

*Федеральное государственное автономное образовательное учреждение высшего образования
«Российский университет дружбы народов»*

Факультет физико-математических и естественных наук

Рекомендовано МССН
02.00.00 «Компьютерные и
информационные науки»

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Наименование дисциплины

Логическое программирование

Рекомендуется для направления подготовки

02.03.02 — Фундаментальная информатика и информационные технологии

Квалификация (степень) выпускника бакалавр

1. Цели и задачи дисциплины:

Целью дисциплины является знакомство слушателей с основными принципами парадигмы логического программирования, формирование навыков решения прикладных и научных задач с использованием инструментальных средств логического программирования.

В рамках дисциплины решаются следующие задачи: изучение теоретических аспектов парадигмы логического программирования и ее применения в задачах искусственного интеллекта, изучение правил и конструкций языка Prolog (синтаксиса, структур и типов данных, предикатов, фактов, правил), формирование навыков создания программ на языке логического программирования.

2. Место дисциплины в структуре ОП ВО:

Дисциплина Логическое программирование относится к части, формируемой участниками образовательных отношений, блока Б1 учебного плана.

В таблице № 1 приведены предшествующие и последующие дисциплины, направленные на формирование компетенций дисциплины в соответствии с матрицей компетенций ОП ВО.

Таблица № 1

Предшествующие и последующие дисциплины, направленные на формирование компетенций

№ п/п	Шифр и наименование компетенции	Предшествующие дисциплины	Последующие дисциплины (группы дисциплин)
Универсальные компетенции			
	УК-12	Математическая логика	Компьютерный практикум по статистическому анализу данных / Компьютерный практикум по интеллектуальным системам
Общепрофессиональные компетенции			
	ОПК-2; ОПК-3; ОПК-6	Математическая логика и комбинаторные алгоритмы, Основы программирования, Технология программирования, Алгоритмы и анализ сложности	Модуль «Научные исследования в области искусственного интеллекта», Компьютерный практикум по статистическому анализу данных / Компьютерный практикум по интеллектуальным системам
Профессиональные компетенции (вид профессиональной деятельности – производственно-технологическая деятельность)			
	ПК-1	Основы программирования, Технология программирования, Алгоритмы и анализ сложности	Модуль «Научные исследования в области искусственного интеллекта», Компьютерный практикум по статистическому анализу данных / Компьютерный практикум по интеллектуальным системам
Профессионально-специализированные компетенции специализации			
	-	-	-

3. Требования к результатам освоения дисциплины:

Процесс изучения дисциплины направлен на формирование следующих компетенций: УК-12; ОПК-2; ОПК-3; ОПК-6; ПК-1

УК-12. Способен: искать нужные источники информации и данные, воспринимать, анализировать, запоминать и передавать информацию с использованием цифровых средств, а также с помощью алгоритмов при работе с полученными из различных источников данными с целью эффективного использования полученной информации для решения задач; проводить оценку информации, ее достоверность, строить логические умозаключения на основании поступающих информации и данных.

- УК-12.1 Способен: искать нужные источники информации и данные, воспринимать, анализировать, запоминать и передавать информацию с использованием цифровых средств, а также с помощью алгоритмов при работе с полученными из различных источников данными с целью эффективного использования полученной информации для решения задач; проводить оценку информации, ее достоверность, строить логические умозаключения на основании поступающих информации и данных

ОПК-2. Способен применять компьютерные/суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности.

- ОПК-2.1 Знает основные положения и концепции в области программирования, архитектуру языков программирования, знает основную терминологию, знаком с содержанием Единого Реестра Российских программ
- ОПК-2.2 Умеет анализировать типовые языки программирования, составлять программы
- ОПК-2.3 Имеет практический опыт решения задач анализа, интеграции различных типов программного обеспечения

ОПК-3. Способен к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям.

- ОПК-3.1 Знает методы теории алгоритмов, методы системного и прикладного программирования, основные положения и концепции в области математических, информационных и имитационных моделей
- ОПК-3.2 Умеет соотносить знания в области программирования, интерпретацию прочитанного, определять и создавать информационные ресурсы глобальных сетей, образовательного контента, средств тестирования систем

ОПК-3.3 Имеет практический опыт применения разработки программного обеспечения

ОПК-6. Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности.

- ОПК-6.1 Знает базовые принципы цифровых технологий и методов, необходимых в профессиональной деятельности в области фундаментальной информатики и информационных технологий для: изучения и моделирования объектов профессиональной деятельности, анализа данных, представления информации и пр.
- ОПК-6.2 Умеет применять необходимые в профессиональной деятельности цифровые технологии и методы в области фундаментальной информатики и

информационных технологий для: изучения и моделирования объектов профессиональной деятельности, анализа данных, представления информации и пр.

- ОПК-6.3 Владеет необходимыми в профессиональной деятельности технологиями и методами в области фундаментальной информатики и информационных технологий для: изучения и моделирования объектов профессиональной деятельности, анализа данных, представления информации и пр.

ПК-1 Способен разрабатывать и отлаживать программный код

- ПК-1.1 Знает основы программирования; современные объектно-ориентированные языки программирования; современные структурные языки программирования; языки современных бизнес-приложений
- ПК-1.2 Умеет кодировать на языках программирования; тестировать результаты кодирования
- ПК-1.3 Владеет навыками разработки кода информационной системы; навыками верификации кода информационной системы

В результате изучения дисциплины студент должен:

Знать:

- Основные принципы и понятия парадигмы логического программирования, ее отличия от других парадигм программирования;
- Принципы работы и организации данных логической программы, понятия интенциональной и экстенциональной баз программы;
- Основные конструкции языка Prolog – синтаксис, структуры и типы данных, понятия предиката и дизъюнкции Хорна, встроенные предикаты, факты, правила;
- Общий алгоритм написания программ на языках логического программирования при решении различных задач.

Уметь:

- Использовать инструментальные средства и интегрированные среды разработки для создания программ на языках логического программирования;
- Осуществлять анализ поставленной задачи с точки зрения выделения основных объектов предметной области, их характеристик, связей и типов связей между объектами;
- Применять в исследовательской и прикладной деятельности средства логического программирования для решения поставленных задач.

Владеть:

- Навыками разработки прикладных и научных программ на языках логического программирования;
- Способностью создавать комплексные программы на языке Prolog с использованием средств построения графического интерфейса пользователя;
- Навыками работы в программной системе Eclipse.

4. Объем дисциплины и виды учебной работы

Общая трудоемкость дисциплины составляет 4 зачетных единиц.

Вид учебной работы	Всего часов	Семестры
		4
Аудиторные занятия (всего)	36	36
<i>Лекции</i>		
<i>Практические занятия (ПЗ)</i>		

Семинары (С)		
Лабораторные работы (ЛР)	36	36
Самостоятельная работа (всего)	108	108
Общая трудоемкость (часов)	144	144
(зач. ед.)	4	4

5. Содержание дисциплины

5.1. Содержание разделов дисциплины

№ п/п	Наименование раздела дисциплины	Содержание раздела
1.	Введение в логическое программирование	1. Обзор парадигмы логического программирования: область применения, история, отличия от других парадигм. 2. Исчисление предикатов первого порядка. Дизъюнкция Хорна. Метод резолюций. Факты, правила. 3. Синтаксис языка Prolog. Работа в интерпретаторе Prolog Inference Engine.
2.	Основы языка Prolog	1. Схема работы логической программы. Константы, переменные, арифметические выражения, ввод-вывод. 2. Управление выполнением логической программы: последовательное выполнение, ветвление, цикл. 3. Динамическое преобразование базы фактов. Обработка массивов данных, разработка агрегатных функций.
3.	Продвинутое программирование в Prolog	1. Рекурсия: восходящая, нисходящая. Циклы с использованием рекурсии. Применение рекурсии для решения задач. 2. Встроенные предикаты управления логической программой. Отсечение, fail, succeed. 3. Структуры данных. Множественные типы данных. 4. Списки в Prolog. Унификация списков. Операции со списками. Составные списки. Решение задач с использованием списков.
4.	Комплексные задачи. Графический интерфейс пользователя	1. Решение задач ИИ на языке Prolog. Описание предметной области задачи. Формирование пространств состояний и алгоритмов решений. 2. Графический интерфейс пользователя в Prolog. Типы интерфейсов. Создание интерфейса. Обработка событий. Работа с файлами данных.

5.2. Разделы дисциплин и виды занятий

№ п/п	Наименование раздела дисциплины	Лекц.	Практ. зан.	Лаб. зан.	Семин	СРС	Всего час.
1.	Введение в логическое программирование			4		12	16
2.	Основы языка Prolog			8		30	38

3.	Продвинутое программирование в Prolog			12		36	48
4.	Комплексные задачи. Графический интерфейс пользователя			12		30	42
	Итого			36		108	144

6. Лабораторный практикум

№ п/п	№ раздела дисциплины	Темы лабораторных работ	Трудоемкость (час.)
1.	1	Описание заданной предметной области на языке Prolog с использованием PIE. Выделение объектов и связей между ними.	4
2.	2	Реализация описания предметной области в среде Prolog	4
3.	2	Расширение описания предметной области арифметическими выражениями, функциями; динамическая работа с данными	4
4.	3	Применение рекурсивного подхода для решения логических и алгебраических задач	6
5.	3	Расширение описания предметной области структурами данных и списками, организация сложных вычислений при помощи рекурсии	6
6.	4	Реализация алгоритма поиска решений задачи искусственного интеллекта на основе логики и графа состояний	6
7.	4	Разработка программы с графическим интерфейсом пользователя как фасада для разработанного ранее описания предметной области	6

7. Практические занятия (семинары) не предусмотрены.

8. Материально-техническое обеспечение дисциплины:

Компьютерные (дисплейные) классы с доступом к сети Интернет и электронно-образовательной среде Университета для выполнения обучающимися заданий лабораторного практикума, выполнения обучающимися самостоятельной работы и проведения компьютерного тестирования обучающихся (при необходимости).

9. Информационное обеспечение дисциплины:

- а) программное обеспечение
 - продукты Microsoft - операционная система, пакет офисных приложений, MS Teams и др. (подписка Enrollment for Education Solutions (EES)).
 - Программное обеспечение со свободной лицензией (free):
 - браузер Chrome (лицензия Google Chrome Terms of Service)
 - Adobe Reader (лицензия Adobe Software License Agreement)
 - офисный пакет LibreOffice (лицензия MPL-2.0)
 - GNU Prolog (лицензия GPL v2)
- б) базы данных, информационно-справочные и поисковые системы
 - Сайт библиотеки РУДН <http://lib.rudn.ru/>.
 - Сайт ТУИС <http://esystem.pfur.ru/>.

10. Учебно-методическое обеспечение дисциплины:

а) основная литература:

1. Прыкина, Е.Н. Основы логического программирования в среде Турбо Пролог : учебное пособие / Е.Н. Прыкина ; Федеральное агентство по культуре и кинематографии, Кемеровский государственный университет культуры и искусств, Кафедра технологии автоматизированной обработки информации. - Кемерово : КемГУКИ, 2006. - 68 с. - ISBN 5-8154-0130-7 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=227891>

б) дополнительная литература:

1. Рублев, В.С. Языки логического программирования : учебное пособие / В.С. Рублев. - Москва : Интернет-Университет Информационных Технологий, 2008. - 115 с. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=234653>

11. Методические указания для обучающихся по освоению дисциплины (модуля)

Учебным планом на изучение дисциплины отводится один семестр. В течение семестра выполняются лабораторные работы, контрольные мероприятия. В конце семестра проводится итоговый контроль знаний.

В рамках дисциплины студенты выполняют курсовую работу по одной из тем, представленных в приложении.

11.1. Методические указания по самостоятельному освоению теоретического материала по дисциплине

Теоретический материал дисциплины охватывает темы, указанные в разделе 5.1 программы дисциплины. В ТУИС (<http://esystem.pfur.ru>) по темам лекций размещены дополнительные материалы. Также рекомендуется изучать разделы литературы, указанные в рамках каждой темы курса (полный список литературы приведен в п.10 программы).

В ТУИС также размещены прикладные задачи, относящиеся к каждому разделу курса. Рекомендуется проработка реализации таких задач с использованием интерпретатора PIE и среды Prolog.

11.2. Методические указания по выполнению лабораторных работ

Задания по лабораторным работам выполняются индивидуально каждым студентом в дисплейных классах в соответствии с календарным планом и методическими указаниями по выполнению лабораторных работ по дисциплине.

Лабораторные работы направлены на формирование практических навыков у студентов по решению различных задач с использованием логического программирования.

По результатам выполнения каждой лабораторной работы студентом готовится отчет. Отчеты в электронном виде сдаются студентом на проверку через соответствующий раздел ТУИС (<http://esystem.pfur.ru>).

11.3. Методические указания по подготовке к контрольным мероприятиям

Контрольные мероприятия по дисциплине проводятся в форме онлайн-тестирований и оценки результатов выполнения лабораторных работ. Задания (темы) для подготовки к прохождению тестирований размещены в соответствующем разделе ТУИС (<http://esystem.pfur.ru>).

Итоговый контроль в форме опроса проводится по темам всех разделов дисциплины. Вопросы для подготовки к промежуточному и итоговому контролю размещены в соответствующем разделе ТУИС (<http://esystem.pfur.ru>).

12. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине (модулю)

ФОС по дисциплине представлен в приложении к данной программе.
Программа составлена в соответствии с требованиями ОС ВО РУДН.

Зав. кафедрой информационных технологий

Руководитель программы

Заведующий кафедрой
прикладной информатики
и теории вероятностей, д.т.н., проф.



Ю.Н. Орлов



К.Е. Самуйлов

Федеральное государственное автономное образовательное учреждение высшего образования
«Российский университет дружбы народов»

Факультет физико-математических и естественных наук

Кафедра информационных технологий

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

Логическое программирование

(наименование дисциплины)

02.03.02 — Фундаментальная информатика и информационные технологии
(код и наименование направления подготовки)

(наименование профиля подготовки)

бакалавр

Квалификация (степень) выпускника

Паспорт фонда оценочных средств по дисциплине Логическое программирование

Направление: 02.03.02 — Фундаментальная информатика и информационные технологии

Код контролируемой компетенции или ее части	Контролируемый раздел дисциплины	Контролируемая тема дисциплины	ФОСы (формы контроля уровня освоения ООП)			Баллы темы	Баллы раздела
			Аудиторная работа		Экзамен/Зачет		
			Выполнение ЛР	Тест			
УК-12; ОПК-2; ОПК-3; ОПК-6; ПК-1	Раздел 1: Введение в логическое программирование	Тема 1: Обзор парадигмы логического программирования: область применения, история, отличия от других парадигм.		2	1	3	18
		Тема 2: Исчисление предикатов первого порядка. Дизъюнкция Хорна. Метод резолюций. Факты, правила.		2	1	3	
		Тема 3: Синтаксис языка Prolog. Работа в интерпретаторе Prolog Inference Engine.	10	1	1	12	
	Раздел 2: Основы языка Prolog	Тема 1: Схема работы логической программы. Константы, переменные, арифметические выражения, ввод-вывод.		2	1	3	28
		Тема 2: Управление выполнением логической программы: последовательное выполнение, ветвление, цикл.	10	2	1	13	
		Тема 3: Динамическое преобразование базы фактов. Обработка массивов данных, разработка агрегатных функций.	10	1	1	12	
	Раздел 3: Продвинутое программирование в Prolog	Тема 1: Рекурсия: восходящая, нисходящая. Циклы с использованием рекурсии. Применение рекурсии для решения задач.	10	1	1	12	29
		Тема 2: Встроенные предикаты управления логической программой. Отсечение, fail, succeed.		1	1	2	
		Тема 3: Структуры данных. Множественные типы данных.		1	1	2	
		Тема 4: Списки в Prolog. Унификация списков. Операции со списками. Составные списки. Решение задач с использованием списков.	10	2	1	13	

	Раздел 4: Комплексные задачи. Графический интерфейс пользователя	Тема 1: Решение задач ИИ на языке Prolog. Описание предметной области задачи. Формирование пространств состояний и алгоритмов решений.	10	5		15	25
		Тема 2: Графический интерфейс пользователя в Prolog. Обработка событий	10			10	
	ИТОГО:		70	20	10	100	100

Процесс изучения дисциплины направлен на формирование следующих компетенций: УК-12; ОПК-2; ОПК-3; ОПК-6; ПК-1

УК-12. Способен: искать нужные источники информации и данные, воспринимать, анализировать, запоминать и передавать информацию с использованием цифровых средств, а также с помощью алгоритмов при работе с полученными из различных источников данными с целью эффективного использования полученной информации для решения задач; проводить оценку информации, ее достоверность, строить логические умозаключения на основании поступающих информации и данных.

- УК-12.1 Способен: искать нужные источники информации и данные, воспринимать, анализировать, запоминать и передавать информацию с использованием цифровых средств, а также с помощью алгоритмов при работе с полученными из различных источников данными с целью эффективного использования полученной информации для решения задач; проводить оценку информации, ее достоверность, строить логические умозаключения на основании поступающих информации и данных

ОПК-2. Способен применять компьютерные/суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности.

- ОПК-2.1 Знает основные положения и концепции в области программирования, архитектуру языков программирования, знает основную терминологию, знаком с содержанием Единого Реестра Российских программ
- ОПК-2.2 Умеет анализировать типовые языки программирования, составлять программы
- ОПК-2.3 Имеет практический опыт решения задач анализа, интеграции различных типов программного обеспечения

ОПК-3. Способен к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям.

- ОПК-3.1 Знает методы теории алгоритмов, методы системного и прикладного программирования, основные положения и концепции в области математических, информационных и имитационных моделей
- ОПК-3.2 Умеет соотносить знания в области программирования, интерпретацию прочитанного, определять и создавать информационные ресурсы глобальных сетей, образовательного контента, средств тестирования систем

ОПК-3.3 Имеет практический опыт применения разработки программного обеспечения

ОПК-6. Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности.

- ОПК-6.1 Знает базовые принципы цифровых технологий и методов, необходимых в профессиональной деятельности в области фундаментальной информатики и информационных технологий для: изучения и моделирования объектов профессиональной деятельности, анализа данных, представления информации и пр.
 - ОПК-6.2 Умеет применять необходимые в профессиональной деятельности цифровые технологии и методы в области фундаментальной информатики и информационных технологий для: изучения и моделирования объектов профессиональной деятельности, анализа данных, представления информации и пр.
 - ОПК-6.3 Владеет необходимыми в профессиональной деятельности технологиями и методами в области фундаментальной информатики и информационных технологий для: изучения и моделирования объектов профессиональной деятельности, анализа данных, представления информации и пр.
- ПК-1 Способен разрабатывать и отлаживать программный код
- ПК-1.1 Знает основы программирования; современные объектно-ориентированные языки программирования; современные структурные языки программирования; языки современных бизнес-приложений
 - ПК-1.2 Умеет кодировать на языках программирования; тестировать результаты кодирования
 - ПК-1.3 Владеет навыками разработки кода информационной системы; навыками верификации кода информационной системы

Балльно-рейтинговая система оценки уровня знаний

Таблица соответствия баллов и оценок

Баллы БРС	Традиционные оценки РФ	Оценки ECTS
95 - 100	5	A
86 - 94		B
69 - 85	4	C
61 - 68	3	D
51 - 60		E
31 - 50	2	FX
0 - 30		F
51-100	Зачет	Passed

Правила применения БРС

1. Раздел (тема) учебной дисциплины считаются освоенными, если студент набрал более 50 % от возможного числа баллов по этому разделу (теме).
2. Студент не может быть аттестован по дисциплине, если он не освоил все темы и разделы дисциплины, указанные в сводной оценочной таблице дисциплины.
3. По решению преподавателя и с согласия студентов, не освоивших отдельные разделы (темы) изучаемой дисциплины, в течение учебного семестра могут быть повторно проведены мероприятия текущего контроля успеваемости или выданы дополнительные учебные задания по этим темам или разделам. При этом студентам за данную работу засчитывается минимально возможный положительный балл (51 % от максимального балла).
4. При выполнении студентом дополнительных учебных заданий или повторного прохождения мероприятий текущего контроля полученные им баллы засчитываются за конкретные темы. Итоговая сумма баллов не может превышать максимального количества баллов, установленного по данным темам (в соответствии с приказом Ректора № 564 от 20.06.2013). По решению преподавателя предыдущие баллы, полученные студентом по учебным заданиям, могут быть аннулированы.
5. График проведения мероприятий текущего контроля успеваемости формируется в соответствии с календарным планом курса. Студенты обязаны сдавать все задания в сроки, установленные преподавателем.
6. Время, которое отводится студенту на выполнение мероприятий текущего контроля успеваемости, устанавливается преподавателем. По завершении отведенного времени студент должен сдать работу преподавателю, вне зависимости от того, завершена она или нет.
7. Использование источников (в том числе конспектов лекций и лабораторных работ) во время выполнения контрольных мероприятий возможно только с разрешения преподавателя.
8. Отсрочка в прохождении мероприятий текущего контроля успеваемости считается уважительной только в случае болезни студента, что подтверждается наличием у него медицинской справки, заверенной круглой печатью в поликлинике № 25, предоставляемой преподавателю не позднее двух недель после выздоровления. В этом случае выполнение контрольных мероприятий осуществляется после

выздоровления студента в срок, назначенный преподавателем. В противном случае, отсутствие студента на контрольном мероприятии признается не уважительным.

9. Студент допускается к итоговому контролю знаний с любым количеством баллов, набранных в семестре.
10. Итоговая контроль знаний оценивается из 10 баллов независимо от числа баллов за семестр.
11. Если в итоге за семестр студент получил менее 31 балла, то ему выставляется оценка F и студент должен повторить эту дисциплину в установленном порядке. Если же в итоге студент получил 31-50 баллов, т. е. FX, то студенту разрешается добор необходимого (до 51) количества баллов путем повторного одноразового выполнения предусмотренных контрольных мероприятий, при этом по усмотрению преподавателя аннулируются соответствующие предыдущие результаты. Ликвидация задолженностей проводится в период с 07.02 по 28.02 (с 07.09 по 28.09) по согласованию с деканатом.

Примерный перечень оценочных средств

п/п	Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в фонде
<i>Аудиторная работа</i>			
1	Лабораторная работа	Система практических заданий, направленных на формирование практических навыков у обучающихся	Фонд практических заданий
2	Тест	Система стандартизированных заданий (вопросов), позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося.	База тестовых заданий
3	Экзамен	Оценка работы студента в течение семестра (года, всего срока обучения и др.) и призван выявить уровень, прочность и систематичность полученных им теоретических и практических знаний, приобретения навыков самостоятельной работы, развития творческого мышления, умение синтезировать полученные знания и применять их в решении практических задач.	Примеры заданий/вопросов, пример экзаменационного билета

Учебным планом на изучение дисциплины отводится один семестр. В дисциплине предусмотрены лабораторный практикум и тестирования. В конце семестра проводится итоговый контроль знаний.

Оценивание результатов освоения дисциплины производится в соответствии с балльно-рейтинговой системой. По дисциплине предусмотрен экзамен.

Итоговый контроль знаний по дисциплине проводится в форме опроса. При необходимости экзамен может проводиться в форме письменного ответа на вопросы из билетов.

Критерии оценки по дисциплине

Итоговая оценка выставляется по сумме набранных баллов за семь лабораторных работ и четыре тестирования. На промежуточную аттестацию отводится 10 баллов, и она является обязательной для всех студентов.

95-100 баллов:

- полное выполнение лабораторных работ и прохождение тестирований;
- решение всех заданий промежуточной аттестации;
- активное участие в мероприятиях, предусмотренных программой дисциплины;
- систематизированное, глубокое и полное освоение навыков и компетенций по всем разделам программы дисциплины;
- использование научной терминологии, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать обоснованные выводы;
- умение эффективно использовать методику программы дисциплины в постановке и решении научных и профессиональных задач;
- выраженная способность самостоятельно и творчески решать поставленные задачи;
- полная самостоятельность и творческий подход при изложении материала по программе дисциплины;
- полное и глубокое усвоение основной и дополнительной литературы, рекомендованной программой дисциплины и преподавателем.

86- 94 балла:

- полное выполнение лабораторных работ и прохождение тестирований;
- частичное решение заданий промежуточной аттестации;
- участие в мероприятиях, предусмотренных программой дисциплины;
- систематизированное, глубокое и полное освоение навыков и компетенций по всем разделам программы дисциплины;
- использование научной терминологии, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать обоснованные выводы;
- умение эффективно использовать методику программы дисциплины в постановке и решении научных и профессиональных задач;
- способность самостоятельно решать поставленные задачи в нестандартных производственных ситуациях;
- усвоение основной и дополнительной литературы, рекомендованной программой дисциплины и преподавателем.

69-85 баллов:

- частичное выполнение лабораторных работ, частичное прохождение тестирований;
- частичное решение заданий промежуточной аттестации;
- участие в мероприятиях, предусмотренных программой дисциплины;
- систематизированное и полное освоение навыков и компетенций по всем разделам программы дисциплины;
- умение использовать методику программы дисциплины в постановке и решении научных и профессиональных задач;
- усвоение основной литературы, рекомендованной программой дисциплины.

51-68 баллов:

- частичное выполнение лабораторных работ, частичное прохождение тестирований;
- участие в мероприятиях, предусмотренных программой дисциплины;

- систематизированное и полное освоение навыков и компетенций по всем разделам программы дисциплины;
- удовлетворительное умение использовать методику программы дисциплины в постановке и решении научных и профессиональных задач;
- удовлетворительное усвоение основной литературы;

31 - 50 баллов – НЕ ЗАЧТЕНО:

- частичное выполнение лабораторных работ, частичное прохождение тестирований;
- недостаточно полный объем навыков и компетенции в рамках программы дисциплины;
- неумение использовать в практической деятельности научной терминологии, изложение ответа на вопросы с существенными стилистическими и логическими ошибками;
- слабое умение использовать методику программы дисциплины в постановке и решении научных и профессиональных задач;
- удовлетворительное усвоение основной литературы.

0-30 баллов, НЕ ЗАЧТЕНО:

- отсутствие умений, навыков, знаний и компетенции в рамках программы дисциплины;
- невыполнение лабораторных заданий; отказ от ответа по программе дисциплины;
- игнорирование занятий по дисциплине по неуважительной причине.

Комплект экзаменационных билетов

Дисциплина Логическое программирование
(наименование дисциплины)

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №1

1. Записать на языке Prolog следующую теорию (в виде набора фактов и правил):
Вася друг Пети;
Петя друг Вити;
Саша соперник Вити;
Друзья наших друзей – наши друзья;
Соперники наших друзей – наши соперники.
2. На языке Prolog реализовать следующий предикат, согласно его описанию и объявлению:
Предикат, вычисляющий факториал числа методом нисходящей рекурсии:
fact : (integer Number) -> integer Fact determ (i).

Составитель

Заведующий кафедрой

Ю. Н. Орлов

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №2

1. Записать на языке Prolog следующую теорию (в виде набора фактов и правил):
Федор отец Анны;
Василий брат Федора;
Александр отец Василия;
Отец и сын (дочь), братья – это родственники;
Родственник нашего родственника – наш родственник.
2. На языке Prolog реализовать следующий предикат, согласно его описанию и объявлению:
Предикат, вычисляющий N-ое число Фибоначчи методом нисходящей рекурсии:
fib : (integer Number) -> integer Fib determ (i).

Составитель

Заведующий кафедрой

Ю. Н. Орлов

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №3

1. Записать на языке Prolog следующую теорию (в виде набора фактов и правил):
Простуда – серьезное заболевание;
У Ани и Пети насморк;
У Пети температура;
Если у человека насморк и температура, то у него простуда;
Если у человека серьезное заболевание, то он должен пойти к врачу.
2. На языке Prolog реализовать следующий предикат, согласно его описанию и объявлению:
Предикат, возвращающий последний элемент строкового списка:
last : (string List) -> string Last determ (i).*

Составитель

Заведующий кафедрой

Ю. Н. Орлов

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №4

1. Записать на языке Prolog следующую теорию (в виде набора фактов и правил):
Java и C++ – языки программирования;
Математика – точная наука;
Петя знает Java;
Маша знает математику;
Если человек знает язык программирования, то он может работать программистом;
Если человек знает точную науку, то он может работать инженером.
2. На языке Prolog реализовать следующий предикат, согласно его описанию и объявлению:
Предикат, вычисляющий максимальный элемент целочисленного списка:
max : (integer List) -> integer Result determ (i).*

Составитель

Заведующий кафедрой

Ю. Н. Орлов

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №5

1. Записать на языке Prolog следующую теорию (в виде набора фактов и правил):
Петя обучается на 3 курсе;
Петя пишет курсовую на кафедре ИТ;
Вася обучается на 2 курсе;
Вася обучается на кафедре СТ;
Если студент на 3 или 4 курсе и пишет курсовую на некоторой кафедре, то он на ней специализируется;
Если студент специализируется на кафедре, то он защищает на ней дипломную работу.
2. На языке Prolog реализовать следующий предикат, согласно его описанию и объявлению:
Предикат, вычисляющий факториал числа методом восходящей рекурсии:
fact : (integer Number) -> integer Fact determ (i).

Составитель

Заведующий кафедрой

Ю. Н. Орлов

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №6

1. Записать на языке Prolog следующую теорию (в виде набора фактов и правил):
Маше нравятся цветы;
Аня любит сына;
Сын Ани любит Машу;
Оля любит всех, кого любит Аня;
Нам нравится все, что нравится человеку, которого мы любим.
2. На языке Prolog реализовать следующий предикат, согласно его описанию и объявлению:
Предикат, вычисляющий N-ое число Фибоначчи методом восходящей рекурсии:
fib : (integer Number) -> integer Fib determ (i).

Составитель

Заведующий кафедрой

Ю. Н. Орлов

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №7

1. Какие значения, если они существуют, будут присвоены переменным при сопоставлении двух шаблонов (шаблона с выражением):

Шаблон $[X1, X2, X3|Z]$ и списки

$[a,b,c,d,e]$

$[a,b,c,d]$

$[a,b]$

$[a]$

$[]$

2. На языке Prolog реализовать следующий предикат, согласно его описанию и объявлению:

Предикат, последовательно соединяющий два списка:

*$join : (A * FirstList, A * LastList) \rightarrow A * Result (i, i).$*

Составитель

Заведующий кафедрой

Ю. Н. Орлов

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №8

1. Какие значения, если они существуют, будут присвоены переменным при сопоставлении двух шаблонов (шаблона с выражением):

Шаблон $[X1 | [X2 | Z]]$ и списки

$[a,b,c,d,e]$

$[a,b,c,d]$

$[a,b]$

$[a]$

$[]$

2. На языке Prolog реализовать следующий предикат, согласно его описанию и объявлению:

Предикат, находящий корни квадратного уравнения по значениям a, b, c , если они не комплексные:

$domains$

$quaecres = quaecres(real, real).$

$class predicates$

$quaeq : (real A, real B, real C) \rightarrow quaecres Result determ (i,i,i).$

Составитель

Заведующий кафедрой

Ю. Н. Орлов

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №9

1. Какие значения, если они существуют, будут присвоены переменных при сопоставлении двух шаблонов (шаблона с выражением):

Шаблон $[X1, X2, X3]$ и списки

$[a, b, c, d, e]$

$[a, b, c, d]$

$[a, b]$

$[a]$

$[]$

2. На языке Prolog реализовать следующие предикаты, согласно их описанию и объявлению:

Предикат, вычисляющий длину списка:

*$length : (A * List) \rightarrow integer\ Length\ (i).$*

Предикат, вычисляющий сумму элементов целочисленного списка:

*$sum : (integer * List) \rightarrow integer\ Sum\ (i).$*

Составитель

Заведующий кафедрой

Ю. Н. Орлов

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №10

1. Какие значения, если они существуют, будут присвоены переменным при сопоставлении двух шаблонов (шаблона с выражением):

Шаблон $[X1, X2 | []]$ и списки

$[a, b, c, d, e]$

$[a, b, c, d]$

$[a, b]$

$[a]$

$[]$

2. На языке Prolog реализовать следующий предикат, согласно его описанию и объявлению:

Предикат, вычисляющий факториал числа методом нисходящей рекурсии:

$fact : (integer\ Number) \rightarrow integer\ Fact\ determ\ (i).$

Составитель

Заведующий кафедрой

Ю. Н. Орлов

Примерный перечень вопросов для опроса в ходе итогового контроля знаний

по дисциплине Логическое программирование

1. Краткое описание парадигмы логического программирования
2. Основные механизмы языка Prolog
3. Понятие предиката
4. Логика предикатов первого порядка, метод резолюций
5. Хорновский дизъюнкт
6. Объекты и отношения в логической программе
7. Именованние предикатов, переменных и констант. Символическое имя
8. Понятие и представление факта в логической программе
9. Понятие и представление правила в логической программе
10. Понятие утверждения в базе знаний
11. Структура базы знаний логической программы
12. Реляционные операции на Prolog
13. Процесс доказательства цели. И-ИЛИ-дерево
14. Алгоритм работы интерпретатора Prolog
15. Процесс унификации аргументов предикатов
16. Разделы программы в Visual Prolog
17. Консольный ввод и вывод в Visual Prolog
18. Управление выполнением программы. Реализация цепочки на Prolog
19. Управление выполнением программы. Реализация выбора среди альтернатив на Prolog
20. Управление выполнением программы. Реализация цикла на Prolog
21. Описание и назначение предиката fail
22. Назначение раздела domains программы на Visual Prolog
23. Назначение раздела class facts программы на Visual Prolog
24. Назначение раздела class predicates программы на Visual Prolog
25. Назначение раздела clauses программы на Visual Prolog
26. Типизация фактов (single, determ, nondeterm)
27. Типизация предикатов (determ, nondeterm, procedure, multi, failure)
28. Объявление и использование поточности предикатов
29. Области видимости в Visual Prolog
30. Динамическая работа с базой фактов. Предикаты assert, retract
31. Реализация накопление суммы и подсчета произведения элементов на Prolog
32. Описание и назначение предиката repeat
33. Описание и назначение встроенного предиката отсечение (!)
34. Многошаговый итерационный процесс. Способы реализации на Prolog
35. Нисходящая рекурсия в Prolog
36. Восходящая рекурсия в Prolog
37. Структуры данных в Prolog. Определение и использование
38. Понятие списка в Prolog. Формы записи списков
39. Процесс унификации списков в различных формах записи
40. Принципы реализации предикатов по работе со списками (содержит, длина, сумма)
41. Функционал List Comprehension
42. Предикаты, возвращающие значения – объявление и реализация
43. Процесс решения задач поиска в пространстве состояний на Prolog

Комплект заданий для выполнения лабораторных работ

по дисциплине Логическое программирование

Лабораторная работа №1. Описание заданной предметной области на языке Prolog с использованием PIE. Выделение объектов и связей между ними.

Лабораторная работа выполняется в интерпретаторе пролога PIE (Prolog Inference Engine). PIE создан в среде Prolog и включен в пакет примеров, поставляемых со средой; его синтаксис соответствует стандартам Edinburgh/ISO.

Задание: описать указанную предметную область, указать не менее 10 фактов и 3 правил. Привести примеры запросов.

Результатом лабораторной работы является текстовый файл «*.PRO», который используется для консультирования интерпретатора, и набор запросов, выполняемых в диалоговом окне интерпретатора.

Лабораторная работа №2. Реализация описания предметной области в среде Prolog.

Лабораторная работа выполняется в среде Eclipse и предполагает перенос описания предметной области, реализованной ранее в PIE, на язык Prolog Eclipse. Для выполнения работы создается пустой проект типа Console. Программирование осуществляется в файле-реализации класса Main – main.pro.

Задание: трансформировать описание предметной области и запросы к базе знаний, реализованные в рамках лабораторной работы 1, в программу на языке Prolog. Требования:

- весь код должен быть описан в файле main.pro;
- используемые факты и предикаты должны быть предварительно корректно объявлены в соответствующих разделах;
- запросы к базе знаний должны быть реализованы в предикате run.

Результатом лабораторной работы является файл main.pro, описывающий заданную предметную область на языке Prolog.

Программа должна корректно компилироваться и исполняться без ошибок и исключений. Допускаются предупреждения о возможной оптимизации типов предикатов и элиминации избыточных переменных.

Лабораторная работа №3. Расширение описания предметной области арифметическими выражениями, функциями; динамическая работа с данными.

Лабораторная работа выполняется в среде Eclipse и является продолжением лабораторной работы 2.

Задание: расширить описание предметной области, используемой при написании лабораторной работы №1. Добавить необходимые правила для вычисления заданных значений. Требования:

- программа должна быть реализована в среде Eclipse;
- необходимо использовать именованную базу фактов;
- использовать консультирование из файла (динамическая загрузка данных);
- расширить базу фактов минимум до 15. При необходимости внести изменения в структуру описания предметной области;
- реализовать по меньшей мере 4 правила. Одно из этих правил должно использоваться для вычисления значений - суммы, среднего или количества;
- запросы реализовать в предикате run.
- Результатами лабораторной работы являются:
- файл исходного кода главного класса "main.pro";
- файл данных, используемый для консультирования программы (в формате txt).

Программа должна корректно компилироваться и исполняться без ошибок и исключений. Допускаются предупреждения о возможной оптимизации типов предикатов и элиминации избыточных переменных.

Лабораторная работа №4. Применение рекурсивного подхода для решения логических и алгебраических задач.

Лабораторная работа выполняется в среде Eclipse и является обособленной программой, не связанной с определенной ранее предметной областью студента.

Задание: реализовать набор предикатов, осуществляющих решение логических и алгебраических задач с использованием нисходящей и восходящих рекурсий. Требования:

- программа должна быть реализована в среде Eclipse;
- предикаты должны быть реализованы рекурсивным образом;
- запросы реализовать в предикате `run`.

Результатом лабораторной работы является файл исходного кода главного класса "main.pro".

Программа должна корректно компилироваться и исполняться без ошибок и исключений. Допускаются предупреждения о возможной оптимизации типов предикатов и элиминации избыточных переменных.

Возможные варианты заданий в рамках лабораторной работы:

1. Предикат, вычисляющий факториал числа
2. Предикат, вычисляющий значение числа Фибоначчи по его порядковому номеру
3. Предикат, вычисляющий функцию возведения числа в степень
4. Предикат, вычисляющий сумму/среднее набора значений базы знаний
5. Предикат, вычисляющий минимум/максимум набора значений базы знаний
6. Предикат, находящий решение задачи о Ханойской башне
7. Предикат, находящий решение задачи о Ведрах с водой
8. Предикат, находящий решение задачи об Упаковке ранца
9. Предикат, находящий решение задачи о Поиске кратчайшего пути

Лабораторная работа №5. Расширение описания предметной области структурами данных и списками, организация сложных вычислений при помощи рекурсии.

Лабораторная работа выполняется в среде Eclipse и является продолжением лабораторных работ 1, 2 и 3.

Задание: расширить и переделать описание предметной области, используемой при написании предыдущих лабораторных работ, с применением функционала по работе со списками и предикатами, возвращающими значения. Реализовать необходимые правила, применяя предикаты по работе со списками, в том числе:

- функционал List Comprehension при генерации решений в виде списков;
- вывод данных предметной области как поэлементный вывод списка;
- поиск данных предметной области как поиск элементов в списке;
- вычисление количества и максимального / среднего / минимального значений характеристик при помощи обработки списков данных;
- получение значений реализовать через предикаты, возвращающие значения.
- Результатами лабораторной работы являются:
- файл исходного кода главного класса "main.pro";
- файл данных, используемый для консультирования программы (в формате txt).

Программа должна корректно компилироваться и исполняться без ошибок и исключений. Также должны быть устранены все предупреждения о возможной оптимизации, возникающие при запуске программы.

Лабораторная работа №6. Реализация алгоритма поиска решений задачи искусственного интеллекта на основе логики и графа состояний.

Лабораторная работа выполняется в среде Eclipse и является обособленной программой, не связанной с определенной ранее предметной областью студента.

Задание: реализовать предикат, осуществляющий решение некоторой задачи искусственного интеллекта как поиск в пространстве состояний. Требования:

- программа должна быть реализована в среде Eclipse;
- решаемая задача должна быть формализована как набор состояний и возможностей переходов между этими состояниями;
- должен быть реализован один из алгоритмов поиска пути в пространстве состояний;
- в предикате `run` должен быть реализован запрос, представляющий собой постановку задачи.

Результатом лабораторной работы является файл исходного кода главного класса "main.pro".

Программа должна корректно компилироваться и исполняться без ошибок и исключений. Допускаются предупреждения о возможной оптимизации типов предикатов и элиминации избыточных переменных.

Возможные варианты задач, решаемых в рамках лабораторной работы:

- задача о поиске кратчайшего пути в лабиринте
- задача о преобразовании (эквивалентности) математических формул
- задача о наполнении ведер
- задача об упаковке ранца оптимальным образом
- задача о составлении расписания занятий

Лабораторная работа №7. Разработка программы с графическим интерфейсом пользователя как фасада для разработанного ранее описания предметной области.

Лабораторная работа выполняется в среде Eclipse с использованием инструментальных средств разработки графического интерфейса типа MDI. Для выполнения работы создается новый проект. Программирование осуществляется как в файле `main.pro`, так и в других файлах, реализующих работу графического интерфейса.

Задание: разработать графический интерфейс пользователя для реализованного ранее класса `Main`, описывающего предметную область (лабораторная работа №5). Требования:

- программа представляет собой графический интерфейс между пользователем и функционалом, реализованным ранее;
- возможен запуск и получение ответов от всех предикатов в рамках графического интерфейса;
- возможно динамическое внесение значений, необходимых для работы предикатов;
- возможна динамическая загрузка и выгрузка базы фактов через соответствующие пункты меню графического интерфейса.

Результатами лабораторной работы являются:

- Исполняемый файл программы;
- Файл данных, используемый для консультирования программы (в формате `txt`) [Опционально].

Примерный перечень предметных областей для выполнения лабораторных работ

- Товары, клиенты, покупки. Клиенты покупают товары;
- Кинотеатры города, кинофильмы, показы, сеансы;
- Подписка на газеты, журналы, подписчики;
- Прием пациентов в поликлинике, медицинские карты, диагнозы, назначенные лекарства;
- Студенты, группы, факультеты, старосты, стипендии, успеваемость;
- Аптеки, адреса, телефоны, база лекарств – наименования, цена, количество;
- Города, государства, части света, население.

Примерный перечень тестовых заданий

по дисциплине Логическое программирование

1. Для каждого из описанных ниже сопоставлений укажите, унифицируются ли указанные выражения, или нет.
 1. друг(X, "Петя") $\leftrightarrow$ друг(Y, Z)
 2. размеры(12, 15, 18) $\leftrightarrow$ размеры(12, X, X)
 3. обед("Салат", "Суп", X) $\leftrightarrow$ обед("Салат", "Суп")
 4. магазин(1, "Пятерочка", "ул. Земная, 15") $\leftrightarrow$ магазин(Id, "Пятерочка", Addr).
2. Определите тип для каждого из указанных ниже утверждений - факт или правило.
 - a. person(2, "Pam", female).
 - b. father(X, Y) :- parent(Xid, Yid), person(Xid, X, _), person(Yid, Y, male).
 - c. parent(7, 3).
 - d. father("Ann", "Max").
 - e. greater(X, Y) :- Y > X.
 - f. greater(0, _).
3. Пусть в базе фактов имеется набор утверждений s(1), s(2), ..., s(100). Также,
$$L = [\text{result}(Y) \mid s(X), Y = \text{result}(X)] .$$
Какое значение примет переменная L после выполнения запроса?
 - a. [1, 2, ..., 100]
 - b. [s(1), s(2), ..., s(100)]
 - c. [result(1), result(2), ..., result(100)]
 - d. [result(result(1)), result(result(2)), ..., result(result(100))]
4. В логической программе задан набор фактов:
студент(1, "Петя") .
студент(2, "Вася") .
студент(3, "Федя") .
студент(4, "Лена") .
учится_в(1, "НИ-202") .
учится_в(2, "НИ-201") .
учится_в(3, "НИ-201") .
учится_в(4, "НИ-202") .
Какое значение примет переменная X после выполнения следующего выражения:
$$X = [G \mid \text{студент}(S, _), \text{учится_в}(S, G)] .$$
 - a. ["НИ-202", "НИ-201", "НИ-201", "НИ-202"]
 - b. ["НИ-201", "НИ-201", "НИ-202", "НИ-202"]
 - c. ["НИ-202", "НИ-201"]
 - d. [НИ-202, НИ-201, НИ-201, НИ-202]
 - e. "НИ-202"

5. Что обозначает ключевое слово `single`, указанное после объявления факта в разделе `class facts` некоторого класса языка Visual Prolog?
 - a. Факт может встретиться в базе программы 0 и более раз
 - b. Факт может встретиться в базе программы 0 или 1 раз
 - c. Факт может встретиться в базе программы 1 и более раз
 - d. Факт всегда встречается в базе программы, только 1 раз

6. Координаты клеток некоторого лабиринта задаются парой целых чисел (X, Y) . В программе определен предикат

```
passable(X, Y),
```

который для заданной пары координат возвращает Истину, если клетка является проходимой, и Ложь в противном случае. Предикат, генерирующий множество смежных клеток для заданной, реализован следующим образом:

```
% adj( ТекущаяКлетка_X, ТекущаяКлетка_Y, СмежнаяКлетка_X,
СмежнаяКлетка_Y )
adj(X, Y, X, Y + 1) :- passable(X, Y + 1).
adj(X, Y, X + 1, Y) :- passable(X + 1, Y).
adj(X, Y, X, Y - 1) :- passable(X, Y - 1).
adj(X, Y, X - 1, Y) :- passable(X - 1, Y).
```

Какой вид перемещений описывает данный предикат смежности состояний?

- a. Шахматным конем
 - b. По любой диагонали на один шаг
 - c. Вверх, вправо, вниз и влево на один шаг
 - d. Вверх, вправо, вниз и влево на два шага
 - e. Вверх, вправо, вниз, влево и по диагонали на один шаг
7. Именем переменной в Prolog может являться
 - a. Неразрывная цепочка букв (латинских, русских), цифр или символа подчеркивания, начинающаяся с прописной буквы или символа подчеркивания.
 - b. Неразрывная цепочка букв (латинских, русских), цифр или символа подчеркивания, начинающаяся со строчной буквы.
 - c. Цепочка букв (латинских, русских), цифр или символа подчеркивания, начинающаяся с прописной буквы (начало с символа подчеркивания не допускается).
 8. Заданы несколько предикатов, описывающих решение задачи поиска в пространстве состояний (состояние - пара чисел (X, Y)).

`state(X, Y)` - факт, сохраняющий состояния (X, Y) , которые уже были рассмотрены.

`adj(X, Y, AX, AY)` - правило, которое для заданного состояния (X, Y) возвращает очередное смежное состояние (AX, AY) . Правило может быть согласовано 0 и более раз (у одного состояния может быть множество смежных состояний).

Предикат поиска пути в пространстве состояний реализован следующим образом:


```

find(Gx, Gy) :-
    assert(state(0, 0)),
    find(0, 0, Gx, Gy).
find(X, Y, X, Y) :-
    writef("(%, %)\n", X, Y),
    !.
find(X, Y, Gx, Gy) :-
    adj(X, Y, X1, Y1),
    not(state(X1, Y1)),
    assert(state(X1, Y1)),
    find(X1, Y1, Gx, Gy),
    writef("(%, %)\n", X, Y),
    !.

```

Какой алгоритм поиска пути в пространстве состояний реализует указанный предикат?

- a. Поиск в глубину
 - b. Поиск в ширину
 - c. Алгоритм Дейкстры
 - d. Алгоритм A*
9. Какие из указанных парадигм программирования реализует классический язык логического программирования Prolog (не Visual Prolog)?
- a. Логическое
 - b. Функциональное
 - c. Объектно-ориентированное
 - d. Вероятностное
10. Выберите правильный метод реализации предиката, вычисляющего факториал числа, методом нисходящей рекурсии.
- a. Код:


```

fact(0, 1) :- !.
fact(N, F1 * N) :- N > 0, fact(N - 1, F1).

```
 - b. Код:


```

fact(0, 1).
fact(N, F1 * N) :- N > 0, fact(N - 1, F1).

```
 - c. Код:


```

fact(N, F1 * N) :- N > 0, fact(N - 1, F1).
fact(0, 1) :- !.

```
 - d. Код:


```

fact(0, 1) :- !.
fact(N, F1 * N) :- N > 0, fact(N + 1, F1).

```
 - e. Код:


```

fact(N, F) :- N = 0, F = 1, !.
fact(N, F1 * N) :- N < 0, fact(N - 1, F1).

```

11. Для чего предназначен раздел `domains` в описании класса логической программы Visual Prolog?

- a. Раздел позволяет описывать собственные типы и структуры данных, которые в дальнейшем будут использоваться в коде класса
- b. Раздел позволяет описывать собственные факты, которые в дальнейшем будут использоваться в коде класса
- c. Раздел позволяет описывать собственные предикаты, которые в дальнейшем будут использоваться в виде правил в коде класса
- d. Раздел позволяет описывать собственные константы, которые в дальнейшем будут использоваться в коде класса

12. С какими из перечисленных термов унифицируется терм `[1, X, 3 | T]`?

- a. `[1, 2]`
- b. `[1, 2, 3]`
- c. `[1, 3, 3, 4, 5]`
- d. `[1, 3, 2, 4]`
- e. `[2, X, 3, 4, 5]`

13. В логической программе заданы два предиката:

```
sump( [ ] ) = 0.
```

```
sump ( [H | T] ) = 2 * H + sump(T) .
```

```
p(L) = sump( [X || X = list:getMember_nd(L), X mod 3 = 0] ) .
```

Какой результат будет выведен в консоль при выполнении следующего выражения:

```
stdio::write( p( [1, 3, 5, 6] ) ) .
```

14. В логической программе реализован предикат `p()`:

```
class predicates
```

```
    p : (integer*, integer*, integer [out]) determ.
```

```
clauses
```

```
    p([], [], 0) .
```

```
    p([H1 | T1], [H2 | T2], X) :-
```

```
        Y = (H1 + 1) * 2 * H2,
```

```
        p(T1, T2, R),
```

```
        X = Y + R.
```

Какой результат `X` будет получен при выполнении запроса:

```
p([1, 2, 0], [2, 0, 1], X)
```

15. Пусть в нашей базе фактов содержится 10 фактов `s(<число>)` (например, `s(1)`, `s(2)`, ...) и 12 фактов `m(<число>)` (например, `m(1)`, `m(2)`, ...).

Сколько строк будет выведено в консоль при выполнении следующего предиката:

```
print() :- s(X), m(Y), stdio::writef("s: % ; m: %", X, Y),  
stdio::nl, fail.
```

16. Пусть в нашей базе фактов содержится 15 фактов `s(<число>)` (например, `s(1)`, `s(2)`, ...) и 12 фактов `m(<число>)` (например, `m(1)`, `m(2)`, ...).

Сколько строк будет выведено в консоль при выполнении следующего предиката:

```
print() :- s(X), !, m(Y), !, stdio::writef("s: % ; m: %", X,
Y), stdio::nl, fail.
```

17. Верно ли следующее утверждение: "Каждое утверждение, определяющее отношение в логической программе, является либо фактом, либо правилом"?

18. Верно ли, что следующие две структуры унифицируются друг с другом:

```
предмет(Код, "Логическое программирование", место(часть(X),
блок("Б1")), компетенции(["ОПК-1", "ПК-1", "ПК-2"]), часы(Y)).
<-->
```

```
предмет(33, "Логическое программирование",
место(часть(вариативная), блок("Б2")), компетенции(C),
часы(Z)).
```