

Министерство науки и высшего образования Российской Федерации
 Документ подписан простым электронным способом
 Информация о владельце: Федеральное государственное бюджетное образовательное учреждение
 ФИО: Игнатенко Виталий Иванович высшего образования
 Должность: Проректор по образовательной деятельности и молодежной политике
 Дата подписания: 24.06.2025 20:19:40 «Заполярный государственный университет им. Н.М. Федоровского»
 Уникальный программный ключ: (ЗГУ)
 a49ae343af5448d45d7e3e1e499659da8109ba78

УТВЕРЖДАЮ
 Проректор по ОД и МП
 _____ Игнатенко В.И.

Программирование и алгоритмизация

рабочая программа дисциплины (модуля)

Закреплена за кафедрой **Информационные системы и технологии**

Учебный план 15.03.04_бак_оч-заоч_АП-2025+.plx
 Направление подготовки: Автоматизация технологических процессов и производств

Квалификация **бакалавр**

Форма обучения **очно-заочная**

Общая трудоемкость **3 ЗЕТ**

Часов по учебному плану 108
 в том числе:
 аудиторные занятия 16
 самостоятельная работа 74
 часов на контроль 18

Виды контроля в семестрах:
 зачеты 4

Распределение часов дисциплины по семестрам

Семестр (<Курс>.<Семестр на курсе>)	4 (2.2)		Итого	
Неделя	16			
Вид занятий	УП	РП	УП	РП
Лекции	8	8	8	8
Практические	8	10	8	10
Итого ауд.	16	18	16	18
Контактная работа	16	18	16	18
Сам. работа	74	86	74	86
Часы на контроль	18	4	18	4
Итого	108	108	108	108

Программу составил(и):

к.т.н. Доцент А.А. Попкова _____

Рабочая программа дисциплины

Программирование и алгоритмизация

разработана в соответствии с ФГОС:

Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 15.03.04 Автоматизация технологических процессов и производств (приказ Минобрнауки России от 09.08.2021 г. № 730)

Рабочая программа одобрена на заседании кафедры

Информационные системы и технологии

Протокол от 28.03.2025г. № 6

Срок действия программы: уч.г.

Зав. кафедрой к.э.н., доцент Беляев И.С.

Визирование РПД для исполнения в очередном учебном году

к.э.н., доцент Беляев И.С. _____ 2026 г.

Рабочая программа пересмотрена, обсуждена и одобрена для
исполнения в 2026-2027 учебном году на заседании кафедры
Информационные системы и технологии

Протокол от _____ 2026 г. № ____
Зав. кафедрой к.э.н., доцент Беляев И.С.

Визирование РПД для исполнения в очередном учебном году

к.э.н., доцент Беляев И.С. _____ 2027 г.

Рабочая программа пересмотрена, обсуждена и одобрена для
исполнения в 2027-2028 учебном году на заседании кафедры
Информационные системы и технологии

Протокол от _____ 2027 г. № ____
Зав. кафедрой к.э.н., доцент Беляев И.С.

Визирование РПД для исполнения в очередном учебном году

к.э.н., доцент Беляев И.С. _____ 2028 г.

Рабочая программа пересмотрена, обсуждена и одобрена для
исполнения в 2028-2029 учебном году на заседании кафедры
Информационные системы и технологии

Протокол от _____ 2028 г. № ____
Зав. кафедрой к.э.н., доцент Беляев И.С.

Визирование РПД для исполнения в очередном учебном году

к.э.н., доцент Беляев И.С. _____ 2029 г.

Рабочая программа пересмотрена, обсуждена и одобрена для
исполнения в 2029-2030 учебном году на заседании кафедры
Информационные системы и технологии

Протокол от _____ 2029 г. № ____
Зав. кафедрой к.э.н., доцент Беляев И.С.

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ	
1.1	Цель: формирование у будущих бакалавров практических навыков по информатике и программированию решения различных задач, по основам алгоритмизации вычислительных процессов, развитие умения работы с персональным компьютером на высоком пользовательском уровне; создание необходимой основы для использования современных средств вычислительной техники и пакетов прикладных программ при изучении студентами естественнонаучных, общепрофессиональных и специальных дисциплин в течение всего периода обучения
1.2	Задачи: освоение предусмотренного программой теоретического материала и приобретение практических навыков использования программных и технических средств
1.3	изучение методов программирования для овладения знаниями в области технологии программирования
1.4	подготовка к осознанному использованию как языков программирования, так и методов программирования

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП	
Цикл (раздел) ООП:	Б1.О
2.1 Требования к предварительной подготовке обучающегося:	
2.1.1	Математика
2.1.2	Компьютерное моделирование, часть 1
2.1.3	Информационные технологии
2.1.4	Аналитическая геометрия и линейная алгебра
2.1.5	Математический анализ
2.1.6	Офисные информационные технологии
2.1.7	Информационные технологии
2.1.8	Компьютерное моделирование, часть 1
2.1.9	Аналитическая геометрия и линейная алгебра
2.1.10	Математический анализ
2.1.11	Офисные информационные технологии
2.1.12	Информационные технологии
2.2 Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:	
2.2.1	Автоматизация управления жизненным циклом продукции
2.2.2	Компьютерное моделирование, часть 2
2.2.3	Проектирование автоматизированных систем
2.2.4	CASE средства при проектировании систем управления
2.2.5	Автоматизация технологических процессов и производств
2.2.6	Автоматизация управления жизненным циклом продукции
2.2.7	Компьютерное моделирование, часть 2
2.2.8	Проектирование автоматизированных систем
2.2.9	CASE средства при проектировании систем управления
2.2.10	Автоматизация технологических процессов и производств

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)	
ОПК-14.1: Способен разрабатывать алгоритмы, пригодные для практического применения	
Знать:	
Уметь:	
Владеть:	
ОПК-14.2: Способен разрабатывать компьютерные программы, пригодные для практического применения	
Знать:	
Уметь:	
Владеть:	
ОПК-6.1: Использует основные положения информационной и библиографической культуры, информационно-коммуникационные технологии, применяемые для решения стандартных задач профессиональной деятельности	
Знать:	

Уметь:
Владеть:
ОПК-6.2: Использует информационно-коммуникационные технологии для решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры
Знать:
Уметь:
Владеть:
ОПК-6.3: Обладает навыками применения информационно-коммуникационных технологий при решении типовых задач профессиональной деятельности на основе информационной и библиографической культуры
Знать:
Уметь:
Владеть:
ОПК-4.1: Владеет современными информационными технологиями и основными программными продуктами, использует для моделирования технологических процессов
Знать:
Уметь:
Владеть:
ОПК-4.2: Применяет современные информационные технологии и программные средства при моделировании технологических процессов и решении других инженерно-технических задач в профессиональной сфере
Знать:
Уметь:
Владеть:
ОПК-4.3: Владеет навыками использования информационных технологий, программных средств для моделирования технологических процессов, а так же решения других инженерно-технических задач в профессиональной сфере
Знать:
Уметь:
Владеть:

В результате освоения дисциплины обучающийся должен

3.1 Знать:	
3.1.1	основные источники информации; основы применения источников информации; основные устройства ввода и вывода информации, принципы хранения цифровой информации; понимать роль ИТ и ИС в развитии и становлении грамотного и саморазвивающегося общества; перспективы развития ИТ и ИС; основы понятия информатики (данные, информация, знания, информационные процессы, информационные системы и технологии), алгоритмизации и программирования (понятия, свойства и методы программирования), особенности функционирования операционных, сервисных и сетевых программных и технических средств (форматы данных, конвертация и сжатие данных, безопасность информации); физические основы элементной базы компьютерной техники и средств передачи информации; международные и отечественные стандарты в области информационных систем и технологий; основные компоненты и законы аппарата булевой алгебры
3.2 Уметь:	
3.2.1	формулировать цели и функции задачи и находить пути её решения; использовать основные устройства формирования, ввода и вывода текстовой и графической информации; презентовать результаты учебных и научных достижений; выполнять анализ прикладной задачи на логическом и алгоритмическом уровне, конструировать сложные алгоритмы с применением аппарата математической логики; документировать процессы создания информационных систем на стадиях жизненного цикла; применять вычислительные средства, эффективные алгоритмы и современные инструментальные средства, поддерживающие производительные технологии программирования, при решении профессиональных и проблемно-ориентированных задач
3.3 Владеть:	
3.3.1	культурой мышления и анализа тенденций развития ИТ и ИС; навыками выбора необходимого устройства ввода и вывода графической и текстовой информации, навыками использования текстовых редакторов, электронных таблиц, браузеров; навыками кодирования данных в основных системах счисления, использования пакетов прикладных программ и технических средств последнего поколения для решения профессиональных задач; различными технологиями программирования для реализации эффективных алгоритмов инженерных задач

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)							
Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Компетенции	Литература	Инте ракт.	Примечание
	Раздел 1.						
1.1	ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ ЛИНЕЙНОЙ, ВЕТВЯЩЕЙСЯ и ЦИКЛИЧЕСКОЙ СТРУКТУРЫ /Лек/	4	5		Л1.2 Л1.5 Л1.6Л2.4 Л2.7 Э1	0	
1.2	Составлении блок-схемы алгоритма линейной, ветвящейся и циклической структуры /Пр/	4	5		Л1.2 Л1.5 Л1.6Л2.1 Л2.7 Э1	0	
1.3	Составлении блок-схемы алгоритмов разветвляющейся и циклической структуры /Ср/	4	0		Л1.5 Л1.7Л2.1 Л2.6	0	
1.4	ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ ОБРАБОТКИ МАССИВОВ /Лек/	4	1		Л1.3 Л1.4 Л1.5 Л1.6Л2.4 Л2.7	0	
1.5	Разработка, программирование вычислительного процесса обработки массивов /Пр/	4	1		Л1.2 Л1.5 Л1.7Л2.1 Л2.3 Л2.6	0	
1.6	Составлении блок-схемы алгоритма и разработка программ обработки массивов /Ср/	4	0		Л1.2 Л1.5	0	
1.7	ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ С ИСПОЛЬЗОВАНИЕМ ПОДПРОГРАММ /Лек/	4	2		Л1.1 Л1.3 Л1.5 Л1.6Л2.7	0	
1.8	Разработка и программирования алгоритмов с использованием подпрограмм, приобретение дальнейших навыков по отладке и тестированию программ. /Пр/	4	2		Л1.5 Л1.7Л2.2 Л2.4	0	
1.9	Составлении блок-схемы алгоритма итерационной циклической структуры /Ср/	4	16		Л1.4 Л1.5Л2.1	0	
1.10	Овладение практическими навыками работы с массивами, особенностями их ввода и вывода, приобретение дальнейших навыков по организации программ циклической структуры с использованием приемов программирования. /Ср/	4	18		Л1.2 Л1.5Л2.2 Л2.5 Л2.6	0	
1.11	Разработка алгоритмов с использованием классов и объектов /Пр/	4	2		Л1.5 Л1.6 Л1.7Л2.4	0	
1.12	Составлении блок-схемы алгоритма обработки двумерных массивов /Ср/	4	18		Л1.2 Л1.5Л2.5	0	
1.13	Овладение навыками алгоритмизации и программирования задач и использованием подпрограмм пользователя различных видов, овладение навыками написания подпрограмм и обращения к ним, выбора параметров подпрограмм.	4	12		Л1.2 Л1.4 Л1.5Л2.7	0	
1.14	Изучение принципа модульного программирования /Ср/	4	22		Л1.3 Л1.5Л2.1 Л2.5	0	
1.15	/Зачёт/	4	4			0	

5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

5.1. Контрольные вопросы и задания

КОНТРОЛЬНЫЕ ВОПРОСЫ (1 семестр зачет с оценкой)

Тема 1

1. Какие типы величин используются в языке программирования?
2. Указать диапазон значений величин целого и действительного типов.
3. Какие имена переменных допустимы в программе? Как задать тип переменной в программе?
4. Указать имена стандартных функций для вычисления e^x , $\sin x$, $\cos x$, $\ln x$, $|x|$.
5. Можно ли в качестве операнда в арифметическом выражении использовать а) имя массива; б) имя стандартной функции, например $\sin(Y)$; в) имя символьной переменной или переменной логического типа?
6. Назвать последовательность действий при выполнении арифметического оператора присваивания. Допустимо ли использование величин разных типов в арифметическом выражении?
7. Написать арифметический оператор присваивания для вычисления значения.
8. Указать старшинство выполнения операций при вычислении арифметического выражения.
9. Указать средства, имеющиеся в языке программирования для управления размещением данных на строке. Как организовать вывод значений, сопровождая выводимое числовое значение наименованием переменной? Как организовать пропуск одной, двух строк при выводе?
10. Как выбрать значения исходных данных для тестового варианта счета?

Тема 2

1. Какие служебные слова могут использоваться при записи условного оператора?
2. Какой тип должно иметь выражение, стоящее в условном операторе между словами if и then?
3. Перечислить действия, реализуемые при выполнении условного оператора.
4. Какие действия выполняются оператором перехода?
5. Что такое вычислительный процесс разветвляющейся структуры?
6. Как организовать разветвление вычислений: а) на две ветви; б) на три ветви?
7. Составить последовательность операторов для вычисления величины $z = 0$, если $x < -2$; $z = 1$, если $-2 \leq x \leq 2$; $z = -1$, если $x > 2$.
8. Зачем необходимо при отладке программы тестировать все ветви алгоритма?
9. Пусть дана следующая последовательность операторов:
If <выражение> then <оператор 1> else <оператор 2>; <оператор 3>; установите:
а) значение <выражения>, при котором выполняется <оператор 3>;
б) возможность записи после слова then нескольких операторов;
в) условие выполнения оператора <оператор 1>;
г) значение выражения, при котором выполняется <оператор 2>;
д) может ли <выражение> содержать знаки операций сравнения;
е) может ли <выражение> содержать знаки логических операций;
ж) могут ли быть операторы If вложенными.
10. Приведите пример составного оператора.
11. Как называются программы, в основе которых лежит структура следования?
12. Какие операторы входят в состав разветвляющихся программ?

Тема 3

1. Указать отличия в организации циклов с заданным числом повторений от операторов цикла с предусловием и постусловием.
2. Какие средства языка целесообразно использовать для организации циклов с заданным числом повторений?
3. Почему при программировании формулы трапеций и прямоугольников индексированные переменные x_i и $f(x_i)$ можно заменить простыми переменными?
4. Почему начальное значение суммы по формуле трапеции принимается не равное нулю, а по формуле прямоугольников – равное нулю?
5. Указать, какие операторы составляют тело цикла.
6. В чем состоят преимущества использования операторов цикла в программах?
7. Когда проверяется истинность выражения в цикле с предусловием?
8. Сколько раз выполняется оператор в цикле с предусловием, если с самого начала значение выражения равно false?
9. Сколько операторов можно записать между ключевыми словами цикла с постусловием?
10. Какой тип должен иметь параметр цикла в цикле с заданным числом повторений?
11. Чему равно значение параметра цикла for после завершения цикла?
12. Допустим ли выход из тела цикла for?
13. Допустим ли выход из тела цикла while?
14. Допустим ли выход из тела цикла repeat?
15. Какие циклы называются вложенными?

Тема 4

1. Что такое итерационный циклический процесс? Его отличия от цикла с заданным числом повторений.
2. Какие два этапа необходимо выделить при нахождении корней уравнений?
3. В чем заключается сущность метода итераций при уточнении корня? как определить число итераций, необходимых для получения значения корня с требуемой точностью?
4. Каковы условия сходимости методы итераций?
5. Почему при программировании итерационных процессов не используются индексированные переменные для обозначения последовательных приближений к корню? Сколько соседних приближений одновременно используется в вычислениях?

6. Каково условие выхода из цикла при вычислении значения суммы бесконечного ряда?
7. Какие операторы организуют цикл в программе вычисления суммы членов бесконечного ряда?
8. Почему при вычислении значения текущего члена a_n используется простая переменная, а не индексированная?
9. Зачем используются рекуррентные соотношения для вычисления значений члена ряда?

Тема 5

1. Верно ли, что массив – это структурированный тип данных, состоящий из фиксированного числа элементов, имеющих один и тот же тип?
2. Дайте определение индекса. Какие типы данных можно использовать при описании индекса?
3. Может ли левая граница индексов массива быть меньше правой?
4. Верно ли, что название регулярный тип массивы получили за то, что в них объединены логически однородные элементы, упорядоченные по индексам, определяющим положение каждого элемента в массиве?
5. Что называется базовым типом?
6. Какой массив называется одномерным?
7. Верно ли, что одномерный массив соответствует понятию линейной таблицы (вектора)?
8. Существуют ли ограничения на размерность массива?
9. Всегда ли работа с массивом сводится к работе с его компонентами?
10. Можно ли с помощью стандартной процедуры `write(x)` вывести весь массив целиком?
11. Алгоритм «быстрой» сортировки.
12. Алгоритм сортировки «вставкой».
13. Алгоритм сортировки «выбором».
14. Алгоритм сортировки «обменом».
15. Алгоритм «двоичного поиска».

Тема 6

1. Двумерный массив, индексация элементов двумерного массива, объявление, заполнение, печать двумерных массивов.
2. Квадратная матрица, главная и побочная диагонали квадратной матрицы.
3. Индексацией элементов, расположенных над, под, на главной и побочной диагоналями.
4. Основные свойства квадратной матрицы.
5. Верно ли, что двумерный массив соответствует понятию прямоугольной таблицы (матрице, набору векторов)?
6. Существуют ли ограничения на размерность массива?
7. Можно ли в описании массива использовать предварительно определенные константы?
8. Верно ли, что двумерные массивы располагаются в памяти таким образом, что второй индекс изменяется быстрее, чем первый?
9. Верно ли, что матрица расположена в памяти по строкам?
10. Всегда ли работа с матрицей сводится к работе с ее компонентами?
11. Предложите способы вывода элементов двумерного массива.
12. Указать основные правила организации вложенных циклов.
13. Указать способы выхода из внутреннего цикла.
14. Как организовать вывод матрицы в общепринятом виде?
15. Как организовать вывод нижней треугольной матрицы в общепринятом виде?
16. Как организовать ввод матрицы размером $N \times M$ элементов?

Тема 7

1. Дайте определение подпрограммы. Существуют ли подпрограммы без параметров?
2. Указать, при каких условиях целесообразно использование подпрограмм, какие выгоды они предоставляют пользователю.
3. Указать, в чем отличие различных видов подпрограмм пользователя.
4. Указать способы обращения к подпрограммам пользователя.
5. Указать способы передачи параметров в подпрограмму.
6. Указать, как организовать подпрограмму без параметров.
7. Перечислить, как согласуются формальные и фактические параметры.
8. Указать конструкции, которые могут быть формальными и фактическими параметрами.
9. Пояснить, как и куда осуществляется выход из подпрограммы.
10. Какое количество значений возвращает функция?
11. Какие переменные называются локальными?
12. Чем локальные переменные отличаются от глобальных?
13. Может ли в основной программе функция вызываться внутри выражения?
14. Может ли элемент массива быть формальным параметром?
15. Может ли быть функция или процедура параметром подпрограммы?

Тема 8

1. В чем состоит основной принцип модульного программирования?
2. Что обеспечивает принцип скрытия информации?
3. В каком разделе программы указывается подключение модулей?
4. Какие существуют особенности при разработке собственных модулей?
5. В чем состоит отличие сферы действия переменных в модуле и процедуре?
6. Из каких разделов состоит модуль?
7. Что описывается в разделе `interface`?
8. Что описывается в разделе `implementation`?
9. Что описывается в разделе инициализация?
10. Компиляция и сохранение модуля.

КОНТРОЛЬНЫЕ ВОПРОСЫ (2 семестр экзамен)**Тема 1**

1. Что называется файлом?
2. В чем различие между структурой логического и структурой физического файла?
3. В чем состоит сходство и различие между массивом и файлом?
4. По каким признакам классифицируются файлы в Pascal?
5. Что необходимо выполнить для открытия файла?
6. Какие процедуры предназначены для открытия файлов и как они работают?
7. Для чего предназначена процедура Close?
8. Какие процедуры предназначены для работы с каталогами?
9. Какие действия выполняют процедуры Rename и Erase?
10. Каких типов допускаются описания типизированных файлов?
11. Как нумеруются элементы типизированных файлов?
12. По каким правилам выполняется чтение из типизированных файлов?
13. Какой формат имеет процедура Write для типизированных файлов?
14. Какие процедуры и функции предназначены для прямого доступа к элементам типизированного файла?
15. В чем состоят особенности текстовых файлов?
16. В чем отличие пользовательских текстовых файлов от файлов Input и Output?
17. В чем отличие текстового файла от file of Char?
18. Как объявляется нетипизированный файл?
19. В чем отличие нетипизированных файлов от типизированных?
20. Какие процедуры и функции используются для работы с нетипизированными файлами?

Тема 2.

1. Является ли тип string структурированным типом данных? Обоснуйте свой ответ.
2. Что представляет собой тип string: а) последовательность английских букв; б) последовательность русских букв; в) последовательность символов.
3. Какое максимальное возможное количество символов может содержать строка?
4. Каково функциональное назначение типа string?
5. Как при описании строкового типа указывается длина строки?
6. Может ли пользователь задать длину строки? Если да, то каким образом.
7. Всегда ли фактическая длина строки равна объявленной в описании?
8. Перечислите типовые операции над строками.
9. Каким образом в переменную типа string можно поместить какое-то конкретное значение?
10. Может ли в процессе выполнения программы изменяться фактическая длина строки?
11. Может ли в процессе выполнения программы фактическая длина строки стать больше, чем объявлено в описании? Что произойдет в этом случае?
12. Если длина строки не указана при объявлении типа string, то чему она равна по умолчанию?
13. Могут ли переменные строкового типа быть операндами выражений?
14. Верно ли, что тип string аналогичен одномерному массиву символов array[0..n] of char?
15. Чем отличается тип string от одномерного массива символов array[0..n] of char?
16. Верно ли, что к любому символу в строке можно обратиться точно так же, как к элементу одномерного массива?
17. Какие операции отношения можно применять к переменным типа string? Каков их приоритет?
18. Как происходит сравнение строк: а) одинаковой длины; б) разной длины?
19. Пусть s – переменная типа string. Что является результатом действия стандартных операторов read(s), write(s), readln(s), writeln(s)?
20. Перечислите основные процедуры и функции для обработки строк.

Тема 3.

1. Верно ли, что запись относится к структурированным типам данных?
2. Дайте определение поля записи.
3. Верно ли, что поля записи должны иметь один и тот же тип?
4. Как осуществляется ссылка на компоненты записи?
5. Может ли компонент записи быть помечен более чем одним именем?
6. Может ли типом поля записи быть массив?
7. Могут ли записи иметь вложенную структуру?
8. Каков максимально допустимый уровень вложенности записей?
9. Могут ли во вложенных записях имена полей повторяться на разных уровнях вложенности?
10. Верно ли, что значения переменных типа запись можно присваивать переменным того же типа?
11. Могут ли совпадать имена полей различных записей?
12. Могут ли совпадать имена одной и той же записи?
13. Как можно оценить объем памяти, необходимой для хранения записи?
14. Можно ли значения полей записи использовать в выражениях?
15. Дайте определение составного имени. Из каких частей оно состоит и как записывается?
16. Чем запись отличается от массива?
17. Какой оператор используется для упрощения доступа к полям записи?
18. Когда применяется запись с вариантами?
19. Из каких частей состоит запись с вариантами?
20. Сколько вариантных частей может быть в любой записи с вариантами?

21. Верно ли, что вариантная часть расположена за всеми фиксированными полями?
22. Что такое поле признака? Для чего оно используется в записи с вариантами?
23. Как описываются компоненты каждого варианта записи?
24. Сколько вариантов может иметь вариантная часть записи?
25. Как можно оценить объем памяти, выделяемой для записи с вариантами?
26. Почему в вариантной части записи не ставится слово `end`, закрывающее оператор `case...of`?
27. Верно ли, что ключ выбора должен быть стандартным или предварительно объявленным порядковым типом?
28. Как влияет ключ выбора на количество следующих ниже вариантных полей?
29. Можно ли ключу выбора в исполняемой части программы присвоить некоторое значение и таким образом повлиять на выбор полей?
30. Пусть `adres` – переменная типа запись, состоящая из нескольких полей. Можно ли инициализировать значение этой переменной с помощью стандартной процедуры ввода `read(adres)`?
31. Верно ли, что все поля записи должны быть разных типов?
32. Почему при описании записи ее пол могут перечисляться в любом порядке?
33. Верно ли, что названия полей записи могут совпадать с именами переменных, констант и других объектов программы, но не могут совпадать с названиями полей других записей?
34. Почему при обращении к полю записи имя поля должно указываться явно и не может быть задано в виде выражения?

Тема 4.

1. Верно ли, что множество – это набор объектов, сгруппированных по какому-либо признаку или группе признаков?
2. Дайте определение элемента множества.
3. Что называется базовым типом множества?
4. Может ли базовый тип множества быть вещественным?
5. В каких пределах можно изменять количество элементов, входящих во множество?
6. Может ли множество не содержать ни одного элемента?
7. Как обозначается пустое множество?
8. Какой объем памяти занимает один элемент множества?
9. Каково внутреннее представление множества?
10. Может ли множество содержать элементы различных типов?
11. Может ли множество содержать несколько одинаковых элементов?
12. Что называется мощностью множества?
13. Какова мощность множества, базовый тип которого `boolean`?
14. Может ли тип диапазон быть типом множества?
15. Могут ли два множества содержать одинаковые элементы?
16. Верно ли, что два множества эквивалентны тогда и только тогда, когда все их элементы одинаковы, причем порядок следования элементов во множестве не имеет значения?
17. Какие операции допустимы над множествами?
18. Как работает операция `in`?
19. Какие множества считают равными; неравными? Имеет ли значение для сравниваемых множеств порядок следования элементов?
20. Каково назначение операции «больше или равно», «меньше или равно», примененных к данным множественного типа?
21. Что называется объединением множеств?
22. Что называется пересечением множеств?
23. Что называется разностью множеств?
24. Верно ли, что спецификациями элементов конструктора множеств могут быть константы или выражения базового типа, а также тип диапазон того же базового типа?
25. Может ли конструктор множества стоять в правой части оператора присваивания?
26. Можно ли конструктор множества использовать в выражении?

Тема 5 - 8

1. Какие данные относят к данным динамической структуры?
2. Какие существуют разновидности связанных динамических данных?
3. В чем особенность объявления данных динамической структуры?
4. Что выполняет операция разыменования?
5. С помощью каких процедур происходит распределение памяти под динамические переменные?
6. Какие состояния может принимать указательная переменная?
7. В каких случаях указатель может находиться в неопределенном состоянии?
8. В чем различие между состоянием `nil` и неопределенным состоянием?
9. Какие действия выполняют процедуры `New` и `Dispose`?
10. Какие действия выполняют процедуры `Mark` и `Release`?
11. Какие действия выполняют процедуры `Getmem` и `Freemem`?
12. Какие действия выполняют функции `MemAvail` и `MaxAvail`?
13. В чем выражаются динамические свойства несвязанных динамических данных?
14. В чем сходство и различие между линейными и кольцевыми списками?
15. Что представляет собой очередь как структура данных?
16. Что представляет собой стек как структура данных?
17. Что требуется для создания связанных динамических структур данных?
18. В чем состоит особенность описания типов для создания динамических структур данных?
19. Сколько указателей требуется для работы с очередью?
20. Какие действия необходимо выполнить для создания очереди?

21. Как добавить, или удалить элемент очереди?
22. Сколько указателей требуется для работы со стеком?
23. Какие действия необходимо выполнить для создания стека?
24. Как добавить, или удалить элемент стека?
25. При решении, каких прикладных задач удобно представлять набор объектов в виде двоичных деревьев?

5.2. Темы письменных работ

Эссе (20 тем)

1. Роль программирования в современном мире: от науки до повседневной жизни.
2. Этические аспекты использования алгоритмов в программировании.
3. Преимущества и ограничения различных языков программирования.
4. Будущее программирования: тренды и прогнозы развития.
5. Как искусственный интеллект меняет подходы к программированию?
6. Сравнение процедурного и объектно-ориентированного программирования.
7. Влияние алгоритмов на эффективность программного обеспечения.
8. Как программирование помогает решать реальные задачи бизнеса?
9. Роль алгоритмов сортировки и поиска в разработке программ.
10. Безопасность данных в контексте программирования и алгоритмизации.
11. Как блокчейн-технологии используются в программировании?
12. Программирование как инструмент автоматизации бизнес-процессов.
13. Проблемы цифрового неравенства в контексте обучения программированию.
14. Как персонализация влияет на разработку алгоритмов?
15. Роль аналитики данных в создании эффективных алгоритмов.
16. Как программирование поддерживает глобализацию технологий?
17. Проблемы правового регулирования использования алгоритмов.
18. Как программирование помогает малому бизнесу развиваться?
19. Роль алгоритмов в управлении большими данными (Big Data).
20. Этические вопросы использования алгоритмов в социальных сетях.

Рефераты (20 тем)

21. История развития программирования и его влияние на современные технологии.
22. Основные парадигмы программирования: процедурное, объектно-ориентированное, функциональное.
23. Принципы работы рекурсивных алгоритмов и их применение.
24. Принципы создания и управления базами данных через программирование.
25. Как работает система электронных коммуникаций в программировании?
26. Алгоритмы машинного обучения и их реализация в программировании.
27. Принципы работы систем лояльности через алгоритмы.
28. Как используются облачные технологии в программировании?
29. Принципы работы систем управления заказами через программирование.
30. Роль маркетинга в продвижении программных продуктов.
31. Принципы работы систем управления складскими запасами через программирование.
32. Методы защиты персональных данных в программировании.
33. Принципы работы систем управления цепями поставок через алгоритмы.
34. Как работают системы анализа пользовательского поведения через программирование?
35. Принципы работы систем управления финансами в бизнесе через программирование.
36. Принципы работы систем управления рисками через алгоритмы.
37. Как используются технологии IoT в программировании?
38. Принципы работы систем управления документооборотом через программирование.
39. Принципы работы систем управления качеством продукции через алгоритмы.
40. Принципы работы систем управления маркетинговыми кампаниями через программирование.

Курсовые работы (20 тем)

41. Разработка программы для анализа данных с использованием алгоритмов сортировки.
42. Создание системы рекомендаций для клиентов интернет-магазина на основе алгоритмов.
43. Проектирование системы управления рисками через алгоритмы.
44. Разработка программы для анализа больших данных (Big Data) через алгоритмы.
45. Создание системы управления складскими запасами через алгоритмы.
46. Разработка системы электронных платежей для компании с использованием алгоритмов.
47. Проектирование системы управления лояльностью клиентов через алгоритмы.
48. Создание системы управления маркетинговыми кампаниями через алгоритмы.
49. Разработка системы управления цепями поставок через алгоритмы.
50. Проектирование системы управления рисками в бизнесе через алгоритмы.
51. Создание системы управления финансами для компании через алгоритмы.
52. Разработка системы управления документооборотом через алгоритмы.
53. Проектирование системы управления качеством продукции через алгоритмы.
54. Создание системы управления портфелем инвестиций через алгоритмы.
55. Разработка системы управления экологическими показателями через алгоритмы.

56. Проектирование системы управления фермерским хозяйством через алгоритмы.
57. Создание системы управления медицинскими данными через алгоритмы.
58. Разработка системы управления учебным процессом онлайн через алгоритмы.
59. Проектирование системы управления задачами для удаленных команд через алгоритмы.
60. Создание системы управления энергопотреблением здания через алгоритмы.

5.3. Фонд оценочных средств

1. Текущий контроль знаний

Тестовые вопросы

1. Что такое алгоритм? Приведите примеры его применения.
2. Какие основные парадигмы программирования существуют?
3. В чем заключается принцип работы рекурсии? Приведите пример.
4. Как работает алгоритм сортировки пузырьком? Опишите его сложность.
5. Что такое структуры данных? Перечислите основные типы.
6. Как реализуется работа со стеком и очередью в программировании?
7. Что такое объектно-ориентированное программирование? Приведите пример.
8. Как выполняется отладка программного кода?
9. Что такое асимптотическая сложность алгоритма? Приведите пример.
10. Какие методы защиты данных используются в программировании?

Практические задания

11. Напишите программу для вычисления факториала числа с использованием рекурсии.
12. Реализуйте алгоритм сортировки массива методом выбора.
13. Создайте программу для поиска элемента в массиве с использованием бинарного поиска.
14. Напишите программу, демонстрирующую работу со стеком.
15. Разработайте программу для обработки строковых данных (например, подсчет количества слов).

Открытые вопросы

16. Как искусственный интеллект меняет подходы к разработке алгоритмов?
17. Какие ошибки чаще всего допускаются при написании программного кода?
18. Как программирование помогает в снижении затрат на бизнес?
19. Какие факторы следует учитывать при выборе языка программирования?
20. Как программирование поддерживает удаленную работу сотрудников?

2. Промежуточная аттестация

Тестовые вопросы

21. Что такое функциональное программирование? Где оно применяется?
22. Какие методы защиты данных используются в алгоритмах шифрования?
23. В чем заключается концепция "serverless" (безсерверных вычислений) в программировании?
24. Как программирование поддерживает удаленную работу сотрудников?
25. Что такое KPI? Для чего они используются в разработке программного обеспечения?

Практические задания

26. Разработайте систему KPI для отдела разработки с использованием алгоритмов.
27. Создайте план управления рисками для проекта через алгоритмы.
28. Реализуйте систему уведомлений для команды проекта на основе алгоритмов.
29. Настройте интеграцию облачного хранилища с системой анализа данных.
30. Разработайте план миграции данных в облако для анализа.

Ситуационные задачи

31. Компания хочет автоматизировать процесс обработки данных. Как вы будете строить модель алгоритма?
32. У вас есть набор данных о поведении пользователей. Как вы будете их анализировать с помощью алгоритмов?
33. Вам нужно разработать систему для прогнозирования рыночных трендов. Какие алгоритмы вы выберете?
34. Вы работаете над проектом управления рисками. Как алгоритмы могут быть применены в этой задаче?
35. Компания хочет автоматизировать процесс анализа данных. Как вы будете использовать программирование для решения этой задачи?

3. Итоговый контроль

Защита курсового проекта

36. Разработка программы для анализа данных с использованием алгоритмов сортировки.
37. Проектирование системы управления рисками через алгоритмы.
38. Создание системы управления складскими запасами через алгоритмы.
39. Разработка системы рекомендаций для клиентов интернет-магазина.
40. Проектирование системы управления качеством продукции через алгоритмы.

Экзамен

41. Теоретический блок: тестовые вопросы по основным темам курса.
42. Практический блок: выполнение задания на написание программы или алгоритма.
43. Анализ предоставленного кода и исправление ошибок.

Оценка портфолио

44. Сборник выполненных лабораторных работ.
45. Документация по разработанным проектам.
46. Отчеты по практическим заданиям.

4. Дополнительные виды оценочных средств
 47. Устное собеседование по ключевым темам курса.
 48. Анализ кода и исправление ошибок в предоставленном проекте.
 49. Выполнение заданий на оптимизацию производительности алгоритмов.
 50. Разработка прототипа системы управления процессом с использованием алгоритмов.

5.4. Перечень видов оценочных средств

1. Текущий контроль знаний:
- Тесты на знание основ программирования и алгоритмизации (типы данных, структуры данных, парадигмы программирования).
 - Практические задания по написанию программ для решения базовых задач.
 - Написание скриптов для обработки данных с использованием алгоритмов.
 - Лабораторные работы по реализации алгоритмов сортировки, поиска и работы со структурами данных.
2. Промежуточная аттестация:
- Выполнение контрольной работы (например, разработка программы для анализа данных или реализация алгоритма).
 - Рефераты или эссе на заданные темы.
 - Мини-проект (например, разработка системы управления задачами или рекомендательного алгоритма).
3. Итоговый контроль:
- Защита курсового проекта (разработка полноценной программы или системы с использованием алгоритмов).
 - Экзамен в виде теста или выполнения практического задания.
 - Оценка портфолио выполненных лабораторных и практических работ.
4. Дополнительные виды оценочных средств:
- Устное собеседование по ключевым темам курса.
 - Анализ кода и исправление ошибок в предоставленном проекте.
 - Выполнение заданий на оптимизацию производительности алгоритмов или программ.

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

6.1. Рекомендуемая литература

6.1.1. Основная литература

	Авторы, составители	Заглавие, размещение	Издательство, год	Колич-во
Л1.1	Круценюк К.Ю.	Численные методы решения уравнений (Excel, Turbo Pascal, Delphi): учеб. пособие	Нори́льск: НИИ, 2010	51
Л1.2	Фаронов В. В.	Turbo Pascal. 7.0: учеб. пособие	М.: Кнорус, 2013	1
Л1.3	Иопа Н. И.	Информатика (для технических направлений): рекомендовано УМО вузов по университетскому политехн. образованию в качестве учеб. пособия	М.: Кнорус, 2012	1
Л1.4	Иванова Г. С.	Программирование: допущено УМО по универс. политехн. образованию в качестве учебника для студентов вузов	М.: Кнорус, 2014	10
Л1.5	Круценюк К.Ю.	Информатика: лабораторный практикум	Нори́льск: НИИ, 2015	48
Л1.6	Парфилова Н.И., Пылькин А.Н., Трусов Б.Г.	Программирование. Основы алгоритмизации и программирования: рекомендовано ФГБОУ ВПО "Московский гос. техн. ун-т им. Н.Э. Баумана" в качестве учебника для студентов вузов	М.: Академия, 2014	15
Л1.7	Парфилова Н.И., Пылькин А.Н., Трусов Б.Г.	Программирование. Структурирование программ и данных: рекомендовано ФГБОУ ВПО "Московский гос. техн. ун-т им. Н.Э. Баумана" в качестве учебника для студентов вузов	М.: Академия, 2014	3

6.1.2. Дополнительная литература

	Авторы, составители	Заглавие, размещение	Издательство, год	Колич-во
Л2.1	Степанов А.Н.	Информатика. Базовый курс: допущено М-вом образования и науки РВ в качестве учеб. пособия для студентов гуманитарных специальностей вузов	СПб.: Питер, 2010	1
Л2.2	Немнюгин С.А.	Turbo Pascal. Программирование на языке высокого уровня: учебник для вузов	СПб.: Питер, 2007	5

	Авторы, составители	Заглавие, размещение	Издательство, год	Колич-во
Л2.3	Безручко В. Т.	Практикум по курсу "Информатика". Работа в Windows XP, Word 2003, Excel 2003, PowerPoint 2003, Outlook 2003, PROMN Famiky 7.0, Интернет: учеб. пособие для вузов	М.: ФОРУМ-ИНФРА-М, 2008	1
Л2.4	под ред. С.В. Симоновича	Информатика. Базовый курс: рекомендовано М-вом образования и науки РФ в качестве учеб. пособия для студентов вузов	СПб.: Питер, 2011	1
Л2.5	Культин Н.Б.	Программирование в TURBO Pascal 7.0 и Delphi	СПб., 1999	3
Л2.6	Немнюгин С.А., Перколаб Л. В.	Изучаем Turbo Pascal	СПб.: Питер, 2007	1
Л2.7	под ред. Б.Г. Трусова	Информатика и программирование. Основы информатики: рекомендовано ФГБУ "Московский гос. технический ун-т им. Н. Баумана" в качестве учебника для студентов вузов	М.: Академия, 2012	30

6.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"

Э1	Электронный каталог НГИИ http://biblio.norvuz.ru
----	--

6.3.1 Перечень программного обеспечения

6.3.1.1	MS Windows 7 (Номер лицензии 62693665 от 19.11.2013)
6.3.1.2	MS Office Standard 2013 (Номер лицензии 62693665 от 19.11.2013)
6.3.1.3	MS Access 2013 (Номер лицензии 63765822 от 30.06.2014)
6.3.1.4	Mathlab R2010b (Номер лицензии 622090 от 23.12.2009)
6.3.1.5	Lazarus
6.3.1.6	Free Pascal
6.3.1.7	Microsoft Visual Studio 2010 (версия для образовательных учреждений)
6.3.1.8	Pascal ABC.NET

6.3.2 Перечень информационных справочных систем

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

7.1	Аудитории для проведения учебных занятий, предусмотренных программой магистратуры представляют собой помещения, оснащенные оборудованием и техническими средствами обучения, состав которых определяется в рабочих программах дисциплин (модулей).
7.2	Для проведения лекционных занятий предоставляются аудитории, оснащенные специализированной мебелью, демонстрационным оборудованием (проектор, экран, компьютер/ноутбук) с соответствующим программным обеспечением, меловой (и) или маркерной доской.
7.3	Для проведения занятий семинарского типа (семинары, практические занятия) предоставляются аудитории, оснащенные специализированной мебелью, демонстрационным оборудованием (проектор, экран, компьютер/ноутбук) с соответствующим программным обеспечением, меловой (и) или маркерной доской.
7.4	Для проведения групповых (индивидуальных) консультаций предоставляется аудитория, оснащенная специализированной мебелью, меловой (и) или маркерной доской.
7.5	Для проведения мероприятий текущего контроля и промежуточной аттестации - аудитория, оснащенная специализированной мебелью, демонстрационным оборудованием (проектор, экран компьютер/ноутбук) с соответствующим программным обеспечением, меловой (и) или маркерной доской.
7.6	Для проведения практических занятий (лабораторных работ) задействованы специализированные учебные помещения, оснащенные оборудованием:
7.7	
7.8	209 - аудитория для проведения лекционных, практических, семинарских и интерактивных занятий, самостоятельной работы. Мультимедийный класс. Компьютерный класс. (посадочных мест – 45)
7.9	1 проектор Panasonic PT-LB60NTE
7.10	MS Windows XP (Номер лицензии 62693665 от 19.11.2013)
7.11	MS Access 2007 (Номер лицензии 62693665 от 19.11.2013)
7.12	MS Office Standard 2007 (Номер лицензии 62693665 от 19.11.2013)
7.13	CorelDraw Graphics Suite X5 (Номер лицензии 4069593 от 28.07.2010)
7.14	
7.15	403 - аудитория для проведения лекционных, практических, семинарских, интерактивных занятий (мультимедийный класс) (посадочных мест – 22)

7.16	11 компьютеров (Intel Core 2 Duo E6550 2.33GHz, 3Гб ОЗУ, HDD 160 Гб) 1 компьютер (Intel Core i3-2120 3.30GHz, 1Гб ОЗУ, HDD 250 Гб), интерактивная доска iRU, 1 проектор NEC UM361x
7.17	Лицензионное ПО
7.18	MS Windows 7 (Номер лицензии 62693665 от 19.11.2013)
7.19	MS Office Standard 2013 (Номер лицензии 62693665 от 19.11.2013)
7.20	MS Access 2013 (Номер лицензии 63765822 от 30.06.2014)
7.21	RMeasiteach Next Generation (Номер лицензии 1SV-367)
7.22	Бесплатное ПО
7.23	Консультант Плюс (версия для образовательных учреждений)
7.24	Microsoft Visual Studio 2010 (версия для образовательных учреждений)
7.25	Free Pascal
7.26	Pascal ABC.NET
7.27	
7.28	Ауд. 407 - аудитория для проведения лекционных, практических, семинарских занятий, самостоятельной работы. Мультимедийный класс. Компьютерный класс (посадочных мест – 26)
7.29	12 компьютеров (Intel Pentium(R) G850 2.90GHz, 2Гб ОЗУ, HDD 320 Гб), Epson-eb-l255f
7.30	Лицензионное ПО
7.31	MS Windows 7 (Номер лицензии 62693665 от 19.11.2013)
7.32	MS Office Standard 2013 (Номер лицензии 62693665 от 19.11.2013)
7.33	MS Access 2013 (Номер лицензии 63765822 от 30.06.2014)
7.34	Бесплатное ПО
7.35	Microsoft Visual Studio 2010 (версия для образовательных учреждений)
7.36	Lazarus
7.37	Pascal ABC.NET
7.38	Консультант Плюс (версия для образовательных учреждений)
7.39	
7.40	Ауд. 408 - аудитория для проведения лекционных, практических, семинарских занятий, самостоятельной работы. Мультимедийный класс. Компьютерный класс (посадочных мест - 20)
7.41	10 компьютеров (Intel Pentium(R) G3420 3.20GHz, 2Гб ОЗУ, HDD 500 Гб),
7.42	1 Моноблок Shvacher (Платформа Lenovo) QuadCore Intel Core i3-10100T, 3700 MHz (37 x 100) Intel(R) UHD Graphics 630 (1 Гб) 8Гб ОЗУ, SDD 250 Гб
7.43	HDD 1000 Гб,
7.44	1 проектор Panasonic pt-f300vg4
7.45	Лицензионное ПО
7.46	MS Windows 7 (Номер лицензии 62693665 от 19.11.2013)
7.47	MS Office Standard 2007 (Номер лицензии 62693665 от 19.11.2013)
7.48	Mathlab R2010b (Номер лицензии 622090 от 23.12.2009)
7.49	MathCAD 15 (Заказ №2564794 от 25.02.2010)
7.50	MS Office Standard 2013
7.51	Бесплатное ПО
7.52	1С: Предприятие (учебная версия)
7.53	Консультант Плюс (версия для образовательных учреждений)
7.54	AnyLogic Personal Learning Edition
7.55	Microsoft Visual Studio 2010 (версия для образовательных учреждений)
7.56	PascalABC.Net
7.57	Blender
7.58	
7.59	Ауд. 412 - аудитория для проведения лекционных, практических, семинарских и интерактивных занятий, самостоятельной работы. Мультимедийный класс. Компьютерный класс. (посадочных мест - 20)
7.60	10 компьютеров (Intel Pentium(R) G850 2.90GHz, 2Гб ОЗУ, HDD 320 Гб),
7.61	1 Моноблок Shvacher (Платформа Lenovo) QuadCore Intel Core i3-10100T, 3700 MHz (37 x 100) Intel(R) UHD Graphics 630 (1 Гб) 8Гб ОЗУ, SDD 250 Гб
7.62	HDD 1000 Гб,

7.63	1 проектор Epson eb-455wi
7.64	Лицензионное ПО
7.65	MS Windows 7 (Номер лицензии 62693665 от 19.11.2013)
7.66	MS Office Standard 2013 (Номер лицензии 62693665 от 19.11.2013)
7.67	Microsoft Windows 10 Pro
7.68	Бесплатное ПО
7.69	AnyLogic Personal Learning Edition
7.70	Консультант Плюс (версия для образовательных учреждений)
7.71	PascalABC.Net
7.72	Lazarus
7.73	Blender
7.74	
7.75	Ауд. 211 - аудитория для проведения лекционных, практических, семинарских и интерактивных занятий, самостоятельной работы. Мультимедийный класс. Компьютерный класс. (посадочных мест - 18)
7.76	10 компьютеров (Intel Pentium G2120 3.10GHz, 2Гб ОЗУ, HDD 500 Гб)
7.77	Лицензионное ПО
7.78	MS Windows 7 (Номер лицензии 62693665 от 19.11.2013)
7.79	MS Office Standard 2007 (Номер лицензии 62693665 от 19.11.2013)
7.80	MS Access 2007 (Номер лицензии 62693665 от 19.11.2013)
7.81	MathCAD 15 (Заказ №2564794 от 25.02.2010)
7.82	ABBY FineReader 10 (Номер лицензии 94965 от 25.08.2010)
7.83	Mathlab R2010b (Номер лицензии 622090 от 23.12.2009)
7.84	Бесплатное ПО
7.85	1С: Предприятие (учебная версия)
7.86	ArchiCAD 15 (версия для образовательных учреждений)
7.87	Blender
7.88	Консультант Плюс (версия для образовательных учреждений)
7.89	Inkscape
7.90	
7.91	Ауд. 503 - аудитория для проведения лекционных, практических, семинарских и интерактивных занятий. Мультимедийный класс (посадочных мест - 33)
7.92	9 компьютеров (Intel Core 2 Duo E7200 2.53GHz, 3Гб ОЗУ, HDD 320 Гб), 1 проектор acer p1265
7.93	Лицензионное ПО
7.94	MS Windows XP (Номер лицензии 62693665 от 19.11.2013)
7.95	MS Office Standard 2007 (Номер лицензии 62693665 от 19.11.2013)
7.96	MS Access 2007 (Номер лицензии 62693665 от 19.11.2013)
7.97	Mathlab R2010b (Номер лицензии 622090 от 23.12.2009)
7.98	Компас-3D v12 (Номер лицензионного соглашения Кк-10-01126)
7.99	Бесплатное ПО
7.100	Консультант Плюс (версия для образовательных учреждений)
7.101	AutoCAD Education 2012 (версия для образовательных учреждений)
7.102	
7.103	
7.104	
7.105	
7.106	
7.107	
7.108	
7.109	
7.110	
7.111	

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

Планирование и организация времени, необходимого для изучения дисциплины

Важным условием успешного освоения дисциплины является создание системы правильной организации труда, позволяющей распределить учебную нагрузку равномерно в соответствии с графиком образовательного процесса. Большую помощь в этом может оказать составление плана работы на семестр, месяц, неделю, день. Его наличие позволит подчинить свободное время целям учебы, трудиться более успешно и эффективно. С вечера всегда надо распределять работу на завтрашний день. В конце каждого дня целесообразно подвести итог работы: тщательно проверить, все ли выполнено по намеченному плану, не было ли каких-либо отступлений, а если были, по какой причине они произошли. Нужно осуществлять самоконтроль, который является необходимым условием успешной учебы. Если что-то осталось невыполненным, необходимо изыскать время для завершения этой части работы, не уменьшая объема недельного плана. Все задания к лабораторным работам, а также задания, вынесенные на самостоятельную работу, рекомендуется выполнять непосредственно после соответствующей темы лекционного курса, что способствует лучшему усвоению материала, позволяет своевременно выявить и устранить «пробелы» в знаниях, систематизировать ранее пройденный материал, на его основе приступить к овладению новыми знаниями и навыками.

Система обучения основывается на рациональном сочетании нескольких видов учебных занятий (в первую очередь, лекций и лабораторных), работа над которыми обладает определенной спецификой.

Подготовка к лекциям

Знакомство с дисциплиной происходит уже на первой лекции, где от студента требуется не просто внимание, но и самостоятельное оформление конспекта. При работе с конспектом лекций необходимо учитывать тот фактор, что одни лекции дают ответы на конкретные вопросы темы, другие – лишь выявляют взаимосвязи между явлениями, помогая студенту понять глубинные процессы развития изучаемого предмета как в истории, так и в настоящее время.

Конспектирование лекций – сложный вид вузовской аудиторной работы, предполагающий интенсивную умственную деятельность студента. Конспект является полезным тогда, когда записано самое существенное и сделано это самим обучающимся. Не надо стремиться записать дословно всю лекцию. Такое «конспектирование» приносит больше вреда, чем пользы. Целесообразно вначале понять основную мысль, излагаемую лектором, а затем записать ее. Желательно записывать на одной странице листа или оставляя поля, на которых позднее, при самостоятельной работе с конспектом, можно сделать дополнительные записи, отметить непонятные места.

Конспект лекции лучше подразделять на пункты, соблюдая красную строку. Этому в большой степени будут способствовать вопросы плана лекции, предложенные преподавателям. Следует обращать внимание на акценты, выводы, которые делает лектор, отмечая наиболее важные моменты в лекционном материале замечаниями «важно», «хорошо запомнить» и т.п.

Можно делать это и с помощью разноцветных маркеров или ручек, подчеркивая термины и определения.

Целесообразно разработать собственную систему сокращений, аббревиатур и символов. Однако при дальнейшей работе с конспектом символы лучше заменить обычными словами для быстрого зрительного восприятия текста.

Работая над конспектом лекций, всегда необходимо использовать не только учебник, но и ту литературу, которую дополнительно рекомендовал лектор. Именно такая серьезная, кропотливая работа с лекционным материалом позволит глубоко овладеть теоретическим материалом.

Подготовка к лабораторным работам

Подготовку к каждому занятию студент должен начать с ознакомления с планом занятия, который отражает содержание предложенной темы. Тщательное продумывание и изучение вопросов плана основывается на проработке текущего материала лекции, а затем изучения обязательной и дополнительной литературы, рекомендованной к данной теме. Все новые понятия по изучаемой теме необходимо выучить наизусть и внести в глоссарий, который целесообразно вести с самого начала изучения курса.

Результат такой работы должен проявиться в способности студента свободно ответить на теоретические вопросы практикума, его выступлении и участии в коллективном обсуждении вопросов изучаемой темы, правильном выполнении лабораторных работ и самостоятельных работ.

В процессе подготовки к занятиям, студентам необходимо обратить особое внимание на самостоятельное изучение рекомендованной литературы. При всей полноте конспектирования лекции в ней невозможно изложить весь материал из-за лимита аудиторных часов. Поэтому самостоятельная работа с учебниками, учебными пособиями, научной, справочной литературой, материалами периодических изданий и Интернета является наиболее эффективным методом получения дополнительных знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у студентов свое отношение к конкретной проблеме

Подготовка к промежуточной аттестации

При подготовке к промежуточной аттестации целесообразно:

- внимательно изучить перечень вопросов и определить, в каких источниках находятся сведения, необходимые для ответа на них;
- внимательно прочитать рекомендованную литературу;
- составить краткие конспекты ответов (планы ответов).

Методические указания для преподавателей

Рекомендуемые средства, методы обучения, способы учебной деятельности, применение которых для освоения конкретных модулей рабочей учебной программы наиболее эффективно:

– обучение теоретическому материалу рекомендуется основывать на основной и дополнительной литературе, изданных типографским или электронным способом конспектах лекций; рекомендуется в начале семестра ознакомить студентов с программой дисциплины, перечнем теоретических вопросов для текущего промежуточного и итогового контроля знаний, что ориентирует и поощряет студентов к активной самостоятельной работе;

- рекомендуется проводить лекционные занятия с использованием мультимедийной техники (проектора). На первом занятии до студентов должны быть доведены требования к освоению разделов дисциплины, правила выполнения и сдачи лабораторной работы, индивидуального задания (проверочной работы) (ИЗ/ПР), перечень рекомендуемой литературы. Желательно провести обзор тем, которые будут изучены в течение семестра с тем, чтобы студенты более осознанно подходили к выполнению самостоятельной работы и выполнения ИЗ/ПР. Также часть занятий проводятся в активной и интерактивной форме.

Учебный процесс, опирающийся на использование интерактивных методов обучения, организуется с учетом включенности в процесс познания всех студентов группы без исключения. Совместная деятельность означает, что каждый вносит свой особый индивидуальный вклад, в ходе работы идет обмен знаниями, идеями, способами деятельности. Организуются индивидуальная, парная и групповая работа, используется проектная работа, ролевые игры, осуществляется работа с документами и различными источниками информации и т.д.

Интерактивные методы основаны на принципах взаимодействия, активности обучаемых, опоре на групповой опыт, обязательной обратной связи. Создается среда образовательного общения, которая характеризуется открытостью, взаимодействием участников, равенством их аргументов, накоплением совместного знания, возможностью взаимной оценки и контроля.

**Министерство науки и высшего образования РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Заполярный государственный университет им. Н. М. Федоровского»**

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по дисциплине
Программирование и алгоритмизация**

Уровень образования: бакалавриат

Кафедра «Информационные системы и технологии»

Разработчик ФОС:

к.т.н., Доцент, А.А. Попкова

А.А. Попкова

Оценочные материалы по дисциплине рассмотрены и одобрены на заседании кафедры, протокол № 6 от 28.03.2025 г.

Заведующий кафедрой _____ к.э.н., Беляев И.С.

Фонд оценочных средств по дисциплине Программирование и алгоритмизация для текущей/ промежуточной аттестации разработан в соответствии с Федеральным государственным образовательным стандартом высшего образования по специальности / направлению подготовки 15.03.04 Автоматизация технологических процессов и производств на основе Рабочей программы дисциплины Программирование и алгоритмизация, утвержденной решением ученого совета от 28.03.2025 г., Положения о формировании Фонда оценочных средств по дисциплине (ФОС), Положения о проведении текущего контроля успеваемости и промежуточной аттестации обучающихся ЗГУ, Положения о государственной итоговой аттестации (ГИА) выпускников по образовательным программам высшего образования в ЗГУ им. Н.М. Федоровского.

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами образовательной программы

Таблица 1. Компетенции и индикаторы их достижения

Код и наименование компетенции	Индикаторы достижения
ОПК-4 Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности;	ОПК-4.1 Владеет современными информационными технологиями и основными программными продуктами, использует для моделирования технологических процессов
	ОПК-4.2 Применяет современные информационные технологии и программные средства при моделировании технологических процессов и решении других инженерно-технических задач в профессиональной сфере
	ОПК-4.3 Владеет навыками использования информационных технологий, программных средств для моделирования технологических процессов, а так же решения других инженерно-технических задач в профессиональной сфере
ОПК-6 Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий;	ОПК-6.1 Использует основные положения информационной и библиографической культуры, информационно-коммуникационные технологии, применяемые для решения стандартных задач профессиональной деятельности

ОПК-6 Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий;	ОПК-6.2	Использует информационно-коммуникационные технологии для решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры
	ОПК-6.3	Обладает навыками применения информационно-коммуникационных технологий при решении типовых задач профессиональной деятельности на основе информационной и библиографической культуры
ОПК-14 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения.	ОПК-14.1	Способен разрабатывать алгоритмы , пригодные для практического применения
	ОПК-14.2	Способен разрабатывать компьютерные программы, пригодные для практического применения

Таблица 2. Паспорт фонда оценочных средств

№ п/п	Контролируемые разделы (темы) дисциплины	Код результата обучения по дисциплине/ модулю	Оценочные средства текущей аттестации		Оценочные средства промежуточной аттестации	
			Наименовани е	Форма	Наименовани е	Форма
4 семестр						

2. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующие процесс формирования компетенций в ходе освоения образовательной программы.

2.1. Задания для текущего контроля успеваемости

КОНТРОЛЬНЫЕ ВОПРОСЫ (1 семестр зачет с оценкой)

Тема 1

1. Какие типы величин используются в языке программирования?
2. Указать диапазон значений величин целого и действительного типов.
3. Какие имена переменных допустимы в программе? Как задать тип переменной в программе?
4. Указать имена стандартных функций для вычисления e^x , $\sin x$, $\cos x$, $\ln x$, $|x|$.
5. Можно ли в качестве операнда в арифметическом выражении использовать а) имя массива; б) имя стандартной функции, например $\sin(Y)$; в) имя символьной переменной или переменной логического типа?
6. Назвать последовательность действий при выполнении арифметического оператора присваивания. Допустимо ли использование величин разных типов в арифметическом выражении?
7. Написать арифметический оператор присваивания для вычисления значения.

8. Указать старшинство выполнения операций при вычислении арифметического выражения.

9. Указать средства, имеющиеся в языке программирования для управления размещением данных на строке. Как организовать вывод значений, сопровождая выводимое числовое значение наименованием переменной? Как организовать пропуск одной, двух строк при выводе?

10. Как выбрать значения исходных данных для тестового варианта счета?

Тема 2

1. Какие служебные слова могут использоваться при записи условного оператора?

2. Какой тип должно иметь выражение, стоящее в условном операторе между словами if и then?

3. Перечислить действия, реализуемые при выполнении условного оператора.

4. Какие действия выполняются оператором перехода?

5. Что такое вычислительный процесс разветвляющейся структуры?

6. Как организовать разветвление вычислений: а) на две ветви; б) на три ветви?

7. Составить последовательность операторов для вычисления величины $z = 0$, если $x < -2$; $z = 1$; если $-2 \leq x \leq 2$; $z = -1$, если $x > 2$.

8. Зачем необходимо при отладке программы тестировать все ветви алгоритма?

9. Пусть дана следующая последовательность операторов:

If <выражение> then <оператор 1> else <оператор 2>; <оператор 3>; установите:

а) значение <выражения>, при котором выполняется <оператор 3>;

б) возможность записи после слова then нескольких операторов;

в) условие выполнения оператора <оператор 1>;

г) значение выражения, при котором выполняется <оператор 2>;

д) может ли <выражение> содержать знаки операций сравнения;

е) может ли <выражение> содержать знаки логических операций;

ж) могут ли быть операторы If вложенными.

10. Приведите пример составного оператора.

11. Как называются программы, в основе которых лежит структура следования?

12. Какие операторы входят в состав разветвляющихся программ?

Тема 3

1. Указать отличия в организации циклов с заданным числом повторений от операторов цикла с предусловием и постусловием.

2. Какие средства языка целесообразно использовать для организации циклов с заданным числом повторений?

3. Почему при программировании формулы трапеций и прямоугольников индексированные переменные x_i и $f(x_i)$ можно заменить простыми переменными?

4. Почему начальное значение суммы по формуле трапеции принимается не равное нулю, а по формуле прямоугольников – равное нулю?

5. Указать, какие операторы составляют тело цикла.

6. В чем состоят преимущества использования операторов цикла в программах?

7. Когда проверяется истинность выражения в цикле с предусловием?

8. Сколько раз выполняется оператор в цикле с предусловием, если с самого начала значение выражения равно false?

9. Сколько операторов можно записать между ключевыми словами цикла с постусловием?

10. Какой тип должен иметь параметр цикла в цикле с заданным числом повторений?

11. Чему равно значение параметра цикла for после завершения цикла?

12. Допустим ли выход из тела цикла for?

13. Допустим ли выход из тела цикла while?

14. Допустим ли выход из тела цикла repeat?

15. Какие циклы называются вложенными?

Тема 4

1. Что такое итерационный циклический процесс? Его отличия от цикла с заданным числом повторений.

2. Какие два этапа необходимо выделить при нахождении корней уравнений?

3. В чем заключается сущность метода итераций при уточнении корня? как определить число итераций, необходимых для получения значения корня с требуемой точностью?

4. Каковы условия сходимости методы итераций?

5. Почему при программировании итерационных процессов не используются индексированные переменные для обозначения последовательных приближений к корню? Сколько соседних приближений одновременно используется в вычислениях?

6. Каково условие выхода из цикла при вычислении значения суммы бесконечного ряда?

7. Какие операторы организуют цикл в программе вычисления суммы членов бесконечного ряда?

8. Почему при вычислении значения текущего члена a_n используется простая переменная, а не индексированная?

9. Зачем используются рекуррентные соотношения для вычисления значений члена ряда?

Тема 5

1. Верно ли, что массив – это структурированный тип данных, состоящий из фиксированного числа элементов, имеющих один и тот же тип?

2. Дайте определение индекса. Какие типы данных можно использовать при описании индекса?

3. Может ли левая граница индексов массива быть меньше правой?

4. Верно ли, что название регулярный тип массивы получили за то, что в них объединены логически однородные элементы, упорядоченные по индексам, определяющим положение каждого элемента в массиве?

5. Что называется базовым типом?

6. Какой массив называется одномерным?

7. Верно ли, что одномерный массив соответствует понятию линейной таблицы (вектора)?

8. Существуют ли ограничения на размерность массива?

9. Всегда ли работа с массивом сводится к работе с его компонентами?

10. Можно ли с помощью стандартной процедуры write(x) вывести весь массив целиком?

11. Алгоритм «быстрой» сортировки.

12. Алгоритм сортировки «вставкой».

13. Алгоритм сортировки «выбором».

14. Алгоритм сортировки «обменом».

15. Алгоритм «двоичного поиска».

Тема 6

1. Двумерный массив, индексация элементов двумерного массива, объявление, заполнение, печать двумерных массивов.

2. Квадратная матрица, главная и побочная диагонали квадратной матрицы.

3. Индексацией элементов, расположенных над, под, на главной и побочной диагоналями.

4. Основные свойства квадратной матрицы.

5. Верно ли, что двумерный массив соответствует понятию прямоугольной таблицы (матрице, набору векторов)?

6. Существуют ли ограничения на размерность массива?

7. Можно ли в описании массива использовать предварительно определенные константы?
8. Верно ли, что двумерные массивы располагаются в памяти таким образом, что второй индекс изменяется быстрее, чем первый?
9. Верно ли, что матрица расположена в памяти по строкам?
10. Всегда ли работа с матрицей сводится к работе с ее компонентами?
11. Предложите способы вывода элементов двумерного массива.
12. Указать основные правила организации вложенных циклов.
13. Указать способы выхода из внутреннего цикла.
14. Как организовать вывод матрицы в общепринятом виде?
15. Как организовать вывод нижней треугольной матрицы в общепринятом виде?
16. Как организовать ввод матрицы размером $N \times M$ элементов?

Тема 7

1. Дайте определение подпрограммы. Существуют ли подпрограммы без параметров?
2. Указать, при каких условиях целесообразно использование подпрограмм, какие выгоды они предоставляют пользователю.
3. Указать, в чем отличие различных видов подпрограмм пользователя.
4. Указать способы обращения к подпрограммам пользователя.
5. Указать способы передачи параметров в подпрограмму.
6. Указать, как организовать подпрограмму без параметров.
7. Перечислить, как согласуются формальные и фактические параметры.
8. Указать конструкции, которые могут быть формальными и фактическими параметрами.
9. Пояснить, как и куда осуществляется выход из подпрограммы.
10. Какое количество значений возвращает функция?
11. Какие переменные называются локальными?
12. Чем локальные переменные отличаются от глобальных?
13. Может ли в основной программе функция вызываться внутри выражения?
14. Может ли элемент массива быть формальным параметром?
15. Может ли быть функция или процедура параметром подпрограммы?

Тема 8

1. В чем состоит основной принцип модульного программирования?
2. Что обеспечивает принцип скрытия информации?
3. В каком разделе программы указывается подключение модулей?
4. Какие существуют особенности при разработке собственных модулей?
5. В чем состоит отличие сферы действия переменных в модуле и процедуре?
6. Из каких разделов состоит модуль?
7. Что описывается в разделе interface?
8. Что описывается в разделе implementation?
9. Что описывается в разделе инициализация?
10. Компиляция и сохранение модуля.

КОНТРОЛЬНЫЕ ВОПРОСЫ (2 семестр экзамен)

Тема 1

1. Что называется файлом?
2. В чем различие между структурой логического и структурой физического файла?
3. В чем состоит сходство и различие между массивом и файлом?
4. По каким признакам классифицируются файлы в Pascal?
5. Что необходимо выполнить для открытия файла?
6. Какие процедуры предназначены для открытия файлов и как они работают?
7. Для чего предназначена процедура Close?

8. Какие процедуры предназначены для работы с каталогами?
9. Какие действия выполняют процедуры Rename и Erase?
10. Каких типов допускаются описания типизированных файлов?
11. Как нумеруются элементы типизированных файлов?
12. По каким правилам выполняется чтение из типизированных файлов?
13. Какой формат имеет процедура Write для типизированных файлов?
14. Какие процедуры и функции предназначены для прямого доступа к элементам типизированного файла?
15. В чем состоят особенности текстовых файлов?
16. В чем отличие пользовательских текстовых файлов от файлов Input и Output?
17. В чем отличие текстового файла от file of Char?
18. Как объявляется нетипизированный файл?
19. В чем отличие нетипизированных файлов от типизированных?
20. Какие процедуры и функции используются для работы с нетипизированными файлами?

Тема 2.

1. Является ли тип string структурированным типом данных? Обоснуйте свой ответ.
 2. Что представляет собой тип string: а) последовательность английских букв; б) последовательность русских букв; в) последовательность символов.
 3. Какое максимальное возможное количество символов может содержать строка?
 4. Каково функциональное назначение типа string?
 5. Как при описании строкового типа указывается длина строки?
 6. Может ли пользователь задать длину строки? Если да, то каким образом.
 7. Всегда ли фактическая длина строки равна объявленной в описании?
 8. Перечислить типовые операции над строками.
 9. Каким образом в переменную типа string можно поместить какое-то конкретное значение?
 10. Может ли в процессе выполнения программы изменяться фактическая длина строки?
 11. Может ли в процессе выполнения программы фактическая длина строки стать больше, чем объявлено в описании? Что произойдет в этом случае?
 12. Если длина строки не указана при объявлении типа string, то чему она равна по умолчанию?
 13. Могут ли переменные строкового типа быть операндами выражений?
 14. Верно ли, что тип string аналогичен одномерному массиву символов array[0..n] of char?
 15. Чем отличается тип string от одномерного массива символов array[0..n] of char?
 16. Верно ли, что к любому символу в строке можно обратиться точно так же, как к элементу одномерного массива?
 17. Какие операции отношения можно применять к переменным типа string? Каков их приоритет?
 18. Как происходит сравнение строк: а) одинаковой длины; б) разной длины?
 19. Пусть s – переменная типа string. Что является результатом действия стандартных операторов read(s), write(s), readln(s), writeln(s)?
 20. Перечислите основные процедуры и функции для обработки строк.
- #### Тема 3.
1. Верно ли, что запись относится к структурированным типам данных?
 2. Дайте определение поля записи.
 3. Верно ли, что поля записи должны иметь один и тот же тип?
 4. Как осуществляется ссылка на компоненты записи?

5. Может ли компонент записи быть помечен более чем одним именем?
 6. Может ли типом поля записи быть массив?
 7. Могут ли записи иметь вложенную структуру?
 8. Каков максимально допустимый уровень вложенности записей?
 9. Могут ли во вложенных записях имена полей повторяться на разных уровнях вложенности?
 10. Верно ли, что значения переменных типа запись можно присваивать переменным того же типа?
 11. Могут ли совпадать имена полей различных записей?
 12. Могут ли совпадать имена одной и той же записи?
 13. Как можно оценить объем памяти, необходимой для хранения записи?
 14. Можно ли значения полей записи использовать в выражениях?
 15. Дайте определение составного имени. Из каких частей оно состоит и как записывается?
 16. Чем запись отличается от массива?
 17. Какой оператор используется для упрощения доступа к полям записи?
 18. Когда применяется запись с вариантами?
 19. Из каких частей состоит запись с вариантами?
 20. Сколько вариантных частей может быть в любой записи с вариантами?
 21. Верно ли, что вариантная часть расположена за всеми фиксированными полями?
 22. Что такое поле признака? Для чего оно используется в записи с вариантами?
 23. Как описываются компоненты каждого варианта записи?
 24. Сколько вариантов может иметь вариантная часть записи?
 25. Как можно оценить объем памяти, выделяемой для записи с вариантами?
 26. Почему в вариантной части записи не ставится слово `end`, закрывающее оператор `case...of`?
 27. Верно ли, что ключ выбора должен быть стандартным или предварительно объявленным порядковым типом?
 28. Как влияет ключ выбора на количество следующих ниже вариантных полей?
 29. Можно ли ключу выбора в исполняемой части программы присвоить некоторое значение и таким образом повлиять на выбор полей?
 30. Пусть `adres` – переменная типа запись, состоящая из нескольких полей. Можно ли инициализировать значение этой переменной с помощью стандартной процедуры ввода `read(adres)`?
 31. Верно ли, что все поля записи должны быть разных типов?
 32. Почему при описании записи ее пол могут перечисляться в любом порядке?
 33. Верно ли, что названия полей записи могут совпадать с именами переменных, констант и других объектов программы, но не могут совпадать с названиями полей других записей?
 34. Почему при обращении к полю записи имя поля должно указываться явно и не может быть задано в виде выражения?

Тема 4.

1. Верно ли, что множество – это набор объектов, сгруппированных по какому-либо признаку или группе признаков?
2. Дайте определение элемента множества.
3. Что называется базовым типом множества?
4. Может ли базовый тип множества быть вещественным?
5. В каких пределах можно изменять количество элементов, входящих во множество?
6. Может ли множество не содержать ни одного элемента?
7. Как обозначается пустое множество?
8. Какой объем памяти занимает один элемент множества?

9. Каково внутреннее представление множества?

10. Может ли множество содержать элементы различных типов?

11. Может ли множество содержать несколько одинаковых элементов?

12. Что называется мощностью множества?

13. Какова мощность множества, базовый тип которого boolean?

14. Может ли тип диапазон быть типом множества?

15. Могут ли два множества содержать одинаковые элементы?

16. Верно ли, что два множества эквивалентны тогда и только тогда, когда все их элементы одинаковы, причем порядок следования элементов во множестве не имеет значения?

17. Какие операции допустимы над множествами?

18. Как работает операция in?

19. Какие множества считают равными; неравными? Имеет ли значение для сравниваемых множеств порядок следования элементов?

20. Каково назначение операции «больше или равно», «меньше или равно», примененных к данным множественного типа?

21. Что называется объединением множеств?

22. Что называется пересечением множеств?

23. Что называется разностью множеств?

24. Верно ли, что спецификациями элементов конструктора множеств могут быть константы или выражения базового типа, а также тип диапазон того же базового типа?

25. Может ли конструктор множества стоять в правой части оператора присваивания?

26. Можно ли конструктор множества использовать в выражении?

Тема 5 - 8

1. Какие данные относят к данным динамической структуры?

2. Какие существуют разновидности связанных динамических данных?

3. В чем особенность объявления данных динамической структуры?

4. Что выполняет операция разыменования?

5. С помощью каких процедур происходит распределение памяти под динамические переменные?

6. Какие состояния может принять указательная переменная?

7. В каких случаях указатель может находиться в неопределенном состоянии?

8. В чем различие между состоянием nil и неопределенным состоянием?

9. Какие действия выполняют процедуры New и Dispose?

10. Какие действия выполняют процедуры Mark и Release?

11. Какие действия выполняют процедуры Getmem и Freemem?

12. Какие действия выполняют функции MemAvail и MaxAvail?

13. В чем выражаются динамические свойства несвязанных динамических данных?

14. В чем сходство и различие между линейными и кольцевыми списками?

15. Что представляет собой очередь как структура данных?

16. Что представляет собой стек как структура данных?

17. Что требуется для создания связанных динамических структур данных?

18. В чем состоит особенность описания типов для создания динамических структур данных?

19. Сколько указателей требуется для работы с очередью?

20. Какие действия необходимо выполнить для создания очереди?

21. Как добавить, или удалить элемент очереди?

22. Сколько указателей требуется для работы со стеком?

23. Какие действия необходимо выполнить для создания стека?

24. Как добавить, или удалить элемент стека?

25. При решении, каких прикладных задач удобно представлять набор объектов в виде двоичных деревьев?

2.2 Темы письменных работ (эссе, рефераты, курсовые работы и др.)

Эссе (20 тем)

1. Роль программирования в современном мире: от науки до повседневной жизни.
2. Этические аспекты использования алгоритмов в программировании.
3. Преимущества и ограничения различных языков программирования.
4. Будущее программирования: тренды и прогнозы развития.
5. Как искусственный интеллект меняет подходы к программированию?
6. Сравнение процедурного и объектно-ориентированного программирования.
7. Влияние алгоритмов на эффективность программного обеспечения.
8. Как программирование помогает решать реальные задачи бизнеса?
9. Роль алгоритмов сортировки и поиска в разработке программ.
10. Безопасность данных в контексте программирования и алгоритмизации.
11. Как блокчейн-технологии используются в программировании?
12. Программирование как инструмент автоматизации бизнес-процессов.
13. Проблемы цифрового неравенства в контексте обучения программированию.
14. Как персонализация влияет на разработку алгоритмов?
15. Роль аналитики данных в создании эффективных алгоритмов.
16. Как программирование поддерживает глобализацию технологий?
17. Проблемы правового регулирования использования алгоритмов.
18. Как программирование помогает малому бизнесу развиваться?
19. Роль алгоритмов в управлении большими данными (Big Data).
20. Этические вопросы использования алгоритмов в социальных сетях.

Рефераты (20 тем)

21. История развития программирования и его влияние на современные технологии.
22. Основные парадигмы программирования: процедурное, объектно-ориентированное, функциональное.
23. Принципы работы рекурсивных алгоритмов и их применение.
24. Принципы создания и управления базами данных через программирование.
25. Как работает система электронных коммуникаций в программировании?
26. Алгоритмы машинного обучения и их реализация в программировании.
27. Принципы работы систем лояльности через алгоритмы.
28. Как используются облачные технологии в программировании?
29. Принципы работы систем управления заказами через программирование.
30. Роль маркетинга в продвижении программных продуктов.
31. Принципы работы систем управления складскими запасами через программирование.
32. Методы защиты персональных данных в программировании.
33. Принципы работы систем управления цепями поставок через алгоритмы.
34. Как работают системы анализа пользовательского поведения через

программирование?

35. Принципы работы систем управления финансами в бизнесе через программирование.

36. Принципы работы систем управления рисками через алгоритмы.

37. Как используются технологии IoT в программировании?

38. Принципы работы систем управления документооборотом через программирование.

39. Принципы работы систем управления качеством продукции через алгоритмы.

40. Принципы работы систем управления маркетинговыми кампаниями через программирование.

Курсовые работы (20 тем)

41. Разработка программы для анализа данных с использованием алгоритмов сортировки.

42. Создание системы рекомендаций для клиентов интернет-магазина на основе алгоритмов.

43. Проектирование системы управления рисками через алгоритмы.

44. Разработка программы для анализа больших данных (Big Data) через алгоритмы.

45. Создание системы управления складскими запасами через алгоритмы.

46. Разработка системы электронных платежей для компании с использованием алгоритмов.

47. Проектирование системы управления лояльностью клиентов через алгоритмы.

48. Создание системы управления маркетинговыми кампаниями через алгоритмы.

49. Разработка системы управления цепями поставок через алгоритмы.

50. Проектирование системы управления рисками в бизнесе через алгоритмы.

51. Создание системы управления финансами для компании через алгоритмы.

52. Разработка системы управления документооборотом через алгоритмы.

53. Проектирование системы управления качеством продукции через алгоритмы.

54. Создание системы управления портфелем инвестиций через алгоритмы.

55. Разработка системы управления экологическими показателями через алгоритмы.

56. Проектирование системы управления фермерским хозяйством через алгоритмы.

57. Создание системы управления медицинскими данными через алгоритмы.

58. Разработка системы управления учебным процессом онлайн через алгоритмы.

59. Проектирование системы управления задачами для удаленных команд через алгоритмы.

60. Создание системы управления энергопотреблением здания через алгоритмы.

1. Текущий контроль знаний

Тестовые вопросы

1. Что такое алгоритм? Приведите примеры его применения.

2. Какие основные парадигмы программирования существуют?

3. В чем заключается принцип работы рекурсии? Приведите пример.
 4. Как работает алгоритм сортировки пузырьком? Опишите его сложность.
 5. Что такое структуры данных? Перечислите основные типы.
 6. Как реализуется работа со стеком и очередью в программировании?
 7. Что такое объектно-ориентированное программирование? Приведите пример.
 8. Как выполняется отладка программного кода?
 9. Что такое асимптотическая сложность алгоритма? Приведите пример.
 10. Какие методы защиты данных используются в программировании?
- Практические задания
 11. Напишите программу для вычисления факториала числа с использованием рекурсии.
 12. Реализуйте алгоритм сортировки массива методом выбора.
 13. Создайте программу для поиска элемента в массиве с использованием бинарного поиска.
 14. Напишите программу, демонстрирующую работу со стеком.
 15. Разработайте программу для обработки строковых данных (например, подсчет количества слов).
- Открытые вопросы
 16. Как искусственный интеллект меняет подходы к разработке алгоритмов?
 17. Какие ошибки чаще всего допускаются при написании программного кода?
 18. Как программирование помогает в снижении затрат на бизнес?
 19. Какие факторы следует учитывать при выборе языка программирования?
 20. Как программирование поддерживает удаленную работу сотрудников?
2. Промежуточная аттестация
 - Тестовые вопросы
 21. Что такое функциональное программирование? Где оно применяется?
 22. Какие методы защиты данных используются в алгоритмах шифрования?
 23. В чем заключается концепция "serverless" (безсерверных вычислений) в программировании?
 24. Как программирование поддерживает удаленную работу сотрудников?
 25. Что такое KPI? Для чего они используются в разработке программного обеспечения?
 - Практические задания
 26. Разработайте систему KPI для отдела разработки с использованием алгоритмов.
 27. Создайте план управления рисками для проекта через алгоритмы.
 28. Реализуйте систему уведомлений для команды проекта на основе алгоритмов.
 29. Настройте интеграцию облачного хранилища с системой анализа данных.
 30. Разработайте план миграции данных в облако для анализа.
 - Ситуационные задачи
 31. Компания хочет автоматизировать процесс обработки данных. Как вы будете строить модель алгоритма?
 32. У вас есть набор данных о поведении пользователей. Как вы будете их анализировать с помощью алгоритмов?
 33. Вам нужно разработать систему для прогнозирования рыночных трендов. Какие алгоритмы вы выберете?

34. Вы работаете над проектом управления рисками. Как алгоритмы могут быть применены в этой задаче?

35. Компания хочет автоматизировать процесс анализа данных. Как вы будете использовать программирование для решения этой задачи?

3. Итоговый контроль

Защита курсового проекта

36. Разработка программы для анализа данных с использованием алгоритмов сортировки.

37. Проектирование системы управления рисками через алгоритмы.

38. Создание системы управления складскими запасами через алгоритмы.

39. Разработка системы рекомендаций для клиентов интернет-магазина.

40. Проектирование системы управления качеством продукции через алгоритмы.

Экзамен

41. Теоретический блок: тестовые вопросы по основным темам курса.

42. Практический блок: выполнение задания на написание программы или алгоритма.

43. Анализ предоставленного кода и исправление ошибок.

Оценка портфолио

44. Сборник выполненных лабораторных работ.

45. Документация по разработанным проектам.

46. Отчеты по практическим заданиям.

4. Дополнительные виды оценочных средств

47. Устное собеседование по ключевым темам курса.

48. Анализ кода и исправление ошибок в предоставленном проекте.

49. Выполнение заданий на оптимизацию производительности алгоритмов.

50. Разработка прототипа системы управления процессом с использованием алгоритмов.

1. Текущий контроль знаний:

- Тесты на знание основ программирования и алгоритмизации (типы данных, структуры данных, парадигмы программирования).

- Практические задания по написанию программ для решения базовых задач.

- Написание скриптов для обработки данных с использованием алгоритмов.

- Лабораторные работы по реализации алгоритмов сортировки, поиска и работы со структурами данных.

2. Промежуточная аттестация:

- Выполнение контрольной работы (например, разработка программы для анализа данных или реализация алгоритма).

- Рефераты или эссе на заданные темы.

- Мини-проект (например, разработка системы управления задачами или рекомендательного алгоритма).

3. Итоговый контроль:

- Защита курсового проекта (разработка полноценной программы или системы с использованием алгоритмов).

- Экзамен в виде теста или выполнения практического задания.

- Оценка портфолио выполненных лабораторных и практических работ.

4. Дополнительные виды оценочных средств:

- Устное собеседование по ключевым темам курса.
- Анализ кода и исправление ошибок в предоставленном проекте.
- Выполнение заданий на оптимизацию производительности алгоритмов или программ.