

Документ подписан простой электронной подписью Информация о владельце: ФИО: Таскаев Сергей Валерьевич Должность: Ректор Дата подписания: 2025.06.05 Уникальный программный ключ: 054c0182970293149c21699f0009940292896664	МИНОБНАУКИ РОССИИ Федеральное государственное бюджетное образовательное учреждение высшего образования «Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ») Фонд оценочных средств для промежуточной аттестации по дисциплине (модулю) «Логическое программирование» по направлению подготовки (специальности) 02.03.02 «Фундаментальная информатика и информационные технологии» направленности (профилю) «Математические и алгоритмические основы интеллектуальных систем» ФГБОУ ВО «ЧелГУ»	стр. 1
---	---	--------

**Фонд оценочных средств для промежуточной аттестации  
по дисциплине (модулю)**

**Логическое программирование**

Направление подготовки (специальность)

**02.03.02 Фундаментальная информатика и информационные технологии**

Направленность (профиль)

**Математические и алгоритмические основы интеллектуальных систем**

Присваиваемая квалификация (степень)

**бакалавр**

Форма обучения

**очная**

Год набора

**2026**

Троицк, 2026 г.



## Содержание

1. Паспорт фонда оценочных средств
2. Перечень формируемых компетенций
  - 2.1. Компетенции, закреплённые за дисциплиной
3. Содержание оценочных средств по дисциплине
  - 3.1. Виды оценочных средств
  - 3.2. Содержание оценочных средств
4. Порядок проведения и критерии оценивания промежуточной аттестации
  - 4.1. Порядок проведения промежуточной аттестации
  - 4.2. Критерии оценивания промежуточной аттестации по видам оценочных средств
  - 4.3. Результаты промежуточной аттестации и уровни сформированности компетенций



## 1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

Направление подготовки: 02.03.02 «Фундаментальная информатика и информационные технологии»

Направленность (профиль): Математические и алгоритмические основы интеллектуальных систем

Дисциплина: *Логическое программирование*

Семестр изучения: 7 семестр

Форма промежуточной аттестации: Экзамен -7с.

## 2. ПЕРЕЧЕНЬ ФОРМИРУЕМЫХ КОМПЕТЕНЦИЙ

### 2.1. Компетенции, закреплённые за дисциплиной

Изучение дисциплины «Логическое программирование» направлено на формирование следующих компетенций:

Коды компетенции согласно ФГОС (ОПОП ВО)	Содержание компетенций согласно ФГОС (ОПОП ВО)	Индикаторы достижения компетенции согласно ОПОП	Перечень планируемых результатов обучения по дисциплине
УК-4	Способен осуществлять деловую коммуникацию в устной и письменной формах на государственном языке Российской Федерации и иностранном(ых) языке(ах)	УК -4.1 Имеет представление о правилах и принципах деловой устной и письменной коммуникации на государственном языке российской федерации и иностранном(ых) языке(ах) УК -4.2 Демонстрирует умение осуществлять деловую коммуникацию в устной и письменной формах, использовать методы и навыки делового общения УК -4.3 Имеет навыки делового общения на государственном языке российской федерации и иностранном(ых) языке(ах)	Знать: Для реализации УК-4.1: особенности логического программирования как методологией решения трудно формализуемых задач; тенденции и перспективы развития языков и инструментальных средств логического программирования Уметь: Для реализации УК-4.2: использовать методы логического программирования для построения декларативной модели предметной области Владеть: Для реализации УК-4.3: синтаксисом логического языка Пролог
ПК-2	Способность к осуществлению интеграции программных модулей и компонент и проверки работоспособности программного продукта на основе международных и профессиональных	ПК-2.1. Обладает знаниями о методах и средствах сборки модулей и компонент программного обеспечения, о разработке процедур для развертывания программного обеспечения, миграции и преобразования данных, о создании программных	Знать: Для реализации ПК-2.1: основные понятия, методы, алгоритмы и средства основ программирования логического языка Пролог. Уметь: Для достижения ПК-2.2: применять теории, методы, алгоритмы, системы и средства языка программирования Пролог



	стандартов информационных технологий, современных парадигм и методологий, инструментальных и вычислительных средств, методов и механизмов оценки и анализа функционирования средств и систем информационных технологий	интерфейсов; о методах и механизмах оценки и анализа функционирования средств и систем информационных технологий; о международных и профессиональных стандартах информационных технологий, о современных парадигмах и методологиях, инструментальных и вычислительных средствах. ПК-2.2. Демонстрирует умения: применять методы и средства сборки модулей и компонент программного обеспечения, разработки процедур для развертывания программного обеспечения, миграции и преобразования данных, создания программных интерфейсов; проводить проверку и оценку работоспособности программного продукта. ПК-2.3. Имеет практический опыт (навыки): сборки модулей и компонент программного обеспечения, разработки процедур для развертывания программного обеспечения, миграции и преобразования данных, создания программных интерфейсов; оценки работоспособности программного продукта.	при решении профессиональных задач. Владеть: Для достижения ПК-2.3: общепрофессиональными знаниями теории, методов, систем и средств для решения практических задач в области логического программирования с использованием Пролога.
ПК-3	Способность к разработке требований и проектированию программного обеспечения на основе применения базовых математических знаний и информационных технологий при решении проектно-технических и прикладных задач	ПК-3.1. Обладает знаниями о методах и средствах проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов. ПК-3.2. Демонстрирует умения: разрабатывать требования к программному продукту, применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов. ПК-3.3. Имеет практический опыт (навыки):	Знать: для достижения ПК-3.1: основы логических алгоритмов; основные элементы логического языка Пролог; последовательность и этапы разработки прикладных программ; принципы, методы и способы написания и отладки программ. Уметь: Для достижения ПК-3.2: составить информационную модель по поставленной задаче; разработать алгоритм; использовать язык Пролог для создания программных продуктов; применять фундаментальные



		проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов.	алгоритмы для оптимизации программ, направленных на решение прикладных задач. Владеть: Для достижения ПК-3.3:навыками применения современного программного инструментария для разработки и анализа программных решений; навыками разработки и тестирования прикладных программ на языке Пролог.
--	--	--	--

### 3 СОДЕРЖАНИЕ ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

#### 3.1 Виды оценочных средств

№ п/п	Контролируемые разделы	Контролируемые компетенции	Наименование оценочного средства для текущего контроля	Наименование оценочного средства на промежуточной аттестации
1	Раздел 1. Основы логического программирования	УК-4, ПК-3, ПК-2 (знания, умения навыки)	Изучение теоретического материала. Ответы на вопросы, изучаемые самостоятельно лабораторные работы	Собеседование по вопросам к экзамену (тема№1) Практическая задача
2	Раздел 2. Начальные сведения о языке Пролог.	УК-4, ПК-3, ПК-2 (знания, умения навыки)	Изучение теоретического материала. Ответы на вопросы, изучаемые самостоятельно лабораторные работы	Собеседование по вопросам к экзамену (тема№2) Практическая задача
3	Раздел 3. Рекурсия в Прологе.	УК-4, ПК-3, ПК-2 (знания, умения навыки)	Изучение теоретического материала. Ответы на вопросы, изучаемые самостоятельно лабораторные работы	Собеседование по вопросам к экзамену (тема№3) Практическая задача
4	Раздел 4. Управление доказательством.	УК-4, ПК-3, ПК-2 (знания, умения навыки)	Изучение теоретического материала. Ответы на вопросы, изучаемые самостоятельно лабораторные работы	Собеседование по вопросам к экзамену (тема№4) Практическая задача
5	Раздел 5. Применение Пролога для решения прикладных задач.	УК-4, ПК-3, ПК-2 (знания, умения навыки)	Изучение теоретического материала. Ответы на вопросы, изучаемые самостоятельно лабораторные работы	Собеседование по вопросам к экзамену (тема№5) Практическая задача

*Примечание:* типовые задания, критерии и показатели оценивания в рамках текущего контроля представлены в рабочей программе по дисциплине. Полные комплекты оценочных средств и контрольно-  
© ФГБОУ ВО «ЧелГУ»



измерительных материалов хранятся на кафедре и являются учебно-методическими материалами ограниченного (конфиденциального) пользования.

### 3.2 Содержание оценочных средств

#### Список вопросов к экзамену

Вопросы к экзамену по темам

Тема 1. Основы логического программирования.

1. Дайте определение понятия «логическое программирование»
2. Перечислите основные элементы языка логики предикатов 1 порядка.
3. Что такое фраза Хорна? Перечислите типы фраз Хорна, используемые при построении логических программ
4. Опишите идею метода резолюции.
5. Что такое база знаний логической программы? Чем она отличается от базы данных?
6. Что такое недетерминированность логической программы? В чем она проявляется?
7. В чем заключается различие между ложными и недоказуемыми утверждениями?
8. Что такое побочный эффект?
9. Что такое унификация? В чем различие свободных и связанных переменных?
10. Опишите алгоритм метода резолюции.

Тема 2. Начальные сведения о языке Пролог.

1. Опишите структуру Пролог-программы. Какие секции программы являются обязательными? Почему?
2. Дайте определение анонимной переменной, перечислите ее особенности и случаи употребления.
3. Дайте определение области видимости. Какова область видимости переменных логической программы?
4. Что такое входо-выходная разметка? В чем отличие входо-выходной разметки аргументов в декларациях подпрограмм на процедурных языках и декларациях предикатов логической программы.
5. Что такое декларативный стиль программирования?
6. Почему логические программы называют недетерминированными?
7. В чем заключается стандартная стратегия доказательства?



8. Как влияет наличие предопределенной стратегии доказательства на декларативность логических программ?

9. Перечислите эмпирические правила построения эффективных определений логической программы, исходя из стандартной стратегии вывода.

10. Оцените вычислительную эффективность стандартной стратегии доказательства.

Тема 3. Рекурсия в прологе.

1. Дайте определение рекурсии. Почему использование рекурсии увеличивает выразительную силу языков логического программирования?

2. Назовите основные источники рекурсии. Существует ли между ними какая-либо связь?

3. Продемонстрируйте примеры рекурсивных определений. Сформулируйте основные правила их построения.

4. Продемонстрируйте примеры рекурсивных термов. Сформулируйте основные правила их построения.

5. Дайте определение рекурсивного термина «список». Определите синтаксис встроенной реализации списков в прологе и правила их унификации.

6. Приведите примеры построения определений, использующих списки.

7. В чем заключается основная проблема использования рекурсии?

Тема 4. Управление доказательством.

1. Назовите основные мотивы управления доказательством логической программы.

2. Назовите предикаты управления доказательством и дайте их краткую характеристику.

3. В чем заключается связь между предикатами управления доказательством и используемой стратегией вывода?

4. Каков эффект воздействия предиката «отсечение» на стандартную стратегию доказательства? Приведите пример.

5. Перечислите основные случаи использования предиката «отсечение». Приведите примеры.

6. Сформулируйте рекомендации по использованию предиката «отсечение».

7. В чем заключается положительный эффект отсечения?

8. Перечислите основные случаи использования предиката «неудача». Приведите примеры.



9. В чем заключается положительный эффект от использования предиката «неудача»?

10. В чем заключаются недостатки использования предикатов управления доказательством в логической программе?

Тема 5. Применение Пролога для решения прикладных задач.

1. Дайте определение спецификации. Приведите простейший эквивалент этого определения на языке Пролог.

2. Дайте декларативную интерпретацию утверждением для предикатов сжатия спецификационного списка.

3. Напишите версию утверждений для предикатов сжатия спецификационного списка, не использующую предикат отсечения.

4. Дайте определение «интеллектуальных» задач. Приведите примеры интеллектуальных задач.

5. Дайте определение пространства состояний. Какие особенности логических языков делают их удобным инструментом описания задач поиска в пространстве состояний?

6. Какую стратегию поиска в пространстве состояний реализует машина вывода Пролога? Какие еще стратегии поиска возможны?

7. В чем заключается суть метода порождения и проверки? Продемонстрируйте его на примере решения задачи о перевозчике.

8. Покажите на примере задачи о восьми ферзях реализацию предикатов порождения и проверки.

### Примерный список задач к экзамену

1. Создайте предикат, заменяющий в исходном списке первое вхождение заданного значения другим.
2. Создайте предикат, заменяющий в исходном списке все вхождения заданного значения другим.
3. Создайте предикат, порождающий по заданному натуральному числу  $N$  список, состоящий из натуральных чисел от 1 до  $N$  (по возрастанию).
4. Создайте предикат, порождающий по заданному натуральному числу  $N$  список, состоящий из натуральных чисел от  $N$  до 1 (по убыванию).
5. Создайте предикат, порождающий по заданному натуральному числу  $N$  список, состоящий из  $N$  случайных натуральных чисел из промежутка от 1 до 100.



6. Создайте предикат, порождающий по заданным числам  $N$ ,  $M$ ,  $K$  список, состоящий из  $N$  случайных натуральных чисел из промежутка от  $M$  до  $K$ .
7. Создайте предикат, порождающий по заданным числам  $M$ ,  $K$  список, состоящий из случайного количества случайных чисел из промежутка от  $M$  до  $K$ .
8. Создайте предикат, порождающий список, состоящий из случайного количества случайных чисел.
9. Создайте предикат, который увеличивает элементы исходного списка на единицу.
10. Создайте предикат, переводящий список цифр от 0 до 9 в список соответствующих им названий (строк).
11. Создайте предикат, переводящий список чисел в список соответствующих им названий.
12. Создайте предикат, переводящий список цифр от 0 до 9 в список соответствующих им римских чисел.
13. Создайте предикат, переводящий список арабских чисел в список соответствующих им римских чисел.
14. Создайте предикат, переводящий список римских чисел в список соответствующих им арабских чисел.
15. Создайте предикат, удваивающий значения элементов списка.
16. Создайте предикат, преобразующий список, элементами которого являются числа, в список, элементы которого неотрицательны.
17. Создайте предикат, преобразующий исходный список в список позиций отрицательных элементов.
18. Создайте предикат, удаляющий из исходного списка элементы с четными номерами.
19. Создайте предикат, который разделит исходный список из целых чисел на два списка: список положительных чисел и список отрицательных чисел.
20. Создайте предикат, разделяющий исходный список на два подсписка. В первый из них должны попасть элементы с нечетными номерами, во второй - элементы с четными номерами.
21. Создайте предикат, вычисляющий по списку и числу, подсписок исходного списка, начинающийся с элемента с указанным номером.
22. Создайте предикат, осуществляющий удаление указанного количества последних элементов исходного списка.
23. Создайте предикат, осуществляющий разделение исходного списка на два подсписка. В первый из них должно попасть указанное количество элементов из начала списка, во второй - оставшиеся элементы.



24. Создайте предикат, осуществляющий разделение исходного списка на два подсписка. В первый из них должно попасть указанное количество элементов с конца списка, во второй - оставшиеся элементы.
25. Создайте предикат, находящий предпоследний элемент списка.
26. Создайте предикат, удаляющий предпоследний элемент списка.
27. Создайте предикат, заменяющий в исходном списке два подряд идущих одинаковых элемента одним.
28. Создайте предикат, удаляющий в исходном списке все повторные вхождения элементов.
29. Создайте предикат, осуществляющий перестановку двух элементов списка с заданными номерами.
30. Создайте предикат, генерирующий все перестановки элементов списка, указанного в качестве первого аргумента предиката.
31. Создайте предикат, осуществляющий циклический сдвиг элементов списка на один влево (вправо).
32. Создайте предикат, осуществляющий циклический сдвиг элементов списка на заданное количество шагов влево (вправо).
33. Создайте предикат, осуществляющий поэлементное перемножение соответствующих элементов двух исходных списков.
34. Создайте предикат, вычисляющий скалярное произведение векторов, заданных списками целых чисел.
35. Создайте предикат, осуществляющий подсчет числа вхождений каждого элемента исходного списка. Ответом должен быть список пар, в которых первая компонента - элемент исходного списка, вторая - число его вхождений в первоначальный список.
36. Создайте предикат, определяющий первую позицию подсписка в списке.
37. Создайте предикат, добавляющий элементы одного списка во второй список, начиная с заданной позиции.
38. Создайте предикат, возвращающий по списку и двум числам  $M$  и  $N$  подсписок исходного списка, состоящий из элементов с номерами от  $M$  до  $N$ .
39. Создайте предикат, формирующий список простых чисел, не превосходящих данного числа.
40. Создайте предикат, транспонирующий матрицу, заданную списком списков.



## 4. ПОРЯДОК ПРОВЕДЕНИЯ И КРИТЕРИИ ОЦЕНИВАНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

### Порядок проведения промежуточной аттестации

Для допуска на экзамен по дисциплине студент должен набрать 40-60 баллов Эта сумма складывается из выполнения лабораторных работ индивидуальных домашних заданий и проектов, рефератов. Экзамен проводится в форме собеседования по вопросам к экзамену и выполнения практического задания на языке Пролог.

При дистанционном обучении устный опрос, в том числе защита курсовых работ, проводятся в Microsoft Teams. Практические задания и письменные ответы размещаются в системе Moodle. Тестирование осуществляется в системе Moodle.

#### 4.1. Критерии оценивания промежуточной аттестации по видам оценочных средств

Критерии оценки знаний студентов на экзамене по дисциплине:

«Отлично» – выставляется студенту в том случае, если он:

- глубоко и правильно усвоил программный материал, последовательно, грамотно и логически стройно его излагает;
- владеет основными понятиями и конструкциями языка программирования, не только типовыми алгоритмами решения; методами разработки программы и задач
- умеет строить математические модели, увязывать теорию с практикой, показывает умение применять знания.

«Хорошо» – выставляется студенту, если:

- он твердо знает программный материал, грамотно и по существу его излагает;
- владеет основными понятиями дисциплины;
- не допускает существенных ошибок, но и испытывает затруднения в выводах и поиске алгоритма;
- умеет применять основные операторы для решения задачи.

«Удовлетворительно» – выставляется студенту в том случае, если он:

- имеет знания только основного материала, но не умеет в полном объеме применить их на практике;
- допускает ошибки, недостаточно правильные формулировки;



- с трудом увязывает основные положения с практикой.
- «Неудовлетворительно» – выставляется студенту в том случае, если он:
  - не знает основополагающих вопросов изучаемого дисциплины или значительной части программного материала;
  - допускает ошибки, обнаруживает неумение их исправлять;
  - не может увязать теорию с практикой.

При необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на зачете.

#### 4.3. Результаты промежуточной аттестации и уровни сформированности компетенций

Код компетенции	Планируемые результаты обучения по дисциплине	Критерии оценивания			
		Отлично Высокий уровень освоения проверяемых компетенций	Хорошо Средний уровень освоения проверяемых компетенций	Удовлетворительно Базовый уровень освоения проверяемых компетенций	Неудовлетворительно Недостаточный уровень освоения проверяемых компетенций
УК-4	<b>Знает</b> особенности логического программирования как методологией решения трудно формализуемых задач; тенденции и перспективы развития языков и инструментальных средств логического программирования	Знает особенности логического программирования как методологией решения трудно формализуемых задач; тенденции и перспективы развития языков и инструментальных средств логического программирования	Знает особенности логического программирования как методологией решения трудно формализуемых задач; тенденции и перспективы развития языков и инструментальных средств логического программирования, но допускает незначительные ошибки	Знает в ограниченном объеме особенности логического программирования как методологией решения трудно формализуемых задач; тенденции и перспективы развития языков и инструментальных средств логического программирования	Не знает особенности логического программирования как методологией решения трудно формализуемых задач; тенденции и перспективы развития языков и инструментальных средств логического программирования
	<b>Умеет</b> использовать методы логического программирования для построения	<b>Умеет</b> использовать методы логического программирования для построения декларативной	<b>Умеет</b> использовать методы логического программирования для построения декларативной	<b>Умеет</b> использовать методы логического программирования для построения декларативной	Не умеет использовать методы логического программирования для построения декларативной модели предметной



	декларативной модели предметной области	модели предметной области	модели предметной области, но допускает незначительные ошибки	модели предметной области в ограниченном объеме	области
	<b>Владеет</b> синтаксисом логического языка Пролог	<b>Владеет</b> синтаксисом логического языка Пролог	<b>Владеет</b> синтаксисом логического языка Пролог., но допускает незначительные ошибки	<b>Владеет</b> синтаксисом логического языка Пролог, в ограниченном объеме	Не Владеет синтаксисом логического языка Пролог
ПК-2	<b>Знает:</b> основные понятия, методы, алгоритмы и средства основ программирования логического языка Пролог.	<b>Знает</b> основные понятия, методы, алгоритмы и средства основ программирования логического языка Пролог	<b>Знает</b> основные понятия, методы, алгоритмы и средства основ программирования логического языка Пролог, но допускает незначительные ошибки	<b>Знает в ограниченном объеме</b> основные понятия, методы, алгоритмы и средства основ программирования логического языка Пролог	Не знает основные понятия, методы, алгоритмы и средства основ программирования логического языка Пролог
	<b>Умеет</b> применять теории, методы, алгоритмы, системы и средства языка программирования Пролог при решении профессиональных задач.	<b>Умеет</b> применять теории, методы, алгоритмы, системы и средства языка программирования Пролог при решении профессиональных задач.	<b>Умеет</b> применять теории, методы, алгоритмы, системы и средства языка программирования Пролог при решении профессиональных задач, но допускает незначительные ошибки	<b>Умеет</b> применять теории, методы, алгоритмы, системы и средства языка программирования Пролог при решении профессиональных задач, в ограниченном объеме	Не умеет, применять теории, методы, алгоритмы, системы и средства языка программирования Пролог при решении профессиональных задач..
	<b>Владеет</b> общепрофессиональными знаниями теории, методов, систем и средств для решения практических задач в области логического программирования с использованием Пролога.	<b>Владеет</b> общепрофессиональными знаниями теории, методов, систем и средств для решения практических задач в области логического программирования с использованием Пролога.	<b>Владеет</b> общепрофессиональными знаниями теории, методов, систем и средств для решения практических задач в области логического программирования с использованием Пролога., но допускает ошибки	<b>Владеет</b> общепрофессиональными знаниями теории, методов, систем и средств для решения практических задач в области логического программирования с использованием Пролога., в ограниченном объеме	<b>Не владеет</b> общепрофессиональными знаниями теории, методов, систем и средств для решения практических задач в области логического программирования с использованием Пролога.



ПК-3	<p>Знает основы логических алгоритмов; основные элементы логического языка Пролог; последовательность и этапы разработки прикладных программ; принципы, методы и способы написания и отладки программ.</p>	<p><b>Знает</b> основы логических алгоритмов; основные элементы логического языка Пролог; последовательность и этапы разработки прикладных программ; принципы, методы и способы написания и отладки программ.</p>	<p><b>Знает</b> основы логических алгоритмов; основные элементы логического языка Пролог; последовательность и этапы разработки прикладных программ; принципы, методы и способы написания и отладки программ., но допускает незначительные ошибки</p>	<p><b>Знает в ограниченном объеме</b> основы логических алгоритмов; основные элементы логического языка Пролог; последовательность и этапы разработки прикладных программ; принципы, методы и способы написания и отладки программ.</p>	<p>Не знает основы логических алгоритмов; основные элементы логического языка Пролог; последовательность и этапы разработки прикладных программ; принципы, методы и способы написания и отладки программ.</p>
	<p>Умеет составить информационную модель по поставленной задаче; разработать алгоритм; использовать язык Пролог для создания программных продуктов; применять фундаментальные алгоритмы для оптимизации программ, направленных на решение прикладных задач.</p>	<p><b>Умеет</b> составить информационную модель по поставленной задаче; разработать алгоритм; использовать язык Пролог для создания программных продуктов; применять фундаментальные алгоритмы для оптимизации программ, направленных на решение прикладных задач.</p>	<p><b>Умеет</b> составить информационную модель по поставленной задаче; разработать алгоритм; использовать язык Пролог для создания программных продуктов; применять фундаментальные алгоритмы для оптимизации программ, направленных на решение прикладных задач., но допускает незначительные ошибки</p>	<p><b>Умеет</b> составить информационную модель по поставленной задаче; разработать алгоритм; использовать язык Пролог для создания программных продуктов; применять фундаментальные алгоритмы для оптимизации программ, направленных на решение прикладных задач. в ограниченном объеме</p>	<p>Не умеет, составить информационную модель по поставленной задаче; разработать алгоритм; использовать язык Пролог для создания программных продуктов; применять фундаментальные алгоритмы для оптимизации программ, направленных на решение прикладных задач.</p>
	<p>Владет навыками применения современного программного инструментария для</p>	<p><b>Владет</b> навыками применения современного программного инструментария для разработки и</p>	<p><b>Владет</b> навыками применения современного программного инструментария для разработки и</p>	<p><b>Владет</b> навыками применения современного программного инструментария для разработки и</p>	<p><b>Не Владет</b> навыками применения современного программного инструментария для разработки и</p>



разработки и анализа программных решений; навыками разработки и тестирования прикладных программ на языке Пролог.	анализа программных решений; навыками разработки и тестирования прикладных программ на языке Пролог.	анализа программных решений; навыками разработки и тестирования прикладных программ на языке Пролог., но допускает ошибки	анализа программных решений; навыками разработки и тестирования прикладных программ на языке Пролог., в ограниченном объеме	анализа программных решений; навыками разработки и тестирования прикладных программ на языке Пролог..
---	--	---	---	---

Особенности проведения процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья обозначены в рабочей программе дисциплины (модуля).

Уровни сформированности компетенций определяются следующим образом:

1. Высокий уровень сформированности компетенций соответствует оценке отлично, предполагает формирование компетенций на высоком уровне, готовность к самостоятельной профессиональной деятельности:

- глубоко и правильно усвоил программный материал, последовательно, грамотно и логически стройно его излагает;
- владеет основными математическими методами и алгоритмами решения задач;
- уметь строить математические модели, увязывать теорию с практикой, показывает умение применять знания.

2. Средний уровень соответствует оценке хорошо, предполагает формирование компетенций на достаточном уровне:

- он твердо знает программный материал, грамотно и по существу его излагает;
- владеет основными математическими методами;
- не допускает существенных ошибок, но и испытывает затруднения в выводах и доказательствах;
- умеет применять основные положения и формулы для решения задач.

3. Базовый уровень соответствует оценке удовлетворительно, предполагает формирование компетенций на начальном уровне:

- имеет знания только основного материала, но не умеет делать выводы и доказательства;
- допускает ошибки, недостаточно правильные формулировки;
- с трудом увязывает основные положения с практикой



4. Недостаточный уровень соответствует оценке  
неудовлетворительно.

- не знает основополагающих вопросов изучаемого курса или значительной части программного материала;
- допускает ошибки, обнаруживает неумение их исправлять;
- не может увязать теорию с практикой.

