

Энгельсский технологический институт (филиал) федерального государственного
бюджетного образовательного учреждения высшего образования
«Саратовский государственный технический университет имени Гагарина Ю.А.»

Кафедра «Естественные и математические науки»

Оценочные материалы по дисциплине

«Функциональное и логическое программирование»

направления подготовки

09.03.01 «Информатика и вычислительная техника»

профиль: «Программное обеспечение средств вычислительной техники и
автоматизированных систем»

Энгельс 2025

1. Перечень компетенций и уровни их сформированности по дисциплинам (модулям), практикам в процессе освоения ОПОП ВО

В процессе освоения образовательной программы у обучающегося в ходе изучения дисциплины «Функциональное и логическое программирование» должна сформироваться компетенция: ОПК-8

Критерии определения сформированности компетенций на различных уровнях их формирования

Индекс компетенции	Содержание компетенции
ОПК-8	Способен разрабатывать алгоритмы и программы, пригодные для практического применения

Код и наименование индикатора достижения компетенции	Виды занятий для формирования компетенции	Оценочные средства для оценки уровня сформированности компетенции
ИД- 3 опк-8 Понимает основные принципы логического и функционального программирования, способен разрабатывать алгоритмы и программы на языках Prolog и LISP, пригодные для практического использования	Лекции, практические занятия, самостоятельная работа	Вопросы для устного опроса Задания для практических работ. Вопросы к зачету

Уровни освоения компетенции

Уровень освоения компетенции	Критерии оценивания
Продвинутый (отлично)	Знает основные принципы логического и функционального программирования. Умеет использовать основные принципы логического и функционального программирования при разработке алгоритмов и программ на языках Prolog и LISP. Владеет способностью использовать принципы логического и функционального программирования при самостоятельном решении практических задач профессиональной деятельности по программированию на языках Prolog и LISP..
Повышенный (хорошо)	Знает не в полной мере основные принципы логического и функционального программирования. Умеет не в полной мере использовать основные принципы логического и функционального программирования при разработке алгоритмов и программ на языках Prolog и LISP. Владеет способностью использовать не в полной мере принципы логического и функционального программирования при решении практических задач профессиональной деятельности по программированию на языках Prolog и LISP..
Пороговый (базовый) (удовлетворительно)	Знает - на низком уровне основные принципы логического и функционального программирования Умеет с трудом использовать основные принципы логического и

	функционального программирования при разработке простых алгоритмов и программ на языках Prolog и LISP. Владеет способностью использовать принципы логического и функционального программирования при решении простых практических задач профессиональной деятельности по программированию на языках Prolog и LISP при помощи преподавателя
--	--

2. Методические, оценочные материалы и средства, определяющие процедуры оценивания сформированности компетенций (элементов компетенций) в процессе освоения ОПОП ВО

2.1 Оценочные средства для текущего контроля

Вопросы для устного опроса

Тема 1. Введение.

1. Опишите понятие декларативного программирования
2. Опишите понятие логического программирования
3. Дайте определение предикату

Тема 2. Логическое программирование.

1. Как реализуются факты в языке Prolog?
2. Как реализуются правила в языке Prolog?
3. Что означает символ запятой в языке Prolog?
4. Что означает символ точка с запятой в языке Prolog?
5. Что означает комбинация символов двоеточие и тире в языке Prolog?
6. Опишите задачи раздела predicates.
7. Опишите задачи раздела clauses.
8. Опишите задачи раздела goal.
9. Опишите понятие составной цели. 1
10. Опишите два способа реализации операции ИЛИ в Prolog программах.
11. Как обозначаются переменные в языке Prolog?
12. Чем отличаются декларативные переменные от процедурных?
13. Какие типы данных предоставляет Prolog?

Тема 3. Функциональное программирование.

1. Парадигмы программирования. Функциональное программирование.
2. Где используется функциональное программирование?
3. Какие термины применяются в функциональном программировании?
4. В чем преимущества функционального программирования?
5. Чем функциональное программирование отличается от объектно ориентированного?
6. На чем основано функциональное программирование?
7. Какие языки поддерживают функциональное программирование?
8. Для чего нужны функциональные языки программирования?
9. В чем отличие функционального программирования от процедурного?

10. Как работает функциональное программирование?

Тема 4. Язык функционального программирования ЛИСП.

1. Лисп: история языка, основные диалекты, S-выражения.
2. Scheme и Racket: основные типы значений, идентификаторы и связывание, процедуры, условные выражения и предикаты.
3. Рекурсия. Линейно рекурсивный и линейно итеративный процессы, древовидная рекурсия.
4. Лямбда-выражения и замыкания. Процедуры высших порядков.
5. Основные понятия λ -исчисления: аппликация и абстракция, свободные и связанные переменные, каррирование.
6. Что такое атом?
7. Какие символы не могут встречаться в именах атомов?
8. Что такое точечная пара и список?
9. При каких условиях список (a b c) имеет значение?
10. Как присвоить атому a значение a?
11. Почему не нужно кватировать строки и числа?
12. Бывают ли функции без аргументов и как они вызываются?
13. В чем состоит разница между записью a и (a)?

Задания к практическим работам

Практическая работа №1

Создание программы с графическим интерфейсом

Цель работы: изучение средств создания приложений в Prolog.

Задания:

1. Задать несколько фактов, описываемых предикатами: родитель (string, string), ребенок (string, string), мужчина (string), женщина (string). Создать окно, содержащее два списка. Первый список должен содержать всех людей, заданных предикатами “мужчина”, “женщина”. При выборе элемента первого списка во втором списке должны отображаться его братья и сестры.
2. Аналогично заданию 1 должны отображаться все предки выбранного.
3. Аналогично заданию 1 должны отображаться все потомки выбранного.
4. Аналогично заданию 1 должны отображаться все родители и дети выбранного.
5. Задать несколько фактов вида: человек (фамилия, профессия). Создать окно, содержащее два списка. Первый – список профессий. При выборе элемента во втором списке должны отображаться все люди, владеющие этой профессией.
6. Задать несколько фактов вида: человек (имя, возраст). Создать окно, содержащее два списка. Первый – список всех людей. По нажатию на кнопку самый старший переносится во второй список.
7. Задать несколько фактов вида: страна (название, город), континент (название, страна). Создать окно, содержащее два списка. Первый – список континентов. При выборе элемента во втором списке должны отображаться города этого континента.
8. Задать несколько фактов вида: страна (название, город), континент (название, страна). При выборе города из списка название соответствующего континента должно быть отображено в статическом тексте.

9. Задать несколько фактов вида: город (название, река), море (название, река). При выборе моря из раскрывающегося списка во втором списке должен отображаться перечень городов, стоящих на реках, впадающих в это море.
10. Задать несколько фактов вида: город (наименование, улица). Создать окно, содержащее два списка. Первый – список городов. При выборе элемента во втором списке должны отображаться все улицы этого города.
11. Задать несколько фактов вида: маршрут (город, город, расстояние). Выбирая два города из двух раскрывающихся списков, получить в поле ввода кратчайшее расстояние между ними. Города не обязательно должны быть соседними.
12. Задать несколько фактов вида: человек (имя, [список_детей]). Создать окно, содержащее два списка. Первый – список всех людей. При выборе элемента во втором списке отображать перечень его предков.
13. Задать несколько фактов вида: человек (имя, возраст). Используя предикаты рисования, отобразить графически возраст всех людей.
14. Задать несколько фактов вида: спортсмен (имя, вид_спорта). Используя предикаты рисования, отобразить графически количество людей, занимающихся каждым видом спорта.

Практическая работа №2

Средства создания графического интерфейса

Задания:

1. Написать программу, имеющую в окне поле ввода, кнопку и список. При нажатии на кнопку должны выполняться следующие действия: анализируется целое число, введенное в поле ввода, и выводится список его свойств: четное/нечетное, простое/непростое, положительное/отрицательное.
2. Написать программу, имеющую в окне поле ввода, кнопку и список. При нажатии на кнопку должны выполняться следующие действия: анализируется целое число, введенное в поле ввода, и выводится список его свойств: квадрат целого/не квадрат целого, факториал целого/не факториал целого, круглое/некруглое.
3. Написать программу, имеющую в окне поле ввода, кнопку и список. При нажатии на кнопку должны выполняться следующие действия: анализируется вещественное число, введенное в поле ввода, и выводится список его свойств: целое/дробное, положительное/отрицательное, дробная часть меньше 0,5/больше 0,5.
4. Написать программу, имеющую в окне поле ввода, кнопку и список. При нажатии на кнопку текст, введенный в поле ввода, должен быть разбит на лексемы с помощью предиката `fronttoken`. Лексемы должны быть выведены в список.
5. Написать программу, имеющую в окне два поля ввода и кнопку. При нажатии на кнопку текст, введенный в первое поле ввода, должен быть разбит на лексемы с помощью предиката `fronttoken`. Лексемы должны выводиться во второе поле ввода с паузой в 1 секунду.
6. Написать программу, имеющую Поле_ввода1, Поле_ввода2, Поле_ввода3 и кнопку. Вводя числовые значения в два любых поля ввода, при нажатии на

кнопку следует получить в третьем поле ввода значение согласно формуле $\text{Поле_ввода1} + \text{Поле_ввода2} = \text{Поле_ввода3}$.

7. Написать программу, имеющую Поле_ввода1, Поле_ввода2, Поле_ввода3 и кнопку. Вводя текст в первое поле ввода, при нажатии на кнопку следует получить во втором поле ввода первую лексему, а в третьем поле ввода – остальной текст из первого поля ввода.
8. Написать программу: в первом списке – перечень имен, во второй список при нажатии на кнопку переносятся все мужские или все женские имена из первого списка в зависимости от состояния флажка.
9. Написать программу: в первом списке – перечень городов, во второй список при нажатии на кнопку переносятся все города с миллионным населением или все российские города из первого списка в зависимости от состояния флажка.
10. Написать программу: в первом списке – перечень рек, во второй список при нажатии на кнопку переносятся все реки длиной больше заданной или меньше заданной из первого списка в зависимости от состояния переключателя. Длину задавать в поле ввода.
11. Задать несколько фактов вида: игрок (имя). Вывести в список все возможные пары игроков.
12. Написать программу: в первом списке – перечень имен, во втором списке – все мужские или все женские имена из первого списка (изменяется при нажатии на кнопку).
13. Задать несколько фактов вида: спортсмен (имя, вид_спорта). Вывести в список все возможные пары противников, занимающихся одним видом спорта.

Практическая работа №3

Предикаты элементов управления

Цель работы: изучение предикантов, применение их на практике

Задания:

1. Изучить предикаты win_SetBrush/2 и draw_FloodFill/3. При нажатии на кнопку закрашивать окно цветом, выбираемым из списка.
2. Изучить предикат win_GetChildWindows/1. Написать программу, имеющую кнопку, два поля ввода и список. Вывести в список идентификаторы элементов окна. При выборе элемента передавать ему фокус ввода.
3. Изучить предикаты win_GetFont/1 и win_SetFont/2. Написать программу, имеющую поле ввода с любым текстом и кнопку выбора шрифта для поля ввода.
4. Изучить предикаты win_GetFont/1 и win_SetFont/2. Написать программу, имеющую окно редактирования с возможностью изменения шрифта через меню.
5. Изучить предикат win_GetTextSelection/3. В поле ввода вводить любой текст. При нажатии на кнопку выбранный текст из поля ввода разобрать по символам и вывести символы в список в порядке убывания.
6. Изучить предикат lbox_SetColumnWidth/2. При нажатии на кнопку добавлять в многоколоночный список текст, введенный в поле ввода.

7. Изучить предикат `Ibox_SetTabStops/2`. Вывести в список факты вида: человек(имя, возраст). В зависимости от состояния флажка сортировать список по имени или по возрасту.
8. Изучить события, получаемые окном от мыши (см. Эксперт окон и диалоговых окон). Написать программу, содержащую список, куда будут вноситься все получаемые окном события от мыши.
9. Изучить события, получаемые окном от клавиатуры (см. Эксперт окон и диалоговых окон). Написать программу, содержащую список, куда будут вноситься все получаемые окном события от клавиатуры.
10. Написать программу, содержащую два поля ввода и список, куда будут вноситься все получаемые окном события от полей ввода.
11. Изучить предикаты `win_Move/2` и `dlg_Note/2` и события окна `e_Move` и `e_Size`. Написать программу, которая при изменении размеров или расположения окна будет придавать окну первоначальный вид с выдачей сообщения об изменении

Практическая работа №4

Составление простейшей программы на Прологе.

В задании должны быть рассмотрены два варианта блока `goal`: вне текста программы и внутри текста программы.

Задания:

1. Известна система родственных связей с отношением РОДИТЕЛЬ. Имена в ней не повторяются. Разработать программу логического вывода, позволяющую определить, кто кому приходится отцом, матерью, братом, сестрой, дядей, тетей, дедушкой, бабушкой.

2. Известны следующие факты:

Преподаватель и учитель имеют высшее образование. Преподаватель ВУЗа имеет высшее образование и аспирантуру. Также известно, что

Джон работает дворником.

Том работает учителем.

Глория – продавец.

Майкл – преподаватель ВУЗа.

Билл – инженер.

Мери – учитель.

Лиз – инженер.

Составить программу, отвечающую на следующие вопросы:

Имеет ли Майкл высшее образование?

Закончил ли Том аспирантуру?

Кто имеет высшее образование?

Кем работает Лиз?

Может ли инженер быть преподавателем ВУЗа?

3. На основании известной таксономической системы (а) составить программу, отвечающую на вопросы:

Кто кормит детей молоком?

Является ли собака позвоночным животным?

Плотоядная ли кошка?

Какие семейства относятся к млекопитающим?
Кормят ли молоком животные, имеющие острые зубы?
Относятся ли копытные к позвоночным?
Все ли кто лает, имеют острые зубы?
Имеет ли позвоночник корова?

Таксономическая система (а):

Каждое млекопитающее – позвоночное
Каждое плотоядное – млекопитающее
Корова мычит
Каждое копытное – млекопитающее
Позвончатое имеет позвоночник
Семейство псовых и кошачьих относятся к плотоядным
Кошка мурлычет
Собака лает
Копытное имеет плоские зубы
Кошка относится к семейству кошачьих
Собака относится к семейству псовых
Млекопитающее кормит детей молоком
Плотоядное имеет острые зубы
Корова относится к копытным

4. Составить программу, осуществляющую подбор партнера и партнерши в бальных танцах на основании следующих правил:

Рост мальчика > роста девочки, но не более, чем на 10 см
Танцевальный класс мальчика выше или равен классу девочки

Известно, есть партнеры:

Боб (рост 175, класс «С»)
Энтони (рост 180, класс «В»)
Поль (рост 160, класс «Д»)
Ник (рост 160, класс «Е»)
Фред (рост 176, класс «А»)

Партнерши:

Мэри (рост 172, класс «В»)
Энн (рост 170, класс «С»)
Луиза (рост 165, класс «Д»)
Натали (рост 168, класс «С»)
Клаудиа (рост 158, класс «А»)

Практическая работа №5
Реализация рекурсии на ПРОЛОГе.

Письменный отчет по работе кроме текста задания и исходного текста программы содержит пошаговое описание реализуемой рекурсии для 5 шагов.

Задания:

1. $y = a^n + a^{n-1} + \dots + a^2 + a + 1$, a и n вводятся с клавиатуры

2. $y = \sqrt{10 + \sqrt{10 + \sqrt{10 + \dots + \sqrt{10}}}}$, n –раз

3. $y = na^{n-1} + (n-1)a^{n-2} + \dots + 2a + 1$

4. $y = x^2 + x^4 + x^6 + \dots$

5. $y = \sin(x) + \sin(\sin(x)) + \dots + \sin(\sin(\dots(\sin(x))))$

6. $\frac{\sin(1)}{\cos(1)} + \frac{\sin(1)\sin(2)}{\cos(1)\cos(2)} + \dots + \frac{\sin(1)\sin(2) \cdot \sin(n)}{\cos(1)\cos(2) \cdot \cos(n)}$

7. $\cos(x) + \cos(x^2) + \dots + \cos(x^n)$

Практическая работа №6
Работа со списками на Turbo Prolog.

Цель: Научиться решать типовые задачи со списками, используя принципы логического программирования.

Задания:

1. Дан список из 10 элементов. Упорядочить его в порядке возрастания или убывания.
2. Дан список из 10 элементов. Получить список, циклически сдвинутый на один элемент влево, например, исходный список $[1,2,3,4,5]$, результат $[2,3,4,5,1]$.
3. Перевести список чисел от 1 до 12 в соответствующие названия месяцев.
4. Дан список из 10 элементов. Найти максимальный элемент списка.
5. Дан список из 10 элементов. Выделить все одинаковые элементы списка в отдельный список.
6. Дан список из произвольного числа элементов. Установить является ли количество элементов четным или нет.
7. Дан список из 5 элементов. Установить, является ли список палиндромом: т.е. читается ли он одинаково, как справа налево, так и слева направо, например, $[a,b,c,b,a]$.
8. Дан список из 10 элементов. Подсчитать количество положительных и отрицательных элементов в списке.
9. Дан список из 10 элементов. Найти сумму элементов в списке.
10. Дан список из 10 элементов. Найти количество элементов в списке, принадлежащих заданному диапазону.

Практическая работа №7
Программирование на языке ЛИСП

Задания:

1. Определите функцию возводящую число a в степень n .
2. Определите функцию возвращающую значение факториала числа n .

Практическая работа №8

Программирование на языке ЛИСП

Задания:

1. Определите функцию удаляющую из списка l n-й элемент. Первым элементом списка считается голова списка.
2. Повторите задание 4 для списка с обратной нумерацией.
3. Определите функцию добавляющую в список l элемент k в качестве n-го элемента. Первым элементом считается голова списка.
4. Повторите задание 6 для списка с обратной нумерацией элементов.
5. Определите функцию моделирующую "детскую считалочку". В кругу стоит n человек имеющих разные имена, в "считалочке" m слов. Функция должна возвращать последовательность выбывания игроков из круга.

2.2 Оценочные средства для промежуточного контроля

Вопросы к зачету

1. Понятие процедурного, функционального и логического программирования, их сравнительный анализ.
2. Традиционное программирование и логика Хоара.
3. Соответствие между функциональными и императивными программами.
4. Рекурсивные и примитивно-рекурсивные функции.
5. Рекурсивное программирование.
6. Рекурсивные функции и лямбда-исчисление А.Черча.
7. Программирование в функциональных обозначениях.
8. S-выражения.
9. Анализ структуры термов.
10. Нормальные алгоритмы Маркова.
11. Приемы программирования и организации рекурсии на языке ЛИСП.
12. Реализация S-выражений и лямбда-исчисления в языке ЛИСП.
13. Логика предикатов.
14. Металогические предикаты.
15. Внелогические предикаты.
16. Метод резолюций. SDL-резолюция.
17. Вычислительная модель.
18. Недетерминированное программирование.
19. Неполные структуры данных.
20. Логическая программа: основные конструкции, операционная и декларативная семантика, интерпретация, корректность.
21. Хорновская логическая программа.
22. Синтаксис Пролога.
23. Методы поиска.
24. Обработка нечетких данных.
25. Применения функционального программирования.

26. Применение логического программирования в задачах искусственного интеллекта.

Практические задания для зачета

1. Определить функцию $REPL(n) = (n \ n-1 \ ? \ 2 \ 1)$
2. Определить функцию $maxeven(t)$ вычисления максимума среди четных элементов в линейном списке t из положительных чисел
3. Определить функцию $ISSORT(v)$, где v - линейный список из вещественных чисел. Значением функции является $true$ (T), если линейный список отсортирован отношением $\leq$, и $false$ (NIL) - в противном случае
4. Определить функцию $ISALLEV(v)$, где v - линейный список из целых чисел. Значением функции является $true$ (T), если все элементы линейного списка четные числа, и $false$ (NIL) - в противном случае.
5. Определить функцию $ISALLOD(v)$, где v - линейный список из целых чисел. Значением функции является $true$ (T), если все элементы линейного списка нечетные, и $false$ (NIL) - в противном случае.
6. Определить функцию $FEXP(v,k)$, где v - линейный список из целых чисел, k - целое число ≥ 0 . Значением функции является линейный список $(v_1^1, v_2^2, \dots)$, где v_i^i - возведение элемента v_i в i -ую степень.
7. Определить функцию $EXPMAX(v)$, где v - линейный список из целых чисел. Значением функции является линейный список $(v_1^{max(v)}, v_2^{max(v)}, \dots)$, то есть все элементы списка v возвести в число, которое является максимальным элементом списка v .
8. Определить функцию $countr(s,a)$, значением которой является количество вхождений атома a в линейный список s . Например, $s=(a,b,c,a)$, $countr(s,a)=2$.
9. Запрограммировать функцию $Mult(s,a)$, значением которой является новый список, элементы которого получаются в результате умножения элементов числового списка s на a . Например, $s=(b,c,d)$, $Mult(s,a)=(b*a,c*a,d*a)$
10. Определить функцию $IsL(s,a)$, где $IsL(s,a)=T$, если атом a является последним элементом линейного списка s , и $IsL(s,a)=NIL$, если атом a не является последним элементом линейного списка s . Предполагается, что количество элементов списка $s \geq 1$. Если атом a не входит в список s , то $IsL(s,a)=NIL$. Например, $s=(a,b,c,d)$, $IsL(s,d)=T$, $s=(a,b,c)$, $IsL(s,x)=NIL$, $s=(a,b,c)$, $IsL(s,b)=NIL$

2.3. Итоговая диагностическая работа по дисциплине

ЗАДАНИЯ ДЛЯ ДИАГНОСТИЧЕСКОЙ РАБОТЫ ПО ДИСЦИПЛИНЕ «ФУНКЦИОНАЛЬНОЕ И ЛОГИЧЕСКОЕ ПРОГРАММИРОВАНИЕ»

Номер задания	Правильный ответ *	Содержание вопроса	Компетенция	Код и наименование индикатора достижения компетенции
1.	2	Какое из перечисленных свойств не присуще функциональному программированию как методу организации процессов? эффективность реализации алгоритмов? 1. эффективность реализации алгоритмов 2. близость структур данных к специфике оборудования 3. абстрагирование данных и программ их обработки	ОПК-8	ИД- 3 опк-8 Понимает основные принципы логического и функционального программирования, способен разрабатывать алгоритмы и программы на языках Prolog и LISP, пригодные для практического использования
2	3	Какой метод решения задач особенно характерен для функционального программирования? 1. преобразование состояний памяти 2. обработка сигналов 3. символьная обработка информации	ОПК-8	ИД- 3 опк-8
3	3	Какая особенность функционального подхода дает путь к достижению надежности программ? 1. высокий уровень языковых средств 2. разнообразие встроенных функций и библиотек 3. доказательность основных построений при разработке универсальных функций	ОПК-8	ИД- 3 опк-8
4	2	Решение какой задачи было воспринято как ответ на вопрос о реализуемости искусственного разума? 1. автоматическая проверка доказательств теорем 2. «Элиза» - программа собеседник имитирующая диалог с психоаналитиком 3. программа игры в шахматы 4. Auto-CAD - программный комплекс средств инженерной компьютерной графики	ОПК-8	ИД- 3 опк-8

		для нужд проектирования: расчеты, чертежи и эскизы		
5		Какой из перечисленных механизмов реализации языка Лисп способствовал практическому успеху функционального программирования? 1. автоматизация повторного использования памяти – «сбор мусора» 2. математическая основа исходных понятий 3. отсутствие "синтаксического сахара" в представлении программ	ОПК-8	ИД- 3 опк-8
6	2, 4, 5	Какие атомы, представляющие функции, входят в базис языка Лисп? 1. delete 2. cons 3. subst 4. atom 5. eq 6. assos	ОПК-8	ИД- 3 опк-8
7	1	Какой из перечисленных механизмов языка Лисп унаследован новыми языками программирования, например Java и другие? 1. механизм повторного использования памяти – «сборка мусора» 2. предельно простой синтаксис 3. самоопределение языка программирования	ОПК-8	ИД- 3 опк-8
8	1, 2	Какие структуры данных обрабатываются программами, написанными на языке Лисп? 1. списки и атомы 2. s- выражения 3. вектора 4. таблицы	ОПК-8	ИД- 3 опк-8
9	2	Режим доказательства предиката add(X,Y,Z) при конкретизированных X,Z: 1. Сложение 2. Вычитание 3. построение всех комбинаций 4. проверка верности суммы	ОПК-8	ИД- 3 опк-8
10	4	Укажите синтаксис Mercury для объявления прямой суммы: 1. :- type t == int 2. :- type t == T1->T2 3. :- type t == T1 * T2 4. :- type t --> op1(T1); op2(T2)	ОПК-8	ИД- 3 опк-8

11	3	Определите детерминизм предиката Prolog, у которого максимальное количество решений равно единице и отказ возможен: 1. failure 2. multi 3. semidet 4. det	ОПК-8	ИД- 3 опк-8
12	3	Определите детерминизм предиката Prolog, у которого решений нет и отказ возможен: 1. det 2. semide 3. failure 4. multi	ОПК-8	ИД- 3 опк-8
13		Перечислите основные особенности программирования на языке Lisp?	ОПК-8	ИД- 3 опк-8
14		Раскройте основные функции языка Lisp?	ОПК-8	ИД- 3 опк-8
15		Определить функцию $\text{sumev}(v)$, значением которой является сумма всех четных элементов из списка положительных целых чисел v	ОПК-8	ИД- 3 опк-8
16		В чем заключается связь между математической логикой и программированием?	ОПК-8	ИД- 3 опк-8
17		Приведите пример рекурсии на языке Prolog	ОПК-8	ИД- 3 опк-8