потоцкий.-М.: Кнорус,2014.-206с. 3. <i>Баврин, И. И.</i> Дискретная математика. Учебник и задачник [Электронный ресурс]: для СПО / И. И. Баврин. — М.: Издательство Юрайт, 2017. — 209 с. - URL: https://www.biblio-online.ru/book/46422B2A-1497-4FFD-8A53-143190428418 4. <i>Попов, А. М.</i> Информатика и математика [Электронный ресурс]: учебник и практикум для СПО / А. М. Попов, В. Н. Сотников, Е. И. Нагаева; под ред. А. М. Попова. — 3-е изд., перераб. и доп. — М.: Издательство Юрайт, 2016. — 430 с. - URL: https://www.biblio-online.ru/book/552AAA0D-1F2E-434C-8F8B-690CA6280464 5. <i>Судоплатов, С. В.</i> Математическая логика и теория алгоритмов [Электронный ресурс]: учебник и практикум / С. В. Судоплатов, Е. В. Овчинникова. — 5-е изд., стер. — М.: Издательство Юрайт, 2016. — 255 с. - URL: https://www.biblio-online.ru/book/71FA118B-CFD5-48BD-BC6F-073BDCA2806F 6. <i>Перельман, Я. И.</i> Веселые задачи [Электронный ресурс] / Я. И. Перельман. — М.: Издательство Юрайт, 2017. — 292 с. - URL: https://www.biblio-online.ru/book/AF291AE5-1CDF-4521-A4A0-4AC499D7D144 7. Баврин, И.И. Математическая обработка информации [Электронный ресурс]: учебник / И.И. Баврин. - М.: Прометей, 2016. - 261 с. - URL: http://biblioclub.ru/index.php?page=book_red&id=439182&sr=1 8. <i>Черпаков, И. В.</i> Основы программирования [Электронный ресурс]: учебник и практикум для СПО / И. В. Черпаков. — М.: Издательство Юрайт, 2017. — 219 с. - URL: https://www.biblio-online.ru/book/F79BE55A-C6F1-439D-9ED5-0D78A50B403F
2	«Черепашка», Робот, мозаика.	
3	Языки алгоритмов: Естественный (словесная запись), Формулы, Псевдокод, Структурограммы, Синтаксические диаграммы	
4	Условие - вопрос, имеющий два варианта ответа: да или нет. Запись ветвления выполняется в двух формах: полной и неполной.	
5	Массив задается именем, типом данных и размерностью - максимально возможное количество элементов в массиве.	
6	Задача: вычислить факториалы первых 10 натуральных чисел:	
7	Сортировка файлов. Через рабочий массив или файл.	
8	Управление программой с помощью событий.	
9	Локализация переменных внутри процедуры.	

Кроме перечисленных источников студент может воспользоваться поисковыми системами сети Интернет по теме самостоятельной работы.

2.4.5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

Самостоятельная работа студентов является важнейшей формой учебно-познавательного процесса.

Основная цель самостоятельной работы студента при изучении дисциплины – закрепить теоретические знания, полученные в ход лекционных занятий, а также сформировать практические навыки подготовки в области алгоритмизации задач.

Самостоятельная работа студента в процессе освоения дисциплины включает:

- изучение основной и дополнительной литературы по курсу;
- самостоятельное изучение некоторых вопросов (конспектирование);
- работу с электронными учебными ресурсами;
- изучение материалов периодической печати, интернет ресурсов;
- подготовку к тестированию;
- подготовку к практическим (лабораторным) занятиям,
- самостоятельное выполнение домашних заданий.

Для помощи в самостоятельной работе рекомендуется применять электронный учебник (учебное пособие) **Программирование.СНМ**, разработанное Левиным Л.Л.

Для освоения данной дисциплины и выполнения предусмотренных учебной программой курса заданий по самостоятельной работе студент может использовать следующее учебно-методическое обеспечение:

- обучающие видеофильмы и программы по тематике решаемых задач из **Видеотеки программирования** филиала (225 единиц);
- программу компьютерного обучения и контроля “**ЭкзамL**”;
- электронный учебник по прикладному программированию;
- методические рекомендации преподавателя к лекционному материалу;
- методические рекомендации преподавателя к практическим занятиям;
- методические рекомендации преподавателя к выполнению самостоятельных домашних заданий.

Началом организации любой самостоятельной работы должно быть привитие навыков и умений грамотной работы с учебной и научной литературой. Этот процесс, в первую очередь, связан с нахождением необходимой для успешного овладения учебным материалом литературой. Студент должен уметь пользоваться фондами библиотек и справочно-библиографическими изданиями.

Студенты для полноценного освоения учебного курса должны составлять конспекты как при прослушивании его теоретической (лекционной) части, так и при подготовке к практическим (лабораторным) занятиям. Желательно, чтобы конспекты лекций записывались в логической последовательности изучения курса и содержались в одной тетради.

3. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Для реализации компетентностного подхода предусматривается использование в учебном процессе компьютерных активных и интерактивных форм проведения аудиторных и внеаудиторных занятий с целью формирования и развития профессиональных навыков обучающихся.

В процессе преподавания применяются образовательные технологии развития критического мышления. Обязательны компьютерные практические работы по разделам дисциплины.

В учебном процессе наряду с традиционными образовательными технологиями используются электронные учебники, компьютерное обучение, тестирование, учебные видеофильмы, тематические презентации, интерактивные технологии.

3.1. Образовательные технологии при проведении лекций

№	Тема	Виды применяемых образовательных технологий	Кол-во час
1	2	3	4
1	Этапы решения задач на ЭВМ	Компьютерные технологии обучения, активное обучение, тестирование.	8*
2	Алгоритмы и исполнители		6*
3	Алгоритм. Свойства алгоритмов		6*
4	Алгоритмическая конструкция ветвления		6*
5	Алгоритмическая конструкция цикла		6*
6	Процедуры и функции		6*
7	Алгоритмы для файлов		6*
8	Объектно - ориентированное программирование		6*
9	Использование функций в приближенных вычислениях		6*
Итого по курсу			56
в том числе интерактивное обучение*			56*

3.2. Образовательные технологии при проведении практических занятий (лабораторных работ)

№	Тема занятия	Кол. час	Виды применяемых образовательных технологий
1	Алгоритм в виде блок-схемы	8*	Компьютерные технологии обучения. Активное обучение. Дискуссия по теоретическим вопросам. Решение задач индивидуально. Решение задач малыми группами. Разбор решения
2	Программная система «Исполнители»	6*	
3	Графический вид (язык блок-схем). ГОСТ 19.003-80, ИСО 1028-73. Регламент использования графических примитивов.	6*	
4	Программирование ветвлений. Пример: найти наименьшее из трех чисел.	6*	
5	Использование циклов с параметром для обработки массивов	6*	
6	Объявление функции и обращение к функции.	6*	
7	Файловые операции, описание, открытие, чтение, запись, управление указателем.	6*	
8	Управление объектами на форме и из программы.	6*	

9	Использование процедур и функций для повышения надежности программ.	6*	задач.
	Итого по курсу	56	
	в том числе интерактивное обучение*	56*	

4. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ ДИСЦИПЛИНЫ

4.1. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Реализация учебной дисциплины осуществляется в специально оборудованном компьютерном классе.

Оборудование учебного кабинета:

- мультимедийный проектор, экран;
- персональный компьютер, динамики;
- выход в Интернет;
- учебная мебель;
- доска учебная;

Наглядные пособия:

1. Видеофильм Программное прикладное обеспечение.mp4
2. Видеофильм Классификация ПО.mp4
3. 3 видеофильма Программирование игры пинг-понг. mp4
4. 2 видеофильма Проектирование меню приложения. mp4
5. 2 видеофильма Использование компоненты MediaPlayer. mp4
6. 2 видеофильма Создаём Браузер mp4
7. Видеофильм Как работать с папками в Delphi.mp4

Электронные ресурсы:

1. Технология разработки прикладного программного обеспечения
<https://www.monographies.ru/ru/book/view?id=141>
2. Справочник Delphi <http://delphimaster.net/> Delphi Master Search Archive
3. Учебник Delphi <http://www.delphi-manual.ru/> Уроки Delphi начинающим с нуля
4. Delphi компоненты. Справочник <http://www.delphisources.ru/>
5. Delphi Форум программистов <http://www.programmersforum.ru/index.php>
6. Он-лайн справочник. Основы Delphi <http://www.delphibasics.ru/>

4.2. Перечень необходимого программного обеспечения

1. Lazarus – визуальная среда программирования (в свободном доступе);
2. PascalABC - визуальная среда программирования (в свободном доступе);
3. PascalABC.NET - визуальная среда программирования (в свободном доступе);
4. WEB-среда разработки PascalABC.NET. URL <http://wde.pascalabc.net/> (в свободном доступе);
5. Среда разработки ПО PortableDelphi. (в свободном доступе);
6. Разработчик инсталляторов InnoSetup. (в свободном доступе);
7. 7-zip архиватор; (лицензия на англ. <http://www.7-zip.org/license.txt>)
8. Adobe Acrobat Reader просмотрщик файлов ; (лицензия - <https://get.adobe.com/reader/?loc=ru&promoid=KLXME>)

9. Adobe Flash Player –графический редактор; (лицензия - <https://get.adobe.com/reader/?loc=ru&promoid=KLXME>)
10. Apache OpenOffice – офисный пакет; (лицензия - <http://www.openoffice.org/license.html>)
11. FreeCommander - проводник; (лицензия - <https://freecommander.com/ru/%d0%bb%d0%b8%d1%86%d0%b5%d0%bd%d0%b7%d0%b8%d1%8f/>)
12. Google Chrome - браузер;(лицензия - https://www.google.ru/chrome/browser/privacy/eula_text.html)
13. LibreOffice – офисный пакет (в свободном доступе);
14. Mozilla Firefox - браузер.(лицензия - <https://www.mozilla.org/en-US/MPL/2.0/>)
15. nanoCAD версия 5.1 локальная (лицензия - серийный номер: NC50B-45103)
16. ЭкзамL – Система компьютерного тестирования <http://Lkub.ru> Левин Л.Л. (в свободном доступе);
17. Программный комплекс "Универсальный тест 4.0.0.1" <http://www.timk.ru/> (в свободном доступе);

5. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ ОП.08 Теория алгоритмов

5.1. Основная литература

1. Баврин, И. И. Дискретная математика. Учебник и задачник [Электронный ресурс]: для СПО / И. И. Баврин. — М.: Издательство Юрайт, 2018. — 209 с. - URL: <https://www.biblio-online.ru/viewer/46422B2A-1497-4FFD-8A53-143190428418#page/1>

5.2 Дополнительная литература

1. Гринченков, Д.В. Математическая логика и теория алгоритмов для программистов: учебное пособие для вузов/Д.В. Гринченков, С.И. Потоцкий.- М.: Кнорус,2014.-206с. 8
2. Попов, А. М. Информатика и математика [Электронный ресурс]: учебник и практикум для СПО / А. М. Попов, В. Н. Сотников, Е. И. Нагаева; под ред. А. М. Попова. — 3-е изд., перераб. и доп. — М.: Издательство Юрайт, 2018. — 430 с. - URL: <https://www.biblio-online.ru/viewer/552AAA0D-1F2E-434C-8F8B-690CA6280464#page/1>
3. Судоплатов, С. В. Математическая логика и теория алгоритмов [Электронный ресурс]: учебник и практикум / С. В. Судоплатов, Е. В. Овчинникова. — 5-е изд., стер. — М.: Издательство Юрайт, 2018. — 255 с. - URL: <https://www.biblio-online.ru/viewer/71FA118B-CFD5-48BD-BC6F-073BDCA2806F#page/1>
4. Перельман, Я. И. Веселые задачи [Электронный ресурс] / Я. И. Перельман. — М.: Издательство Юрайт, 2018. — 292 с. - URL: <https://www.biblio-online.ru/viewer/AF291AE5-1CDF-4521-A4A0-4AC499D7D144#page/1>
5. Черпаков, И. В. Основы программирования [Электронный ресурс]: учебник и практикум для СПО / И. В. Черпаков. — М.: Издательство Юрайт, 2018. — 219 с. - URL: <https://www.biblio-online.ru/viewer/F79BE55A-C6F1-439D-9ED5-0D78A50B403F#page/1>
6. Татарников, О. В. Элементы линейной алгебры [Электронный ресурс]: учебник и практикум для СПО / О. В. Татарников, А. С. Чуйко, В. Г. Шершнева; под общ. ред. О. В. Татарникова. — М.: Издательство Юрайт, 2016. — 334 с. - URL: <https://www.biblio-online.ru/viewer/067047A5-3AC0-48DE-AD94-D99496C1BBBC#page/1>
7. Дехтярь, М.И. Основы дискретной математики [Электронный ресурс]/ М.И. Дехтярь. - 2-е изд., испр. - М.: Национальный Открытый Университет «ИНТУИТ», 2016. - 184 с. - URL: http://biblioclub.ru/index.php?page=book_view_red&book_id=428981

5.3 Периодические издания

1. Среднее и профессиональное образование
2. Компьютер Пресс
3. Открытые системы.- URL: <http://biblioclub.ru/index.php?page=journal&jid=436083>
4. Информатика в школе .- URL: <http://dlib.eastview.com/browse/publication/18988>
5. Программные продукты и системы.- URL: <http://dlib.eastview.com/browse/publication/64086>

6. Информатика и образование.- URL: <http://dlib.eastview.com/browse/publication/18946>
7. Системный администратор.- URL: <http://dlib.eastview.com/browse/publication/66751>
8. Computerword Россия.- URL: <http://dlib.eastview.com/browse/publication/64081>
9. Мир ПК.- URL: <http://dlib.eastview.com/browse/publication/64067>
10. Информационно-управляющие системы.- URL: <http://dlib.eastview.com/browse/publication/71235>
11. Журнал сетевых решений LAN.- URL: <http://dlib.eastview.com/browse/publication/64078>
12. Информатика и образование.- URL: <http://dlib.eastview.com/browse/publication/18946>
13. Windows IT Pro/ Re.- URL: <http://biblioclub.ru/index.php?page=journal&jid=138741>
14. Прикладная информатика.- URL: http://elibrary.ru/title_about.asp?id=25599

5.4 Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. ЭБС «Университетская библиотека ONLINE». – URL: www.biblioclub.ru
2. ЭБС издательства «Лань». – URL: <https://e.lanbook.com>
3. ЭБС «Юрайт». – URL: <http://www.biblio-online.ru/>
4. ЭБС «BOOK.ru» <https://www.book.ru>
5. ЭБС «ZNANIUM.COM» www.znanium.com
6. Электронный каталог Научной библиотеки КубГУ. – URL: <http://212.192.134.46/MegaPro/Catalog/Home/Index>
7. Электронная библиотека «Издательского дома «Гребенников» - URL: www.grebennikon.ru
8. Научная электронная библиотека (НЭБ) «eLibrary.ru». - URL: <http://www.elibrary.ru>
9. Базы данных компании «Ист Вью». - URL: <http://dlib.eastview.com>
10. Лекториум ТВ». - URL: <http://www.lektorium.tv/>
11. Национальная электронная библиотека «НЭБ». - URL: <http://нэб.рф/>
12. КиберЛенинка: научная электронная библиотека. – URL: <http://cyberleninka.ru/>
13. Единое окно доступа к образовательным ресурсам : федеральная ИС свободного доступа. – URL: <http://window.edu.ru>.
14. Справочно-правовая система «Консультант Плюс» - URL <http://www.consultant.ru>

6. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Методические указания по выполнению практических работ

Перечень практических заданий

1. Постановка задачи. Построение линейного алгоритма.
2. Среда программирования.
3. Линейное программирование.
4. Работа с функциями. Форматный вывод.
5. Программирование с использованием строковых процедур и функций.
6. Операции над строковыми данными.
7. Условный оператор.
8. Полный условный оператор.
9. Программирование с использованием неполных ветвлений.
10. Сложные разветвляющиеся структуры.
11. Вложенные условные операторы.
12. Вложенные ветвления.
13. Использование оператора варианта.
14. Циклические структуры. Цикл с параметром.
15. Встроенный отладчик.
16. Арифметический цикл. Обработка данных во время ввода.
17. Циклические структуры. Основные стандартные алгоритмы.
18. Сочетание условного оператора и цикла.
19. Цикл с предусловием. Вычисление членов ряда.
20. Циклические структуры. Обработка рядов чисел
21. Цикл с предусловием и его использование при решении задач.
22. Цикл с постусловием
23. Обработка строк с использованием оператора цикла.
24. Обработка числовых данных в строке с использованием оператора цикла.
25. Условный оператор в цикле и после него.
26. Сложные циклические структуры с дополнительным условием.
27. Действия над элементами массива.
28. Сортировка массивов. Оценка сложности алгоритмов

К каждой задаче разработать блок-схему алгоритма.

7. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ КОНТРОЛЯ УСПЕВАЕМОСТИ

7.1. Паспорт фонда оценочных средств

Код и наименование элемента **знаний**, контролируемые компетенции

31	основные модели алгоритмов;	ПК 1.1
32	методы построения алгоритмов;	ПК 1.2
33	методы вычисления сложности работы алгоритмов;	ПК 1.1, 1.2.

Код и наименование элемента **умений**, контролируемые компетенции

У1	разрабатывать алгоритмы для конкретных задач;	ПК 1.1
У2	определять сложность работы алгоритмов;	ПК 1.2

Для оценки вышеуказанных знаний и умений используются программы, разработанные и отлаженные обучающимся, которые представлены преподавателю. Дополнительный контроль проводится с помощью тематических тестов и собеседований.

7.2. Критерии оценки знаний

Контроль и оценка результатов освоения дисциплины осуществляется преподавателем в процессе проведения практических занятий, тестирования, а также выполнения обучающимися индивидуальных самостоятельных заданий и курсовых работ.

Тест. Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося. Тест оценивается по количеству правильных ответов, по времени выполнения, весу (сложности) заданий (не менее 50%).

Критерии оценки знаний студентов в целом по дисциплине:

«отлично» - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений;

«хорошо» - выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;

«удовлетворительно» - выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;

«неудовлетворительно» - выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

7.3. Оценочные средств для проведения текущей аттестации

В данном разделе приводятся образцы оценочных средств. Полный комплект оценочных средств приводится в Фонде оценочных средств.

Текущий контроль проводится в форме:

- тестирование по теоретическому и практическому материалу
- практическая работа – разработка и отладка программы
- защита выполненного задания,
- индивидуальный устный опрос.

Форма аттестации	Знания	Умения	Практический опыт (владение)	Личные качества студента	Примеры оценочных средств
Устный (письменный) опрос по темам	Контроль знаний по закладкам Delphi с встроенными компонентами	Оценка умения применять компоненты для разработки программ	Оценка навыков разработки законченных программ	Оценка способности оперативно и качественно отвечать на поставленные вопросы	Контрольные вопросы по темам прилагаются
Практические работы	Контроль знания основ программирования свойств и событий компонент Delphi	Оценка умения работать с графическими компонентами Delphi	Оценка навыков работы прикладными программными средствами	Оценка способности качественно решать задачи и аргументировать результаты	Темы работ прилагаются
Тестирование	Контроль знаний по определенным темам	Оценка умения различать конкретные понятия алгоритмов	Оценка навыков логического анализа и синтеза при сопоставлении конкретных понятий	Оценка способности оперативно и качественно отвечать на поставленные вопросы	Вопросы прилагаются

Балльно-рейтинговая система контроля

В основу фиксирования достижений учащихся положена компьютерная программа «**Рейтинг-автомат**», разработанная Левиным Л.Л., в которой имеются две главные связанные таблицы: 1. Список студентов по горизонтали и по вертикали Дата/Код КИМ/Сложность (вес) КИМ. 2. Список кодов тем занятий (КИМ) и содержание. При заполнении таблиц происходит автоматический пересчет баллов, набранных каждым студентом, ранжирование списка по набранным баллам, печать индивидуальных экзаменационных билетов с перечислением тем, пропущенных студентом. За посещение студентом занятия ему начисляется 1 балл. В качестве КИМ в значительной степени используются различные компьютерные тестовые системы, набранные баллы из которых заносятся в первую таблицу. Сложность КИМ назначает преподаватель.

Для текущего и итогового контроля применяется компьютерная программа «**ЭкзамL**», разработанная Левиным Л.Л. для компьютерного контроля и тестирования, работающая в режиме обучения и контроля. Характеристики тем, заложенных в программу, приведены ниже.

Программирование

1. Логические основы языков программирования. Паскаль 67 вопросов
2. Pascal Общие принципы языка. Основные понятия 70 вопросов

3. Pascal Ввод данных, оператор присваивания, вывод данных 37 вопросов
4. Pascal If Case: условные операторы, управление выбором. Логические выражения 36 вопросов
5. Pascal Типы данных. Простые операторы 27 вопросов
6. Pascal Стандартные функции, использование. Основные конструкции 74 вопросов
7. Pascal Операторы цикла и условные 47 вопросов
8. Pascal Циклы 40 вопросов
9. Pascal Строки, процедуры и функции обработки их 25 вопросов
10. Pascal Данные, операции и функции 43 вопроса
11. Pascal Процедуры и функции пользователя 25 вопросов
12. Pascal Процедуры функции и массивы 27 вопросов
13. Pascal Основные конструкции языка. Текущий контроль 99 вопросов
14. Массивы 27 вопросов
15. Массивы, записи, множества 46 вопросов
16. Массивы, перечислимые данные, множества, записи, файлы 48 вопросов
17. Файлы, записи 20 вопросов
18. Файлы типизированные, текстовые, нетипизированные 28 вопросов
19. Delphi Введение 30 вопросов
20. Delphi. Компоненты, свойства, события 81 вопрос
21. Delphi Основы и 11 программ 38 вопросов АГПА 2012 г.
22. Delphi Обязательный минимум 50 вопросов
23. Delphi Стандартные функции 30 вопросов
24. Delphi Объекты свойства события 41 вопрос
25. Delphi Проект диалоговые окна графика 36 вопросов
26. Delphi Теория ООП 40 вопросов
27. Delphi Базы данных, Язык SQL 27 вопросов
28. Delphi Техник-программист Квалификационные тесты 114 вопросов
29. Pascal, Delphi, Примеры 40 вопросов
30. Сумма языков программирования: Delphi, Pascal, SQL, HTML 119 вопросов

7.4. Оценочные средств для проведения промежуточной аттестации

Промежуточная аттестация

Форма аттестации	Знания	Умения	Иметь практический опыт	Личные качества студента	Примеры оценочных средств
Экзамен, тест по теме, комплексный тест по предметам	Контроль знания базовых положений в прикладном программировании	Оценка умения понимать специальную терминологию	Оценка навыков логического анализа задачи придумывать алгоритм.	Оценка способности грамотно и четко излагать материал	Вопросы: прилагаются
		Оценка умения разрабатывать алгоритмы и писать программы на языках высокого уровня.	Оценка навыков Переработки алгоритма в компьютерную программу	Оценка способности грамотно и четко излагать ход работы программы и аргументировать результаты	Задачи прилагаются

7.4.1. Примерные вопросы для проведения промежуточной аттестации

Раздел 1. Определение и свойства алгоритма. Место алгоритмов в повседневной жизни.

Примеры вопросов для устного опроса:

1. Приведите примеры алгоритмов из повседневной жизни, сложных, разных видов.
2. Что называется алгоритмом?
3. Какие свойства отличают алгоритмы от остальных конструкций?
4. Какие виды алгоритмов вы знаете?
5. Изобразите графически работу циклического алгоритма.
6. В чем заключается необходимость изучения алгоритмов?

Раздел 2. Линейные алгоритмы.

Примеры вопросов для устного опроса (в начале урока):

1. Что называется алгоритмом?
2. Какие виды алгоритмов вы знаете?
3. Какие свойства отличают алгоритмы от остальных конструкций?

Примеры вопросов для устного опроса (в конце урока):

1. Что называется линейным алгоритмом?
2. Приведите примеры линейных алгоритмов.
3. Изобразите графически работу линейного алгоритма.

Раздел 3. Ветвления.

Примеры вопросов для устного опроса (в начале урока):

1. Что называется алгоритмом?
2. Какие виды алгоритмов вы знаете?
3. Какие свойства отличают алгоритмы от остальных конструкций?
4. В чем заключаются особенности работы линейного алгоритма?

Примеры вопросов для устного опроса (в конце урока):

1. Что называется условным алгоритмом?
2. Приведите примеры условных алгоритмов.
3. Изобразите графически работу условного алгоритма.
4. В чем отличия условных алгоритмов от линейных?

Раздел 4. Циклические алгоритмы. Циклы в сочетании с ветвлениями.

Примеры вопросов для устного опроса (в начале урока):

1. Что называется алгоритмом?
2. Какие виды алгоритмов вы знаете?
3. Какие свойства отличают алгоритмы от остальных конструкций?
4. В чем отличия условных алгоритмов от линейных?

Примеры вопросов для устного опроса (в конце урока):

1. Что называется циклическим алгоритмом?
2. Приведите примеры циклических алгоритмов.
3. Изобразите графически работу цикла в сочетании с ветвлением.
4. Проведите сравнительный анализ всех видов алгоритмов.
5. Какие алгоритмы, на ваш взгляд, лучше.

7.4.2. Примерные экзаменационные задачи

Вопросы к экзамену

Пример итогового теста по теории алгоритмов:

1. Важнейший принцип структурного программирования:

- любой алгоритм имеет дискретную структуру;
- алгоритм любой сложности можно построить с помощью следующих базовых структур: линейной, ветвящейся, циклической;
- современный компьютер — это единство аппаратных средств и программного обеспечения;
- сущность формализации решаемой задачи заключается в составлении алгоритма;
- в качестве обязательного этапа создания программы выступает ее тестирование и отладка.

2. Алгоритм — это:

3. Алгоритм называется линейным, если:

- при его выполнении многократно повторяются одни и те же действия;
- последовательность выполнения его команд зависит от истинности тех или иных условий;
- его команды выполняются друг за другом независимо от каких-либо условий;
- он представим в табличной форме;
- в нем используются исключительно операторы присваивания.

4. Алгоритм включает в себя ветвление, если:

- при его выполнении многократно повторяются одни и те же действия;
 - последовательность выполнения его команд зависит от истинности тех или иных условий;
 - его команды выполняются друг за другом независимо от каких-либо условий;
 - он представим в табличной форме;
 - он составлен так, что в каждой программной строке записан только один оператор.
5. Алгоритм называется циклическим, если: при его выполнении многократно
- повторяются одни и те же действия;
 - последовательность выполнения его команд зависит от истинности тех или иных условий;
 - его команды выполняются друг за другом независимо от каких-либо условий;
- представим в табличной форме;
- он составлен так, что в каждой программной строке записан только один оператор.

6. Что можно назвать алгоритмом?

- схему движения автобусов;
- инструкцию по использованию микроволновой печи;
- расписание уроков;
- схему электрической цепи;
- чертеж дома;
- инструкцию по использованию акварельной краски.

Контрольная работа по дисциплине "**Теория алгоритмов**" является итоговой формой контроля знаний студентов, самостоятельной работой студентов на завершающем этапе изучения данной дисциплины.

Цель контрольной работы - закрепление теоретических знаний по курсу, получение практических навыков составления бизнес-плана.

Контрольная работа представляет собой разработанный студентом алгоритм и написание программы для выбранной задачи.

Объем контрольной работы - 5-10 стр.; время, отводимое на подготовку – от 2 недель до месяца.

Работа должна состоять из следующих частей:

- Постановка задачи.
- Описание алгоритма решения.
- Блок-схема алгоритмы.
- Текст программы на языке Паскаль.
- Описание встретившихся проблем и ошибок.
- Демонстрация на компьютере работающей программы.
- Описание возможных усовершенствований программы.
- Приложения (если необходимо).

В приложениях помещаются по необходимости иллюстрированные материалы, имеющие вспомогательное значение (схемы, диаграммы и т.п.).

Выполненная контрольная работа проверяется преподавателем, затем защищается студентом и оценивается в форме зачета и экзамена.

8. ДОПОЛНИТЕЛЬНОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Приложение 1. Краткий конспект лекционных занятий

Задачи:

Формирование базовых знаний по алгоритмизации, рациональные методах разработки алгоритмов;

Изучение базовых алгоритмических структур;

Изучение объектов алгоритмов;

Знакомство с этапами построения математических моделей;

Алгоритмы

Этапы решения задач на ЭВМ



Рассмотрим, из чего складывается процесс решения задачи на ЭВМ, и какие необходимо пройти для этого этапы.

I этап – постановка задачи. На этом этапе должна быть определена предметная область задачи, определены цели задачи, необходимый объем исходной информации, проведено описание каждого исходного данного. Предложен общий подход к решению.

II этап – математическое описание задачи. Цель этого этапа – создать такую математическую модель, которая может быть реализована на компьютере, выбрать оптимальный метод решения.

Под математической моделью будем понимать описание объекта или процесса математическими формулами, связывающими их количественные параметры. При описании математических моделей используются различные системы обозначений.

III этап – алгоритмизация задачи. Главная особенность всех вычислений машины состоит в том, что в основе ее работы лежит программный принцип управления. Это означает, что для решения, как самой простой, так и самой сложной задачи пользователю необходимо использовать перечень инструкций или команд, следуя которым шаг за шагом ЭВМ выдаст необходимый результат.

Таким образом, для того, чтобы решать задачу на ЭВМ, ее необходимо сначала, алгоритмизировать. Именно алгоритмический принцип и лежит в основе работы всех ЭВМ.

На основе математической модели (имеющихся расчетных формул) разрабатывается алгоритм решения. Чаще всего алгоритм разрабатывается на основе блок – схемы с

четко определенной последовательностью действий.

IV этап – программирование. Программа – это представление алгоритма с помощью специальных символов, воспринимаемых компьютером. Каждому блоку алгоритма соответствует определенная последовательность операторов. Программа обеспечивает возможность реализации алгоритма и поставленной задачи. При составлении программы возможно уточнение и изменение алгоритма.

V этап – ввод программы в ЭВМ. На этом этапе программу необходимо набрать в интегрированной среде программирования и сохранить на диске.

VI этап – разработка контрольного примера. Для того, чтобы убедиться в правильности составленной программы, необходимо разработать тестовую задачу, проверяющую все ветви алгоритма. Это совокупность таких исходных данных, на основании которых заранее определяются выходные данные.

VII этап – отладка программы. Программа и исходные данные контрольного примера обрабатываются на компьютере и, если контрольный пример работает неправильно, то необходимо найти ошибки, допущенные в программе и вновь проверить ее на контрольном примере.

VIII этап – получение и анализ результатов. После устранения всех ошибок, выявленных тестовой задачей, можно перейти к получению результатов поставленной задачи. Полученные в результате решения результаты необходимо проанализировать.

Алгоритмы

Слово «алгоритм» знакомо многим читателям. Его применяют широко и не только в области вычислительной техники и программирования. В повседневной жизни приходится решать различные задачи. Задачи могут быть бытовые (прибраться в комнате, приготовить обед, перейти улицу и т.п.), учебные (решить систему уравнений, построить график) или профессиональные (изготовить деталь на станке, рассчитать заработную плату). Все задачи можно разделить по различным признакам: математические, научные, инженерные, экономические. Одни задачи решаются легко, другие нет. Решить задачу означает получить результат, отвечающий целям данной задачи. Для каждой задачи должно быть известно, что считать результатом. В процессе решения задачи выполняются определенные действия над исходными данными. Совокупность этих действий может быть задана настолько подробно, что ее исполнение становится чисто механическим процессом. Полученная инструкция может использоваться для решения однотипных задач. Такую инструкцию называют алгоритмом.

Понятие алгоритма в информатике является фундаментальным, т.е. таким, которое не определяется через другие, еще более простые понятия.

Содержание понятия алгоритма можно определить следующим образом:

Алгоритм – точное предписание, задающее алгоритмический процесс, начинающийся с произвольного исходного данного и направленный на получение результата, определенного этим исходным данным.

Алгоритм позволяет чисто механически решать любую конкретную задачу из некоторого класса однотипных задач.

Алгоритмический процесс – процесс последовательного преобразования объектов дискретными шагами.

Каждый алгоритм характеризуют независимые параметры:

- совокупность возможных исходных данных (исходные данные могут изменяться в определенных пределах);

- совокупность возможных промежуточных результатов (на каждом шаге должно быть известно, что считать результатом);
- совокупность результатов;
- правило начала;
- правило непосредственной переработки;
- правило окончания;
- правило извлечения результата.

Свойства алгоритма

Понятность. Каждый алгоритм создается для конкретного исполнителя. Чтобы исполнитель мог решить поставленную перед ним задачу, используя алгоритм, он должен уметь выполнить каждое его указание, понимать суть управления. Под «понятностью» алгоритмов понимают указания, понятные исполнителю.

Детерминированность (однозначность). Будучи понятным, алгоритм не должен все же содержать предписаний, смысл которых может восприниматься неоднозначно. В алгоритмах недопустимы такие ситуации, когда после выполнения очередного предписания алгоритма исполнителю неясно, какое из них должно выполняться на следующем шаге.

Под однозначностью алгоритмов понимается единственность толкования правил выполнения действий и порядка их выполнения.

Дискретность. Под дискретностью понимают возможность разбиения алгоритма на отдельные элементарные действия, выполнение которых человеком или машиной не вызывает сомнения.

Массовость. Очень важно, чтобы составленный алгоритм обеспечивал решение не одной частной задачи, а мог выполнять решение широкого класса задач данного типа. Под массовостью алгоритмов подразумевается возможность их применения для решения целого класса конкретных задач, отвечающих общей постановке задачи.

Конечность. Выполнение действий, заданных алгоритмом, состоит из конечного числа шагов.

Результативность. Выполнение алгоритма должно завершаться получением определенных результатов. Должно быть известно какой результат должен быть получен через конечное число шагов.

Способы представления алгоритмов

Существует несколько способов представления или записи алгоритмов, отличающихся наглядностью, компактностью, формализацией. Алгоритм может формироваться в виде схемы, текста или программы.

Мы будем рассматривать представление алгоритмов в виде структурных схем – блок-схем, когда отдельные его действия (этапы) изображаются при помощи различных геометрических фигур (блоков), а связи между этапами указываются при помощи стрелок, соединяющих эти фигуры. В блок-схеме отображаются шаги, которые должны выполняться компьютером. Каждому блоку соответствует свой оператор на языке программирования. По составленной блок-схеме пишется программа на языке программирования и реализуется в дальнейшем на компьютере. Существуют также специальные пакеты программ, позволяющие получать результаты непосредственно только по блок-схеме (не используя язык программирования).

Каждый блок имеет свою форму, назначение и определенные размеры. Блоки соединяются соединительными стрелками. Внутри блоков записываются операторы.

Условные графические обозначения блоков

Объекты алгоритмов

У каждой конкретной задачи есть свои реальные объекты (данные, которые подлежат обработке). Каждый объект имеет свои характеристики, свойства или атрибуты. В процессе решения могут возникнуть и вспомогательные объекты. К объектам относятся константы, переменные, файлы, массивы. Каждый объект имеет свой тип (например, число может быть целого типа или вещественного).

Под **константой** будем понимать объект алгоритма, который имеет определенный фиксированный тип и фиксированное, неизменяемое значение.

Под **переменной** будем понимать объект, который имеет определенный фиксированный тип, но значение переменной может меняться на протяжении вычислительного процесса.

Под **массивом** будем понимать упорядоченную совокупность данных одного типа. Доступ к каждому элементу массива осуществляется по его порядковому номеру (индексу).

Каждый объект в алгоритме должен иметь свое имя – **идентификатор**.

Для обработки объектов алгоритма используют операторы, которые представляют закодированную форму инструкции. Различают простые и составные операторы. К простым относятся операторы присваивания, ввода-вывода, перехода. Группу составных (сложных) операторов представляют операторы условия, циклические, присоединения. Для пояснения алгоритмов используют комментарии.

Базовые алгоритмические конструкции

К базовым конструкциям алгоритмов относятся три основные структуры:

- линейная;
- разветвляющаяся;
- циклическая.

Алгоритм любой сложности состоит из комбинации этих базовых структур.

Линейные алгоритмы

Алгоритм называется линейным, если все действия в нем выполняются последовательно одно за другим.

Типовая блок-схема линейного алгоритма. Линейный алгоритм может содержать блоки ввода и вывода данных, блоки вычислений – действий.

Блоки ввода-вывода могут располагаться в любом месте алгоритма. Все действия данного алгоритма выполняются последовательно одно за другим. К линейным алгоритмам относится большинство расчетных инженерных задач.

Разветвляющиеся алгоритмы

Алгоритм называется разветвляющимся, если выполнение действий в алгоритме происходит после выполнения поставленного условия по одной или другой ветви.

Если условие выполняется, то дальнейшее вычисления алгоритма происходят по ветви «Да», если не выполняется, по ветви «Нет». В каждом варианте может быть не одно действие, а несколько. После проверки одного условия, может стоять новая проверка условия, которая тоже будет иметь два выхода.

Циклические алгоритмы

Циклическая структура обеспечивает повторяющуюся реализацию содержащегося в ней функционального узла. Каждый цикл имеет следующие характеристики:

- параметр цикла;
- тело цикла (повторяющиеся действия);
- приращение параметра цикла (изменением его на определенный шаг);
- условие выхода из цикла.

Если заранее известно число повторений в цикле, то такой цикл называют **арифметическим** и его реализуют при помощи типовой схемы «цикл с параметром». Если число повторений в цикле неизвестно, то цикл называют **итерационным**. Итерационные циклы можно организовывать двумя структурами:

- с **постусловием** ;
- с **предусловием**.

Цикл с постусловием выполняется пока условие «ложно». Этот цикл выполнится всегда хотя бы один раз, так как первая проверка выхода из цикла происходит после выполнения «тела цикла». Цикл с предусловием выполняется пока условие «истинно» и может не выполниться ни одного раза, если при первой проверке условие выхода выполняется.

Внутри цикла может содержаться функциональный узел на проверку условия . Допускается неограниченное соединение базовых структур, их вложение друг в друга, например, цикл с условием, вложенные циклы. Такое соединение позволяет проектировать сложные алгоритмы. Каждой структуре соответствуют свои операторы языка программирования: линейной – операторы присваивания, разветвляющейся – условные операторы, циклической – операторы цикла.

Этапы разработки алгоритма

Разработка любого алгоритма состоит из многих взаимосвязанных этапов. На каждом этапе решаются свои конкретные проблемы, которые в конечном итоге определяют общий результат поставленной задачи.

На первом этапе анализа необходимо понять задачу и определить все составляющие, необходимые для разработки алгоритма. На этом этапе выясняются все исходные данные и требуемые результаты, определяются типы объектов.

Пошаговая детализация задачи позволяет свести трудную задачу к последовательности более простых задач.

Макет исходных данных представляет собой форму исходных данных с записью конкретных значений. В качестве выходного документа служит форма макета вывода или печати результатов, в которой должны быть предусмотрены все поясняющие тексты на выходе.

Таблица идентификаторов может быть представлена в произвольной форме, но она должна содержать все объекты задачи с указанием имен и типов.

Таблица не закрывается и в процессе решения задачи всегда может быть дополнена. При разработке программы таблица используется для записи раздела описаний данных.

Пример 1. Рассмотрим задачу вычисления площади треугольника по трем заданным сторонам.

Постановка задачи. Даны стороны треугольника. Вычислить его площадь.

Математическая модель.

Входные данные: стороны треугольника.

Выходные данные: площадь треугольника.

Дополнительные параметры: полупериметр.

Расчетные формулы:

Данный алгоритм повторяет линейную структуру. Но всегда ли такой алгоритм будет работать? Нет, так как при вычислении площади мы имеем дело с вычислением квадратного корня, подкоренное выражение которого может оказаться отрицательным. Это говорит о том, что не из любых трех отрезков можно построить

треугольник. На уточняющем этапе разработки алгоритма необходимо учесть случай существования треугольника и в алгоритме поставить проверку на существование треугольника. Треугольник будет существовать, если сумма двух сторон больше третьей стороны. При такой постановке задачи алгоритм уже будет реализовывать разветвляющуюся структуру (рис. 9). Для исключения варианта некорректного ввода данных, в случае невыполнения условия, сделаем переход на блок ввода данных .
Пример 2. Определить, принадлежит ли точка с координатами (x,y) окружности с центром в начале координат и радиусом r .

Математическая модель.

Входные данные: радиус окружности;
координаты точки.

Выходные данные: сообщение о принадлежности точки окружности.

Дополнительные параметры: расстояние от точки до центра окружности.

Для определения расчетных формул составим геометрическую модель задачи

Точка будет принадлежать окружности при условии, что расстояние от нее до центра окружности будет меньше или равно радиусу окружности.

Расчетные формулы:

Расстояние от точки до центра окружности

Составим таблицу идентификаторов

Алгоритм задачи. Структура алгоритма повторяет типовую разветвляющуюся структуру.

Пример 3. Теперь усложним задачу. Пусть задано n точек. Определить количество точек, принадлежащих окружности. Для решения этой задачи требуется ввести новые объекты. Занесем их в таблицу идентификаторов:

Составим алгоритмы двумя способами: в первом случае используем цикл с параметром, а во втором цикл с постусловием. Циклический блок объединяет следующие действия:

- задание начального значения параметра цикла (переменной i присваивается начальное значение равно 1);
- приращение переменной на 1 ($i=i+1$);
- проверка условия выхода из цикла (как только i достигает значения n , цикл завершается.)

Во втором алгоритме все эти действия показаны в отдельных блоках.

В цикле осуществляется ввод координат точек и подсчет количества точек, принадлежащих окружности.

Приложение 2. Иллюстративный материал

Преподавание курса сопровождается интерактивным просмотром учебных видеofilьмов. Часть оглавления видеотеки представлена ниже. Общий объем 7,5 Гб.

Видеотека программирования

[Delphi]	<Папка> 30.05.2017 17:43 —
[Office]	<Папка> 30.05.2017 12:52 —
[Pascal]	<Папка> 30.05.2017 12:18 —
[Архивация]	<Папка> 30.05.2017 12:23 —
[Архитектура компьютерных систем]	<Папка> 30.05.2017 16:20 —
[Базы данных]	<Папка> 30.05.2017 12:31 —
[Инфокоммуникационные системы и сети]	<Папка> 30.05.2017 16:20 —
[Логика Подготовка к ЕГЭ]	<Папка> 30.05.2017 16:20 —
[ОС]	<Папка> 30.05-2017 16:20

[Основы программирования]	<Папка> 30.05.2017 16:20
[Сайтостроение]	<Папка> 30.05.2017 16:20 —
[Файлы]	<Папка> 30.05.2017 16:20
Объекты операционной системы.mp4	80 929 499 07.09.2015 18:44 -a-
Основы программирования Виды алгоритмов.mp4	40 074 121 03.04.2015 10:59 -a-
Ofile_system.avi	200 247 388 11.02.2015 18:40 -a-
02. Основной навык-умение хранить информацию.mp4	54 574 403 19.12.2013 18:22 -a
01. Знакомство с ОС Windows.mp4	46 463 046 16 12.2013 20:57 -a-
01. Количество информации.mp4	34 449 975 08.02.2012 14:23 -э-
446 033 Кбайт в 6 файлах/файле	

Видеотека программирования Delphi

[0 Закладки и компоненты]	<Папка> 30.05.2017 11:33
[Menu]	<Папка>30.05.2017 11:38 —
[Player1]	<Папка> 30.05.2017 11:45
[Браузер]	<Папка>30.05.2017 11:47 —
[Графика]	<Папка>30.05.2017 12:36 —
[Игра Пин-Понг]	<Папка>30.05.2017 11:37 —
[Массив]	<Папка>30.05.2017 11:33 —
[Обработчик событий примеры]	<Папка>30.05.2017 17:33 -a-
[ООП]	<Папка>30.05.2017 17:33 —
[Профессия]	<Папка>30.05.2017 12:04 —
[Редактор текстовый]	<Папка> 30.05.2017 11:49
[Сортировка]	<Папка>30.05.2017 17:32 —
[Условный оператор]	<Папка>30.05.2017 11:34 —
[Цикл]	<Папка>30.05.2017 11:34 —

Ограничить количество знаков после запятой.mp4	38097 459 31.03.2016 11:11 -a-
Работать с папками в Delphi.mp4	71477 658 31.03.2016 11:04 -a-
Создать форму любой формы в Delphi.mp4	36 950 269 31.03.2016 10:35 -a-
Создать связанные combobox.mp4	52 184 309 31.03.2016 10:26 -a-
Создать многооконное приложение.mp4	20 955 495 31.03.2016 10:11 -a-
Объекты операционной системы.mp4	80 929 499 07.09.2015 18:44 -a-
02. Основной навык-умение хранить информацию.mp4	54 574 403 19.12.2013 18:22 -a-
01. Знакомство с ОС Windows.mp4	46 463 046 16.12.2013 20:57 -a-
392 218 Кбайт в 8 файлах/файле	

Приложение 3. Презентации.

Папка F:\Uni\2019_20\Теория алгоритмов\РП\2019\Презентации На 30.08.2019

№	Имя файла	Байт	Дата
1	1 Линейный алгоритм..ppt	1292288	14.09.2014
2	3 Алгоритмы.pps	345088	04.11.2013
3	test-algoritm.ppt	456704	11.02.2015
4	Алгоритмизация и языки программирования.ppt	1507840	19.12.2015
5	Логика примеры.pptx	73889	15.02.2015
6	Машина Тьюринга.ppt	2135040	10.10.2015
7	Основы алгебры высказываний. Логические операции.pptx	2490393	10.06.2014
8	Основы логики.ppt	513024	05.11.2012
9	презентация к игре.pptx	113611	16.01.2012

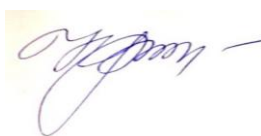
ЛИСТ

изменений рабочей учебной программы по дисциплине
ОП.08 Теория алгоритмов

Дополнения и изменения, вносимые в рабочую программу дисциплины

Основания внесения дополнений и изменений	Раздел РПД, в который вносятся изменения	Содержание вносимых дополнений, изменений
Предложение работодателя		
Предложение составителя программы		
Другие основания		

Составитель: преподаватель



Т.П. Кривошеенко

подпись

Утвержден на заседании предметной (цикловой) комиссии профессиональных дисциплин специальности Программирование в компьютерных системах протокол № 10 от «27» мая 2020 г.

Председатель предметной (цикловой) комиссии профессиональных дисциплин специальности Программирование в компьютерных системах



Л.А. Благова

« 27 » мая 2020 г.

Зам. директора по УР филиала



Т. А. Резуненко

«27» мая 2020 г.

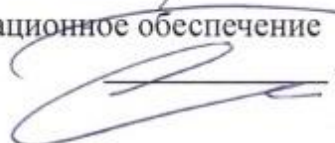
Заведующая сектором библиотеки



Л. Г. Соколова

«27» мая 2020 г.

Инженер-электроник (программно-информационное обеспечение образовательной программы)



А. В. Сметанин

«27» мая 2020 г.

РЕЦЕНЗИЯ

**на рабочую программу по учебной дисциплине
ОП 0.8 «ТЕОРИЯ АЛГОРИТМОВ»** по специальности 09.02.03 «Программирование в компьютерных системах» СПО, разработанную, преподавателем **Кривошеенко Татьяной Петровной**.

Структура рабочей программы соответствует существующему уровню развития технологий алгоритмизации программирования, она включает в себя описание алгоритмических средств, современного программного обеспечения и соответствующих технических средств.

Рецензируемая программа предусматривает освоение профессиональных компетенций: (ПК 1.1. ПК 1.2.) и видов деятельности, в соответствии с ФГОС № 804 от «28» июля 2014 г.

Программа содержит: основные понятия и принципы алгоритмизации задач, конструкции и стандартные приёмы алгоритмизации, основы объектно-ориентированного и визуального программирования, необходимый инструментарий и технологии. Программа имеет достаточную степень полноты и законченности для изучения предмета в условиях СПО.

Дисциплина «Теория алгоритмов» предусматривает приобретение навыков в разработке алгоритмов решения базовых задач на языках Паскаль и Дельфи с применением соответствующих сред.

В программе отражены основные теоретические и практические свойства алгоритмов, что даёт возможность получить необходимые знания о содержании и сущности алгоритмизации, декомпозиции задач, о современном состоянии и тенденциях развития программного обеспечения, компьютерной техники.

Структура программы соответствует современным требованиям. Содержание проработано с достаточной степенью подробности и законченности. Пояснительная записка раскрывает цели программы, включает в себя краткую характеристику её предметного содержания.

В программе имеется необходимый список учебных материалов.

Применение в учебном процессе балльно- рейтинговой системы оценивания знаний и программ тестирования также положительно влияет на усвоение учащимися материала дисциплины.

Программа учебной дисциплины продумана и ориентирована на подготовку обучающихся к использованию полученных навыков в своей профессиональной деятельности.

Таким образом, рабочая программа содержит все необходимые элементы рекомендуемой структуры, обладает достаточной полнотой и законченностью, является ценным практическим документом при преподавании дисциплины «Теория алгоритмов».

Рецензент:

Директор ООО «Современные
информационные технологии»

А.В.Сметанин

РЕЦЕНЗИЯ

на рабочую программу по учебной дисциплине **ОП 0.8 «ТЕОРИЯ АЛГОРИТМОВ»** по специальности 09.02.03 «Программирование в компьютерных системах» СПО, разработанную, преподавателем **Кривошеенко Татьяной Петровной**.

По структуре рабочая программа соответствует практическому уровню развития технологий алгоритмизации программирования, она включает в себя описание алгоритмических средств, современного программного обеспечения и соответствующих технических средств.

Рецензируемая программа предусматривает освоение профессиональных компетенций: (ПК 1.1. ПК 1.2.) и видов деятельности, в соответствии с ФГОС № 804 от «28» июля 2014 г.

В программе содержатся: основные понятия и принципы алгоритмизации задач, конструкции и стандартные приёмы алгоритмизации, основы объектно-ориентированного и визуального программирования, необходимый инструментарий и технологии. Программа имеет достаточную степень полноты и законченности для изучения предмета в условиях СПО.

«Теория алгоритмов» как дисциплина предусматривает приобретение навыков в разработке алгоритмов решения базовых задач на языках Паскаль и Дельфи с применением соответствующих сред.

В рецензируемой программе отражены основные теоретические и практические свойства алгоритмов, что даёт возможность получить необходимые знания о содержании и сущности алгоритмизации, декомпозиции задач, о современном состоянии и тенденциях развития программного обеспечения.

По структуре программа соответствует современным требованиям. Содержание проработано с достаточной степенью подробности и законченности. Пояснительная записка раскрывает цели программы, включает в себя краткую характеристику её предметного содержания.

В программе приводится необходимый список учебных материалов.

Следует отметить: применение в учебном процессе балльно- рейтинговой системы оценивания знаний и программ тестирования также положительно влияет на усвоение учащимися материала дисциплины.

Программа учебной дисциплины достаточно продумана и ориентирована на подготовку обучающихся к использованию полученных навыков алгоритмизации в своей профессиональной деятельности.

Следовательно, рабочая программа содержит все необходимые элементы рекомендуемой структуры, обладает достаточной полнотой и законченностью, является ценным практическим документом при преподавании дисциплины «Теория алгоритмов».

Рецензент:

Программист ГБУЗ

"Геленджикский психоневрологический диспансер"
министерства здравоохранения Краснодарского края

Е.В.Мельников