

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Крюков Вадим Николаевич

Должность: Проректор по образовательной деятельности и молодежной политике

Дата подписания: 15.06.2026 16:11:05

Уникальный программный ключ:

1b0adb7fd710f6a0725d90c58682bd0c52f25b2

Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Заполярье 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 10.04.2026 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 10.04.2026 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	Кодрезультатаобучения по дисциплине/ модулю	Оценочные средства текущей		Оценочные средства промежуточно	
			Наименование	Форма	Наименование	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 $\text{SIN}(Y)$; 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 описании 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.