

**Оценочные средства для проведения аттестации  
по дисциплине «Программирование в биотехнических системах»  
для обучающихся 2023 года поступления  
по образовательной программе  
12.03.04. «Биотехнические системы и технологии»,  
профиль «Клиническая инженерия» (бакалавриат), форма обучения очная  
2024-2025 учебный год.**

Промежуточная аттестация по дисциплине проводится в форме зачета. Промежуточная аттестация проводится в форме собеседования.

**Примеры тестовых заданий:**

Готовность выполнять расчет и проектирование деталей, компонентов и узлов биотехнических систем, биомедицинской и экологической техники в соответствии с техническим заданием с использованием средств автоматизации проектирования (ПК-20)

1.1 Что из перечисленного НЕ является свойством алгоритма:

- а) Дискретность
- б) Детерминированность
- в) Многозначность**
- г) Понятность
- д) Массовость

1.2 Эвристический алгоритм – это:

- а) это такой алгоритм, в котором достижение конечного результата программы действий однозначно не предопределено, так же как не обозначена вся последовательность действий, не выявлены все действия исполнителя.**
- б) набор команд (указаний), выполняемых последовательно во времени друг за другом.

- в) алгоритм, содержащий хотя бы одно условие, в результате проверки которого ЭВМ обеспечивает переход на один из двух возможных шагов.
- в) алгоритм, который дает программу решения задачи несколькими путями или способами, приводящими к вероятному достижению результата.
- г) алгоритм, предусматривающий многократное повторение одного и того же действия (одних и тех же операций) над новыми исходными данными.
- д) алгоритм, ранее разработанный и целиком используемый при алгоритмизации конкретной задачи.

### 1.3 Вспомогательный (подчиненный) алгоритм – это

- а) это такой алгоритм, в котором достижение конечного результата программы действий однозначно не предопределено, так же как не обозначена вся последовательность действий, не выявлены все действия исполнителя.
- б) набор команд (указаний), выполняемых последовательно во времени друг за другом.
- в) алгоритм, содержащий хотя бы одно условие, в результате проверки которого ЭВМ обеспечивает переход на один из двух возможных шагов.
- г) алгоритм, который дает программу решения задачи несколькими путями или способами, приводящими к вероятному достижению результата.
- д) алгоритм, предусматривающий многократное повторение одного и того же действия (одних и тех же операций) над новыми исходными данными.
- е) алгоритм, ранее разработанный и целиком используемый при алгоритмизации конкретной задачи.**

### 1.4 Разветвляющийся алгоритм – это:

- а) это такой алгоритм, в котором достижение конечного результата программы действий однозначно не предопределено, так же как не обозначена вся последовательность действий, не выявлены все действия исполнителя.
- б) набор команд (указаний), выполняемых последовательно во времени друг за другом.
- в) алгоритм, содержащий хотя бы одно условие, в результате проверки которого ЭВМ обеспечивает переход на один из двух возможных шагов.**

- г) алгоритм, который дает программу решения задачи несколькими путями или способами, приводящими к вероятному достижению результата.
- д) алгоритм, предусматривающий многократное повторение одного и того же действия (одних и тех же операций) над новыми исходными данными.
- е) алгоритм, ранее разработанный и целиком используемый при алгоритмизации конкретной задачи.

#### 1.5 Циклический алгоритм – это:

- а) это такой алгоритм, в котором достижение конечного результата программы действий однозначно не предопределено, так же как не обозначена вся последовательность действий, не выявлены все действия исполнителя.
- б) набор команд (указаний), выполняемых последовательно во времени друг за другом.
- в) алгоритм, содержащий хотя бы одно условие, в результате проверки которого ЭВМ обеспечивает переход на один из двух возможных шагов.
- г) алгоритм, который дает программу решения задачи несколькими путями или способами, приводящими к вероятному достижению результата.
- д) алгоритм, предусматривающий многократное повторение одного и того же действия (одних и тех же операций) над новыми исходными данными.**
- е) алгоритм, ранее разработанный и целиком используемый при алгоритмизации конкретной задачи.

#### 1.6 Линейный алгоритм – это:

- а) это такой алгоритм, в котором достижение конечного результата программы действий однозначно не предопределено, так же как не обозначена вся последовательность действий, не выявлены все действия исполнителя.
- б) набор команд (указаний), выполняемых последовательно во времени друг за другом.**
- в) алгоритм, содержащий хотя бы одно условие, в результате проверки которого ЭВМ обеспечивает переход на один из двух возможных шагов.
- г) алгоритм, который дает программу решения задачи несколькими путями или способами, приводящими к вероятному достижению результата.

- д) алгоритм, предусматривающий многократное повторение одного и того же действия (одних и тех же операций) над новыми исходными данными.
- е) алгоритм, ранее разработанный и целиком используемый при алгоритмизации конкретной задачи.

1.7 Вероятностный (стохастический) алгоритм – это:

- а) это такой алгоритм, в котором достижение конечного результата программы действий однозначно не предопределено, так же как не обозначена вся последовательность действий, не выявлены все действия исполнителя.
- б) набор команд (указаний), выполняемых последовательно во времени друг за другом.
- в) алгоритм, содержащий хотя бы одно условие, в результате проверки которого ЭВМ обеспечивает переход на один из двух возможных шагов.
- г) алгоритм, который дает программу решения задачи несколькими путями или способами, приводящими к вероятному достижению результата.**
- д) алгоритм, предусматривающий многократное повторение одного и того же действия (одних и тех же операций) над новыми исходными данными.
- е) алгоритм, ранее разработанный и целиком используемый при алгоритмизации конкретной задачи.

1.8 К какому из способов задания алгоритмов относят задание с помощью блок-схем.

- а) графический**
- б) словесный
- в) формульно-словесный
- г) програмный

1.9 Множество М называется разрешимым

- а) если существует алгоритм, позволяющий перечислить все элементы этого множества (возможно с повторениями).
- б) тогда и только тогда, когда оно само и его дополнение эффективно перечислимы.

**в) если для него существует алгоритм, решающий проблему вхождения слова  $x$  в  $M$ .**

1.10 Множество  $M$  называется эффективно перечислимым, если

**а) если существует алгоритм, позволяющий перечислить все элементы этого множества (возможно с повторениями).**

б) тогда и только тогда, когда оно само и его дополнение эффективно перечислимы.

в) если для него существует алгоритм, решающий проблему вхождения слова  $x$  в  $M$ .

**Перечень контрольных вопросов:**

<b>№</b>	<b>Вопросы для промежуточной аттестации</b>	<b>Проверяемые компетенции</b>
1.	Понятие алгоритма. Основные свойства алгоритмов.	ОПК-1, ОПК-2, ОПК-5, ПК-20
2.	Классификация алгоритмов. Формы записи алгоритмов.	ОПК-1, ОПК-2, ОПК-5, ПК-20
3.	Понятие исполнителя алгоритмов. Алгоритм и вычислительный процесс.	ОПК-1, ОПК-2, ОПК-5, ПК-20
4.	Блок-схемы описания алгоритмов. Основные алгоритмические структуры.	ОПК-1, ОПК-2, ОПК-5, ПК-20
5.	Основная проблема теории алгоритмов. Классическая теория алгоритмов.	ОПК-1, ОПК-2, ОПК-5, ПК-20
6.	Нормальные алгорифмы Маркова. Принцип нормализации.	ОПК-1, ОПК-2, ОПК-5, ПК-20
7.	Рекурсивные функции. Тезис Черча. Суперпозиция.	ОПК-1, ОПК-2, ОПК-5, ПК-20
8.	Рекурсивные функции. Примитивная рекурсия.	ОПК-1, ОПК-2, ОПК-5, ПК-20
9.	Рекурсивные функции. Тезис Клини. Частично рекурсивные функции. Минимизация.	ОПК-1, ОПК-2, ОПК-5, ПК-20

10.	Рекурсивные функции. Классы рекурсивных функций.	ОПК-1, ОПК-2, ОПК-5, ПК-20
11.	Рекурсивные функции. Способы доказательства рекурсивности.	ОПК-1, ОПК-2, ОПК-5, ПК-20
12.	Машина Тьюринга. Тезис Тьюринга. Работа машины Тьюринга.	ОПК-1, ОПК-2, ОПК-5, ПК-20
13.	Машина Поста. Особенности работы и программирования машины Поста.	ОПК-1, ОПК-2, ОПК-5, ПК-20
14.	Машина Тьюринга, внешний и внутренний алфавит. Программирование машины Тьюринга.	ОПК-1, ОПК-2, ОПК-5, ПК-20
15.	Программа машины Тьюринга. Методы программирования базовых алгоритмов.	ОПК-1, ОПК-2, ОПК-5, ПК-20
16.	Нормальные алгоритмы. Конструирование нормальных алгоритмов.	ОПК-1, ОПК-2, ОПК-5, ПК-20
17.	Естественный и формальный языки. Построение формальных языков.	ОПК-1, ОПК-2, ОПК-5, ПК-20
18.	Грамматики формального языка. Определение, построение, исследование.	ОПК-1, ОПК-2, ОПК-5, ПК-20
19.	Общее понятие исчисления. Иерархия языков и грамматик по Хомскому.	ОПК-1, ОПК-2, ОПК-5, ПК-20
20.	Распознаватели и задача разбора.	ОПК-1, ОПК-2, ОПК-5, ПК-20
21.	Классификация распознавателей по типам языков.	ОПК-1, ОПК-2, ОПК-5, ПК-20
22.	Синтаксические диаграммы алгоритмического языка. Описание грамматики с помощью метасимволов.	ОПК-1, ОПК-2, ОПК-5, ПК-20
23.	Нотации Бекуса-Наура.	ОПК-1, ОПК-2, ОПК-5, ПК-20
24.	Языки и машины. Методы построения машинных языков.	ОПК-1, ОПК-2, ОПК-5, ПК-20
25.	Основы теории автоматов. Конечные и бесконечные автоматы, детерминированные и недетерминированные автоматы.	ОПК-1, ОПК-2, ОПК-5, ПК-20
26.	Автоматы с магазинной памятью.	ОПК-1, ОПК-2, ОПК-5, ПК-20
27.	Сложность алгоритма. Основные меры	ОПК-1, ОПК-2, ОПК-5,

	сложности вычисления.	ПК-20
28.	Оценки сложности алгоритмов. Верхняя и нижняя оценки.	ОПК-1, ОПК-2, ОПК-5, ПК-20
29.	Полнота алгоритма. Основы теории NP-полноты.	ОПК-1, ОПК-2, ОПК-5, ПК-20
30.	Оценки полноты алгоритмов и задач.	ОПК-1, ОПК-2, ОПК-5, ПК-20
31.	Применение теории NP-полноты для анализа сложности проблем.	ОПК-1, ОПК-2, ОПК-5, ПК-20
32.	Алгоритмически неразрешимые проблемы. Примеры алгоритмически неразрешимых проблем в математике и информатике.	ОПК-1, ОПК-2, ОПК-5, ПК-20
33.	Алгоритмически неразрешимые проблемы. Экстраалгоритм. Теорема распознавания применимости алгоритма.	ОПК-1, ОПК-2, ОПК-5, ПК-20
34.	Вычислимые функции.	ОПК-1, ОПК-2, ОПК-5, ПК-20
35.	Универсальная машина Тьюринга. Композиция МТ.	ОПК-1, ОПК-2, ОПК-5, ПК-20
36.	Теорема о неподвижной точке.	ОПК-1, ОПК-2, ОПК-5, ПК-20
37.	Взаимосвязь математической теории алгоритмов с теоретической	ОПК-1, ОПК-2, ОПК-5, ПК-20

Рассмотрено на заседании кафедры клинической инженерии и технологий искусственного интеллекта «23» мая 2024 г., протокол №10.

Заведующий кафедрой



С.А.Безбородов