

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Комсомольский-на-Амуре государственный университет»

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ<sup>1</sup>**  
**по дисциплине**

**«Логическое программирование»**

Направление подготовки	<i>09.03.02 Информационные системы и технологии</i>
Направленность (профиль) образовательной программы	<i>«Проектирование и реализация информационных систем и технологий»</i>

Обеспечивающее подразделение
<i>Кафедра «ПУРИС»</i>

Разработчик ФОС:

Доцент, кандидат технических наук  
(должность, степень, ученое звание)

Абарникова Е.Б.  
(ФИО)

Согласовано:

Заведующий кафедрой  
(должность, степень, ученое звание)

Петрова А.Н.  
(ФИО)

<sup>1</sup> В данном документе представлены типовые оценочные средства. Полный комплект оценочных средств, включающий все варианты заданий (тестов, контрольных работ и др.), предлагаемых обучающемуся, хранится на кафедре в бумажном и электронном виде.

**1 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций**

Таблица 1 – Компетенции и индикаторы их достижения

Код и наименование компетенции	Индикаторы достижения	Планируемые результаты обучения по дисциплине
<b>Общепрофессиональные</b>		
ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий	<p>ОПК-6.1 Знает методы алгоритмизации, языки и технологии программирования, пригодные для практического применения в области информационных систем и технологий</p> <p>ОПК-6.2 Умеет применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий</p> <p>ОПК-6.3 Владеет навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач</p>	<p>Знает основные понятия, конструкции, структуры и логические высказывания языка логического программирования. Умеет применять знания о языке логического программирования для решения задач Имеет представление о среде визуальной разработки приложений.</p> <p>Умеет использовать функциональные возможности элементов среды визуальной разработки.</p> <p>Умеет обосновывать выбор того или иного элемента среды визуальной разработки.</p> <p>Обладает навыками разработки элементарных приложений в среде логического программирования.</p>

Таблица 2 – Паспорт фонда оценочных средств

Контролируемые разделы (темы) дисциплины	Формируемая компетенция	Наименование оценочного средства	Показатели оценки
Раздел 1 Введение в логическое программирование	ОПК-6	Лабораторная работа 1	<p>Имеет представление об основных понятиях логического программирования.</p> <p>Имеет представление о среде визуальной разработки приложений.</p> <p>Умеет использовать функциональные возможности элементов среды визуальной разработки.</p> <p>Умеет обосновывать выбор того или иного</p>

			элемента среды визуальной разработки.
	ОПК-6	Опорный конспект 1	Имеет представление об основных понятиях логического программирования. Умеет проводить сравнительный анализ объектов предметной области по нескольким критериям
Раздел 2 Основные структуры, методы и приемы программирования на языке Пролог	ОПК-6	Лабораторные работы 2-4	Имеет представление об основных понятиях, конструкциях, структурах и логических высказываниях языка логического программирования Умеет анализировать поставленную задачу. Умеет разрабатывать логические модели и конструкции для поставленной задачи. Умеет выбирать, оценивать, применять при разработке средства и методы логического программирования. Умеет обосновывать реализованные решения. Обладает навыками разработки приложений в среде логического программирования.
	ОПК-6	Краткий конспект 1-7	Имеет представление об основных понятиях, конструкциях, структурах и логических высказываниях языка логического программирования
	ОПК-6	Опорный конспект 2	Знает основные понятия, конструкции, структуры и логические высказывания языка логического программирования.

			Умеет применять знания о языке логического программирования для решения задач
	ОПК-6	Мини-тест	Знает основные понятия, конструкции, структуры и логические высказывания языка логического программирования
Графы, деревья. Поиск пути на графе пространства состояний	ОПК-6	Лабораторная работа 5 РГР	Знает основные методы и приемы решения трудноформализуемых задач. Знает основные возможности визуальной среды разработки приложений. Умеет применять знания о языке и среде визуальной разработки для реализации поставленной задачи. Умеет выбирать, оценивать, применять при разработке средства и методы логического программирования. Имеет навык использования логических моделей, применяемых при разработке ПО. Обладает навыками разработки приложений в среде логического программирования
	ОПК-6	Краткий конспект 8	Знает основные методы и приемы решения трудноформализуемых задач

**2 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующие процесс формирования компетенций**

Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, представлены в виде технологической карты дисциплины (таблица 3).

Таблица 3 – Технологическая карта

Наименование оценочного средства	Сроки выполнения	Шкала оценивания	Критерии оценивания
<b>Промежуточная аттестация в форме «Зачет с оценкой»</b>			
Лабораторные работы (5 работ)	В течение семестра	10 баллов	10 баллов - студент правильно и полностью выполнил задание. Показал отличные знания, умения и навыки в рамках освоенного учебного материала. 7 баллов - студент выполнил задание не в срок. Показал хорошие знания, умения и навыки в рамках освоенного учебного материала. 4 баллов - студент выполнил задание не в срок. Показал удовлетворительные знания, умения и навыки в рамках освоенного учебного материала. 0 баллов - задание не выполнено
Опорный конспект (2 конспекта)	В течение семестра	5 баллов	5 баллов - задание выполнено полностью без ошибок, в срок, допустимо наличие некоторых неточностей. 0,1- 4,9 баллов - задание выполнено не в срок 0 баллов – задание не выполнено
Краткий конспект лекции (8 конспектов)	В течение семестра	2 баллов	2 балла - задание выполнено полностью без ошибок, в срок, допустимо наличие некоторых неточностей. 0,1- 1,9 баллов - задание выполнено не в срок 0 баллов – задание не выполнено
Мини-тест	6 неделя	4 балла	4 балла - тест выполнен полностью без ошибок, в срок. 0,1- 3,9 баллов - тест выполнен полностью, не в срок 0 баллов – тест не выполнен
Расчетно-графическая работа	В течение семестра	20 баллов	20 баллов - студент правильно и полностью выполнил задание. Показал отличные знания, умения и навыки в рамках освоенного учебного материала. 16 баллов - студент выполнил задание не в срок. Показал хорошие знания, умения и навыки в рамках освоенного учебного материала. 11 баллов - студент выполнил

			задание не в срок. Показал удовлетворительные знания, умения и навыки в рамках освоенного учебного материала. 0 баллов - задание не выполнено
<b>ИТОГО:</b>		100 баллов	
<p><b>Критерии оценки результатов обучения по дисциплине:</b>  0 – 64 % от максимально возможной суммы баллов – «неудовлетворительно» (недостаточный уровень для промежуточной аттестации по дисциплине);  65 – 74 % от максимально возможной суммы баллов – «удовлетворительно» (пороговый (минимальный) уровень);  75 – 84 % от максимально возможной суммы баллов – «хорошо» (средний уровень);  85 – 100 % от максимально возможной суммы баллов – «отлично» (высокий (максимальный) уровень)</p>			

**3 Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующие процесс формирования компетенций в ходе освоения образовательной программы**

**Задания для текущего контроля успеваемости**

**Пример задания на лабораторную работу 1**

Описать структуру и функциональные возможности одного из экспертов Visual Prolog (по выбору студента). Разработать элементарное приложение, осуществляющее ввод-вывод текстовой информации.

**Пример задания на лабораторную работу 2**

Разработать приложение для построения ряда чисел (по заданию преподавателя). Границы ряда должны задаваться в режиме диалога с пользователем.

Возможные варианты для построения рядов:

- 1) Ряд простых чисел
- 2) Ряд чисел Фиббоначи
- 3) Ряд квадратов чисел, кратных 3
- 4) Ряд квадратов чисел, кратных 5
- 5) Ряд квадратов чисел, кратных 7
- 6) Ряд целых нечетных чисел
- 7) Ряд целых четных чисел
- 8) Треугольник Паскаля
- 9) Ряд чисел, кратных 5
- 10) Ряд чисел, кратных 7

**Пример задания на лабораторную работу 3  
(реализуется в форме практической подготовки)**

Разработать приложение для выполнения операций со списками и бинарными деревьями (по заданию преподавателя). Все исходные данные должны задаваться в режиме диалога с пользователем.

Возможные варианты операций со списками и бинарными деревьями:

- 1) вычисление длины списка с помощью восходящей рекурсии
- 2) реверс списка
- 3) добавление элемента в список по указанному номеру

- 4) объединение двух списков в один в указанной последовательности
- 5) разбиение списка на два подсписка по длине
- 6) вычисление длины бинарного дерева
- 7) определение принадлежности элемента бинарному дереву
- 8) проверка двух деревьев на изоморфность
- 9) обход бинарного дерева по схеме: Левое поддереву-Корень-Правое поддереву
- 10) добавление элемента в бинарное дерево
- 11) замена элемента в бинарном дереве

#### **Пример задания на лабораторную работу 4 (реализуется в форме практической подготовки)**

Разработать приложение, реализующее обработку данных на основе цикла (вид цикла и реализуемая функция выдаются преподавателем). Все исходные данные должны задаваться в режиме диалога с пользователем.

Возможные варианты формулировки заданий:

- 1) реализовать построчный вывод информации на экран из файла с помощью `repeat`-цикла
- 2) реализовать построчный вывод информации на экран из файла с помощью цикла, управляемого отказом
- 3) реализовать построчный вывод информации на экран из файла с помощью интерактивного цикла
- 4) реализовать имитатор предиката `asserta` с помощью любого цикла
- 5) реализовать имитатор предиката `assertz` с помощью любого цикла
- 6) реализовать имитатор предиката `retract` с помощью любого цикла
- 7) реализовать построчную запись информации в текстовый файл с помощью `repeat`-цикла
- 8) реализовать построчную запись информации в текстовый файл с помощью управляемого отказом
- 9) реализовать построчную запись информации в текстовый файл с помощью интерактивного цикла

#### **Пример задания на лабораторную работу 5 (реализуется в форме практической подготовки)**

Разработать приложение, реализующее простейшую экспертную систему. В экспертной системе реализовать 2 режима – обучения и проведения консультаций. Компоненту объяснения выводов реализовать только в режиме объяснения, без обоснования.

#### **Пример задания на расчетно-графическую работу (реализуется в форме практической подготовки)**

Используя «слепые» методы поиска решений на графе пространства состояний, разработать приложение для решения логических задач (по заданию преподавателя). Вывод результата осуществить в отдельное окно. Реализовать систему сообщений об ошибках, которые должна выдавать программа.

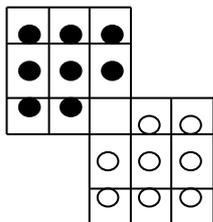
Варианты задач

**Вариант 1.** Обойти ходом шахматного коня шахматную доску таким образом, чтобы в каждой клетке побывать только один раз. Размерность шахматной доски и начальная позиция коня задаются пользователем. В случае если существует несколько решений, показать первое найденное. В случае если решение отсутствует, вывести на экран сообщение об ошибке.

**Вариант 2.** Перевезти с одного берега реки на другой отряд рыцарей и двух оруженосцев, если известно, что в лодку может поместиться или один рыцарь, или не более двух оруженосцев. Количество рыцарей в отряде задается пользователем.

**Вариант 3.** Перевезти через реку трех рыцарей со своими оруженосцами, если известно, что в лодке могут поместиться не более двух человек, и оруженосец не может находиться на берегу без своего рыцаря (или с другими оруженосцами, или один). Допускается вариант, когда рыцарь привозит своего оруженосца, но не выходит на берег, если там уже есть чужие оруженосцы.

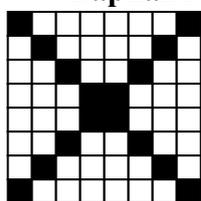
**Вариант 4.** На доске размером 4x4 случайным образом расположены фишки с номерами от 1 до 15. Одна клетка остается пустой. Перемещая соседние фишки, упорядочите их расположение (слева – направо и сверху – вниз). Например, первый ряд – 1,2,3,4, второй – 5,6,7,8 и т.д. «Вытаскивать» фишки и перескакивать через фишку не разрешается.



**Вариант 5.** На поле расположены черные и белые шашки, как показано на рисунке. Требуется, не снимая шашек с поля, перегнать за минимальное количество ходов все черные шашки на места белых, а белые – на места черных. Шашки могут передвигаться вперед и назад, вправо и влево, но не наискось. В тех же направлениях разрешается перепрыгивать через одну шашку на свободную клетку. Две шашки в одну клетку помещать нельзя. Очередность в перемещении белых и черных шашек соблюдать не требуется.

Очередность в перемещении белых и черных шашек соблюдать не требуется.

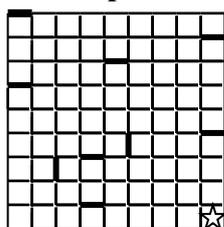
**Вариант 6.** На поле, показанном на рисунке, разместить только в белых клетках 8 звездочек таким образом, чтобы никакие 2 звездочки не находились на одной горизонтали, или вертикали, или какой-либо диагонали. Найти все варианты решения.



**Вариант 7.** Имеется 4 комплекта квадратов четырех цветов. В каждом комплекте – по 4 квадрата, пронумерованные от 1 до 4. Например, имеется синий, зеленый, красный и белый комплекты. Соответственно, квадраты в них обозначим С1 (синий №1), С2, С3, С4 и т.д. Разместить эти квадраты на поле, размером 4x4 клетки таким образом, чтобы в каждом горизонтальном ряду, в каждом вертикальном ряду и в каждой из двух диагоналей находились в каком-либо произвольном порядке квадраты с цифрами 1, 2, 3, 4 и непременно разных окрасок. Найти все варианты решения.

**Вариант 8.** Реализовать игру «Крестики-нолики» в режиме «человек-компьютер» с размерностью игрового поля 5x5 клеток.

**Вариант 9.** На рисунке представлен план лабиринта, в котором из 145 дверей только 9 заперты (они обозначены жирными полосками), а все остальные открыты настежь. Открыть запертую дверь можно в том случае, если она будет точно девятой по счету, то есть если перед этим будет пройдено 8 открытых дверей. При этом должны быть открыты и пройдены все запертые двери, каждая из них открывается сама, если перед этим пройдено ровно 8 открытых дверей. Как только какая-нибудь камера пройдена, все прежде открытые в ней двери наглухо закрываются.



Найти путь в лабиринте. Звездочкой помечен вход в лабиринт.

**Вариант 10.** Реализовать игру «Морской бой» в режиме «человек-компьютер» с размерностью игрового поля 10x10 клеток.

### Возможные вопросы и задания для защиты лабораторных работ и РГР

- 1) Объясните основное отличие идеологии логического программирования от других направлений
- 2) Перечислите основные принципы и структуры языков логического программирования
- 3) Объясните в каких случаях необходимо использовать функциональные возможности А,В,С эксперта М, N, Р
- 4) Перечислите основные конструкции языка Prolog
- 5) Что такое простой абстрактный интерпретатор?

- 6) Перечислите основные секции Prolog-программы
- 7) Объясните назначение, расположение, способы задания и варианты использования секции X Prolog-программы
- 8) Объясните на примере работу механизма поиска с возвратом (backtracking)
- 9) Что такое унификация? Приведите пример унификации.
- 10) Какая структура называется «списком»? Способы задания, основные свойства списков
- 11) Какая структура называется «списком»? Приведите пример одной из основных операций со списками
- 12) Какая структура называется «бинарным деревом»? Способы задания, основные свойства бинарных деревьев
- 13) Какая структура называется «бинарным деревом»? Приведите пример одной из основных операций с бинарными деревьями
- 14) Что такое «составные объекты»? Способы задания и использования составных объектов
- 15) Объясните принципиальное отличие списков и бинарных деревьев от составных объектов
- 16) Дайте определение рекурсии. Виды рекурсий в Prolog'e
- 17) Каковы правила написания рекурсивных правил. Приведите пример
- 18) Объясните, принципиальное отличие восходящей от нисходящей рекурсии
- 19) Определите по фрагменту программы вид рекурсии
- 20) Дайте определение отсечения. Виды отсечений
- 21) Определите по фрагменту программы вид отсечения
- 22) Объясните в каких случаях необходимо использовать отсечения. Приведите пример
- 23) Дайте определение понятию «внелогический предикат».
- 24) Определите по фрагменту программы какие из предикатов относятся к внелогическим/логическим
- 25) Перечислите наиболее используемые циклы в Prolog'e
- 26) Проведите сравнительный попарный анализ наиболее используемых циклов
- 27) Дайте определение понятию «экспертная система»
- 28) Опишите основные компоненты экспертной системы
- 29) Опишите основные режимы функционирования экспертной системы
- 30) В тексте программы выделите компоненты экспертной системы
- 31) Дайте определение понятию «граф пространства состояний»
- 32) Перечислите основные свойства графа пространства состояний
- 33) Дайте классификацию методам поиска решений на графе пространства состояний
- 34) Приведите пример программной реализации графа пространства состояний
- 35) Обоснуйте выбранный метод поиска решений

### **Возможные вопросы мини-теста**

Мини-тест реализован и опубликован в качестве игрового ресурса по адресу: <https://LearningApps.org/view2218331>. Пример тестового задания приведен на рисунке 1.

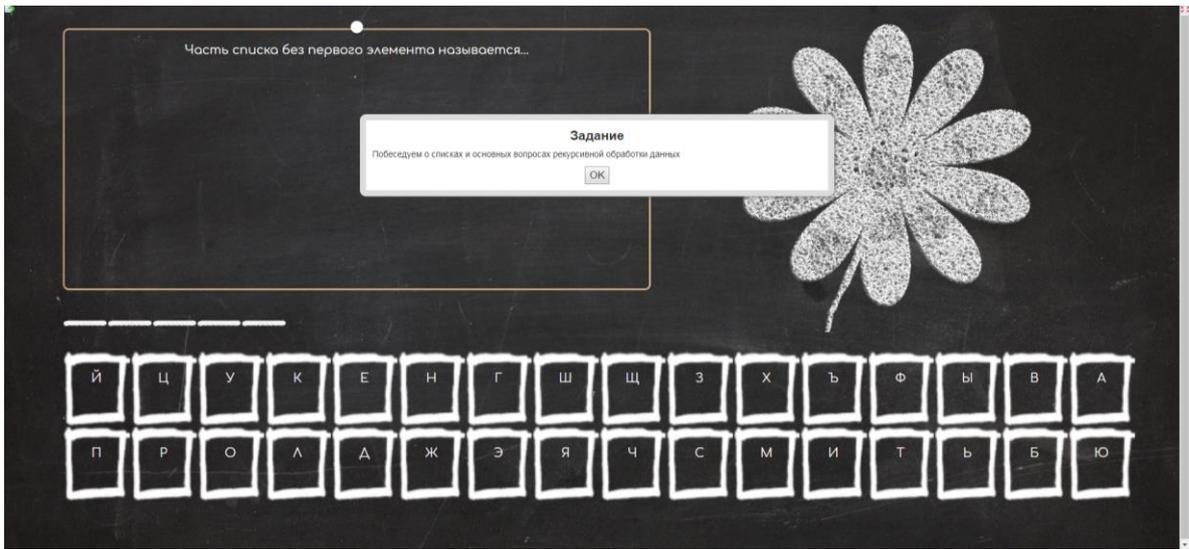


Рисунок 1 – Пример задания в мини-тесте