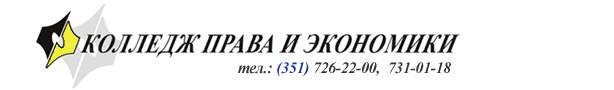
Профессиональное образовательное учреждение

среднего профессионального образования



**РАБОЧАЯ ПРОГРАММА**

**УЧЕБНОЙ ДИСЦИПЛИНЫ**

**Теория Алгоритмов**

по специальности среднего профессионального образования

**09.02.03 «Программирование в компьютерных системах»**

(базовый уровень)

Челябинск

2015 г.

Одобрена:

Цикловой (методической) комиссией

Утверждена:

Директором ПОУ СПО «Колледж права и экономики»

Молодчиком А.В.

Программа учебной дисциплины «Теория алгоритмов» разработана на основе Федеральных государственных образовательных стандартов (далее – ФГОС) по специальности среднего профессионального образования (далее СПО) 09.02.03 Программирование в компьютерных системах – базовый уровень, укрупненная группа 230000 Информатика и вычислительная техника

Организация-разработчик: Профессиональное образовательное учреждение

среднего профессионального образования

«Колледж права и экономики»

Разработчики: Желудок И.А.. – преподаватель

**СОДЕРЖАНИЕ**

|  |  |
| --- | --- |
|  | стр. |
| ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ | 4 |
| СТРУКТУРА и содержание УЧЕБНОЙ ДИСЦИПЛИНЫ | 5 |
| условия реализации учебной дисциплины | 11 |
| Контроль и оценка результатов Освоения учебной дисциплины | 12 |

**1. паспорт ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ**

**Теория Алгоритмов**

**1.1. Область применения программы**

Рабочая программа учебной дисциплины является частью основной профессиональной образовательной программы в соответствии с Федеральным государственным образовательным стандартом по специальности СПО 09.02.03 «Программирование в компьютерных системах» укрупненной группы 230100 «Информатика и вычислительная техника».

Рабочая программа учебной дисциплины может быть использована в дополнительном профессиональном образовании и профессиональной подготовке специалистов в области специальности 230115 «Программирование в компьютерных системах». Квалификация «Техник-программист». Опыт работы не требуется.

**1.2. Место учебной дисциплины в структуре основной профессиональной образовательной программы:**

Теория Алгоритмов входит в цикл общепрофессиональных дисциплин в базовой части. Для ее успешного изучения не требуются знания и умения, выходящие за рамки общего об­разования.

Знание архитектуры компьютерных систем может существенно помочь в научно-исследовательской работе.

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

Целями освоения дисциплины «Теория Алгоритмов» являются: формирование логической и концептуальной культуры студента, освоение общих содержа­тельных системных понятий доказательства и вычисления, их формализации и основ­ных свойств; начальная фундаментальная подготовка в области теории алгоритмов, включая теорию сложности, овладение их современным программным аппаратом для дальнейшего использования в приложениях.

В результате изучения обязательной части цикла обучающийся должен:

**уметь:**

* разрабатывать алгоритмы для конкретных задач;
* определять сложность работы алгоритмов;

**знать:**

* основные модели алгоритмов;
* методы построения алгоритмов;
* методы вычисления сложности работы алгоритмов.

**1.4. Рекомендуемое количество часов на освоение программы учебной дисциплины:**

максимальной учебной нагрузки обучающегося 96 часов, в том числе:

обязательной аудиторной учебной нагрузки обучающегося 64часов;

самостоятельной работы обучающегося 32 часа.

**2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ**

**2.1. Объем учебной дисциплины и виды учебной работы**

|  |  |
| --- | --- |
| **Вид учебной работы** | ***Количество часов*** |
| **Максимальная учебная нагрузка (всего)** | ***96*** |
| **Обязательная аудиторная учебная нагрузка (всего)** | ***64*** |
| в том числе: |  |
| практические занятия | *16* |
| **Самостоятельная работа обучающегося (всего)** | ***32*** |
| в том числе: |  |
| Подготовка рефератов.  Подготовка презентационных материалов.  Выполнение домашней работы: решение задач; составление таблиц; | 8  12  12 |
| ***Итоговая аттестация*** *в форме экзамена* | |

**2.2. Тематический план и содержание учебной дисциплины**

**«Архитектура компьютерных систем»**

|  |  |  |  |
| --- | --- | --- | --- |
| **Наименование разделов и тем** | **Содержание учебного материала, лабораторные и практические работы, самостоятельная работа обучающихся** | **Объем часов** | **Уровень освоения** |
| *1* | *2* | *3* | *4* |
| **Раздел 1. Основы алгоритмизации** |  | **48** |  |
| Тема 1.1. Алгоритмы и величины | Введение. Понятие алгоритма. Свойства, способы записи алгоритмов. Базовые алгоритмические структуры. | **8** | 1 |
| Этапы решения задач на ЭВМ. Данные и величины. | 1 |
| Тема 1.2. Линейные вычислительные алгоритмы | Понятие линейного алгоритма. Примеры линейных алгоритмов. | **10** | 2 |
| Алгоритмические команды присваивания, ввода, вывода данных. Свойства команды присваивания. | 2 |
| **Практические работы** | 2 |
| Анализ линейных алгоритмов математических задач. | 2 |  |
| Составление линейных алгоритмов математических задач. |
| Тема 1.3. Ветвление в вычислительных алгоритмах | Свойство универсальности алгоритма. Общий вид команды ветвления на алгоритмическом языке и в блок-схеме. | **8** | 1 |
| Структурная команда ветвления. Структура вложенных ветвлений. | 2 |
| **Практические работы** | 3 |
| Составление алгоритмов, содержащих ветвление. | 2 |  |
| Составление алгоритмов с вложенным ветвлением. |
| Тема 1.4. Циклы в вычислительных алгоритмах | Алгоритм циклической структуры. | **8** | 1 |
| Понятие итерации. Тело цикла. Шаг цикла. | 1 |
| Команда цикла с предусловием. Использование цикла с предусловием в задачах. | 2 |
| Команда цикла с постусловием. Использование цикла с постусловием в задачах. | 2 |
| **Практические работы** | 3 |
| Анализ и составление алгоритмов с использованием цикла с предусловием. | 2 |  |
| Анализ и составление алгоритмов с использованием цикла с постусловием. |
| **Контрольная работа** по теме «Базовые алгоритмические структуры» | 2 |
| Тема 1.5. Вспомогательные алгоритмы и процедуры | Понятия основного и вспомогательного алгоритмов. Понятие процедуры. | **8** | 1 |
| Обращение к вспомогательному алгоритму и процедуре из основного алгоритма. | 2 |
| Фактические и формальные параметры. | 1 |
| Правила соответствия между фактическими и формальными параметрами. | 2 |
| **Практические работы** | 3 |
| Анализ и составление алгоритмов с вспомогательными алгоритмами. | 2 |  |
| Анализ и составление алгоритмов с процедурами. |
|  | **Самостоятельная работа:** выполнение домашних заданий по разделу 1 | 16 |
| **Раздел 2. Методы построения алгоритмов** |  | **48** |
| Тема 2.1. Основные понятия структурного программирования | Этапы изготовления программного продукта. | **8** | 1 |
| Теорема, лежащая в основе структурного программирования. | 1 |
| Сложный алгоритм. Способы соединения базовых алгоритмических структур. Глубина вложенности структур. | 1 |
| Стандарты изображения блок-схем алгоритмов. Наглядность построения программ. | 1 |
| Декомпозиция задачи. Способы построения алгоритма: метод последовательной детализации и сборочный метод. | 2 |
| Отладка и тестирование алгоритма. | 2 |
| **Практические работы** | 2 |
| Построение и чтение блок-схем сложных алгоритмов. | **2** |  |
| Применение методов отладки, разработка системы тестов для алгоритма. |
| Тема 2.2. Рекурсивные методы построения алгоритмов | Понятие рекурсии. Рекурсивные вспомогательные алгоритмы. | **12** | 1 |
| Задача «Ханойская башня». | 1 |
| **Практические работы** | 2 |
| Использование рекурсивных алгоритмов в вычислительных задачах. | 2 |  |
| Составление алгоритмов с рекурсией. |
| Тема 2.3. Методы перебора в задачах поиска | Проблема поиска информации. Критерий поиска. | **8** | 1 |
| Методы полного перебора и перебора без повторений. Метод перебора с возвратом. | 2 |
| **Практические работы** | 2 |
| Использование метода полного перебора в вычислительных задачах. | 2 |  |
| Использование метода перебора без повторений и перебора с возвратом в вычислительных задачах. |
| Тема 2.4. Сложность алгоритма | Понятия временной и объемной сложности алгоритма. | **8** | 1 |
| Оценка временной сложности алгоритма. | 1 |
| **Практические работы** | 2 |
| Расчет временной сложности алгоритма. | 2 |  |
| Расчет объемной сложности алгоритма. |
| Тема 2.5. Методы сортировки данных | Понятие сортировки данных в массивах. Сортировка простым включением. | **12** | 2 |
| Алгоритм быстрой сортировки. Оценка сложности алгоритмов сортировки. | 2 |
|  | 2 |
|  |  |  |
| Расчет сложности алгоритмов сортировки. |
|  | **Самостоятельная работа:** выполнение домашних заданий по разделу 3  **Примерная тематика внеаудиторной самостоятельной работы**  Эвристические методы. Методы сортировки данных | 16 |
|  |  | **Итого** | 96 |

Для характеристики уровня освоения учебного материала используются следующие обозначения:

1. – ознакомительный (узнавание ранее изученных объектов, свойств);

2. – репродуктивный (выполнение деятельности по образцу, инструкции или под руководством)

3. – продуктивный (планирование и самостоятельное выполнение деятельности, решение проблемных задач)

# **3. условия реализации УЧЕБНОЙ дисциплины**

**3.1. Требования к минимальному материально-техническому обеспечению**

Реализация учебной дисциплины требует наличия учебного кабинета «Теория Алгоритмов».

Лаборатории нет.

Оборудование учебного кабинета:

- посадочные места по количеству обучающихся;

- рабочее место преподавателя;

- учебно-методические материалы по дисциплине «Теория Алгоритмов».

Технические средства обучения:

- компьютер с лицензионным программным обеспечением

- мультимедиа-проектор.

# **3.2. Информационное обеспечение обучения**

**Перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительной литературы**

Основная литература

Гусятников, В.Н. Стандартизация и разработка программных систем : учеб. пособие для студ. вузов / В. Н. Гусятников, А. И. Безруков. - М. : Финансы и статистика, 2010. - 286 с.

Дополнительная литература

* Мальцева А.И. Алгоритмы и рекурсивные функции. М. «Наука», 1985.
* Вентцель Е.С. Исследование операций. Задачи, принципы, методология. - М.: Высшая школа, 2001.
* Аронович А.Б., Афанасьев М.Ю., Суворов Б.П. Сборник задач по исследованию операций. - М.: Изд-во МГУ, 1997.
* Sundaram R.K. A First Curse in Optimization Theory. Cambridge Univ. Press, 1999.
* Подиновский В.В. Математическая теория выработки решений в сложных ситуациях. Учебник. - М.: МО СССР, 1982.
* Томас Ричард. Количественные методы анализа хозяйственной деятельности. - М.: Дело и Сервис, 1999.
* Эддоус М., Стэнсфилд Р. Методы принятия решений. - М.: Аудит ЮНИТИ, 1997.
* Chiang Alpha С. Fundamental Methods of mathematical economics, McGrawhill, 1984
* Матрос Д.Ш. Теория алгоритмов. – М.: БИНОМ, 2008;

Интернет-ресурсы

http://ru.wikipedia.org/wiki/Теория Алгоритмов

# **4. Контроль и оценка результатов освоения УЧЕБНОЙ Дисциплины**

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

|  |  |
| --- | --- |
| **Результаты обучения**  **(освоенные умения, усвоенные знания)** | **Формы и методы контроля и оценки результатов обучения** |
| 1 | 2 |
| **Уметь:** |  |
| Использовать и знать основные понятия теории алгоритмов; | практические работы,  внеаудиторная самостоятельная работа |
| Проводить анализ всего многообразия типов алгоритмов с целью выбора наиболее приемлемого варианта для конкретного использования; | практические работы,  внеаудиторная самостоятельная работа |
| Проводить сравнительный анализ основных параметров алгоритмов; | практические работы,  внеаудиторная самостоятельная работа |
| Разрабатывать алгоритмы и программы для решения вычислительных задач, учитывая необходимую точность получаемого результата. | практические работы,  внеаудиторная самостоятельная работа |
| **Знать:** |  |
| Основы построения алгоритмов:  Технические и эксплуатационные характеристики алгоритмов; | внеаудиторная самостоятельная работа  практические работы, |
| Основные понятия и терминологию в области вычислительной техники; | внеаудиторная самостоятельная работа |
| Классификации алгоритмов; | внеаудиторная самостоятельная работа |

Формы и методы контроля и оценки результатов обучения должны позволять проверять у обучающихся развитие общих компетенций и обеспечивающих их умений.

|  |  |  |
| --- | --- | --- |
| Результаты  (освоенные общие компетенции) | Основные показатели оценки результата | Формы и методы контроля и оценки |
| ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес. | Активность студентов при проведении учебно-воспитательных мероприятий профессиональной направленности | Самооценка результатов собственной деятельности. Публичный рейтинг с целью демонстрации индивидуальных и групповых компетенций |
| ОК 2. Организовывать собственную деятельность, определять методы и способы выполнения профессиональных задач, оценивать их эффективность и качество. | Обоснование выбора и применения методов и способов решения профессиональных задач в области подготовки и организации сетевого взаимодействия на предприятиях | Экспертная оценка сформированности компетенций в ходе практической работы. Обратная связь (анализ и обсуждение результатов деятельности с целью выявления сильных/слабых компетенций студента) |
| ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность | - решение стандартных и нестандартных профессиональных задач в области применения информационных технологий, технических средств, системного ПО. | Диагностика, с целью оценки способностей к анализу, контролю и принятию решений |
| ОК 4. Осуществлять поиск, анализ и оценку информации, необходимой для постановки и решения профессиональных задач, профессионального и личностного развития. | Оперативность поиска и использования необходимой информации для качественного выполнения профессиональных задач, профессионального и личностного развития.  Широта использования различных источников, включая электронные | Количественная оценка результатов практической деятельности. Качественная оценка результатов практической деятельности. |
| ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности | - использовать современные информационно-коммуникационные технологии, пакеты прикладных программ | Практическая работа. Технический тест |
| ОК 6. Работать в коллективе и команде, обеспечивать ее сплочение, эффективно общаться с коллегами, руководством, потребителями. | - взаимодействия с обучающимися, преподавателями, лаборантами в ходе обучения | Самооценка результатов собственной деятельности. Публичный рейтинг с целью демонстрации индивидуальных и групповых компетенций |
| ОК 7. Брать на себя ответственность за работу членов команды (подчиненных), результат выполнения заданий | - самоанализ и коррекция результатов собственной работы | Экспертная оценка сформированности компетенций в ходе практической работы. Обратная связь (анализ и обсуждение результатов деятельности с целью выявления сильных/слабых компетенций студента) |
| ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации. | Планирование внеаудиторной самостоятельной работы при изучении профессионального модуля, выполнение дополнительных творческих заданий при выполнении домашних заданий | Диагностика, с целью оценки способностей к анализу, контролю и принятию решений |
| ОК 9. Ориентироваться в условиях частой смены технологий в профессиональной деятельности. | Проявление интереса к инновациям в области профессиональной деятельности, участие в проектной, конкурсной деятельности | Количественная оценка результатов практической деятельности. Качественная оценка результатов практической деятельности. |
| ПК 1.1 Выполнять разработку спецификаций отдельных компонент | - организация самостоятельных занятий при изучении профессионального модуля | Работа проектных групп с целью оценки ПК связанных с навыками управления рабочей группой |
| ПК 1.2 Осуществлять разработку кода программного продукта на основе  готовых спецификаций на уровне модуля. | - самоанализ и коррекция результатов собственной работы | Анализ достижений с целью выявления зоны ближайшего развития студента |

# **ВОПРОСЫ ДЛЯ ИТОГОВОЙ АТТЕСТАЦИИ**

1. Алгоритм - это...
2. Какими чертами обладает алгоритм? Дать им Определение.
3. Дискретность Алгоритма - это?
4. Абстракция потенциальной осуществимости - это?
5. На какие две части делится Исполнитель? Дать им определение.
6. Какие виды ошибок Вы знаете? Перечислить и дать им определения.
7. Перечислить свойства алгоритма. Дать им определения.
8. Назвать способы записи алгоритмов. И дать им определения.
9. Блок - схема - это...? Основные правила составления блок схем?
10. Назвать и зарисовать условные обозначения основных блоков и дать им пояснение.
11. Классификация алгоритмов. Зарисовать блок схемы и дать им определения.
12. Алгоритмы обработки массивов. Массив это - это... Виды массивов.
13. Дать определение величине. Из чего состоят величины, на что делятся.
14. Операции над величинами. Что такое операнды.
15. Типы величин.
16. Выражение - это.. Типы Выражении. Привести примеры.
17. Команда присваивания. Свойства присваивания.
18. Дать определение табличным величинам. Типы. Примеры.
19. Дать определение машины Поста. Из чего состоит машина. Привести пример.
20. Дать определение машины Тьюринга. Основные отличия машины Тьюринга от Машины Поста.
21. Проектирование Алгоритма. Методы проектирования. Структурированный алгоритм.
22. Дать определение что такое модуль? Свойства модулей.
23. Свойства модульного проектирования? Тестирование алгоритма.
24. Общая структура алгоритмического обеспечения. Построить схему.
25. Основные формы использования алгоритмов.
26. Алгоритм Дейкстера. Дать определение, что такое граф, вершина графа, начало графа, ребро. Привести пример.
27. Дать определение рекурсивной функции. В чем основное отличие рекурсии от зацикливания. Привести пример рекурсии.
28. Оценка сложности алгоритма. Классификация алгоритмов по сложности.
29. Финитный процесс - это...?
30. Алгоритм сортировки слиянием. Определение, свойства.
31. Метод грубой силы, определение и свойства.
32. Проблема соответствии Поста над алфавитом.
33. Асимпотический анализ функции.
34. Теоретический предел трудоемкости задачи
35. Алгоритм точного решения задачи о сумме (метод перебора).
36. Анализ трудоемкости механизма вызова процедуры.
37. Основная теорема о рекуррентных соотношениях.
38. Прямой анализ рекурсивного дерева вызовов.
39. Исторический обзор. Цели и задачи теории алгоритмов.
40. Сложностные классы задач. Примеры NP - полных задач.
41. Построить блок-схему нахождения факториала. Пример реализации на Delphi.
42. Алгоритмы сложения, вычитания, умножения, деления, оптимизированная блок - схема, пример реализации на delphi.
43. Построить блок- схему увеличения числа на единицу до заданного параметра.
44. Написать программу - увеличение переменной С на 5. Возвести переменную B в квадрат.
45. Составить блок схему сравнения двух переменных. Пример составить на Delphi.
46. Найти среднеарифметическое. Блок-схема, пример Delphi.
47. Составить блок схему к следующему выражению:

S:=128

**нц для** i **от** 1 **до** 4

S:=div(S,2)

**кц**

Написать программу на Delphi.

1. . i:=0; S:=0

**нц пока** i<3

i:=i+1;

S:=S+i\*i

**кц**

1. S:=0; N:=125

**нц пока** N>0

S:=S+mod(N,10) | S — сумма цифр

N:=div(N,10) | числа N

**кц**

1. a:=1; b:=1

**нц пока** a+b<10

a:=a+1

b:=b+a

**кц**

S:=a+b

1. а:=1; b:=1; S:=0;

**нц пока** a<=5

a:=a+b; b:=b+a;

S:=S+a+b

**кц**

1. S:=0

**нц для** i **от** 1 **до** 2

**нц для** j **от** 2 **до** 3

S:=S+i+j

**кц**

**кц**

1. Составить алгоритм определния количества положительных чисел среди заданных чисел  a,  b  и  c. Реализовать в delphi.
2. Составить алгоритм упорядочивания по возрастанию последовательность трех чисел  a,  b  и  c. Реализовать в delphi.
3. Составить алгоритм уравнения ((a+b+c)/4)+c/2+b\*3. Пример реализовать на delphi.
4. Составить алгоритм на следующий пример " меньшее из двух заданных неравных чисел увеличить вдвое, а большее оставить без изменения", реализовать в delphi.
5. Составить алгоритм для определения, является ли треугольник с заданными сторонами  a,  b,  c  равнобедренным. Реализовать на Delphi.
6. Составить алгоритм нахождения наименьшего числа в массиве. Реализовать в Delphi.
7. Составить алгоритм уравнения (a+b+c)/2. Пример реализовать на delphi.
8. Составить алгоритм уравнения ((a+b+c)/4)+c/2. Пример реализовать на delphi.