

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Выборнова Любовь Алексеевна
Должность: Ректор
Дата подписания: 03.02.2022 15:17:47
Уникальный программный ключ:
c3b3b9c625f6c113afa2a2c42ba19e05a38b76e

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СЕРВИСА»
(ФГБОУ ВО «ПВГУС»)

Кафедра «Информационный и электронный сервис»

РАБОЧАЯ УЧЕБНАЯ ПРОГРАММА

по дисциплине

«Функциональное и логическое программирование»

для студентов направления подготовки

09.03.04 «Программная инженерия»

направленности (профиля) «Разработка программно-информационных систем»

Рабочая учебная программа по дисциплине «Функциональное и логическое программирование» включена в основную профессиональную образовательную программу направления подготовки 09.03.04 «Программная инженерия» направленности (профиля) «Разработка программно-информационных систем» решением Президиума Ученого совета

Протокол № 4 от 28.06.2018 г.

Начальник учебно-методического отдела _____  Н.М.Шемендук
28.06.2018 г.

Рабочая учебная программа по дисциплине «Функциональное и логическое программирование» разработана в соответствии с Федеральным государственным образовательным стандартом направления подготовки 09.03.04 «Программная инженерия» утвержденный приказом Министерства образования и науки Российской Федерации от 12 марта 2015 г. N 229

Составила: к.т.н., доцент Яницкая Т.С.

СОГЛАСОВАНО:

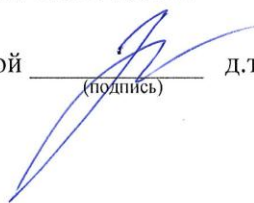
Директор научной библиотеки  В.Н.Еремина

СОГЛАСОВАНО:

Начальник управления информатизации  В.В.Обухов

Рабочая программа утверждена на заседании кафедры «Информационный и электронный сервис»

Протокол № 11 от «27» июня 2018 г.

Заведующий кафедрой  д.т.н., профессор В.И. Воловач
(подпись)

СОГЛАСОВАНО:

Начальник учебно-методического отдела  Н.М.Шемендюк

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

1.1. Цели освоения дисциплины

Целью изучения дисциплины являются получение студентами базовых знаний по логическому программированию и приобретение навыков составления и отладки программ на языке логического программирования для решения задач представления и использования знаний о предметной области.

1.2. В соответствии с видами профессиональной деятельности, на которые ориентирована образовательная программа указанных направлений подготовки, содержание дисциплины позволит обучающимся решать следующие профессиональные задачи:

производственно-технологическая деятельность:

- освоение и применение средств автоматизированного проектирования, разработки, тестирования и сопровождения программного обеспечения;
- освоение и применение методов и инструментальных средств управления инженерной деятельностью и процессами жизненного цикла программного обеспечения.

1.3. Компетенции обучающегося, формируемые в результате освоения дисциплины

В результате освоения дисциплины у обучающихся формируются следующие компетенции:

Код компетенции	Наименование компетенции	Направление подготовки
1	2	3
ПК-3	Владением навыками использования различных технологий программного обеспечения разработки	09.03.04 «Программная инженерия», направленность «Разработка программно-информационных систем»

1.4. Перечень планируемых результатов обучения по дисциплине

Результаты освоения дисциплины	Технологии формирования компетенции по указанным результатам	Средства и технологии оценки по указанным результатам
Знает: ПК-3 Стандартные пакеты прикладных программ, ориентированных на решение научных и проектных задач	Лекции	Собеседование
Умеет: ПК-3 Использовать стандартные пакеты прикладных программ для решения практических задач	Лабораторные работы	Собеседование Защита лабораторных работ

Результаты освоения дисциплины	Технологии формирования компетенции по указанным результатам	Средства и технологии оценки по указанным результатам
Имеет практический опыт: ПК-3 При работе с современными программными средствами подготовки конструкторско-технологической документации	Лекции Лабораторные работы	Защита лабораторных работ

2. Место дисциплины в структуре образовательной программы

Дисциплина относится к вариативной части.

Ее освоение осуществляется в 5 семестре (заочная форма (февраль)), в 6 семестре (очная и заочная формы).

п/п	Наименование дисциплин, определяющих междисциплинарные связи	Код компетенции (й)
09.03.04 «Программная инженерия»		
	Предшествующие дисциплины	
1	Программирование	ПК-1, ПК-3
2	Алгоритмы и структуры данных	ПК-1, ПК-7
	Последующие дисциплины	
3	Тестирование программного обеспечения	ПК-4, ПК-9, ПК-10

3. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу

Распределение фонда времени по семестрам и видам занятий

Виды занятий	очная форма обучения	заочная форма обучения (февраль)	заочная форма обучения
Итого часов	216	216	216
Зачетных единиц	6	6	6
Лекции (час)	24	8	8
Практические (семинарские) занятия (час)	-	-	-
Лабораторные работы (час)	56	14	14
Самостоятельная работа (час)	109	185	185
Курсовой проект (работа) (+,-)	КП	КП	КП
Контрольная работа (+,-)	-	-	-
Экзамен, семестр /час.	6 семестр / 27	5 семестр/ 9	6 семестр/ 9
Зачет, семестр	-	-	-
Контрольная работа, семестр	-	-	-

4. Содержание дисциплины, структурированное по темам (разделам) с указанием

отведенного на них количества академических часов и видов учебных занятий

4.1. Содержание дисциплины

п/п	Раздел дисциплины	Виды учебных занятий, включая самостоятельную работу студентов и трудоемкость (в академических часах)				Средства и технологии оценки
		Лекции, час	Практические (семинарские) занятия, час	Лабораторные работы, час	Самостоятельная работа, час	
1	Парадигмы функционального и логического программирования	2/0,5/ 0,5	-	-	10/37/37	Конспект, сообщение
2	Основы программирования с помощью функций	4/1,5/ 1,5	-	-	22/37/37	Конспект, опрос на лекции
3	Введение в язык Лисп и Haskell	6/2/2	-	28/6/6	26/37/37	Конспект, сообщение, защита лабораторных работ
4	Основы логического программирования	6/2/2	-	12/4/4	26/37/37	Конспект, сообщение, защита лабораторных работ, курсовая работа
5	Использование языка Пролог для решения задач	6/2/2	-	16/4/4	25/37/37	Конспект, сообщение, промежуточное тестирование
	Промежуточная аттестация по дисциплине	24/8/ 8	-	56/14/1 4	109/185/ 185	Экзамен

Примечание:

-/-/-, объем часов соответственно для очной формы обучения, заочной формы обучения (февраль), заочной формы обучения

4.2. Содержание практических (семинарских) занятий

Практические работы планом не предусмотрены

4.3. Содержание лабораторных работ

	Наименование лабораторных работ	Объем часов	Наименование темы дисциплины
1	Лабораторная работа 1. «Знакомство с языком Lisp. Абстракция вычислений»	2/1/1	Введение в язык Лисп и Haskell.
2	Лабораторная работа 2. «Отложенные вычисления и ленивые списки в Лиспе. Нисходящий разбор по заданной LL(1)-грамматике P.»	4/1/1	Введение в язык Лисп и Haskell.
3	Лабораторная работа 3. «Реализация работы с полиномами на функциональном языке.»	4/1/1	Введение в язык Лисп и Haskell.

4	Лабораторная работа 4. «Рекурсивные функции»	6/1/1	Введение в язык Лисп и Haskell.
5	Лабораторная работа 5. «Арифметика и обработка списков в LISP»	6/1/1	Введение в язык Лисп и Haskell.
6	Лабораторная работа 6. «Организация итерационных вычислений в LISP.»	6/1/1	Введение в язык Лисп и Haskell.
7	Лабораторная работа 7. «Ознакомление с языком программирования Prolog»	4/1/1	Основы логического программирования
8	Лабораторная работа 8. «Решение простых логических задач на языке Prolog»	4/1,5/1,5	Основы логического программирования
9	Лабораторная работа 9. «Рекурсивные структуры данных (Prolog)»	4/1,5/1,5	Основы логического программирования
10	Лабораторная работа 10. «Экспертная система на Прологе»	6/1/1	Использование языка Пролог для решения задач
11	Лабораторная работа 11. «Описание предметной области с помощью программы на Прологе»	6/1,5/1,5	Использование языка Пролог для решения задач
12	Лабораторная работа 12. «Управление выполнением программы с помощью предикатов »	4/1,5/1,5	Использование языка Пролог для решения задач
	Итого	54/14/14	

Примечание:

-/-/-, объем часов соответственно для очной формы обучения, заочной формы обучения (февраль), заочной формы обучения

5. Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине

Технологическая карта самостоятельной работы студента

Код реализуемой компетенции	Вид деятельности студентов (задания на самостоятельную работу)	Итоговый продукт самостоятельной работы	Средства и технологии оценки	Объем часов
1	2	3	4	5
09.03.04 «Программная инженерия»				
ПК-3	Выполнение индивидуальных заданий в виде доклада и презентации на заданную тему.	Доклад, презентация	Собеседование	54/90/90
ПК-3	Выполнение индивидуальных заданий в виде краткого конспекта на заданную тему.	Конспект	Собеседование	55/95/95
Итого				109/185/185

Примечание:

-/-/-, объем часов соответственно для очной формы обучения, заочной формы обучения (февраль), заочной формы обучения

Литература:

1. Иванова, Г. С. Технология программирования [Текст]: учеб. для вузов по направлению "Информатика и вычисл. техника" / Г. С. Иванова. - 3-е изд., стереотип. - М. : КноРус, 2016. - 334 с. : ил.;
2. Орлов, С. А. Теория и практика языков программирования [Текст]: учеб. для вузов по направл. "Информатика и вычисл. техника" / С. А. Орлов. - СПб. : Питер, 2014. - 688 с. : ил.;
3. Программирование [Электронный ресурс] : учеб. для студентов вузов по направлению "Пед. образование" : в 2 т. Т. 2 / Э. А. Нигматулина [и др.] под ред. Н. И. Пака. - Документ Adobe Acrobat. - М.: Академия, 2013. - 60,4 МБ, 240 с. : ил. - Режим доступа: <http://elib.tolgas.ru>
4. Цуканова, Н. И. Теория и практика логического программирования на языке Visual Prolog 7 [Текст]: учеб. пособие для вузов по специальности "Прогр. обеспечение вычисл. техники и автоматизир. систем" / Н. И. Цуканова, Т. А. Дмитриева. - М. : Горячая линия - Телеком, 2013. - 232 с. : ил.

Содержание заданий для самостоятельной работы

Вопросы для самоконтроля

1. Понятие декларативного программирования. Основные принципы функционального и логического программирования.
2. Общие свойства функциональных языков: краткость, модульность, строгая типизация, чистота, ленивые вычисления.
3. Основные элементы языка логического программирования пролог: определение, простые и структурные объекты данных.
4. Структура пролог – программы, типы предложений программы, пример логического вывода (реализация дерева родственных отношений).
5. Определение арности предикатов. Признак сопоставимости термов, общая схема доказательства целевого утверждения.
6. Признаки сопоставимости термов. Общая схема доказательства целевого утверждения. Операционная семантика языка Пролог.
7. Рекурсия в прологе: определение рекурсивных правил, использование отсечения.
8. Алгоритм логического вывода пролог - машины на примере вычислений с «откатом».
9. Встроенные предикаты для арифметических вычислений. Встроенные предикаты ввода-вывода, управления базой данных (знаний).
10. Предикаты управления на примере «отсечения».
11. Зеленое и красное «отсечение» на примерах выбора наилучшего фильма и добавления в список отсутствующего элементов.
12. Встроенные предикаты преобразования структур.
13. Структура данных – список. Шаблоны списков, сопоставление.
14. Примеры отношений обработки списков: замена первого элемента списка (`replace_first`), определения вхождения в список (`member`), определения длины списка (`length`).
15. Примеры отношений обработки списков: конкатенации списков (`append`), обращение списка (`reverse`). Использование конкатенации для поиска комбинаций элементов, удаления элементов, нахождения последнего элемента списка.
16. Сортировка списков: метод наивной сортировки, метод сортировки вставками.
17. Сортировка списков: метод пузырьковой сортировки.
18. Реализация повторений (циклов) на прологе: метод обхода без возвратов, метод поиска с возвратом, метод `cat and fail (CAF)`, метод `user define repeat (UDR)`.
19. Деревья в Прологе: определение, построение. Определение отношения «внутри». Вычисление глубины дерева.
20. Двоичные справочники: определение, поиск элементов. Отношение, реализующее проверку: является ли объект двоичным справочником.

21. Добавление элементов в двоичное дерево.
22. Удаление элементов двоичного дерева, удаление листьев, удаление внутренних вершин.
23. Отображение деревьев сверху - вниз (корень наверху, листья внизу), слева – направо (корень самый левый выводимый элемент).
24. Операции в прологе, их приоритет, тип. Определение собственных операций.
25. Основные понятия языка функционального программирования Лисп. Особенности языка.
26. S-выражения в Лиспе. Символьные, числовые атомы. Списки.
27. Функции в Лиспе. Единая префиксная запись, ее достоинства.
28. Вычисление выражений. Функции quote, eval.
29. Использование символов в качестве переменных.
30. Базовые функции Лиспа. Функции работы со списками car, cdr, list.
31. Базовые функции Лиспа. Функции cons, length, atom, eq.
32. Определение функций в языке Лисп. Предикаты.
33. Глобальные и локальные переменные. Задача вычисления сопротивления электрической цепи.
34. Дополнительные функции обработки списков: конкатенации списков(append), обращения списка (reverse), выделения последнего элемента списка (last). Примеры применения.
35. Базовые предикаты Лиспа: сравнения символов (eq), сравнения чисел и символов (eql), общий предикат сравнения (eql). Примеры применения.
36. Логические функции Лиспа: и, или, отрицание. Условное предложение cond. Примеры применения.
37. Условные предложения if, when, unless. Их отличия от общей формы cond. Примеры вычислений с использованием этих функций.
38. Ввод - вывод информации. Чтение и запись информации в файлы, входные и выходные потоки.
39. Локальные переменные в Лиспе, конструкция let.
40. Организация циклических вычислений в Лиспе, предложения loop, do.
41. Рекурсия в Лиспе, правило записи рекурсивных функций. Численная рекурсия.
42. Общая форма рекурсии. CDR – рекурсия.
43. Функционалы в Лиспе, отображающий функционал MAPCAR. Применяющие функционалы.
44. Лямбда – выражения, их структура, применение.
45. Свойства символов. Операторы получения (get), изменения (общая форма setf), получения списка свойств (symbol-plist).
46. Внутреннее представление списков. Сборка мусора.
47. Обработка списков без разрушения, разрушающие функции, проблемы их применения.
48. Параметры функций: необязательные, переменное количество параметров связанных с хвостом списка, ключевые параметры.

6. Методические указания для обучающихся по освоению дисциплины Инновационные образовательные технологии

Вид образовательных технологий, средств передачи знаний, формирования умений и практического опыта	№ темы / тема лекции	№ практического (семинарского) занятия/наименование темы	№ лабораторной работы / цель
Разбор конкретных ситуаций	-	-	№1-12
Слайд-лекции	№ 1-5	-	-

В начале семестра студентам необходимо ознакомиться с технологической картой дисциплины, выяснить, какие результаты освоения дисциплины заявлены (знания, умения, практический опыт). Для успешного освоения дисциплины студентам необходимо выполнить задания, предусмотренные рабочей учебной программой дисциплины и пройти контрольные точки в сроки, указанные в технологической карте (раздел 11). От качества и полноты их выполнения будет зависеть уровень сформированности компетенции и оценка

текущей успеваемости по дисциплине. По итогам текущей успеваемости студенту может быть выставлена оценка по промежуточной аттестации, если это предусмотрено технологической картой дисциплины. Списки учебных пособий, научных трудов, которые студентам следует прочесть и законспектировать, темы лабораторных работ и вопросы к ним, вопросы к экзамену и другие необходимые материалы указаны в разработанном для данной дисциплины учебно-методическом комплексе.

Основной формой освоения дисциплины является контактная работа с преподавателем - лекции, лабораторные работы, консультации (в том числе индивидуальные), в том числе проводимые с применением дистанционных технологий.

По дисциплине часть тем (разделов) изучается студентами самостоятельно. Самостоятельная работа предусматривает подготовку к аудиторным занятиям, выполнение заданий (письменных работ, творческих проектов и др.) подготовку к промежуточной аттестации (экзамену).

На лекционных занятиях и лабораторных работах вырабатываются навыки и умения обучающихся по применению полученных знаний в конкретных ситуациях, связанных с будущей профессиональной деятельностью. По окончании изучения дисциплины проводится промежуточная аттестация (экзамен).

Регулярное посещение аудиторных занятий не только способствует успешному овладению знаниями, но и помогает организовать время, т.к. все виды учебных занятий распределены в семестре планомерно, с учетом необходимых временных затрат.

6.1. Методические указания для обучающихся по освоению дисциплины на лабораторных работах

Лабораторные работы

	Наименование лабораторных работ	Задание по лабораторным работам
1	Лабораторная работа 1. «Знакомство с языком Lisp. Абстракция вычислений»	Требуется реализовать вычисление неэлементарных функций. Создать и отладить код на языке программирования Lisp.
2	Лабораторная работа 2. «Отложенные вычисления и ленивые списки в Лиспе. Нисходящий разбор по заданной LL(1)-грамматике P.»	Написать программу, эмулирующую поведение детерминированного МП-автомата, составленного по заданной LL(1)-грамматике. Проверку принадлежности строки языка выполнить в виде поиска допускающего состояния в истории вычислений автомата. Историю вычислений реализовать в виде отложенного списка, то есть не допускается одновременное хранение в памяти всех промежуточных состояний автомата.
3	Лабораторная работа 3. «Реализация работы с полиномами на функциональном языке»	Разработать программу на языке LISP в соответствии с вариантом задания.
4	Лабораторная работа 4. «Рекурсивные функции»	Разработать программу на языке LISP, для реализации заданной функции в соответствии с вариантом задания согласно номеру зачетной книжки.
5	Лабораторная работа 5. «Арифметика и обработка списков в LISP»	Разработать программу на языке LISP, реализующую заданную преподавателем функцию из предложенных вариантов заданий (19 вариантов)
6	Лабораторная работа 6. «Организация	1. Написать функцию, реализующую

итерационных вычислений в LISP.»

вычисления, заданные формулой:

$$\sum_{k=1}^n k^k$$

$k=1$ Пример:

>(fun 3)

32; $1^1 + 2^2 + 3^3$

2. Написать функцию, использующую метод Ньютона для вычисления квадратного корня. Метод Ньютона вычисления квадратного корня из числа x начинается с выбора начального приближения y . Это приближение считается достаточно точным, если $|x - y^2| \leq \text{err}$, где err – некоторая заранее определенная погрешность. В противном случае более точным приближением будет $1/2 * (y + x/y)$, которое можно вычислить и точно так же подвергнуть проверке на погрешность. Для получения абсолютной величины числа используйте функция `abs`:
(`abs -5`) = 5.

Пример:

; вычисления квадратного корня из числа 5

; начальное приближение 1

; уровень допустимой погрешности 0.05

>(newton 5 1 0.05)

2.2380953

3. Написать функции для вычисления N -го числа последовательности Фибоначчи. Каждое следующее число последовательности Фибоначчи вычисляется как сумма двух предыдущих. Два первых числа равны единице.

4. Напишите функцию, которая в качестве параметра получает список и возвращает последний элемент этого списка.

Пример: >(last1 '(a b c))

c

5. Напишите функцию `my-nth`, которая получает в качестве параметра список l и число n , и возвращает n -ый элемент списка l . Нумерация элементов списка начинается с нуля.

Примеры: >(my-nth 2 '(0 1 2 3))

2

6. Напишите функцию `fringe`, которая получает в качестве параметра список, возможно содержащий подсписки, и выдает все не списочные элементы списка в том же порядке, в котором они расположены в исходном списке. Для выполнения этого задания Вам могут потребоваться встроенные предикаты:

`atom` - возвращает Т, если аргумент

		<p>является атомом, NIL в противном случае; <i>listp</i> - возвращает T, если аргумент является списком.</p> <p>Примеры: <code>>(fringe '(a b (c) (d (e) f)))</code> (A B C D E F)</p> <p>7. Вычислить сумму ряда целых чётных чисел от n до $2n$. Определить предикат, проверяющий, обладает ли символ заданным свойством.</p>
7	Лабораторная работа 7. «Ознакомление с языком программирования Prolog»	<p>Необходимо описать предметную область «родственные отношения». Для этого задайте в качестве фактов следующие отношения между объектами предметной области:</p> <ol style="list-style-type: none"> 1. <code>parents(X, Y)</code> – X является родителем Y; 2. <code>man(X)</code> – X мужчина; 3. <code>woman(X)</code> – X женщина. <p>Определите в качестве правил (используя отношения <code>parents</code>, <code>man</code>, <code>woman</code>) следующие отношения:</p> <ol style="list-style-type: none"> 1. <code>sister(X, Y)</code> – X является сестрой Y; 2. <code>brother(X, Y)</code> – X является братом Y; 3. <code>father(X, Y)</code> – X является отцом Y; 4. <code>mother(X, Y)</code> – X является матерью Y; 5. <code>grandfather(X, Y)</code> – X является дедушкой Y; 6. <code>grandmother(X, Y)</code> – X является бабушкой Y.
8	Лабораторная работа 8. «Решение простых логических задач на языке Prolog»	<ol style="list-style-type: none"> 1. Написать функцию, использующую метод Ньютона для вычисления квадратного корня. Метод Ньютона вычисления квадратного корня из числа x начинается с выбора начального приближения y. Это приближение считается достаточно точным, если $x - y^2 \leq \text{err}$, где err – некоторая заранее определенная погрешность. В противном случае более точным приближением будет $1/2 * (y + x/y)$, которое можно вычислить и точно так же подвергнуть проверке на погрешность. Для получения абсолютной величины числа используйте функция <code>abs</code>: <code>abs(-5) = 5</code>. ?- <code>newton(5, 1, 0.05, Z)</code> <code>Z=2.2380953</code> 2. Написать рекурсивную и итерационную программы возведения числа в степень. Предполагается использование только натуральных чисел. Возведение в степень представить как повторяющееся умножение. 3. Написать функции для вычисления N-го числа последовательности

		<p>Фибоначчи. Каждое следующее число последовательности Фибоначчи вычисляется как сумма двух предыдущих. Два первых числа равны единице.</p>
9	<p>Лабораторная работа 9. «Рекурсивные структуры данных (Prolog)»</p>	<p>1. Написать программу нахождения последнего элемента списка. Для нахождения последнего элемента списка используйте двуарный предикат <code>last(type_of_elements, list)</code>. Первый аргумент этого предиката задает последний элемент списка, второй – список, в котором производится поиск последнего элемента. Т. о. цель <code>last(X, L)</code> согласуется с базой данных, если элемент <code>X</code> является последним элементом списка <code>L</code>. Например: ?- last(3, [1, 2, 3]) Yes ?- last(X, [1, 2, 3]) 3</p> <p>2. Написать программу исключения первого вхождения элемента в список. Для исключения первого вхождения некоторого элемента в список используйте предикат <code>exclude(type_of_elements, list, list)</code>. Цель <code>exclude(X, Y, Z)</code> исключает первое вхождение элемента <code>X</code> в список <code>Y</code>, формируя новый список <code>Z</code>. Например: ?- exclude(2, [1, 2, 3, 2], Z) Z=[1, 3, 2]</p> <p>3. Написать программу обращения списка. Обращенный список можно получить путем присоединения головы этого списка к обращенному хвосту. Для решения задачи используйте двуарный предикат <code>reverse(list, list)</code>. Первый аргумент этого предиката задает начальный список, второй – обращенный. Например: ?- reverse([1, 2, 3], Y) Y=[3, 2, 1]</p> <p>4. Написать программу поиска минимального элемента списка и его порядкового номера в этом списке. Для решения задачи используйте тринарный предикат <code>minimum(type_of_elements, integer, list)</code>. Первый аргумент в этом</p>

		<p>предикате задает минимальный элемент списка, второй – индекс этого элемента в списке, третий – список, в котором производится поиск минимального элемента. Т. о. цель $\text{minimum}(X, N, L)$ согласуется с базой данных, если X является минимальным элементом списка L, а N – порядковый номер элемента X в этом списке. Например:</p> <p>?- $\text{minimum}(1, 4, [2, 5, 3, 1, 7])$ Yes ?- $\text{minimum}(X, N, [2, 5, 3, 1, 7])$ $X=1, N=4$</p> <p>5. Написать программу сортировки элементов списка с помощью прямого включения. При сортировке включением каждый элемент списка рассматривается отдельно и включается в новый список на соответствующее место. Алгоритм этой сортировки следующий:</p> <p>FOR $i := 2$ TO n DO $x := a[i]$; включение x на соответствующее место среди $a[1] \dots a[i]$ END</p> <p>Пример сортировки списка с помощью прямого включения: начальный список 44 55 12 42 $i = 2$ 44 55 12 42 $i = 3$ 12 44 55 42 $i = 4$ 12 42 44 55</p> <p>Для решения задачи используйте двуарный предикат $\text{insertion_sort}(\text{list}, \text{list})$. Первый аргумент в этом предикате задает начальный список, второй – отсортированный.</p> <p>Например: ?- $\text{insertion_sort}([2, 5, 3, 1, 7], Y)$ $Y=[1, 2, 3, 5, 7]$</p>
10	Лабораторная работа 10. «Экспертная система на Прологе»	<p>Реализовать экспертную систему по заданной предметной области в соответствии с номером задания. При этом количество описываемых объектов должно быть не менее 12, а характеризующих их атрибутов — не менее .8. Рассмотреть реализацию экспертной системы как базирующуюся на логике и как базирующуюся на правилах. Предметная область задана в соответствии</p>

		с вариантом задания согласно последним цифрам номера зачетной книжки.
11	Лабораторная работа 11. «Описание предметной области с помощью программы на Прологе»	Разработать программу на языке Prolog для реализации заданной преподавателем предметной области. Разработка структурной схемы программы, отладка листинга выполненной программы.
12	Лабораторная работа 12. «Управление выполнением программы с помощью предикатов»	Разработать программу на языке Prolog, используя действия предикатов неудачи и отсечения. Создание программного кода с помощью методов организации повторного выполнения группы задач.

6.2. Методические указания для выполнения контрольных работ (письменных работ)

Контрольные работы учебным планом не предусмотрены.

6.3. Методические указания для выполнения курсовых проектов

Курсовой проект по дисциплине по правилам оформления должен соответствовать требованиям к курсовым проектам, утвержденным на кафедре.

Цель курсовой проекта — получить навыки в основах функционального или логического программирования вне рамок искусственных ограничений лабораторных работ, в приложении к решению конкретных интересных задач. Отдельным видом курсовых проекта являются исследования, посвященные изучению отдельных библиотек, средств разработки и расширений языков Prolog. Приветствуются работы, посвященные изучению и сравнительному анализу других языков функционального и логического программирования, не вошедших в программу курса.

В качестве темы курсовой работы может выступать исследование любой достаточно сложной библиотеки или средства из списка библиотек и инструментов языка функционального или логического программирования.

Отчет по курсовому проекту, включающему исследование, должен представлять собой руководство по использованию системы или библиотеки в практическом программировании и содержать: обоснование актуальности рассматриваемой проблемы; основной подход к ее решению, примененный в рассматриваемой библиотеке; общую структуру библиотеки; средства (функции) по категориям; описание применения отдельных функций для решения задач с исходным кодом и примерами результатов работы.

Отчет по курсовому проекту, включающему разработку приложения, должен содержать подробную постановку задачи, словесное описание подхода к решению, особенности решения отдельных частей задачи, исходный код. Непременное условие для сдачи такой курсовой — работающая по заданной спецификации программа.

Список тем для выполнению курсовых проектов:

1. Профилирование программ на логическом языке.
2. Автоматизированное тестирование программ на логическом языке.
3. Отладка программ на логическом языке.
4. Работа с MIDI-интерфейсом.
5. Создание музыкальных произведений.
6. Средства цифровой обработки сигналов.
7. Синтезирование звука и звуковых эффектов.
8. Представление виртуального звукового пространства в 3D.
9. Механизмы распределенных вычислений в программах на языке Haskell.
10. Представление и обработка сложных структур данных в программах на Haskell.

11. Средства работы с регулярными выражениями.
12. Работа с СУБД.
13. Средства автоматизации разработки игровых программ.
14. Разработка графического интерфейса пользователя средствами библиотеки
15. Компьютерная графика в программах.
16. Работа с векторной графикой.
17. Специализированный язык генерации и обработки изображений ...
18. Моделирование аналоговых электронных схем.
19. Моделирование цифровых электронных схем.
20. Специализированный язык описания электронных схем.
21. Обработка естественного языка.
22. Моделирование физических процессов средствами ...
23. Моделирование систем средствами ...
24. Символьные вычисления и преобразование формул средствами ...
25. Использование языков для решения математических задач.
26. Обработка статистических данных.
27. Построение диаграмм и графиков средствами ...
28. Библиотеки алгоритмов преобразования геометрических объектов.
29. Средства библиотеки ... для работы в компьютерной сети.
30. Работа с низкоуровневыми функциями операционной системы.
31. Анализ программ на языке ...
32. Генерация web-сайтов средствами библиотеки...
33. Разбор и преобразование XML.
34. Разбор и преобразование HTML.
35. Искусственный интеллект пошаговой стратегической игры.
36. Логическая игра.
37. Игра «Крестики-нолики» с переменным размером игрового поля и выигрывающей цепочки.
38. Игра «Жизнь».
39. Игра в морской бой.
40. Программа графического представления позиций в шахматах и шашках.
41. Игра «Бумажный футбол».
42. Система описания текстовых презентаций.
43. Эмулятор среды более низкого уровня.
44. Система управления контентом сайта.
45. Система генерации сайтов на основе сценариев.
46. Система верификации сайтов на основе правил.
47. Программа обработки и трансформации XML-документов.
48. Робот для извлечения информации web-сайтов.
49. Поисковая система web-сайтов.
50. Разработка программы ассоциативного поиска и категоризации текстов.
51. Встроенный язык описания и выполнения интерактивных презентаций.
52. Программа-калькулятор с функциональностью инженерного калькулятора Windows.
53. Система вычисления и упрощения арифметических выражений с объяснением способа получения ответа.
54. Система символьного дифференцирования, интегрирования и упрощения некоторых типов выражений.
55. Система решения вычислительных задач по геометрии.
56. Автоматизированная система составления расписаний.
57. Автоматизированная система планирования встреч и совещаний.
58. Экспертная системы по классификации объектов некоторой предметной области.
59. Экспертная системы по ремонту/настройке устройств некоторой предметной области.

В результате выполнения курсового проекта студент может набрать от 12 до 15 баллов в зависимости от сложности задания и качества его выполнения.

Минимальное число баллов - работа выполнена, но имеются замечания по ее

выполнению (погрешности непринципиального характера), и/или есть неточности при защите проекта. Максимальное число баллов - работа выполнена в полном соответствии с выданным заданием, и студент ответил на все вопросы при защите проекта.

Выбор темы курсового проекта

Обучающийся по своему желанию выбирает тему курсового проекта из утвержденной тематики. При возникновении у обучающегося затруднений с выбором темы, подбором литературы, составлением плана проекта необходимую помощь в этих вопросах ему оказывает преподаватель, являющийся руководителем выполнения курсового проекта.

Выбранная тема курсового проекта фиксируется на кафедре в приложении к распоряжению декана факультета. Тема курсового проекта не должна повторяться в одной учебной группе. Для студентов заочной формы обучения выбор темы курсового проекта производится из утвержденной тематики в соответствии с одним или двумя последними номерами зачетной книжки.

7. Фонд оценочных средств для проведения текущего контроля успеваемости и промежуточной аттестации обучающихся по дисциплине (экзамен)

Фонды оценочных средств, позволяющие оценить уровень сформированности компетенций и результаты освоения дисциплины, представлены следующими компонентами:

Код оцениваемой компетенции (или ее части)	Тип контроля	Вид контроля	Количество элементов
ПК-3	<i>текущий</i>	<i>устный опрос</i>	<i>88-103</i>
ПК-3	<i>промежуточный</i>	<i>тест</i>	<i>1-87</i>

7.1. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины

Результаты освоения дисциплины	Оценочные средства (перечень вопросов, заданий и др.)
Знает: ПК-3 Стандартные пакеты прикладных программ, ориентированных на решение научных и проектных задач	<ol style="list-style-type: none"> 1. Общие свойства функциональных языков: краткость, модульность, строгая типизация, чистота, ленивые вычисления. 2. Признаки сопоставимости термов. Общая схема доказательства целевого утверждения. 3. Рекурсия: определение рекурсивных правил, использование отсечения. 4. Встроенные предикаты для арифметических вычислений. Встроенные предикаты ввода-вывода, управления базой данных (знаний). 5. Сортировка списков: метод пузырьковой сортировки. 6. Ввод - вывод информации. Чтение и запись информации в файлы, входные и выходные потоки. 7. Сортировка списков: метод наивной сортировки, метод сортировки вставкам. 8. Деревья: определение, построение.

	<p>Определение отношения «внутри». Вычисление глубины дерева.</p> <p>9. Структура данных – список. Шаблоны списков, сопоставление.</p>
<p>Умеет: ПК-3</p> <p>Использовать стандартные пакеты прикладных программ для решения практических задач</p>	<ol style="list-style-type: none"> 1. Глобальные и локальные переменные. Задача вычисления сопротивления электрической цепи. 2. Дополнительные функции обработки списков: конкатенации списков(append), обращения списка (reverse), выделения последнего элемента списка (last). Примеры применения. 3. Базовые предикаты Лиспа: сравнения символов (eq), сравнения чисел и символов (eq1), общий предикат сравнения (equal). Примеры применения. 4. Логические функции Лиспа: и, или, отрицание. Условное предложение cond. Примеры применения. 5. Условные предложения if, when, unless. Их отличия от общей формы cond. Примеры вычислений с использованием этих функций. 6. Ввод - вывод информации. Чтение и запись информации в файлы, входные и выходные потоки. 7. Локальные переменные в Лиспе, конструкция let. 8. Организация циклических вычислений в Лиспе, предложения loop, do. 9. Рекурсия в Лиспе, правило записи рекурсивных функций. Численная рекурсия. 10. Общая форма рекурсии. CDR – рекурсия. 11. Функционалы в Лиспе, отображающий функционал MAPCAR. Применяющие функционалы. 12. Лямбда – выражения, их структура, применение. 13. Свойства символов. Операторы получения (get), изменения (общая форма setf), получения списка свойств (symbol-plist). 14. Внутреннее представление списков. Сборка мусора. 15. Обработка списков без разрушения, разрушающие функции, проблемы их применения. 16. Параметры функций: необязательные, переменное количество параметров связанных с хвостом 17. Профилирование программ на логическом языке. 18. Автоматизированное тестирование программ на логическом языке. 19. Отладка программ на логическом языке. 20. Работа с MIDI-интерфейсом. 21. Создание музыкальных произведений. 22. Средства цифровой обработки сигналов. 23. Синтезирование звука и звуковых эффектов. 24. Представление виртуального звукового пространства в 3D. 25. Механизмы распределенных вычислений в программах на языке Haskell. 26. Представление и обработка сложных структур данных в программах на Haskell. 27. Средства работы с регулярными выражениями.

	<p>28. Работа с СУБД.</p> <p>29. Средства автоматизации разработки игровых программ.</p> <p>30. Разработка графического интерфейса пользователя средствами библиотеки</p>
<p><i>Имеет практический опыт: ПК-3</i></p> <p>При работе с современными программными средствами подготовки конструкторско-технологической документации</p>	<ol style="list-style-type: none"> 1. Понятие декларативного программирования. Основные принципы функционального и логического программирования. 2. Общие свойства функциональных языков: краткость, модульность, строгая типизация, чистота, ленивые вычисления. 3. Основные элементы языка логического программирования пролог: определение, простые и структурные объекты данных. 4. Структура пролог – программы, типы предложений программы, пример логического вывода (реализация дерева родственных отношений). 5. Определение арности предикатов. Признак сопоставимости термов, общая схема доказательства целевого утверждения. 6. Признаки сопоставимости термов. Общая схема доказательства целевого утверждения. Операционная семантика языка Пролог. 7. Рекурсия в прологе: определение рекурсивных правил, использование отсечения. 8. Алгоритм логического вывода пролог - машины на примере вычислений с «откатом». 9. Встроенные предикаты для арифметических вычислений. Встроенные предикаты ввода-вывода, управления базой данных (знаний). 10. Предикаты управления на примере «отсечения». 11. Зеленое и красное «отсечение» на примерах выбора наилучшего фильма и добавления в список отсутствующего элементов. 12. Встроенные предикаты преобразования структур. 13. Структура данных – список. Шаблоны списков, сопоставление. 14. Примеры отношений обработки списков: замена первого элемента списка (relace_first), определения вхождения в список (member), определения длины списка (length). 15. Примеры отношений обработки списков: конкатенации списков (append), обращение списка (reverse). Использование конкатенации для поиска комбинаций элементов, удаления элементов, нахождения последнего элемента списка. 16. Сортировка списков: метод наивной сортировки, метод сортировки вставками. 17. Сортировка списков: метод пузырьковой сортировки. 18. Реализация повторений (циклов) на прологе: метод обхода без возвратов, метод поиска с возвратом, метод cat and fail (CAF), метод user define repeat (UDR). 19. Деревья в Прологе: определение, построение.

	<p>Определение отношения «внутри». Вычисление глубины дерева.</p> <p>20. Двоичные справочники: определение, поиск элементов. Отношение, реализующее проверку: является ли объект двоичным справочником.</p> <p>21. Добавление элементов в двоичное дерево.</p> <p>22. Удаление элементов двоичного дерева, удаление листьев, удаление внутренних вершин.</p> <p>23. Отображение деревьев сверху - вниз (корень наверху, листья внизу), слева – направо (корень самый левый выводимый элемент).</p> <p>24. Операции в прологе, их приоритет, тип. Определение собственных операций.</p> <p>25. Основные понятия языка функционального программирования Лисп. Особенности языка.</p> <p>26. S-выражения в Лиспе. Символьные, числовые атомы. Списки.</p> <p>27. Функции в Лиспе. Единая префиксная запись, ее достоинства.</p> <p>28. Вычисление выражений. Функции quote, eval.</p> <p>29. Использование символов в качестве переменных.</p> <p>30. Базовые функции Лиспа. Функции работы со списками car, cdr, list.</p> <p>31. Базовые функции Лиспа. Функции cons, length, atom, eq.</p> <p>32. Определение функций в языке Лисп. Предикаты.</p>
--	--

7.2. Методические рекомендации к определению процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Рабочая учебная программа дисциплины содержит следующие структурные элементы:

- перечень компетенций, формируемых в результате изучения дисциплины в процессе освоения образовательной программы;

- типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности в процессе освоения образовательной программы (далее–задания). Задания по каждой компетенции, как правило, не должны повторяться.

Требования по формированию задания на оценку ЗНАНИЙ:

- обучающийся должен воспроизводить и объяснять учебный материал с требуемой степенью научной точности и полноты;

- применяются средства оценивания компетенций: тестирование, вопросы по основным понятиям дисциплины и т.п.

Требования по формированию задания на оценку УМЕНИЙ:

- обучающийся должен решать типовые задачи (выполнять задания) на основе воспроизведения стандартных алгоритмов решения;

- применяются следующие средства оценивания компетенций: простые ситуационные задачи (задания) с коротким ответом или простым действием, упражнения, задания на соответствие или на установление правильной последовательности, эссе и другое.

Требования по формированию задания на оценку навыков и (или) ОПЫТА ДЕЯТЕЛЬНОСТИ:

- обучающийся должен решать усложненные задачи (выполнять задания) на основе приобретенных знаний, умений и навыков, с их применением в определенных ситуациях;

- применяются средства оценивания компетенций: задания требующие многошаговых решений как в известной, так и в нестандартной ситуациях, задания, требующие поэтапного решения и развернутого ответа, ситуационные задачи, проектная деятельность, задания

расчетно-графического типа. Средства оценивания компетенций выбираются в соответствии с заявленными результатами обучения по дисциплине.

Процедура выставления оценки доводится до сведения обучающихся в течение месяца с начала изучения дисциплины путем ознакомления их с технологической картой дисциплины, которая является неотъемлемой частью рабочей учебной программы по дисциплине.

В результате оценивания компетенций по дисциплине студенту начисляются баллы по шкале, указанной в рабочей учебной программе по дисциплине.

7.3. Описание показателей и критериев оценивания компетенций, описание шкал оценивания

Успешность усвоения дисциплины характеризуется качественной оценкой на основе листа оценки сформированности компетенций, который является приложением к зачетно-экзаменационной ведомости при проведении промежуточной аттестации по дисциплине.

Критерии оценивания компетенций

Компетенция считается сформированной, если теоретическое содержание курса освоено полностью; при устных собеседованиях студент исчерпывающе, последовательно, четко и логически стройно излагает учебный материал; свободно справляется с задачами, вопросами и другими видами заданий, требующих применения знаний, использует в ответе дополнительный материал; все предусмотренные рабочей учебной программой задания выполнены в соответствии с установленными требованиями, студент способен анализировать полученные результаты; проявляет самостоятельность при выполнении заданий, качество их выполнения оценено числом баллов от 86 до 100, что соответствует *повышенному уровню* сформированности компетенции.

Компетенция считается сформированной, если теоретическое содержание курса освоено полностью; при устных собеседованиях студент последовательно, четко и логически стройно излагает учебный материал; справляется с задачами, вопросами и другими видами заданий, требующих применения знаний; все предусмотренные рабочей учебной программой задания выполнены в соответствии с установленными требованиями, студент способен анализировать полученные результаты; проявляет самостоятельность при выполнении заданий, качество их выполнения оценено числом баллов от 61 до 85,9, что соответствует *пороговому уровню* сформированности компетенции.

Компетенция считается несформированной, если студент при выполнении заданий не демонстрирует знаний учебного материала, допускает ошибки, неуверенно, с большими затруднениями выполняет практические работы, не демонстрирует необходимых умений, доля невыполненных заданий, предусмотренных рабочей учебной программой составляет 55 %, качество выполненных заданий не соответствует установленным требованиям, качество их выполнения оценено числом баллов ниже 61, что соответствует *допороговому уровню*.

Шкала оценки уровня освоения дисциплины

Качественная оценка может быть выражена: в процентном отношении качества усвоения дисциплины, которая соответствует баллам, и переводится в уровневую шкалу и оценки «отлично» / 5, «хорошо» / 4, «удовлетворительно» / 3, «неудовлетворительно» / 2, «зачтено», «не зачтено». Преподаватель ведет письменный учет текущей успеваемости студента в соответствии с технологической картой по дисциплине.

Шкала оценки результатов освоения дисциплины, сформированности компетенций

Шкалы оценки уровня сформированности компетенции (й)		Шкала оценки уровня освоения дисциплины		
<i>Уровневая шкала оценки компетенций</i>	<i>100 балльная шкала, %</i>	<i>100 балльная шкала, %</i>	<i>5-балльная шкала, дифференцированная оценка/балл</i>	<i>недифференцированная оценка</i>

допороговый	ниже 61	ниже 61	«неудовлетворительно» / 2	Не зачтено
пороговый	61-85,9	70-85,9	«хорошо» / 4	зачтено
		61-69,9	«удовлетворительно» / 3	зачтено
повышенный	86-100	86-100	«отлично» / 5	зачтено

8. Учебно-методическое и информационное обеспечение дисциплины

8.1. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Списки основной литературы

1. Иванова, Г. С. Технология программирования [Текст]: учеб. для вузов по направлению "Информатика и вычисл. техника" / Г. С. Иванова. - 3-е изд., стереотип. - М.: КноРус, 2016. - 334 с. : ил.;

2. Орлов, С. А. Теория и практика языков программирования [Текст]: учеб. для вузов по направл. "Информатика и вычисл. техника" / С. А. Орлов. - СПб.: Питер, 2014. - 688 с. : ил.;

Списки дополнительной литературы

5. Абдикеев, Н. М. Проектирование интеллектуальных систем в экономике [Текст]: учеб. для вузов по специальности "Приклад. информатика (по областям)" / Н. М. Абдикеев под ред. Н. П. Тихомирова; Рос. экон. акад. им. Г. В. Плеханова. - М.: Экзамен, 2004. - 526 с.: ил.

6. Глухих, И. Н. Интеллектуальные информационные системы [Текст]: учеб. пособие для высш. проф. образования / И. Н. Глухих Тюмен. гос. ун-т. - М.: Академия, 2010. - 110 с.

7. Ездаков, А. Л. Функциональное и логическое программирование [Текст]: учеб. пособие / А. Л. Ездаков. - 2-е изд. - М.: БИНОМ. Лаб. знаний, 2011. - 119 с. : табл.

8. Нейронные сети. STATISTICA Neural Networks: Методология и технологии современного анализа данных [Текст]: [справ. изд.] / под ред. В. П. Боровикова. - 2-е изд., перераб. и доп. - М.: Горячая линия - Телеком, 2008. - 392 с. : ил.

9. Подчукаев, В. А. Теория информационных процессов и систем [Текст]: учеб. пособие для вузов по специальности "Информ. системы и технологии" / В. А. Подчукаев. - М.: Гардарики, 2007. - 207 с. : ил.

4. Программирование [Электронный ресурс]: учеб. для студентов вузов по направлению "Пед. образование": в 2 т. Т. 2 / Э. А. Нигматулина [и др.] под ред. Н. И. Пака. - Документ Adobe Acrobat. - М.: Академия, 2013. - 60,4 МБ, 240 с. : ил. - Режим доступа: <http://elib.tolgas.ru>

5. Рыбина, Г. В. Основы построения интеллектуальных систем [Текст]: учеб. пособие по специальности "Приклад. информатика" и др. экон. специальностям / Г. В. Рыбина. - М.: Финансы и статистика, 2010. - 432 с. : ил.

6. Сергиевский, Г. М. Функциональное и логическое программирование [Электронный ресурс]: учеб. пособие для вузов по направлению "Информатика и вычисл. техника" / Г. М. Сергиевский, Н. Г. Волчёнков. - Документ Adobe Acrobat. - М.: Академия, 2010. - 57,5 МБ, 319 с. : ил. - Режим доступа: <http://elib.tolgas.ru>

7. Сошников, Д. В. Функциональное программирование на F# [Текст] / Д. В. Сошников. - М.: ДМК Пресс, 2011. - 190 с. : ил.

8. Судоплатов, С. В. Математическая логика и теория алгоритмов [Текст]: учеб. для вузов по направлениям "Информатика и вычисл. техника", "Информ. системы", "Физ.-мат. образование" / С. В. Судоплатов, Е. В. Овчинникова Новосибир. гос. техн. ун-т. - М. [и др.]: ИНФРА-М [и др.], 2008. - 224 с. : схем.

5. Цуканова, Н. И. Теория и практика логического программирования на языке Visual Prolog 7 [Текст]: учеб. пособие для вузов по специальности "Прогр. обеспечение вычисл. техники и автоматизир. систем" / Н. И. Цуканова, Т. А. Дмитриева. - М. : Горячая линия - Телеком, 2013. - 232 с. : ил.

5. Ясницкий, Л. Н. Введение в искусственный интеллект [Текст] : учеб. пособие для вузов по специальности "Математика" / Л. Н. Ясницкий. - М. : Академия, 2005. - 175 с. : ил..

8.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет" (далее - сеть "Интернет"), необходимых для освоения дисциплины **Интернет-ресурсы**

1. ИНТУИТ. Национальный Открытый Университет [Электронный ресурс]. – Режим доступа: <http://www.intuit.ru/>. – Загл. с экрана.

2. Образовательные ресурсы Интернета. Информатика [Электронный ресурс]. - Режим доступа: <http://www.alleng.ru/edu/comp.htm>. - Загл. с экрана.

3. Электронная библиотека. Техническая литература [Электронный ресурс]. - Режим доступа: <http://techliter.ru/>. – Загл. с экрана.

4. Электронная библиотечная система Поволжского государственного университета сервиса [Электронный ресурс]. - Режим доступа: <http://elib.tolgas.ru/>. - Загл. с экрана.

5. Электронно-библиотечная система Znanium.com [Электронный ресурс]. - Режим доступа: <http://znanium.com/>. – Загл. с экрана.

9. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем

Краткая характеристика применяемого программного обеспечения

п/п	Программный продукт	Характеристика	Назначение при освоении дисциплины
	Текстовый и табличный процессоры.	Прикладная компьютерная программа, предназначенная для производства (включая набор, редактирование, форматирование, иногда печать) любого вида печатной информации, а также для обеспечения работы с большими таблицами чисел.	Выполнение и оформление отчетов по лабораторным работам
	Среда программирования Prolog	Пролог (<i>англ.</i> Prolog) — язык и система логического программирования, основанные на языке предикатов математической логики дизъюнктов Хорна, представляющей собой подмножество логики предикатов первого порядка.	Выполнение лабораторных работ
	Пакет Microsoft Office 2007	Пакет Microsoft Office –набор приложений, предназначенных для обработки электронной документации на персональном компьютере.	Выполнение и оформление отчетов по лабораторным работам

		Компоненты офисных пакетов распространяются, как правило, только вместе, имеют схожий интерфейс и хорошо взаимодействуют друг с другом.	
--	--	---	--

10. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Для проведения занятий лекционного типа используются специальные помещения - учебные аудитории, укомплектованные специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации.

Для проведения лабораторных работ используются учебные аудитории, оснащенные персональным компьютером с операционной системой семейства Ms Windows NT не ниже версии 5.1, языком программирования Lisp, средой программирования Prolog, пакетом Microsoft Office 2007.

Для текущего контроля и промежуточной аттестации используются специальные помещения - учебные аудитории, укомплектованные специализированной мебелью, и (или) компьютерные классы, оснащенные компьютерной техникой с возможностью подключения к сети Интернет и обеспечением доступа в электронную информационно-образовательную среду университета.

Для самостоятельной работы обучающихся используются специальные помещения - учебные аудитории для самостоятельной работы, оснащенные компьютерной техникой с возможностью подключения к сети Интернет и обеспечением доступа в электронную информационно-образовательную среду университета.

11. Примерная технологическая карта дисциплины «Функциональное и логическое программирование»

Факультет информационно-технического сервиса
кафедра «Информационный и электронный сервис»
направление подготовки 09.03.04 «Программная инженерия»
направленности (профиля) «Разработка программно-информационных систем»

	Виды контрольных точек	Кол-во контр. точек	Кол-во баллов за 1 контр. точку	График прохождения контрольных точек																зач. неделя
				Февраль				Март				Апрель				Май				
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
Обязательные:																				
.1	посещение лекционных занятий	6	2			+		+		+		+		+		+				
.2	активная работа на практических занятиях	5	7				+		+		+		+		+					
.3	промежуточное тестирование	1	10										+							
.4	итоговое тестирование	1	15															+		
Творческий рейтинг:																				
.1	самостоятельное решение задач								+		+		+							
.2	выполнение и защита практической работы с элементами исследования	4	5												+					
.3	подготовка докладов, рефератов, сообщений	1	8										+							
	Форма контроля																		Экзамен	

