

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Макаренко Елена Владимировна

Должность: Ректор

Дата подписания: 11.11.2024 11:11:26

Уникальный программный ключ:

c098bc0c1041cb2a4cf926cf171d6715d99a6ae00adc8e27b55cbe1e2dbd7c78

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования «Ростовский государственный экономический университет (РИНХ)»
Финансово-экономический колледж



Р. А. Сычев

2023г.

Рабочая программа дисциплины Основы алгоритмизации и программирования

Специальность

09.02.07

Информационные системы и программирование

Форма обучения	очная
Часов по учебному плану	194
в том числе:	
аудиторные занятия	184
самостоятельная работа	4

Ростов-на-Дону
2023

Распределение часов дисциплины по семестрам

Семестр (<Курс>.<Семестр на курсе>)	3 (2.1)		4 (2.2)		Итого	
	Неделя		21			
Вид занятий	уп	рп	уп	рп	уп	рп
Лекции	32	32	46	46	78	78
Лабораторные			4	4	4	4
Практические	42	42	60	60	102	102
Итого ауд.	74	74	110	110	184	184
Контактная работа	74	74	110	110	184	184
Сам. работа	2	2	2	2	4	4
Промежуг. аттестаця			6	6	6	6
Итого	76	76	118	118	194	194

ОСНОВАНИЕ

Федеральный государственный образовательный стандарт среднего профессионального образования по специальности 09.02.07 Информационные системы и программирование (Приказ Министерства образования и науки Российской Федерации от 9 декабря 2016 г. № 1547)

Рабочая программа составлена по образовательной программе 09.02.07 Информационные системы и программирование для набора 2022 года

программа среднего профессионального образования

Учебный план утвержден учёным советом вуза от 29.08.2023 протокол № 1

Программу составил(и): Преподаватель, Ермилова Е. В.

Председатель ЦМК: Горелько Е. А.

Рассмотрено на заседании ЦМК от 30.08.2023 протокол № 1

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ	
1.1	Целью освоения дисциплины является получения навыков, разрабатывать алгоритмы для конкретных задач, использовать программы для графического отображения алгоритмов, определять сложность работы алгоритмов, работать в среде программирования, реализовывать построенные алгоритмы в виде программ на конкретном языке программирования, оформлять код программы в соответствии со стандартом кодирования, выполнять проверку, отладку кода программы

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ	
Цикл (раздел) ООП:	ОП
2.1	Требования к предварительной подготовке обучающегося:
2.1.1	Информатика
2.1.2	Основы программирования
2.2	Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:
2.2.1	Инструментальные средства разработки программного обеспечения
2.2.2	Математическое моделирование
2.2.3	Разработка кода информационных систем
2.2.4	Тестирование информационных систем

3. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ	
3.1 Знать	
ОК-01: Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам	
- актуальный профессиональный и социальный контекст, в котором приходится работать и жить; основные источники информации и ресурсы для решения задач и проблем в профессиональном и/или социальном контексте; - алгоритмы выполнения работ в профессиональной и смежных областях; методы работы в профессиональной и смежных сферах; структуру плана для решения задач; порядок оценки результатов решения задач профессиональной деятельности	
ОК-02: Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности	
- номенклатуру информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации; формат оформления результатов поиска информации	
ОК 04: Эффективно взаимодействовать и работать в коллективе и команде.	
На удовлетворительном уровне основные направления психологии, психологию личности и малых групп, психологию общения, особенности делового общения, нормы общения в коллективе;	
ОК-05: Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста.	
- приемы грамотного изложения своих мыслей и оформления документов по профессиональной тематике на государственном языке, проявлять толерантность в рабочем коллективе	
ОК-09: Пользоваться профессиональной документацией на государственном и иностранном языках	
- правила построения простых и сложных предложений на профессиональные темы; основные общеупотребительные глаголы (бытовая и профессиональная лексика); лексический минимум, относящийся к описанию предметов, средств и процессов профессиональной деятельности; особенности произношения; правила чтения текстов профессиональной направленности	
3.2 Уметь	

ОК 01: Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам

- применять различные методы моделирования для анализа и проектирования ПО; - выбирать подходы в зависимости от задач и контекста.

ОК-02: Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности

- использовать номенклатуру информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации, а также формат оформления результатов поиска информации.

ОК 04: Эффективно взаимодействовать и работать в коллективе и команде.

особенности делового общения, нормы общения в коллективе;

На удовлетворительном уровне эффективно работать в команде, поддерживать оптимальный психологический климат в лечебно-профилактическом учреждении, выстраивать позитивный стиль общения и вести деловую беседу в соответствии с этическими нормами;

ОК-05: Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста

- эффективно выражать свои мысли на устном и письменном языке; - учитывать особенности аудитории при коммуникации.

ОК-09: Пользоваться профессиональной документацией на государственном и иностранном языках

Понимать общий смысл четко произнесенных высказываний на известные темы (профессиональные и бытовые), понимать тексты на базовые профессиональные темы; участвовать в диалогах на знакомые общие и профессиональные темы; строить простые высказывания о себе и о своей профессиональной деятельности; кратко обосновывать и объяснять свои действия (текущие и планируемые); писать простые связные сообщения на знакомые или интересующие профессиональные темы

3.3 Владеть

ОК 01: Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам

- навыками работы с современными инструментами моделирования ПО; - умением оценивать эффективность выбранных методов.

ОК-02: Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности

- номенклатурой информационных источников, применяемых в профессиональной деятельности, а также приемами структурирования информации и форматом оформления результатов поиска информации

ОК 04: Эффективно взаимодействовать и работать в коллективе и команде.

- способностью своевременно выполнять письменные и устные рекомендации руководства, способностью признавать чужое мнение и критику в свой адрес, навыкам выбора стиля общения в соответствии с ситуацией;

ОК-05: Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста

- навыками построения аргументации и убеждения; - умением адаптировать свой стиль коммуникации под различные ситуации.

ОК-09: Пользоваться профессиональной документацией на государственном и иностранном языках

- основной профессиональной терминологией на государственном и иностранном языке, выполняя анализ документов в процессе решения профессиональных задач.

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Компетенции	Литература	Примечание
	Раздел 1. Основы алгоритмизации					
1.1	Тема 1.1 Эволюция языков программирования, их классификация /Лек/	3	2	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.Л2.1 Э1 Э2	

1.2	Тема 1.2. Структурное программирование. Общие принципы построения алгоритмов /Лек/	3	2	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
1.3	Составление блок-схем линейных и разветвляющихся алгоритмов /Пр/	3	4	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
1.4	Тема 1.3. Основные алгоритмические конструкции /Лек/	3	2	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
1.5	Составление блок-схем циклических алгоритмов /Пр/	3	2	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
1.6	Составление блок-схем циклических алгоритмов /Ср/	3	2	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
	Раздел 2. Программирование на алгоритмическом языке					
2.1	Тема 2.1. Основные элементы языка Turbo Pascal /Лек/	3	4	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
2.2	Знакомство с интегрированной средой программирования Turbo Pascal. /Пр/	3	2	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
2.3	Ввод и отладка программ линейной структуры по образцу /Пр/	3	2	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
2.4	Ввод и отладка программ с использованием структуры «ветвление» по образцу /Пр/	3	2	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	

2.5	Тема 2.2. Операторы языка Простые операторы. Синтаксис операторов присваивания, безусловного перехода, пустого оператора. Синтаксис операторов ввода-вывода. Управление символьным выводом на экран. Логические выражения в управляющих операторах. Условный оператор IF. Оператор выбора CASE. Операторы повтора: оператор цикла с параметром. Операторы повтора: оператор цикла с предусловием. Операторы повтора: оператор цикла с постусловием. Вложенные операторы цикла. /Лек/	3	10	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
2.6	Составление программ линейной структуры. /Пр/	3	2	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
2.7	Составление программ разветвляющейся структуры. /Пр/	3	2	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
2.8	Составление программ разветвляющейся структуры с использованием оператора выбора /Пр/	3	2	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
2.9	Составление программ циклической структуры. Оператор цикла с параметром. /Пр/	3	2	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
2.10	Составление программ циклической структуры. Оператор цикла с предусловием /Пр/	3	2	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
2.11	Составление программ циклической структуры. Оператор цикла с постусловием /Пр/	3	2	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
2.12	Составление программ циклической структуры. Вложенные операторы цикла /Пр/	3	2	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	

2.13	Тема 2.3. Процедуры и функции Понятие подпрограммы. Процедуры и функции, их сущность, назначение, различие. Организация процедур, стандартные процедуры. Процедуры, определенные пользователем: синтаксис, передача аргументов Формальные и фактические параметры. Процедуры с параметрами, описание процедур. Функции: способы организации и описание Вызов функций, рекурсия. Программирование рекурсивных алгоритмов. /Лек/	3	6	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
2.14	Составление программ с использованием процедур и функций /Пр/	3	6	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
2.15	Тема 2.4. Строки и множества /Лек/	3	4	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
2.16	Работа со строковыми переменными. Использование стандартных функций и процедур для работы со строками. /Пр/	3	4	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
2.17	Тема 2.5. Массивы, записи, указатели /Лек/	3	2	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
2.18	Обработка одномерных массивов. /Пр/	3	4	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
2.19	Обработка двумерных массивов. /Пр/	3	2	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
2.20	Использование стандартных функций для работы с массивами /Пр/	4	6	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
2.21	Обработка двумерных массивов. /Пр/	4	4	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
2.22	Тема 2.6. Организация ввода- вывода данных. Работа с файлами /Лек/	4	10	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	

2.23	Выполнение операций с файлом последовательного доступа. /Пр/	4	2	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09.	Л1.1Л2.1 Э1 Э2	
2.24	Выполнение операций с файлом произвольного доступа. /Пр/	4	2	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
2.25	Разработка программ с чтением и записью файлов разных типов. /Пр/	4	2	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
2.26	Использование стандартных процедур и функций для работы с файлами. /Пр/	4	2	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
2.27	Обработка массивов /Лаб/	4	4	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
	Раздел 3. Программирование в объектно-ориентированной среде					
3.1	Тема 3.1. Основные принципы объектно-ориентированного программирования (ООП) /Лек/	4	6	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
3.2	Тема 3.2. Интегрированная среда разработки /Лек/	4	8	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
3.3	Изучение интегрированной среды разработчика /Пр/	4	4	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
3.4	Создание простого проекта. /Пр/	4	6	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
3.5	Тема 3.3. Этапы разработки приложения /Лек/	4	4	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
3.6	Тема 3.4. Иерархия классов /Лек/	4	4	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
3.7	Объявление класса, создание экземпляров класса. /Пр/	4	2	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	

3.8	Создание наследованного класса. /Пр/	4	2	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
3.9	Перегрузка методов. /Пр/	4	2	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
3.10	Тема 3.5. Визуальное событийно-управляемое Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение. Дополнительные элементы управления. Свойства компонентов (элементов управления). Виды свойств. Синтаксис определения свойств. Категория свойств. /Лек/	4	8	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
3.11	Создание проекта с использованием кнопочных компонентов. /Пр/	4	4	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
3.12	Создание проекта с использованием компонентов для работы с текстом. /Пр/	4	4	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
3.13	Создание проекта с использованием компонентов ввода и отображения чисел, дат и времени. /Пр/	4	4	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
3.14	Создание проекта с использованием компонентов стандартных диалогов и системы меню /Пр/	4	4	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
3.15	Тема 3.6. Разработка оконного приложения /Лек/	4	6	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
3.16	Разработка оконного приложения. /Пр/	4	2	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
3.17	Разработка оконного приложения с несколькими формами. /Пр/	4	4	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	
3.18	Разработка многооконного приложения. /Пр/	4	4	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.1Л2.1 Э1 Э2	

3.19	Разработка приложения /Ср/	4	2	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.Л2.1 Э1 Э2	
3.20	Экзамен	4	6	ОК 01. ОК 02. ОК 04. ОК 05. ОК 09	Л1.Л2.1 Э1 Э2	

5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

5.1. Фонд оценочных средств для проведения промежуточной аттестации

Промежуточная аттестация проходит в форме экзамена. Перечень вопросов к экзамену:

1. Дайте понятие алгоритма, перечислите свойства и виды алгоритмов. Приведите пример простого алгоритма.
2. Перечислите способы записи алгоритмов, приведите примеры.
3. Опишите основные принципы объектно-ориентированного программирования.
4. Знакомство со средой программирования.
5. Перечислите основные типы данных языка Pascal
6. Опишите синтаксис оператора присваивания в языке Pascal
7. Опишите синтаксис условного оператора в языке Pascal
8. Опишите использование операций сравнения и логических операций в условном операторе, приведите примеры.
9. Опишите синтаксис оператора цикла с параметром
10. Опишите синтаксис оператора цикла с условием
11. Составление программ линейной структуры.
12. Составление программ разветвляющейся структуры.
13. Составление программ циклической структуры
14. Обработка одномерных массивов.
15. Опишите алгоритмы сортировки в одномерном массиве. Приведите как минимум два разных алгоритма.
16. Обработка двумерных массивов.
17. Вычисление новых массивов.
18. Удаление элементов в массиве.
19. Поиск наибольшего/наименьшего элемента массива и его номера.
20. Вложенные циклы (обработка матриц).
21. Сортировка методом “установки”.
22. Сортировка методом “пузырька”.
23. Работа со строками.
24. Работа с данными типа множество.
25. Файлы последовательного доступа.
26. Типизированные файлы.
27. Нетипизированные файлы.
28. Организация процедур.
29. Организация функций.
30. Применение рекурсивных функций.
31. Программирование модуля.
32. Создание библиотеки подпрограмм.
33. Использование указателей для организации связанных списков.
34. Изучение интегрированной среды разработчика.
35. Создание проекта с использованием компонентов для работы с текстом.
36. Создание проекта с использованием компонентов ввода и отображения чисел, дат и времени.
37. События компонентов (элементов управления), их сущность и назначение.
38. Создание процедур на основе событий.
39. Создание проекта с использованием кнопочных компонентов.
40. Создание проекта с использованием компонентов стандартных диалогов и системы меню.
41. Разработка функциональной схемы работы приложения.
42. Разработка оконного приложения с несколькими формами.
43. Разработка игрового приложения.

Критерии оценивания:

5 баллов выставляется студентам за полный и правильный ответ на все вопросы билета с логическим обоснованием аргументов, в ответе нет ошибок.

4 балла выставляется студентам, если вопросы билета раскрыты полностью, но обоснования доказательства недостаточны, при этом допущены две-три несущественные ошибки, исправленные по требованию преподавателя.

3 балла ставится студентам за правильный ответ на вопросы билета, при этом допущено более одной ошибки по изложению фактов или более двух-трех недочетов в ответе.

2 балла ставится студентам, если допущены существенные ошибки, показавшие, что обучающийся не обладает обязательными умениями по данной теме в полной мере.

5.2. Фонд оценочных средств для проведения текущего контроля

Представлен в Приложении 1 к рабочей программе дисциплины

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)**6.1. Рекомендуемая литература****6.1.1. Основная литература**

	Авторы,	Заглавие	Издательство, год	Количество
Л1.1	Трофимов, В. В	Основы алгоритмизации и программирования: учебник для среднего профессионального образования: учебник для среднего	Юрайт, 2022	https://urait.ru/bcode/493261 неограниченный доступ зарегистрированным пользователям

6.1.2. Дополнительная литература

	Авторы,	Заглавие	Издательство, год	Количество
Л2.1	Кудрина Е. В.	Основы алгоритмизации и программирования на языке с: Учебное пособие для СПО	Юрайт, 2022	https://urait.ru/bcode/494914 неограниченный доступ зарегистрированным пользователям

6.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

Э1	Единое окно доступа к образовательным ресурсам https://elementy.ru/catalog/8602/Edinoe okno dostupa k obrazovatelnyim resursam window edu ru		
Э2	Учебный курс – Инструменты, алгоритмы и структуры данных https://intuit.ru/studies/professional skill improvements/1497/courses/539/info		

6.3. Перечень программного обеспечения

6.3.1	Интернет-браузер - Chromium
6.3.2	Офисный пакет - LibreOffice

6.4 Перечень информационных справочных систем

6.4.1	ИСС «КонсультантПлюс»
6.4.2	ИСС «Гарант»

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

7.1	Помещения для проведения всех видов работ, предусмотренных учебным планом, укомплектованы необходимой специализированной учебной мебелью и техническими средствами обучения
-----	---

8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

Методические указания по освоению дисциплины представлены в Приложении 2 к рабочей программе

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

ОП.04 Основы алгоритмизации и программирования

1. Описание показателей и критериев оценивания компетенций на различных этапах их.

1.1. Показатели и критерии оценивания компетенций:

УУД, составляющие компетенцию	Показатели оценивания	Критерии оценивания	Средства оценивания
ОК 01.: Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.			
<p>Знать:</p> <ul style="list-style-type: none"> - актуальный профессиональный и социальный контекст, в котором приходится работать и жить; основные источники информации и ресурсы для решения задач и проблем в профессиональном и/или социальном контексте; - алгоритмы выполнения работ в профессиональной и смежных областях; методы работы в профессиональной и смежных сферах; структуру плана для решения задач; порядок оценки результатов решения задач профессиональной деятельности 	<p>Сформировавшиеся систематические знания о актуальный профессиональный и социальный контекст, в котором приходится работать и жить; основные источники информации и ресурсы для решения задач и проблем в профессиональном и/или социальном контексте</p>	<p>Уровень знания алгоритмов выполнения работ в профессиональной и смежных областях; методы работы в профессиональной и смежных сферах; структуру плана для решения задач; порядок оценки результатов решения задач профессиональной деятельности</p>	<p>Т(1-40), ПЗ (1-12), ЛР</p>
<p>Уметь:</p> <ul style="list-style-type: none"> - применять различные методы моделирования для анализа и проектирования ПО; - выбирать подходы в зависимости от задач и контекста. 	<p>Сформировавшиеся систематические умения применять различные методы моделирования для анализа и проектирования ПО</p>	<p>Уровень умения Выбора подходов в зависимости от задач и контекста.</p>	<p>Т(1-40), ПЗ (1-12), ЛР</p>
<p>Владеть:</p> <ul style="list-style-type: none"> - навыками работы с современными инструментами моделирования ПО; - умением оценивать эффективность выбранных методов. 	<p>Сформировавшиеся систематические владения о Требованиях к безопасности сервера базы данных.</p>	<p>Уровень владения о Государственных стандартах и требований к обслуживанию баз данных.</p>	<p>Т(1-40), ПЗ (1-12), ЛР</p>

ОК 02.: Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности

<p>Знать: - номенклатуру информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации; формат оформления результатов поиска информации</p>	<p>Сформировавшиеся систематические знания на высоком уровне об основных понятиях автоматизированной обработки информации, общий состав и структуры персональных компьютеров и вычислительных систем, базовых системных программных продуктах и пакетах прикладных программ в области профессиональной деятельности;</p>	<p>Уровень знания на высокопрофессиональном уровне об основных понятиях автоматизированной обработки информации, общем составе и структуре персональных компьютеров и вычислительных систем, базовых системных программных продуктах и пакетах прикладных программ в области профессиональной деятельности;</p>	<p>Т(1-40), ПЗ (1-12), ЛР</p>
<p>Уметь: – использовать номенклатуру информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации, а также формат оформления результатов поиска информации.</p>	<p>Сформировавшиеся систематические умения на высоком уровне использовать технологии сбора, размещения, хранения, накопления, преобразования и передачи данных в профессионально ориентированных информационных системах, выделять профессионально-значимую профессиональную информацию, использовать основные методы и приемы обеспечения информационной безопасности;</p>	<p>Уровень умения на высокопрофессиональном уровне использовать технологии сбора, размещения, хранения, накопления, преобразования и передачи данных в профессионально ориентированных информационных системах, выделять профессионально-значимую профессиональную информацию, использовать основные методы и приемы обеспечения информационной безопасности;</p>	<p>Т(1-40), ПЗ (1-12), ЛР</p>
<p>Владеть: - номенклатурой информационных источников, применяемых в профессиональной деятельности, а также приемами структурирования информации и форматом оформления результатов поиска информации</p>	<p>Сформировавшиеся систематические владения на высоком уровне использование технологии сбора, размещения, хранения, накопления, преобразования и передачи данных в профессионально ориентированных информационных системах, выделять профессионально-значимую профессиональную информацию, использовать основные методы и приемы обеспечения информационной безопасности;</p>	<p>Уровень владения на высокопрофессиональном уровне использовать технологии сбора, размещения, хранения, накопления, преобразования и передачи данных в профессионально ориентированных информационных системах, выделять профессионально-значимую профессиональную информацию, использовать основные методы и приемы обеспечения информационной безопасности;</p>	<p>Т(1-40), ПЗ (1-12), ЛР</p>

ОК 04: Эффективно взаимодействовать и работать в коллективе и команде.

<p>Знать: На удовлетворительном уровне основные направления психологии, психологию личности и малых групп, психологию общения, особенности делового общения, нормы общения в коллективе;</p>	<p>Сформировавшиеся систематические знания на высоком уровне основные направления психологии, психологию личности и малых групп, психологию общения, особенности делового общения, нормы общения в коллективе;</p>	<p>Уровень знания на высокопрофессиональном уровне основных направлений психологии, психологию личности и малых групп, психологию общения, особенности делового общения, нормы общения в коллективе;</p>	<p>Т(1-40), ПЗ (1-12), ЛР</p>
<p>Уметь: На удовлетворительном уровне эффективно работать в команде, поддерживать оптимальный психологический климат в лечебно-профилактическом учреждении, выстраивать позитивный стиль общения и вести деловую беседу в соответствии с этическими нормами;</p>	<p>Сформировавшиеся систематические умения на высоком уровне эффективно работать в команде, поддерживать оптимальный психологический климат, выстраивать позитивный стиль общения и вести деловую беседу в соответствии с этическими нормами;</p>	<p>Уровень умения на высокопрофессиональном уровне эффективно работать в команде, поддерживать оптимальный психологический климат, выстраивать позитивный стиль общения и вести деловую беседу в соответствии с этическими нормами;</p>	<p>Т(1-40), ПЗ (1-12), ЛР</p>
<p>Владеть: На удовлетворительном уровне способностью своевременно выполнять письменные и устные рекомендации руководства, способностью признавать чужое мнение и критику в свой адрес, навыкам выбора стиля общения в соответствии с ситуацией;</p>	<p>Сформировавшиеся систематические владения на высоком уровне способностью своевременно выполнять письменные и устные рекомендации руководства, способностью признавать чужое мнение и критику в свой адрес, навыкам выбора стиля общения в соответствии с ситуацией;</p>	<p>Уровень владения на высокопрофессиональном уровне способностью своевременно выполнять письменные и устные рекомендации руководства, способностью признавать чужое мнение и критику в свой адрес, навыкам выбора стиля общения в соответствии с ситуацией;</p>	<p>Т(1-40), ПЗ (1-12), ЛР</p>
<p>ОК 05.: Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста.</p>			
<p>Знать: - приемы грамотного изложения своих мыслей и оформления документов по профессиональной тематике на государственном языке, проявлять толерантность в рабочем коллективе</p>	<p>Сформировавшиеся систематические знания на высоком уровне состава, функции и возможности использования информационных и телекоммуникационных технологий в профессиональной деятельности, методы и средства сбора, обработки, хранения, передачи и накопления информации, правила ведения деловой беседы в соответствии с этическими нормами;</p>	<p>Уровень знания на высокопрофессиональном уровне состава, функции и возможности использования информационных и телекоммуникационных технологий в профессиональной деятельности, методы и средства сбора, обработки, хранения, передачи и накопления информации, правила ведения деловой беседы в соответствии с этическими нормами;</p>	<p>Т(1-40), ПЗ (1-12), ЛР</p>

<p>Уметь: - эффективно выражать свои мысли на устном и письменном языке; - учитывать особенности аудитории при коммуникации.</p>	<p>Сформировавшиеся систематические умения на высоком уровне использовать в профессиональной деятельности различные виды программного обеспечения, в том числе специального, применять компьютерные и телекоммуникационные средства, представлять информацию в различных формах с использованием разнообразного программного обеспечения;</p>	<p>Уровень умения на высокопрофессиональном уровне использовать в профессиональной деятельности различные виды программного обеспечения, в том числе специального, применять компьютерные и телекоммуникационные средства, представлять информацию в различных формах с использованием разнообразного программного обеспечения;</p>	<p>Т(1-40), ПЗ (1-12), ЛР</p>
<p>Владеть: - навыками построения аргументации и убеждения; - умением адаптировать свой стиль коммуникации под различные ситуации.</p>	<p>Сформировавшиеся систематические владения на высоком уровне навыком поиска информации в сети Интернет и на различных электронных носителях, созданием различных макетов презентаций, навыками представления информации в различных формах с использованием разнообразного программного обеспечения;</p>	<p>Уровень владения на высокопрофессиональном уровне навыком поиска информации в сети Интернет и на различных электронных носителях, созданием различных макетов презентаций, навыками представления информации в различных формах с использованием разнообразного программного обеспечения;</p>	<p>Т(1-40), ПЗ (1-12), ЛР</p>
<p>ОК 9.: Пользоваться профессиональной документацией на государственном и иностранном языках.</p>			
<p>Знать: - правила построения простых и сложных предложений на профессиональные темы; основные общеупотребительные глаголы (бытовая и профессиональная лексика); лексический минимум, относящийся к описанию предметов, средств и процессов профессиональной деятельности; особенности произношения; правила чтения текстов профессиональной направленности</p>	<p>Сформировавшиеся систематические знания основные общеупотребительные глаголы (бытовая и профессиональная лексика); лексический минимум, относящийся к описанию предметов, средств и процессов профессиональной деятельности на государственном и иностранном языках</p>	<p>Уровень знания правила чтения текстов и составления профессиональной направленности на государственном и иностранном языках</p>	<p>Т(1-40), ПЗ (1-12), ЛР</p>

<p>Уметь: Понимать общий смысл чётко произнесённых высказываний на известные темы (профессиональные и бытовые), понимать тексты на базовые профессиональные темы; участвовать в диалогах на знакомые общие и профессиональные темы; строить простые высказывания о себе и о своей профессиональной деятельности; кратко обосновывать и объяснять свои действия (текущие и планируемые); писать простые связные сообщения на знакомые или интересующие профессиональные темы</p>	<p>Сформировавшиеся систематические умения участвовать в диалогах на знакомые общие и профессиональные темы; строить простые высказывания о себе и о своей профессиональной деятельности</p>	<p>Уровень умения писать простые связные тексты документов, изучать и пользоваться профессиональной документацией на государственном и иностранном языках</p>	<p>Т(1-40), ПЗ (1-12), ЛР</p>
<p>Владеть: - основной профессиональной терминологией на государственном и иностранном языке, выполняя анализ документов в процессе решения профессиональных задач.</p>	<p>Сформировавшиеся систематические владения правилами и стандартами составления профессиональной документации на государственном и иностранном языках</p>	<p>Уровень владения на высокопрофессиональном уровне правилами и стандартами составления профессиональной документации на государственном и иностранном языках</p>	<p>Т(1-40), ПЗ (1-12), ЛР</p>

Т – тестовые задания, ПЗ – практические задания, ЛР- лабораторная работа

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

Тестовые задания:

1. Что такое алгоритм?
 - a) Набор инструкций для приготовления еды
 - b) Способ решения конкретной задачи с помощью последовательности действий
 - c) Программа для компьютера
 - d) Математическая формула

2. Какой из следующих языков программирования не является языком высокого уровня?
 - a) Python
 - b) Java
 - c) C++
 - d) Ассемблер

3. Что такое переменная в программировании?
 - a) Элемент программы, который может принимать разные значения
 - b) Однозначный символ или слово, обозначающее операцию

- c) Комментарий в коде
- d) Ошибка в программе

4. Какой из следующих циклов выполняется, пока условие истинно?

- a) for
- b) while
- c) do-while
- d) switch

5. Что такое функция в программировании?

- a) Фрагмент программы, выполняющий определенные действия
- b) Переменная, которая хранит только числовые значения
- c) Условный оператор
- d) Тип данных

6. Что такое массив в программировании?

- a) Набор однотипных переменных, объединенных общим именем
- b) Набор разнотипных переменных, объединенных общим именем
- c) Функция, которая возвращает только одно значение
- d) Условная конструкция

7. Какой из следующих операторов используется для сравнения двух значений на равенство?

- a) =
- b) ==
- c) !=
- d) <>

8. Что такое рекурсия в программировании?

- a) Вызов функции из другой функции
- b) Вызов функции самой себя
- c) Обращение к элементам массива по индексу
- d) Операция присваивания

9. Какой из следующих типов данных может хранить только целые числа?

- a) int
- b) float
- c) char
- d) boolean

10. Что такое отладка в программировании?

- a) Процесс написания кода
- b) Процесс исправления ошибок в коде
- c) Процесс компиляции кода
- d) Процесс тестирования кода

11. Что такое асимптотическая сложность алгоритма?

- a) Сложность алгоритма при малых входных данных
- b) Сложность алгоритма при средних входных данных
- c) Сложность алгоритма при больших входных данных
- d) Сложность алгоритма при любых входных данных

12. Какая нотация используется для описания асимптотической сложности алгоритма?

- a) O-нотация

- b) Омега-нотация
- c) Тета-нотация
- d) Все вышеперечисленные варианты

13. Какова асимптотическая сложность алгоритма сортировки пузырьком?

- a) $O(n)$
- b) $O(n \log n)$
- c) $O(n^2)$
- d) $O(2^n)$

14. Что такое двоичное дерево поиска?

- a) Дерево, в котором каждый узел имеет не более двух потомков
- b) Дерево, в котором каждый узел имеет ровно два потомка
- c) Дерево, в котором левый потомок меньше корня, а правый потомок больше корня
- d) Дерево, в котором левый потомок больше корня, а правый потомок меньше корня

15. Что такое стек в программировании?

- a) Список, в котором элементы добавляются и удаляются только с одного конца
- b) Список, в котором элементы добавляются и удаляются только с двух концов
- c) Список, в котором элементы добавляются и удаляются в произвольном порядке
- d) Список, в котором элементы добавляются и удаляются только с середины

16. Что такое очередь в программировании?

- a) Список, в котором элементы добавляются и удаляются только с одного конца
- b) Список, в котором элементы добавляются и удаляются только с двух концов
- c) Список, в котором элементы добавляются и удаляются в произвольном порядке
- d) Список, в котором элементы добавляются и удаляются только с середины

17. Что такое рекурсивный алгоритм?

- a) Алгоритм, который вызывает сам себя
- b) Алгоритм, который вызывает другой алгоритм
- c) Алгоритм, который не имеет вызова других алгоритмов
- d) Алгоритм, который вызывается только один раз

18. Что такое хеш-таблица?

- a) Структура данных, которая хранит пары ключ-значение
- b) Структура данных, которая хранит только ключи
- c) Структура данных, которая хранит только значения
- d) Структура данных, которая хранит произвольные данные

19. Что такое граф в программировании?

- a) Структура данных, состоящая из множества вершин и множества ребер
- b) Структура данных, состоящая из множества вершин и одного ребра
- c) Структура данных, состоящая из множества ребер и одной вершины
- d) Структура данных, состоящая из множества вершин без ребер

20. Что такое обход графа в глубину?

- a) Обход всех вершин графа, начиная с первой и заканчивая последней
- b) Обход всех вершин графа, начиная с последней и заканчивая первой
- c) Обход всех вершин графа, начиная с какой-либо вершины и заканчивая ее же
- d) Обход всех вершин графа, начиная с какой-либо вершины и заканчивая любой другой

21. Что такое алгоритм Дейкстры?

- a) Алгоритм поиска кратчайшего пути в графе
- b) Алгоритм сортировки
- c) Алгоритм поиска цикла в графе
- d) Алгоритм обхода графа в ширину

22. Что такое бинарный поиск?

- a) Алгоритм поиска элемента в неупорядоченном массиве
- b) Алгоритм поиска элемента в упорядоченном массиве
- c) Алгоритм сортировки
- d) Алгоритм обхода дерева

23. Что такое быстрая сортировка?

- a) Алгоритм сортировки, основанный на сравнении соседних элементов
- b) Алгоритм сортировки, основанный на разбиении массива на подмассивы
- c) Алгоритм сортировки, основанный на подсчете элементов
- d) Алгоритм сортировки, основанный на вставке элементов

24. Что такое динамическое программирование?

- a) Метод программирования, основанный на использовании динамических библиотек
- b) Метод программирования, основанный на использовании динамической памяти
- c) Метод решения задач, основанный на разбиении их на подзадачи
- d) Метод решения задач, основанный на использовании динамических структур данных

25. Что такое жадный алгоритм?

- a) Алгоритм, который всегда делает выбор, наиболее выгодный в текущий момент
- b) Алгоритм, который всегда делает выбор, наиболее выгодный в будущем
- c) Алгоритм, который всегда делает выбор, наиболее выгодный в прошлом
- d) Алгоритм, который всегда делает выбор, наиболее выгодный в целом

26. Что такое задача о ранце?

- a) Задача о поиске кратчайшего пути в графе
- b) Задача о сортировке элементов
- c) Задача о выборе оптимального набора предметов с ограниченным объемом
- d) Задача о поиске цикла в графе

27. Что такое задача о коммивояжере?

- a) Задача о поиске кратчайшего пути в графе
- b) Задача о сортировке элементов
- c) Задача о выборе оптимального набора предметов с ограниченным объемом
- d) Задача о поиске оптимального маршрута, проходящего через все вершины графа

28. Что такое задача о покрытии множества?

- a) Задача о поиске кратчайшего пути в графе
- b) Задача о сортировке элементов
- c) Задача о выборе оптимального набора предметов, покрывающих все элементы множества
- d) Задача о поиске цикла в графе

29. Что такое задача о разрезании отрезка?

- a) Задача о поиске кратчайшего пути в графе
- b) Задача о сортировке элементов
- c) Задача о выборе оптимального набора точек разрезания отрезка
- d) Задача о поиске цикла в графе

30. Что такое задача о назначениях?

- a) Задача о поиске кратчайшего пути в графе
- b) Задача о сортировке элементов
- c) Задача о выборе оптимального набора пар элементов, удовлетворяющих определенным условиям
- d) Задача о поиске цикла в графе

31. Что такое асимптотическая нотация Big O?

- a) Нотация, используемая для описания наилучшего случая алгоритма
- b) Нотация, используемая для описания наихудшего случая алгоритма
- c) Нотация, используемая для описания среднего случая алгоритма
- d) Нотация, используемая для описания любого случая алгоритма

32. Что такое асимптотическая нотация Big Omega?

- a) Нотация, используемая для описания наилучшего случая алгоритма
- b) Нотация, используемая для описания наихудшего случая алгоритма
- c) Нотация, используемая для описания среднего случая алгоритма
- d) Нотация, используемая для описания любого случая алгоритма

33. Что такое асимптотическая нотация Big Theta?

- a) Нотация, используемая для описания наилучшего случая алгоритма
- b) Нотация, используемая для описания наихудшего случая алгоритма
- c) Нотация, используемая для описания среднего случая алгоритма
- d) Нотация, используемая для описания любого случая алгоритма

34. Что такое линейное время работы алгоритма?

- a) Время работы алгоритма, линейно зависящее от количества элементов входных данных
- b) Время работы алгоритма, квадратично зависящее от количества элементов входных данных
- c) Время работы алгоритма, экспоненциально зависящее от количества элементов входных данных
- d) Время работы алгоритма, логарифмически зависящее от количества элементов входных данных

35. Что такое логарифмическое время работы алгоритма?

- a) Время работы алгоритма, линейно зависящее от количества элементов входных данных
- b) Время работы алгоритма, квадратично зависящее от количества элементов входных данных
- c) Время работы алгоритма, экспоненциально зависящее от количества элементов входных данных
- d) Время работы алгоритма, логарифмически зависящее от количества элементов входных данных

36. Что такое квадратичное время работы алгоритма?

- a) Время работы алгоритма, линейно зависящее от количества элементов входных данных
- b) Время работы алгоритма, квадратично зависящее от количества элементов входных данных
- c) Время работы алгоритма, экспоненциально зависящее от количества элементов входных данных
- d) Время работы алгоритма, логарифмически зависящее от количества элементов входных данных

37. Что такое экспоненциальное время работы алгоритма?

- a) Время работы алгоритма, линейно зависящее от количества элементов входных данных
- b) Время работы алгоритма, квадратично зависящее от количества элементов входных данных
- c) Время работы алгоритма, экспоненциально зависящее от количества элементов входных данных
- d) Время работы алгоритма, логарифмически зависящее от количества элементов входных данных

38. Что такое поиск в ширину?

- a) Алгоритм обхода графа, основанный на обходе всех вершин на одном уровне перед переходом к следующему уровню
- b) Алгоритм обхода графа, основанный на обходе всех вершин на одном уровне после перехода к следующему уровню
- c) Алгоритм обхода графа, основанный на обходе всех вершин в глубину перед переходом к следующей

вершине

d) Алгоритм обхода графа, основанный на обходе всех вершин в глубину после перехода к следующей вершине

39. Что такое поиск в глубину?

a) Алгоритм обхода графа, основанный на обходе всех вершин на одном уровне перед переходом к следующему уровню

b) Алгоритм обхода графа, основанный на обходе всех вершин на одном уровне после перехода к следующему уровню

c) Алгоритм обхода графа, основанный на обходе всех вершин в глубину перед переходом к следующей вершине

d) Алгоритм обхода графа, основанный на обходе всех вершин в глубину после перехода к следующей вершине

40. Что такое обход графа в глубину с рекурсией?

a) Алгоритм обхода графа, основанный на обходе всех вершин на одном уровне перед переходом к следующему уровню

b) Алгоритм обхода графа, основанный на обходе всех вершин на одном уровне после перехода к следующему уровню

c) Алгоритм обхода графа, основанный на обходе всех вершин в глубину перед переходом к следующей вершине с использованием рекурсии

d) Алгоритм обхода графа, основанный на обходе всех вершин в глубину после перехода к следующей вершине с использованием рекурсии

Критерии оценивания:

- 5 баллов выставляется, если правильные ответы даны на 85- 100% вопросов

- 4 балла выставляется студенту, если правильные ответы даны на 65-84% вопросов

- 3 балла выставляется студенту, если правильные ответы даны на 50-64% вопросов

- 2 балла выставляется студенту, если правильные ответы даны на менее 50% тестовых заданий

Практические задания:

1 семестр

1. Составить алгоритм решения задачи. Описать алгоритмы в словесной форме и с помощью блок-схемы.
 - 1.1. Вычислить значение функции $f(z) = \left(1 + \frac{1}{z^2}\right) - \sin z$ для $z=3$
 - 1.2. Ввести два числа a и b . Если $a < b$, оба числа возвести в квадрат, если $a = b$ - оставить их без изменения. Если $a > b$, возвести эти числа в куб.
2. Составить алгоритм решения задачи. Описать алгоритмы в словесной форме и с помощью блок-схемы.
 - 2.1. Определить количество натуральных чисел, не превышающих n , которые делятся на 11.
 - 2.2. Вычислить $y = \cos x + \cos^2 x + \cos^3 x + \dots + \cos^{30} x$
3. Составить программу с применением условных операторов
 - 3.1. Даны действительные числа a, b, c . Проверить выполняются ли неравенства $a < b < c$.
 - 3.2. Дано натуральное число a . Выяснить, является ли данное число четным.
4. Составить программу с применением операторов цикла
 - 4.1. Используя цикл с предусловием, составить программу табулирования функции $f(x) = x - \sin x$ на отрезке $[a, b]$ с шагом h . Результат представить в виде таблицы, первый столбец которой – значения аргумента, второй – соответствующие значения функции.
 - 4.2. Используя цикл с постусловием, составить программу табулирования функции $f(x) = \operatorname{tg} x$ на отрезке $[a, b]$ с шагом h . Результат представить в виде таблицы, первый столбец которой – значения аргумента, второй – соответствующие значения функции
 - 4.3. Дано натуральное число n . Найти количество цифр данного числа, больших a (a вводится с

клавиатуры).

5. Составить программу с применением вложенных циклов
 - 5.1. Дано натуральное число n . Составить программу вычисления значения выражения $1^1 + 2^2 + \dots + n^n$
 - 5.2. Даны натуральные числа a, b ($a < b$). Получить все простые числа p , удовлетворяющие неравенствам:
 $a \leq p \leq b$.

2 семестр

6. Составить программу для обработки одномерных массивов
 - 6.1. Заполнить одномерный массив случайными числами, найти его наибольший элемент.
 - 6.2. В одномерном массиве, состоящем из n вещественных элементов, вычислить сумму элементов массива, расположенных между первым и последним отрицательными элементами. Элементы массива вводятся с клавиатуры.
 - 6.3. В одномерном массиве, состоящем из n вещественных элементов, вычислить сумму элементов массива, расположенных до последнего положительного элемента. Элементы массива вводятся с клавиатуры.
 - 6.4. Массив $A(n)$ вводится с клавиатуры. Найти среднее арифметическое его элементов с нечетными номерами.
 - 6.5. Дан массив $A(n)$, состоящий из целых чисел. Найти количество и сумму тех элементов, которые делятся на 5 и не делятся на 7. Элементы массива вводятся с клавиатуры.
7. Составить программу для обработки двумерных массивов
 - 7.1. Дана целочисленная квадратная матрица. Вывести ее на экран и определить сумму элементов в тех строках, которые не содержат отрицательных элементов.
 - 7.2. Дана целочисленная прямоугольная матрица. Вывести ее на экран и определить количество столбцов, содержащих хотя бы один нулевой элемент.
 - 7.3. Дана целочисленная прямоугольная матрица. Вывести ее на экран и определить количество положительных элементов в каждом столбце.
 - 7.4. Дана целочисленная прямоугольная матрица. Вывести ее на экран и определить количество четных элементов в каждом столбце.
8. Составить программу с использованием строковых типов данных
 - 8.1. Дана строка символов S . Подсчитать, сколько раз среди данных символов встречается буква x .
 - 8.2. Дана строка символов. Выяснить имеются ли в данной строке рядом стоящие запятая и тире.
 - 8.3. Определить, сколько раз в строке встречается заданное слово.
 - 8.4. Дана строка. Указать те слова, которые содержат хотя бы одну букву t .
 - 8.5. Дана строка символов. Удалить из неё все знаки препинания.
 - 8.6. Дана строка, содержащая текст. Найти длину самого короткого и самого длинного слова.
9. Составить программу с использованием множественного типа данных
 - 9.1. Известны сорта роз, выращиваемые тремя цветоводами: «Анжелика», «Виктория», «Гагарин», «Ave Maria», «Катарина», «Юбилейная». Определить те сорта, которые имеются у каждого из цветоводов; которые есть хотя бы у одного из цветоводов; которых нет ни у одного из цветоводов.
 - 9.2. Имеется множество, содержащее натуральные числа из некоторого диапазона. Сформировать два множества, первое из которых содержит все простые числа, а второе – все составные.
 - 9.3. Дан текст из строчных латинских букв, заканчивающийся точкой. Напечатать все буквы, входящие в текст не менее двух раз.
 - 9.4. Дан текст из строчных латинских букв, заканчивающийся точкой. Напечатать все буквы, входящие в текст по одному разу.
10. Составить программу с использованием процедур
 - 10.1. Написать процедуру перевода числа из градусной меры в радианную.
 - 10.2. Написать процедуру, заменяющую в исходной строке все символы-единицы на символы-нули. Замена выполняется, начиная с заданной позиции строки.
 - 10.3. Написать процедуру, которая находит из двух целых чисел наибольшее число
 - 10.4. Написать процедуру, результатом которой является true, если символ, заданный при обращении к

процедуре - буква, и false в противном случае.

11. Составить программу с использованием функций

11.1. Составить функцию нахождения факториала и вычислить значение выражения:

$$y = \frac{(p - m)! * n!}{(p - n)! * (m - n)!}$$

12. Получить практические навыки в работе с файловым типом данных

12.1. Дан файл f, элементы которого являются действительными числами. Найти сумму и произведение элементов файла.

12.2. Дан файл f, элементы которого являются действительными числами. Найти сумму квадратов элементов файла и последний элемент файла.

Критерии оценивания:

- 5 баллов выставляется, если правильные ответы даны на 85-100% практических заданий
- 4 балла выставляется студенту, если правильные ответы даны на 65-84% практических заданий
- 3 балла выставляется студенту, если правильные ответы даны на 50-64% практических заданий
- 2 балла выставляется студенту, если правильные ответы даны на менее 50% практических заданий.

Лабораторная работа:

Обработка одномерных массивов

Задание: составить программу заданной обработки массива целых чисел. В процессе обработки использовать перестановки элементов внутри массива, не создавая новых массивов. Заполнение исходного массива организовать с помощью генератора случайных чисел, если иное не предусмотрено вариантом задания. Исходный и обработанный массив выводить на экран.

Теоретический материал

При выполнении лабораторной работы будет использоваться структурированный тип данных *массив*, который представляет собой фиксированное количество упорядоченных однотипных компонент, снабженных индексами. Он может быть *одномерным* и *многомерным*. В лабораторной работе №1 будут использоваться одномерные массивы

Тип-массив описывается в разделе описания типов следующим образом:

type<имя типа>= **array**[<тип индекса(индексов)>] **of**<тип компонента>;

Размерность массива может быть любой, компоненты массива могут быть любого (в том числе и структурированного) типа, индекс (индексы) может (могут) быть только интервального или перечисляемого типа. При употреблении в качестве типа индекса типа *Integer* или *Word* можно использовать только его диапазон.

При описании типа индекса (индексов) можно использовать константы, которые должны быть определены до определения типа.

Определенный в разделе описания типов тип-массив можно использовать для описания переменных и типизированных констант. Тип-массив можно вводить непосредственно и при определении соответствующих переменных или типизированных констант.

При задании значений константе-массиву компоненты указываются в круглых скобках и разделяются запятыми, причем, если массив многомерный, внешние круглые скобки соответствуют левому индексу, вложенные в них круглые скобки – следующему индексу и т. д. При этом все компоненты массива должны быть заполнены.

Доступ к компонентам массива осуществляется указанием имени массива, за которым в квадратных скобках помещается значение индекса (индексов) компоненты. В многомерных массивах значения индексов перечисляются через запятую.

Для работы с массивом как с единым целым используется идентификатор массива без указания индексов в квадратных скобках. Массив может участвовать только в операциях отношения («=», «<>») и в операторе присваивания.

Массивы, участвующие в этих действиях, должны быть идентичны по структуре, то есть иметь

одинаковые базовые типы и типы индексов.

Задание значений переменной типа массив может осуществляться с клавиатуры, путем генерирования случайным образом либо иным способом. Когда значения элементов задаются с клавиатуры или генерируются случайным образом, необходима организация цикла, в котором последовательно происходит обращение к каждому элементу массива. Вывод значений элементов массива на экран или в файл также происходит в цикле.

При написании программ необходимо следить за тем, чтобы значения индексов не превышали границ, указанных при объявлении массива, так как выход индекса за границы массива приводит к сбою в работе программы. Контроль значений индексов массивов можно организовать при помощи директивы компилятора `{R+}`, которая приводит к проверке всех индексных выражений на соответствие их значений диапазону индекса.

Пример выполнения лабораторной работы

Для примера рассмотрим следующую задачу. Дан одномерный массив целых чисел. Удалить все отрицательные элементы, расположенные справа от максимального.

Прежде всего, определим алгоритм решения данной задачи. Вначале надо найти максимальный элемент среди элементов массива и запомнить его индекс. Затем, начиная со следующего за максимальным элементом, проверять элементы массива на знак, и если число оказывается отрицательным, то удаляем этот элемент из массива путем «сдвига» элементов массива влево и уменьшаем длину массива. При этом следует заметить, что в памяти переменная типа массив будет занимать столько же места, как и до удаления элементов.

В графическом виде этот алгоритм будет блок-схему, представленную на рис. 5.1.

Теперь пишем код программы.

Поскольку в программе требуется использование генератора случайных чисел, то необходимо подключить стандартный модуль CRT, в котором описана функция генерации случайных чисел.

Uses crt;

В разделе описания типов определим тип-массив:

```
type arr=array[1..15] of integer;
```

В разделе описания переменных описываем необходимые переменные:

```
var a:arr; {массив}
i,j,n: byte; {индексы, длина массива}
```

i_max : byte; {индекс максимального элемента массива} max : integer; {максимальный элемент}

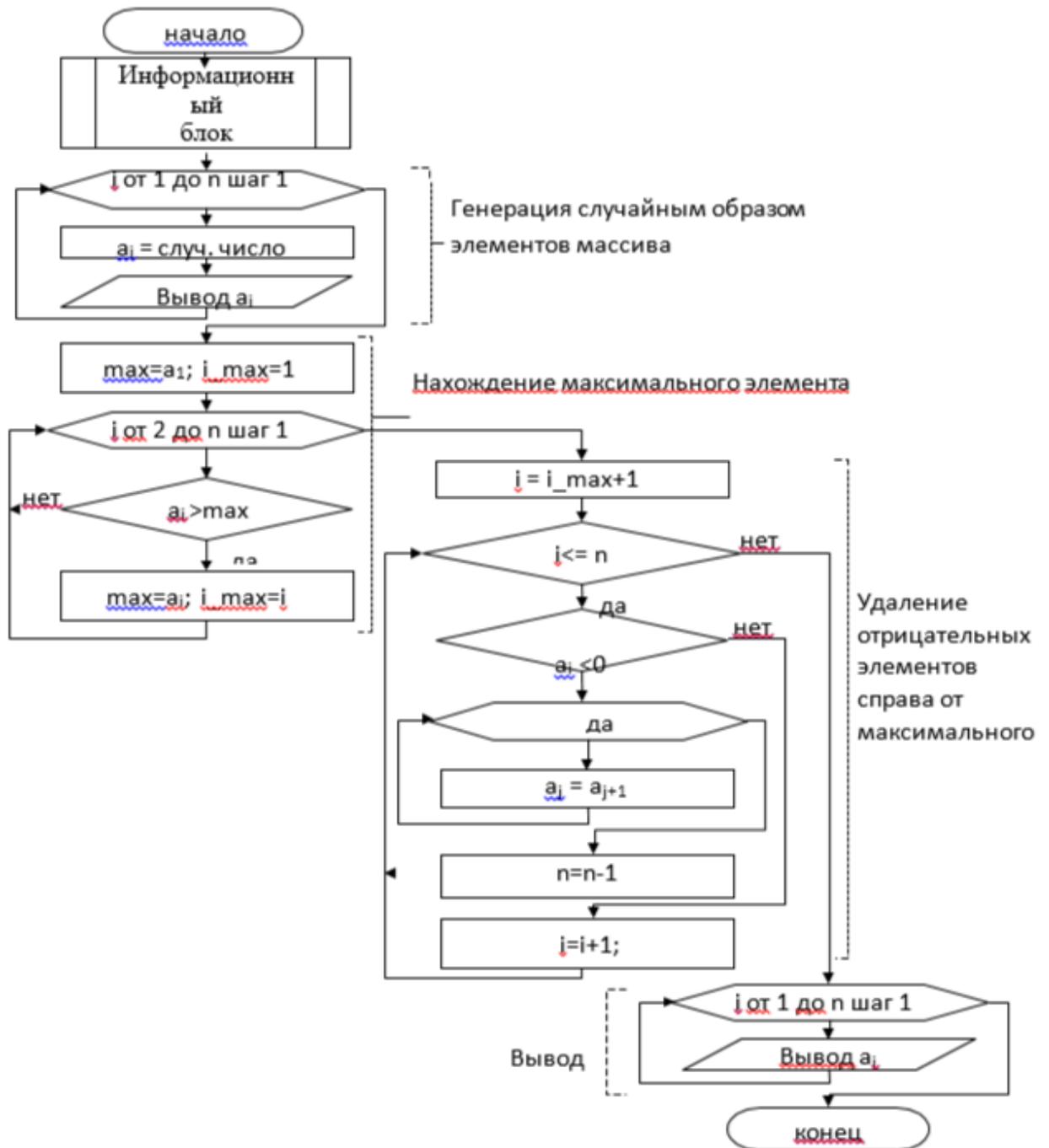


Рис. 5.1. Блок-схема алгоритма удаления из массива отрицательных элементов, расположенных справа от максимального элемента массива

Далее описываем основное тело программы. Как отмечено на рисунке, алгоритм состоит из нескольких частей. Первая часть – генерация исходного массива:

begin

writeln;

writeln(' Автор – Иванов И.П., студент гр. ИС-11');

writeln(' Вариант № 100');

writeln(' Дан одномерный массив целых чисел.');

```
writeln(' Удалить все отрицательные элементы, расположенные справа от  
максимального ');
```

```
writeln;
```

```
{ $R+ }      {Включаем контроль значений индексов} clrscr;
```

```
n:=15;
```

```
randomize;
```

```
writeln('Исходный массив:');
```

```
for i:=1 to n do          {генерация элементов массива} begin
```

```
    a[i]:=-20+random(41); {в диапазоне [-20; 20]} write(a[i]:4);
```

```
end; writeln;
```

Генератор случайных чисел активизируется командой `randomize`. Далее в цикле случайным образом задаются элементы массива. Функция `random(41)` возвращает случайное число в диапазоне от 0 до 40. Получается, что минимально возможное значение элементов массива будет равно -20 (к -20 прибавить 0, сгенерированный функцией `random`), максимально возможное – 20 (-20 плюс число 40, сгенерированное функцией `random`). Сразу же выводим значение элемента массива на экран. В цикле все элементы будут выводиться в строчку, т. к. используется команда вывода `write`. После генерации и вывода всех элементов массива осуществляется переход на новую строку.

Вторая часть алгоритма – поиск максимального элемента. Предположим, что первый элемент – максимальный, тогда запомним значение первого элемента в переменной `max`, а в переменную `i_max`, где будет запоминаться положение максимального элемента, запишем 1. Далее организуется цикл, в котором ищется элемент, больший чем `max`.

```
max:=a[1];
```

```
i_max:=1;
```

```
for i:=2 to n do
```

```
    if a[i]>max then begin
```

```
        max:=a[i];
```

```
        i_max:=i;
```

```
    end;
```

Далее, начиная с элемента, следующим за максимальным, организуется цикл проверки элементов массива на знак (важно заметить, что в данном случае нужно использовать цикл `while`, т. к. длина массива будет изменяться, и, соответственно, число итераций цикла также будет меняться). Если число оказывается отрицательным, то удаляем этот элемент из массива, для чего организуется еще один цикл, в котором на место удаляемого записывается стоящий следом элемент.

Внутренний цикл сдвига можно проиллюстрировать на примере удаления седьмого элемента из массива, содержащего 10 элементов (рис. 5.2).

Исходный массив





Рис. 5.2. Схема удаления элемента из массива

В программе удаление элементов, находящихся справа от максимального, будет выглядеть следующим образом:

```

i:=i_max; while i<=n
dobegin
    if a[i]<0 thenbegin
        for j:=i to n-1 do a[j]:=a[j+1];
        n:=n-1;
    end else
        i:=i+1;
    end;
end;

```

Еще одна тонкость данного фрагмента состоит в том, что переход к следующему элементу ($i:=i+1$) для проверки осуществляется только тогда, когда элемент не удовлетворяет условию. Если же i -й элемент окажется отрицательным, то после удаления на его месте окажется $i+1$ -й элемент, который и нужно проверять на следующей итерации цикла.

Остается вывести результирующий массив на экран.

```

writeln('Результат обработки:');for i:=1 to n do
    write(a[i]:4);readln;
{$R-}    {Выключаем контроль значений индексов}end.

```

Ниже представлен полный текст программы.

```
program lab5_var100;Uses crt;

type arr=array[1..15] of integer;var a:arr;
                                {массив}

    i,j,n: byte; {индексы, длина массива}

    i_max: byte; {индекс максимального элемента массива} max:integer;
    {максимальный элемент}

begin

    writeln;

    writeln('        Автор – Иванов И.П., студент гр. ИС-11');

    writeln('        Вариант № 100');

    writeln(' Дан одномерный массив целых чисел. ');

    writeln(' Удалить все отрицательные элементы, расположенные справа от максимального ');

    writeln;

    {$R+}    {Включение контроля значений индексов} clrscr;
            {Очистка экрана}

    n:=15;

    randomize;

    writeln('Исходный массив:');

    for i:=1 to n do                {генерация элементов массива}begin

        a[i]:=-20+random(41);  {в диапазоне [-20; 20]}write(a[i]:4);

    end; writeln;
    max:=a[1];
    i_max:=1;

    for i:=2 to n do                {поиск максимального элемента массива}if a[i]>max then

        begin

            max:=a[i];
            i_max:=i;

        end;
        i:=i_max+1;

    while i<=n do                    {цикл для проверки элементов на знак }begin

        if a[i]<0 then                {если элемент отрицательный, то}begin
```

```

for j:=i to n-1 do {цикл удаления i-го элемента}a[j]:=a[j+1];

n:=n-1;           {уменьшение длины массива}

end else

i:=i+1;          {иначе переход к следующему элементу}

end;

writeln('Результат обработки:');for i:=1 to n do

write(a[i]:4);readln;

{$R-}           {Выключение контроля значений индексов}end.

```

Варианты заданий

1. Скорректировать массив $A=(a_1, a_2, \dots, a_n)$, переписав в начало массива группу, содержащую наибольшее число подряд идущих положительных элементов. Элементы массива вводить с клавиатуры.
2. В массиве $A=(a_1, a_2, \dots, a_n)$ все элементы, равные нулю, поставить сразу после максимального элемента данного массива. Элементы массива вводятся с клавиатуры.
3. В массиве $A=(a_1, a_2, \dots, a_n)$ все отрицательные элементы отправить в «хвост» массива.
4. В массиве $A=(a_1, a_2, \dots, a_n)$ удалить последнюю группу положительных элементов. Группой называется подряд идущие элементы одного знака, число которых больше или равно 2.
5. В массиве $A=(a_1, a_2, \dots, a_n)$ все положительные элементы, стоящие перед минимальным положительным элементом, переслать в «хвост» массива.
6. В массиве $A=(a_1, a_2, \dots, a_n)$ удалить все подряд идущие отрицательные элементы, идущие вслед за минимальным элементом массива.
7. В массиве $A=(a_1, a_2, \dots, a_n)$ удалить все отрицательные элементы, стоящие перед минимальным элементом массива.
8. В массиве $A=(a_1, a_2, \dots, a_n)$ удалить все элементы, меньшие, чем элемент массива, расположенный слева от максимального.
9. В массиве $A=(a_1, a_2, \dots, a_n)$ вставить новый элемент со значением P вслед за наибольшим из отрицательных элементов этого массива.
10. В массиве $A=(a_1, a_2, \dots, a_n)$ удалить все элементы, стоящие между минимальным положительным и максимальным отрицательным элементами.
11. В массиве $A=(a_1, a_2, \dots, a_n)$ удалить все положительные элементы, имеющие четный порядковый номер, идущие после минимального элемента массива.
12. В массиве $A=(a_1, a_2, \dots, a_n)$ все положительные элементы, начиная со второго положительного, отправить в «хвост» массива.
13. В одномерном массиве $A=(a_1, a_2, \dots, a_n)$ группу, содержащую наибольшее число равных элементов, заменить на максимальный элемент этого массива. После корректировки массив может содержать меньше элементов, чем прежде. Элементы массива вводить с клавиатуры.
14. В одномерном массиве $A=(a_1, a_2, \dots, a_n)$ группу элементов, содержащую наибольшее число подряд идущих отрицательных элементов, переписать в «хвост» массива. Элементы массива вводить с клавиатуры.

15. В одномерном массиве $A=(a_1, a_2, \dots, a_n)$ все отрицательные элементы, имеющие нечетный порядковый номер, отправить в «хвост» массива, т. е. поместить на место последних элементов.
16. В одномерном массиве $A=(a_1, a_2, \dots, a_n)$ все группы элементов, содержащие более 3-х подряд идущих отрицательных элементов, заменить на максимальный элемент. Элементы массива вводить с клавиатуры.
17. В одномерном массиве $A=(a_1, a_2, \dots, a_n)$ все положительные элементы, имеющие четный порядковый номер, переписать в начало массива.
18. В одномерном массиве $A=(a_1, a_2, \dots, a_n)$ группу, содержащую наибольшее число равных элементов, заменить на максимальный элемент этого массива. Элементы массива вводить с клавиатуры.
19. В одномерном массиве $A=(a_1, a_2, \dots, a_n)$ удалить все отрицательные элементы, расположенные между положительными.
20. В одномерном массиве $A=(a_1, a_2, \dots, a_n)$ исключить из массива группу с наибольшим числом подряд идущих положительных элементов. Элементы массива вводить с клавиатуры.
21. В одномерном массиве $A=(a_1, a_2, \dots, a_n)$ отрицательные элементы, имеющие четный порядковый номер, переписать в начало массива.
22. В одномерном массиве $A=(a_1, a_2, \dots, a_n)$ удалить все равные элементы, оставив только один из данных групп равных. Элементы массива вводить с клавиатуры.
23. В одномерном массиве $A=(a_1, a_2, \dots, a_n)$ группу из наибольшего числа подряд идущих нулей заменить на максимальный элемент массива. Элементы массива вводить с клавиатуры.
24. В одномерном массиве $A=(a_1, a_2, \dots, a_n)$ группу, содержащую наибольшее число подряд идущих положительных элементов, переписать в «хвост» массива.
25. В одномерном массиве $A=(a_1, a_2, \dots, a_n)$ все положительные элементы, расположенные между отрицательными, поставить после минимального элемента массива.

Критерии оценивания:

- 5 баллов выставляется, если лабораторная работа выполнена полностью, без погрешностей и замечаний.
- 4 балла выставляется студенту, если лабораторная работа полностью выполнена, есть погрешности и/или замечания.
- 3 балла выставляется студенту, если программный код не оптимален; использованы глобальные переменные; не на все вопросы получены верные ответы.
- 2 балла выставляется студенту, если лабораторная работа не выполнена, программа не работает или выдает неправильный ответ.

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

Процедура оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций состоит из текущего контроля.

Текущий контроль успеваемости проводится с использованием оценочных средств, представленных в п. 2 данного приложения. Результаты текущего контроля доводятся до сведения студентов до промежуточной аттестации и учитываются при оценивании знаний, умений, навыков и (или) опыта деятельности.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

ОП.04. «Основы алгоритмизации и программирования»

Методические указания для студентов по освоению дисциплины ОП.04. «Основы алгоритмизации и программирования» являются частью рабочей программы (РПД) (приложением к рабочей программе).

РПД – рабочая программа, утвержденная директором колледжа для изучения дисциплины ОП.04. «Основы алгоритмизации и программирования». Она определяет цели и задачи дисциплины, формируемые в ходе ее изучения компетенции и их компоненты, содержание изучаемого материала, виды занятий и объем выделяемого учебного времени, а также порядок изучения и преподавания дисциплины.

Для самостоятельной учебной работы студента важное значение имеют разделы «Структура и содержание дисциплины (модуля)» и «Учебно-методическое и информационное обеспечение дисциплины (модуля)». В первом указываются разделы и темы изучаемой дисциплины, а также виды занятий и планируемый объем (в академических часах), во втором – рекомендуемая литература и перечень ресурсов информационно-телекоммуникационной сети "Интернет".

Для подготовки к текущему контролю студенты могут воспользоваться оценочными средствами, представленными в Приложении 1 к рабочей программе дисциплины.

1. Описание последовательности действий студента

Приступая к изучению дисциплины ОП.04. «Основы алгоритмизации и программирования» необходимо в первую очередь ознакомиться содержанием РПД, где в разделе «Структура и содержание дисциплины (модуля)» приведено общее распределение часов аудиторных занятий и самостоятельной работы по темам дисциплины.

Залогом успешного освоения дисциплины является регулярное посещение занятий и выполнение предусмотренных программой заданий. Пропуск одного, а тем более нескольких занятий может осложнить освоение разделов курса.

Лекции имеют целью дать систематизированные основы научных знаний по содержанию дисциплины. При изучении и проработке теоретического материала необходимо:

- повторить законспектированный на лекционном занятии материал и дополнить его с учетом рекомендованной по данной теме литературы;
- при самостоятельном изучении теоретической темы подготовить конспект, используя рекомендованные в РПД литературные источники и электронные образовательные ресурсы.

Практические занятия и лабораторная работа проводятся с целью углубления и закрепления знаний, полученных на лекциях и в процессе самостоятельной работы с учебной литературой.

Выполнение обучающимися практических заданий направлено на:

- обобщение, систематизацию, углубление, закрепление полученных теоретических знаний по конкретным темам дисциплины;
- формирование умений применять полученные знания на практике, реализацию единства интеллектуальной и практической деятельности;

- выработку при решении поставленных задач таких профессионально значимых качеств, как самостоятельность, ответственность, точность, творческая инициатива.

При подготовке к практическому занятию необходимо изучить или повторить лекционный материал по соответствующей теме.

2. Самостоятельная работа студента

Самостоятельная работа студента – самостоятельная учебная деятельность студента, организуемая колледжем и осуществляемая без непосредственного руководства педагога, но по его заданиям и под его контролем.

Цели самостоятельной работы:

- систематизация и закрепление полученных теоретических знаний и практических умений студентов;
- углубление и расширение теоретических знаний;
- формирование умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развитие познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности; формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- воспитание самостоятельности, как личностного качества будущего специалиста.

Самостоятельная работа студента по дисциплины выполняется:

- самостоятельно вне расписания учебных занятий;
- с использованием современных образовательных технологий;
- работа со специальной литературой для подготовки к тестовым, практическим и лабораторным заданиям.

3. Рекомендации по работе с литературой и источниками

Работу с литературой следует начинать с анализа РПД, содержащей список основной и дополнительной литературы, а также знакомства с учебно-методическими разработками.

В случае возникновения затруднений в понимании учебного материала следует обратиться к другим источникам, где изложение может оказаться более доступным.

Работа с литературой не только полезна как средство более глубокого изучения дисциплины, но и является неотъемлемой частью профессиональной деятельности будущего выпускника.