



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Магнитогорский государственный технический университет им. Г.И. Носова»



УТВЕРЖДАЮ  
Директор ИЭиАС  
В.Р. Храмшин

03.03.2021 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)**

***ОСНОВЫ ЛОГИЧЕСКОГО ВЫВОДА ИНФОРМАЦИИ***

Направление подготовки (специальность)  
09.03.01 Информатика и вычислительная техника

Направленность (профиль/специализация) программы  
Проектирование и разработка Web-приложений

Уровень высшего образования - бакалавриат


Форма обучения  
очная

Институт/ факультет	Институт энергетики и автоматизированных систем
Кафедра	Вычислительной техники и программирования
Курс	3
Семестр	5

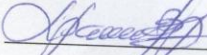
Магнитогорск  
2021 год

Рабочая программа составлена на основе ФГОС ВО - бакалавриат по направлению подготовки 09.03.01 Информатика и вычислительная техника (приказ Минобрнауки России от 19.09.2017 г. № 929)

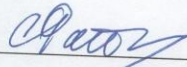
Рабочая программа рассмотрена и одобрена на заседании кафедры Вычислительной техники и программирования 20.01.2021, протокол № 6

Зав. кафедрой  О.С. Логунова

Рабочая программа одобрена методической комиссией ИЭиАС 03.03.2021 г. протокол № 5

Председатель  В.Р. Храмшин

Рабочая программа составлена:  
ст. преподаватель кафедры ВТиП,

 С.И. Файнштейн

Рецензент:

Начальник отдела технологических платформ ООО «Компас Плюс», канд. техн. наук  
Д.С. Сафонов



## Лист актуализации рабочей программы

---

---

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2022 - 2023 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от \_\_\_\_\_ 20\_\_ г. № \_\_\_\_  
Зав. кафедрой \_\_\_\_\_ О.С. Логунова

---

---

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2023 - 2024 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от \_\_\_\_\_ 20\_\_ г. № \_\_\_\_  
Зав. кафедрой \_\_\_\_\_ О.С. Логунова

---

---

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2024 - 2025 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от \_\_\_\_\_ 20\_\_ г. № \_\_\_\_  
Зав. кафедрой \_\_\_\_\_ О.С. Логунова

---

---

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2025 - 2026 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от \_\_\_\_\_ 20\_\_ г. № \_\_\_\_  
Зав. кафедрой \_\_\_\_\_ О.С. Логунова

### 1 Цели освоения дисциплины (модуля)

Целями освоения дисциплины (модуля) «Логическое программирование» является ознакомление студентов с базовыми понятиями и принципами логического программирования и декларативной семантики, формирование представлений о методах и алгоритмах рекурсивного программирования.

Для достижения поставленной цели в курсе «Логическое программирование» решаются задачи:

- изучение логики предикатов первого порядка и ее использование для реализации языка логического программирования;
- изучение языка логического программирования Пролог: синтаксис языка, особенности интерпретации программ, понятие унификации, недетерминированные и детерминированные правила;
- формирование навыков рекурсивного программирования
- формирование навыков решения задач с использованием списков;
- формирование навыков отладки и повышения эффективности логических программ;
- подготовка к изучению интеллектуальных систем.

### 2 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина Основы логического вывода информации входит в часть учебного плана формируемую участниками образовательных отношений образовательной программы.

Для изучения дисциплины необходимы знания (умения, владения), сформированные в результате изучения дисциплин/ практик:

Математическая логика и дискретная математика

Структуры и модели данных

Информатика

Знания (умения, владения), полученные при изучении данной дисциплины будут необходимы для изучения дисциплин/ практик:

Подготовка к сдаче и сдача государственного экзамена

Теория языков программирования

Базы данных Web-приложений

Функциональные языки программирования

### 3 Компетенции обучающегося, формируемые в результате освоения дисциплины (модуля) и планируемые результаты обучения

В результате освоения дисциплины (модуля) «Основы логического вывода информации» обучающийся должен обладать следующими компетенциями:

Код индикатора	Индикатор достижения компетенции
ПК-1	Способность к анализу проблемной ситуации, разработке требований к системе, постановке целей создания, разработке концепции и технического задания на создание Web-приложения, представления концепции, технического задания на Web-приложение и изменений в них заинтересованным лицам
ПК-1.1	Анализирует требования к разработке Web-приложений и базам данных
ПК-1.2	Оценивает качество разработки технических спецификаций на Web-приложения
ПК-1.3	Оценивает качество проекта на разработку Web-приложения и баз данных
ПК-5	Способность к формализации и алгоритмизации поставленных задач, к

написанию программного кода с использованием языков программирования, определения и манипулирования данными и оформлению программного кода в соответствии установленными требованиями	
ПК-5.1	Оценивает качество математической модели при формализации задачи предметной области
ПК-5.2	Оценивает качество разработанных алгоритмов для последующего кодирования
ПК-5.3	Оценивает выбор программных средств для программирования и манипулирования данными в соответствии установленными требованиями



2.1 Стандартные типы данных. Структуры, простые и составные. Альтернативные домены	5	1	2		2	Самостоятельное изучение учебной и научной литературы, работа с электронным учебником, написание программы на языке Prolog	1. Беседа - обсуждение 2. Проверка индивидуальных заданий 3. Устный опрос.	ПК-1.1, ПК-1.2
2.2 Встроенный предикат fail. Встроенный предикат отсечения cut(!). Программирование альтернатив		2	4		4	Самостоятельное изучение учебной и научной литературы, работа с электронным учебником, написание программы на языке Prolog	1. Беседа - обсуждение 2. Проверка индивидуальных заданий 3. Устный опрос.	ПК-1.1, ПК-1.2
2.3 Восходящая и нисходящая рекурсия		2	4		10	Самостоятельное изучение учебной и научной литературы, работа с электронным учебником, написание программы на языке Prolog	1. Беседа - обсуждение 2. Проверка индивидуальных заданий 3. Устный опрос.	ПК-1.1, ПК-1.2
2.4 Списки. Операции со списками. Сортировки списков		2	4		10,05	Самостоятельное изучение учебной и научной литературы, работа с электронным учебником, написание программ на языке Prolog	1. Беседа – обсуждение. 2. Проверка индивидуальных заданий. 3. Устный опрос.	ПК-1.1, ПК-1.2
Итого по разделу		7	14		26,05			
3. Раздел 3. Внелогические средства языка Пролог								
3.1 Стандартные предикаты ввода и вывода, работа с файлами	5	1	4		24	Самостоятельное изучение учебной и научной литературы, работа с электронным учебником, написание программ на языке Prolog	1. Беседа – обсуждение. 2. Проверка индивидуальных заданий. 3. Устный опрос.	ПК-1.1, ПК-1.2

3.2 Базы данных в языке Turbo Prolog. Создание динамической базы данных. Предикаты assert, retract, retractall, save, consult		2	4		18	Самостоятельное изучение учебной и научной литературы, работа с электронным учебни-ком, написание программ на языке Prolog	1. Беседа – обсуждение. 2. Проверка индивидуальных заданий. 3. Устный опрос.	ПК-1.1, ПК-1.2
3.3 Использование динамической базы данных в качестве совокупной глобальной переменной		1	4		14	Самостоятельное изучение учебной и научной литературы, работа с электронным учебни-ком, написание программ на языке Prolog	1. Беседа – обсуждение. 2. Проверка индивидуальных заданий. 3. Устный опрос.	ПК-1.1, ПК-1.2
Итого по разделу		4	12		56			
Итого за семестр		17	34		92,05		зачёт	
Итого по дисциплине		17	34		92,05		зачет	



## **5 Образовательные технологии**

1. Традиционные образовательные технологии, ориентированные на организацию образовательного процесса и предполагающую прямую трансляцию знаний от преподавателя к аспиранту.

Формы учебных занятий с использованием традиционных технологий:

Информационная лекция – последовательное изложение материала в дисциплинарной логике, осуществляемое преимущественно вербальными средствами (монолог преподавателя).

Лабораторная работа – организация учебной работы с реальными материальными и информационными объектами, экспериментальная работа с аналоговыми моделями реальных объектов.

2. Технологии проблемного обучения – организация образовательного процесса, которая предполагает постановку проблемных вопросов, создание учебных проблемных ситуаций для стимулирования активной познавательной деятельности аспирантов.

3. Интерактивные технологии – организация образовательного процесса, которая предполагает активное и нелинейное взаимодействие всех участников, достижение на этой основе лично значимого для них образовательного результата.

Формы учебных занятий с использованием специализированных интерактивных технологий:

Лекция «обратной связи» – лекция–провокация (изложение материала с заранее запланированными ошибками), лекция–беседа, лекция–дискуссия, лекция–конференция.

4. Информационно-коммуникационные образовательные технологии – организация образовательного процесса, основанная на применении программных сред и технических средств работы со знаниями в различных предметных областях.

## **6 Учебно-методическое обеспечение самостоятельной работы обучающихся**

Представлено в приложении 1.

## **7 Оценочные средства для проведения промежуточной аттестации**

Представлены в приложении 2.

## **8 Учебно-методическое и информационное обеспечение дисциплины (модуля)**

### **а) Основная литература:**

1. Сергиевский, Г.М. Функциональное и логическое программирование : учебное пособие [Текст]. / Г. М. Сергиевский, Н. Г. Волченков – М. : Академия, 2010. 318 с.

2. Миков, А.Ю. Основы логического программирования: учебное пособие [Текст]. / А.Ю. Миков, С.И. Файнштейн – М.: ФГУП НТЦ «Информрегистр», 2017. – № 0321702298.

### **б) Дополнительная литература:**

3. Ездаков, А.Л. Функциональное и логическое программирование : учебное пособие [Текст]. / А.Л. Ездаков. – 2-е изд. – М. : БИНОМ. Лаборатория знаний, 2014. – 119 с.

4. Марков, В.Н. Современное логическое программирование на языке Visual Prolog 7.5: Учебник [Текст]. / В.Н. Марков. – БХВ-Петербург, 2016. – 640 с.

5. Цуканова, Н.И. Теория и практика логического программирования на языке Visual Prolog : Учебное пособие [Текст]. / Н.И. Цуканова, Т.А. Дмитриева. – М.: Горячая линия – Телеком, 2011. – 232 с

**в) Методические указания:**

1. Миков, А.Ю. Основы логического программирования: учебное пособие [Текст]. / А.Ю. Миков, С.И. Файнштейн – М.: ФГУП НТЦ «Информрегистр», 2017. – № 0321702298.

**г) Программное обеспечение и Интернет-ресурсы:**

**Программное обеспечение**

Наименование ПО	№ договора	Срок действия лицензии
MS Windows 7 Professional(для классов)	Д-1227-18 от 08.10.2018	11.10.2021
MS Office 2007 Professional	№ 135 от 17.09.2007	бессрочно

**Профессиональные базы данных и информационные справочные системы**

Название курса	Ссылка

**9 Материально-техническое обеспечение дисциплины (модуля)**

Материально-техническое обеспечение дисциплины включает:

Лекционная аудитория ауд. 282 – Мультимедийные средства хранения, передачи и представления информации;

Компьютерные классы Центра информационных технологий ФГБОУ ВПО «МГТУ им. Г.И. Носова» – Персональные компьютеры, объединенные в локальные сети с выходом в Internet, оснащенные современными программно-методическими комплексами для решения задач в области информатики и вычислительной техники;

Аудитории для самостоятельной работы: компьютерные классы; читальные залы библиотеки – ауд. 282 и классы УИТ и АСУ;

Помещения для самостоятельной работы обучающихся, оснащенных компьютерной техникой с возможностью подключения к сети «Интернет» и наличием доступа в электронную информационно-образовательную среду организации – классы УИТ и АСУ;

Помещения для хранения и профилактического обслуживания учебного оборудования – Центр информационных технологий – ауд. 379.

**Перечень вопросов для подготовки к зачету**

1. Декларативные и процедурные языки программирования.
2. Пролог и логика предикатов. Внешние цели.
3. Управление программой. Подцели. Механизм сопоставления.
4. Внутренние подпрограммы унификации.
5. Структура Пролог-программы. Использование внутренних целей.
6. Сокращенные варианты внутренних запросов. Использование в запросах анонимных переменных.
7. Встроенный предикат *fail*. Механизм возврата после неудачи.
8. Стандартные типы данных в языке Turbo Prolog.
9. Структуры, простые и составные. Использование альтернативных доменов.
10. Предикат отсечения ! (cut). Программирование альтернатив.
11. «Зелёные» и «красные отсечения».
12. Детерминированные и недетерминированные предикаты. Управление выполнением программы с помощью отсечений.

13. Детерминированные и недетерминированные предикаты. Повышение эффективности программы с помощью «красных» отсечений.
14. Предикат отрицания *not*.
15. Методы организации рекурсии. Бесконечная рекурсия. Граничное условие рекурсии.
16. Методы организации рекурсии. Восходящая рекурсия.
17. Методы организации рекурсии. Нисходящая рекурсия.
18. Методы организации рекурсии. Отличия между восходящей и нисходящей рекурсией.
19. Применение списков в программе. Внутреннее представление списков. Метод разделения списка на голову и хвост.
20. Поиск элемента в списке.
21. Конкатенация двух списков.
22. Добавление и удаление элемента в списке.
23. Подсписок. Перестановки списка.
24. Компоновка данных в список. Встроенный предикат *findall*.
25. Сортировка списков методом вставки.
26. Сортировка списков методом разделения на два.
27. «Быстрая» сортировка списка.
28. «Сверхбыстрая» сортировка списка.
29. Турбо-Пролог и реляционные базы данных. Описание предикатов динамических БД.
30. Встроенные предикаты *asserta*, *assertz*, *retract*, *retractall*, *save*, *consult* для работы с динамическими базами данных.
31. Использование динамической базы данных в качестве совокупной глобальной переменной. Накопление результатов с помощью вынуждаемого возврата.

**Перечень заданий для подготовки к зачету**

1. К какой парадигме программирования относится ПРОЛОГ?
  - a. Императивной.
  - b. Декларативной
  - c. Функциональной
  - d. Логической.
2. В ПРОЛОГе синтаксически все объекты данных и отношения представляют собой:
  - a. Термы.
  - b. Объекты.
  - c. Утверждения.
  - d. Элементы.
3. Отношение, связывающее объекты данных, называется:
  - a. Предикатом.
  - b. Функтором.
  - c. Атомом.
  - d. Квантором.
4. Что описывает приведенный ниже фрагмент Пролог-программы?  
*repeat.*  
*repeat:- repeat.*
  - a. Правило, выполняющее повторение.
  - b. Правило повторения, основанное на бэктрекинге.
  - c. Правило повторения, определенное пользователем.
  - d. Конструкция некорректна.
5. Какой предикат вызывает откат (бэктрекинг)?
  - a. False
  - b. Fault
  - c. Fail
  - d. Force

6. Выберите типы переменных, которые используются в Turbo Prolog (может быть несколько вариантов):
- a. symbol
  - b. integer
  - c. float
  - d. double
  - e. file
  - f. real
  - g. list
7. Какой из представленных встроенных предикатов динамической базы данных передоказывается?
- a. retract
  - b. asserta
  - c. retractall
  - d. consult
8. В основу языка логического программирования ПРОЛОГ положена ...
- a. модель правил базы знаний;
  - b. модель логических рассуждений на основе базы знаний;
  - c. модель эксперта;
  - d. логическая модель структуры базы знаний;
  - e. нет правильного ответа.
9. Если Иван брат моего отца, то это мой дядя. Это ...
- a. факт;
  - b. правило;
  - c. цель;
  - d. механизм вывода;
  - e. нет правильного ответа;
10. Переменная в Прологе служит для обозначения
- a. конкретного факта;
  - b. различных фактов;
  - c. конкретной цели;
  - d. различных правил;
  - e. различных объектов.
11. В каких случаях в ПРОЛОГЕ употребляется конъюнкция?
- a. в сложных запросах;
  - b. в теле правил;
  - c. в теле правил и в сложных запросах;
  - d. в фактах;
  - e. во всех случаях в ПРОЛОГЕ употребляется только дизъюнкция.
12. Какая директива в Turbo Prolog используется при отладке программы для трассирования?
- a. tracet
  - b. traceroute
  - c. debug
  - d. trace
  - e. tracing
13. Отсечения, которые меняют процедурное поведение программы, но не ее декларативный смысл, называются ...:
14. Отсечения, которые меняют декларативный смысл программы, называются ...:
15. Правило, содержащее само себя в качестве компоненты, называется правилом ... .

16. Правило `read_a_char` демонстрирует простое правило рекурсии, в которое включено условие выхода. Программа циклически считывает символ, введенный пользователем: если этот символ не \$, то он выдается на экран, если этот символ – \$, то программа завершается.

```
read_a_char :-
    readchar(Ch),
    _____,
    write(Ch),
    read_a_char.
```

Заполните пустую строку, чтобы правило работало корректно.

17. Ниже записано правило. Что оно реализует? Запишите последовательность цифр, полученную в случае, если будет сформулирована следующая цель: `write_number(5)`.

```
write_number(8).
write_number(Number) :-
    Number < 8,
    write(Number),
    Next_Number = Number + 1,
    write_number(Next_number).
```

18. Установите соответствие элемента структуры Пролог-программы и его описания:

Раздел	Описание раздела
1. constants	а. описания предикатов динамической базы данных
2. domains	б. определение внутренней цели
3. database	с. описание констант
4. predicates	д. содержит факты и правила
5. goal	е. описание используемых программой предикатов
6. clauses	ф. определение типов данных
	г. определение внешней цели
	h. определение глобальных доменов

19. Имеется следующая Пролог-программа:

```
domains
    thing = book(author,title) ;
    record(artist,album)
    name, author, title, artist, album = symbol
predicates
    owns(name, thing)
clauses
    owns(kahn, book("The Computer and the Brain", "von Neumann")).
    owns(kahn,book("Symbolic Logic","Lewis Carroll")).
    owns(johnson,book("Database: A Primer","C.J.Date")).
    owns(johnson,book("Problem-Solving Methods in AI", "Nils Nilsson")).
```

```
owns(smith,book("Alice in Wonderland", "Lewis Carroll")).
owns(smith,book("Fables of Aesop", "Aesop-Calder")).
owns(bill,book("J.R.R. Tolkein", "Return of the Ring")).
owns(bill,record("Elton John", "Ice Fair")).
```

в данной программе содержится описание коллекций книг и аудиозаписей у владельцев. Сформулируйте внутреннюю цель для вывода всех книг, имена их владельцев узнавать не требуется.

Замечание. Использовать анонимные переменные.

20. Ниже записан фрагмент Пролог-программы для определения максимума из двух чисел (например: запрос  $\text{max}(4,2,4)$  – максимум из 4 и 2 равен 4?):

```
max(X, Y, M):-
  Y>=X, !, ###.
max(X, Y, X):-
  Y < X.
```

Что необходимо записать вместо ###, чтобы программа выдавала верный результат?

21. См. фрагмент кода в предыдущем задании. Какой тип отсечения в нем используется. Ответ запишите одним словом.

22. Ниже записан фрагмент Пролог-программы для подсчета суммы ряда целых чисел от 1 до N:

```
sum_series2(1,1) :- !.
sum_series2(Number,Sum) :-
  Next_number = Number - 1,
  sum_series2(Next_number,Partial_Sum),
  Sum = Number + Partial_sum.
```

Какой тип рекурсии реализован?

23. Иногда, при программировании определенных задач, возникает необходимость собрать данные из базы данных в список для последующей их обработки. Турбо-Пролог содержит встроенный предикат, позволяющий справиться с этой задачей. Требуемый список представляется означенной переменной, являющейся одним из объектов предиката. Запишите этот предикат.

24. Соотнесите встроенные предикаты динамической базы данных и их описание.

Предикат	Описание предиката
1. asserta(Clause)	a. удаление из резидентной части динамической БД одного из ранее внесенных туда утверждений.
2. assertz(Clause)	b. перенос резидентной части динамической БД в долговременную
3. retract(Clause)	c. Занесение нового факта в резидентную часть динамической БД перед всеми уже внесенными утверждениями данного предиката
4. save(file_name)	d. Занесение нового факта в резидентную часть динамической БД после всех уже

	внесенных утверждений данного предиката
--	---

25. Ниже приведена Пролог-программа. Какой будет ответ системы на запрос: `sister(beth, X)`?

Почему? Запишите через запятую все значения, которые принимает X.

*predicates*

```
sister(symbol,symbol)
parent(symbol,symbol)
brother(symbol,symbol)
male(symbol)
female(symbol)
```

*clauses*

```
parent(mary,beth).
parent(mary,bob).
parent(tom,beth).
parent(tom,bob).
parent(tom,liz).
parent(bob,din).
parent(bob,pat).
parent(pat,jim).
male(din).
male(bob).
male(tom).
male(jim).
female(pat).
female(mary).
female(beth).
female(liz).
sister(X,Y):-
    parent(Z,X),
    parent(Z,Y),
    female(X),
    X<>Y.
```

26. Напишите программу, которая запрашивает у пользователя слово, затем букву, и удаляет все вхождения данной буквы во введенном слове.

27. Напишите программу, которая запрашивает список целых чисел и печатает его в обратном порядке.

28. Дополните меню программу «Партийная жизнь»:

- а) новой функцией, осуществляющей просмотр всех членов партии;
- б) новой функцией, показывающей сумму всех сданных членских взносов.

*/\* Программа Партийная жизнь.*

*Назначение: Демонстрация работы с динамической базой данных. База данных допускает следующие операции: добавление, удаление и выборку данных. Выборка включает просмотр данных. Замечание: эта программа:*

- создает динамическую базу данных,
- считывает статическую базу данных в динамическую,

— загружает динамическую базу данных из party.dba,  
— сохраняет динамическую базу данных в файле party.dba в текущем каталоге. \*/

domains

name, payment = symbol

age, pay = integer

database dmember\_party(name, age, pay, payment)

predicates

repeat

do\_mbase

assert\_database

clear\_database

del\_statbase

menu

process(integer)

member\_party(name, age, pay, payment)

error

goal

do\_mbase.

/\* Загрузка фактов статической БД в динамическую \*/

assert\_database :-

member\_party(Name, Age, Pay, Flag),

assertz(dmember\_party(Name, Age, Pay, Flag)),

fail.

assert\_database :- !.

/\* Очистка динамической БД \*/

clear\_database :-

retractall(dmember\_party(\_,\_,\_,\_)).

/\* Удаление фактов статической БД из динамической \*/

del\_statbase :-

member\_party(N,\_,\_,\_),

retract(dmember\_party(N,\_,\_,\_)),

fail.

del\_statbase :- !.

/\* Диалог с этой БД осуществляется по принципу меню.

Меню можно расширить за счет включения новых функций. \*/

do\_mbase :-

assert\_database,

makewindow(1,23,7," PRO PARTY DATABASE ",0,0,25,80),

menu,



*clear\_database.*

*menu :-*

```
repeat,
clearwindow,
write("*****"), nl,
write(" 1.Load database      *"),
nl,
write(" 2.Add a member_party to database  *"),
nl,
write(" 3.Delete a member_party from database *"),
nl,
write(" 4.View a member_party from database  *"),
nl,
write(" 5. Save database      *"),
nl,
write(" 6. Quit from this program      *"),
nl,
write("*****"),
nl,
write("Please enter your choice – 1,2,3,4,5 or 6:"),
readint(Choice),nl,
process(Choice),
Choice = 6, !.
/* Загрузка базы данных из файла */
process(1) :-
    consult("party.dba"), !.

/* Добавление информации о новом члене в БД */
process(2) :-
    makewindow(2,23,7," Add member to DATABASE ",2,20,18,58),
    shiftwindow(2),
    write("Enter member_party name: "),
    readln(Name),
    write("Enter member_party age: "),
    readint(Age),
    write("Enter party dues: "),
    readint(Rub),
    write("Enter answer 'y' or 'n' about payment: "),
    readln(Answer),
    assertz(dmember_party(Name,Age,Rub,Answer)),
    write(Name," has been added to the database."), nl,
    write("Press space bar. "),
    readchar(_),
    removewindow.
```

*/\* Удаление информации о члене партии из БД \*/*

*process(3) :-*

```
    makewindow(3,23,7," Delete member from DATABASE ",10,30,7,40),
    shiftwindow(3),
    write("Enter name to DELETE: "),
    readln(Name),
    retract(dmember_party(Name,_,_)),
    write(Name," has been deleted from the database"), nl, !,
    write("Press space bar"),
    readchar(_),
    removewindow.
```

*/\* Просмотр информации о члене партии \*/*

*process(4) :-*

```
    makewindow(4,23,7," View Window ", 7,30,16,47),
    shiftwindow(4),
    write("Enter name to view: "),
    readln(Name),
    dmember_party(Name,Age,Rub,Yes),
    nl, write(" PARTY MEMBER "),nl,
    nl, write(" Party Name : ",Name),
    nl, write(" Age : ",Age),
    nl, write(" Pay : ",Rub),
    nl, write(" Payment : ",Yes),
    nl, nl, !,
    nl, write("Press space bar"),
    readchar(_),
    removewindow.
```

*process(4) :-*

```
    makewindow(5,23,7," No Luck ",14,7,5,60),
    shiftwindow(5),
    write( "Can't find that member of party in the database."),
    nl, write("Sorry, bye!"), nl, !,
    write("Press space bar"),
    readchar(_),
    removewindow,
    shiftwindow(1).
```

*/\* Сохранение динамической БД в файле "party.dba" \*/*

*process(5) :-*

```
    write("Are you want to save database (y/n)",
    readln(Answer),
```

```

frontchar(Answer,'y',_),
del_statbase,
save("party.dba"), !.

/* Выход из программы */
process(б) :-
    write("Are you sure want to quit (y/n)"),
    readln(Answer),
    upper_lower(Answer),
    frontchar(Answer,'y',_), !.

/* Неправильное обращение к БД */
process(Choice) :-
    Choice < 1,
    error.
process(Choice) :-
    Choice > 6,
    error.

error :-
    write("Please enter a number from 1 to 6."),
    write("(Press the space bar to continue)"),
    readchar(_).

```

### **Задания для оценки сформированности компетенций**

Проверяемая компетенция ПК-1

Задания:

1. К какой парадигме программирования относится ПРОЛОГ?
  - а) Императивной.
  - б) Декларативной.
  - в) Функциональной.
  - г) Объектно-ориентированной.
  
2. Какой стандартный предикат вызывает откат (включает механизм возврата)?
  - а) False.
  - б) Fault.
  - в) Fail.
  - г) Force.
  
3. Если Иван брат моего отца, то это мой дядя. Это ...

- а) Факт.
- б) Правило.
- в) Цель.
- г) Механизм вывода.

Проверяемая компетенция ПК-5

Задания:

4. Какая структура используется для хранения данных в пролог-программе?

- а) Список.
- б) Массив.
- в) Запись.
- г) Указатель.

5. Что такое детерминированная процедура?

- а) Ограниченная по размеру используемой памяти.
- б) Ограниченная по времени использования.
- в) Процедура, возвращающая при возврате только одно решение.
- г) Процедура, возвращающая при возврате все решения.

6. Что такое недетерминированная процедура?

- а) Не ограниченная по размеру используемой памяти.
- б) Не ограниченная по времени использования.
- в) Процедура, возвращающая при возврате только одно решение.
- г) Процедура, возвращающая при возврате все решения.

**Ключ к заданию для оценки сформированности компетенций**

Шифр компетенции	ПК-1			ПК-5		
	1	2	3	4	5	6
№ вопроса						
Правильный вариант ответа	а	в	г	а	в	г

**Критерии оценивания:**

ПК-1:

0 правильных ответов – «неудовлетворительно»,

1 правильный ответ – «удовлетворительно»,

2 правильных ответа – «хорошо»,

3 правильных ответа – «отлично».

ПК-5:

0 правильных ответов – «неудовлетворительно»,

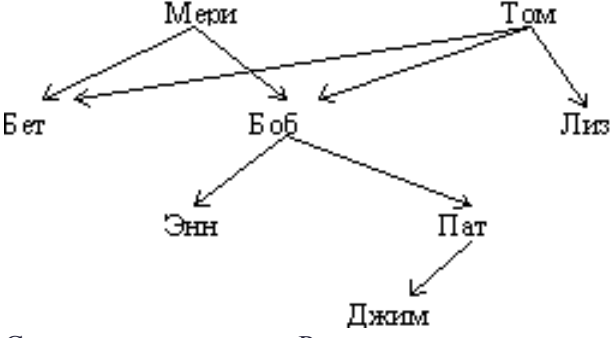
1 правильный ответ – «удовлетворительно»,

2 правильных ответа – «хорошо»,

3 правильных ответа – «отлично».

Код индикатора	Индикатор достижения компетенции	Оценочные средства
<b>ПК-1: Способность к анализу проблемной ситуации, разработке требований к системе, постановке целей создания, разработке концепции и технического задания на создание Web-приложения, представления концепции, технического задания на Web-приложение и изменений в них заинтересованным лицам</b>		
ПК-1.1:	Анализирует требования к разработке Web-приложений и базам данных	Перечень теоретических вопросов 1. Применение списков в программе. Внутреннее представление списков. Метод разделения списка на голову и хвост. 2. Поиск элемента в списке. 3. Конкатенация двух списков. 4. Добавление и удаление элемента в списке. 5. Подсписок. Перестановки списка. 6. Компоновка данных в список. Встроенный предикат findall. 7. Сортировка списков. Практические задания 1. Соберите в одну программу все известные вам правила работы со списками применительно к спискам из целых чисел.
ПК-1.2	Оценивает качество разработки технических спецификаций на Web-приложения	Перечень теоретических вопросов 1. Турбо-Пролог и реляционные базы данных. Описание предикатов динамических БД. 2. Встроенные предикаты asserta, assertz, retract, retractall, save, consult для работы с динамическими базами данных. 3. Использование динамической базы данных в качестве совокупной глобальной переменной. Накопление результатов с помощью вынуждаемого возврата. Практические задания 1. Дополните меню программу «Партийная жизнь»: а) новой функцией, осуществляющей просмотр всех членов партии; б) новой функцией, показывающей сумму всех сданных членских взносов.

ПК-1.3	Оценивает качество проекта на разработку Web-приложения и баз данных	<p>Перечень теоретических вопросов</p> <ol style="list-style-type: none"> <li>1. Предикат отсечения ! (cut). Программирование альтернатив.</li> <li>2. «Зелёные» и «красные отсечения».</li> <li>3. Детерминированные и недетерминированные предикаты. Управление выполнением программы с помощью отсечений.</li> <li>4. Детерминированные и недетерминированные предикаты. Повышение эффективности программы с помощью «красных» отсечений.</li> </ol> <p>Практические задания</p> <ol style="list-style-type: none"> <li>1. Протрассировать выполнение программы «Родственники» с внутренней целью goal sister(beth, X), write(X), nl, fail. Почему bob печатается два раза, а liz один?</li> <li>2. Имеется база данных о результатах партий теннисного матча, которые представлены в программе в виде фактов типа win(tom, john), на первом месте победитель, на втором – проигравший. Определить отношение class, которое будет распределять игроков по категориям: profi – победитель всех сыгранных им матчей; player – выиграл и проиграл хотя бы одну игру; loser – проиграл все матчи; absent – отсутствует в базе данных. Напишите программу двумя способами. В первом способе используйте предикат not и не используйте красные отсечения. Во втором способе, напротив, не пользуйтесь предикатом not, а используйте красные отсечения. Какая программа будет более эффективной? В какой программе нарушается её декларативный смысл? Почему?</li> </ol>
<b>ПК-5:</b> Способность к формализации и алгоритмизации поставленных задач, к написанию программного кода с использованием языков программирования, определения и манипулирования данными и оформлению программного кода в соответствии установленными требованиями		
ПК-5.1	Оценивает качество математической модели при формализации задачи предметной области	<p>Перечень теоретических вопросов</p> <ol style="list-style-type: none"> <li>1. Декларативные и процедурные языки программирования.</li> <li>2. Пролог и логика предикатов. Внешние цели.</li> <li>3. Управление программой. Подцели. Механизм сопоставления.</li> <li>4. Внутренние подпрограммы унификации.</li> <li>5. Структура Пролог-программы. Использование</li> </ol> <p>Практические задания</p> <ol style="list-style-type: none"> <li>1. Генеалогическое дерево имеет следующий вид:</li> </ol>

Код индикатора	Индикатор достижения компетенции	Оценочные средства
		 <pre> graph TD     M[Мэри] --&gt; B[Бет]     M --&gt; Bb[Боб]     M --&gt; T[Том]     T --&gt; Bb     T --&gt; L[Лиз]     Bb --&gt; A[Энн]     Bb --&gt; P[Пат]     P --&gt; J[Джим] </pre> <p>Составить программу «Родственники», содержащую правила определения отца, матери, бабушки, дедушки, предка, сестры, брата, тёти, дяди, племянника, племянницы. Напечатайте всех родственников девушки по имени Бэт с указанием их родства. Указание. Племянника (племянницу) определять по тётё и по дяде с помощью двух правил.</p>
ПК-5.2	Оценивает качество разработанных алгоритмов для последующего кодирования	<p>Перечень теоретических вопросов</p> <ol style="list-style-type: none"> <li>1. Структура Пролог-программы. Использование внутренних целей.</li> <li>2. Сокращенные варианты внутренних запросов. Использование в запросах анонимных переменных.</li> <li>3. Встроенный предикат fail. Механизм возврата после неудачи.</li> </ol> <p>Практические задания</p> <ol style="list-style-type: none"> <li>3. Соберите в одну программу все известные вам правила работы со списками применительно к спискам из целых чисел.</li> <li>4. Дополните меню программу «Партийная жизнь»:       <ol style="list-style-type: none"> <li>а) новой функцией, осуществляющей просмотр всех членов партии;</li> <li>б) новой функцией, показывающей сумму всех сданных членских взносов.</li> </ol> </li> </ol>
ПК-5.3	Оценивает выбор программных средств для программирования и манипулирования данными в соответствии установленными требованиями	<p>Перечень теоретических вопросов</p> <ol style="list-style-type: none"> <li>1. Методы организации рекурсии. Бесконечная рекурсия. Граничное условие рекурсии.</li> <li>2. Методы организации рекурсии. Восходящая рекурсия.</li> <li>3. Методы организации рекурсии. Нисходящая рекурсия.</li> <li>4. Методы организации рекурсии. Отличия между восходящей и нисходящей рекурсией.</li> </ol> <p>Практические задания</p> <ol style="list-style-type: none"> <li>1. Подсчитать сумму чисел от 1 до 7 восходящей и нисходящей рекурсией:       <ol style="list-style-type: none"> <li>а) на языке Turbo Prolog;</li> <li>б) на любом алгоритмическом языке, поддерживающем рекурсию.</li> </ol> </li> <li>2. Напечатать сумму ряда <math>\sum_{n=1}^{\infty} \frac{(-2)^n}{n!}</math>, вычисленную с заданной точностью eps = 0.001 (сумму вычислить и восходящей, и нисходящей рекурсией).</li> </ol>

