

**Филиал федерального государственного бюджетного образовательного учреждения
высшего образования
«Национальный исследовательский университет «МЭИ»
в г. Смоленске**

УТВЕРЖДАЮ
Зам. директора
филиала ФГБОУ ВО «НИУ «МЭИ»
в г. Смоленске
по учебно-методической работе
В.В. Рожков
«2016 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ ОСНОВНЫЕ ПАРАДИГМЫ ПРОГРАММИРОВАНИЯ

(НАИМЕНОВАНИЕ ДИСЦИПЛИНЫ)

Направление подготовки: 09.03.01 «Информатика и вычислительная техника»

**Профиль подготовки: Программное обеспечение средств вычислительной
техники и автоматизированных систем**

Уровень высшего образования: бакалавриат

Нормативный срок обучения: 4 года

Форма обучения: очная

Смоленск – 2016 г.

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

Целью освоения дисциплины является подготовка обучающихся по направлению подготовки 09.03.01 "Информатика и вычислительная техника" посредством обеспечения этапов формирования компетенций, предусмотренных ФГОС, в части представленных ниже знаний, умений и навыков.

Задачами дисциплины является изучение понятийного аппарата и основных теоретических положений и методов дисциплины, ознакомление студентов с исторически сложившимся разнообразием парадигм программирования, определивших развитие современных информационных технологий, систематизация знаний о возможностях и особенностях применения различных парадигм программирования, усвоение студентами понятий, накопленных в различных формах языков программирования и реализованных с помощью систем программирования, развитие навыков применения теоретических знаний для решения практических задач.

В соответствии с поставленными целями после изучения дисциплины «Основные парадигмы программирования» студенты приобретают знания, умения и опыт, которые определяют компетенциями:

ОПК- 1 - способностью устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем.

ОПК-2 - способностью осваивать методики использования программных средств для решения практических задач

ПК -2 способностью разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования

ПК-3 - способностью обосновывать принимаемые проектные решения, осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности.

В результате изучения дисциплины студенты должны:

Знать:

Особенности эволюции средств и методов разработки программ (ОПК-1).

Концепции и идеи современных парадигм программирования и компьютерных языков (ОПК-1).

Понятие об императивной, процедурной, логической, функциональной, объектно – ориентированной и компонентной парадигмах программирования (ОПК-2).

Характеристики парадигм программирования и компьютерных языков (ОПК-2).

Парадигмы низкоуровневого программирования и программирования на языках высокого и сверхвысокого уровня (ОПК-2).

Теоретическую базу средств формализации различных парадигм программирования (ОПК-2).

Математическую базу функциональных языков программирования (ОПК-2).

Место и роль, состояние развития современных функциональных языков (ОПК-2)

Средства и методы программирования, поддерживаемые разными языками и системами программирования (ПК-2).

Сравнительный анализ наиболее известных парадигм программирования и парадигматическая характеристика языков параллельного программирования разного уровня (ПК-2).

Проблемы и направления развития современных программных средств функционального программирования (ПК-2).

Проблематику параллельного программирования (ПК-2).

Основные подходы к выбору языковых средств в процессе разработки программ (ПК-3).

Проблемы жизненного цикла программ (ПК-3).

Представление о взаимосвязи между показателями качества программы и качества процесса ее разработки на базе основных парадигм программирования (ПК-3).

Модели вычислений, включая модели параллелизма, а также методы программирования параллельных процессов (ПК-3).

Проблемы организации высокопроизводительных вычислений (ПК-3).

Подходы к разработке и конструированию программ на базе информационных и компьютерных технологий (ПК-3).

Способы совместного использования различных программных средств на платформе Microsoft .NET (ПК-3)

Уметь:

Использовать низкоуровневые языки и языки логического и функционального программирования (ОПК-1).

Использовать инструментальные средства логического программирования и языки Пролог или Mercury (ОПК-1).

Видеть проблемы развития и применения парадигм программирования в разных аспектах (ОПК-2).

Делать обоснованный выбор парадигм программирования для решения задач разного типа (ОПК-2).

Ориентироваться в современных языках функционального программирования (ОПК-2).

Проектировать и реализовывать программы на одном из языков низкого уровня (ПК-2).

Составлять и отлаживать программы на языках программирования высокого уровня (ПК-2).

Выполнять анализ парадигм программирования и компьютерных языков для удовлетворения требований и создания сценариев использования программного продукта (ПК-3).

Формализовать и представить концепцию разрабатываемой информационной системы (ПК-3).

Прогнозировать трудоемкость жизненного цикла программного продукта в зависимости от выбора парадигм программирования и компьютерных языков (ПК-3).

Владеть:

Популярными парадигмами программирования и компьютерных языков (ОПК-1).

Методикой программирования на основе языков императивной, декларативной и объектно – ориентированной парадигм (ОПК-2).

Пониманием отличий различных парадигм программирования (ОПК-2).

Принципами логического и функционального программирования (ОПК-2).

Основными приемами использования языков ассемблер, Пролог или Mercury, различных версий языка ЛИСП для решения типовых задач (ПК-2).

Средствами формализации наиболее распространенных парадигм императивного, объектно-ориентированного, функционального и логического программирования (ПК-3).

Пониманием связи формальной теории логики предикатов первого порядка с логическим программированием и связи формализованных систем λ - исчислений и комбинаторной логики с функциональным программированием (ПК-3).

Пониманием возможностей и проблем различных языков параллельного программирования (ПК – 3).

2. Место дисциплины в структуре образовательной программы

Дисциплина «Основные парадигмы программирования» является дисциплиной по выбору вариативной части профессионального цикла Б1 (Б1.В.ДВ.8.1) основной образовательной программы подготовки бакалавров по профилю «Вычислительные машины, комплексы, системы и сети», направления 09.03.01 «Информатика и вычислительная техника».

В соответствии с учебным планом по направлению "Информатика и вычислительная техника" дисциплина «Основные парадигмы программирования» базируется на следующих дисциплинах:

- Б1.Б.7 Информатика
- Б1.Б.6 Теория вероятностей и математическая статистика
- Б1.Б.8 Инженерная графика
- Б1.Б.9 ЭВМ и периферийные устройства
- Б1.Б.10 Базы данных
- Б1.Б.14 Высшая математика
- Б1.Б.15 Вычислительная математика
- Б1.Б.16 Электротехника
- Б1.Б.17 Электроника
- Б1.Б.18 Схемотехника
- Б1.В.ОД.1 Программирование
- Б1.В.ОД.2 Дискретная математика
- Б1.В.ОД.3 Теория алгоритмов
- Б1.В.ОД.4 Операционные системы
- Б1.В.ОД.5 Компьютерная графика
- Б1.В.ОД.6 Технология программирования
- Б1.В.ОД.7 Сети и телекоммуникации
- Б1.В.ОД.12 Теория автоматов
- Б1.В.ОД.13 Основы теории управления
- Б1.В.ОД.15 Сопровождение разработки программного обеспечения
- Б1.В.ОД.16 Объектно-ориентированные технологии
- Б1.В.ДВ.3.1 Введение в оптимизацию
- Б1.В.ДВ.3.2 Теория систем
- Б1.В.ДВ.5.1 Прикладная статистика
- Б1.В.ДВ.9.1 Теория формальных языков
- Б1.В.ДВ.9.2 Трансляторы программных языков

Знания, умения и навыки, полученные в процессе изучения дисциплины «Основные парадигмы программирования» являются базой для изучения следующих дисциплин:

- Б1.В.ОД.8 Сетевые технологии
- Б1.В.ОД.9 Микропроцессорные системы
- Б1.В.ОД.10 Защита информации
- Б1.В.ОД.11 Моделирование
- Б1.В.ОД.14 Тестирование программного обеспечения
- Б1.В.ОД.17 Проектирование Web-приложений
- Б1.В.ДВ.4.1 Введение в цифровую обработку сигналов
- Б1.В.ДВ.6.1 Аппаратная реализация алгоритмов
- Б1.В.ДВ.6.2 Технология проектирования устройств на ПЛИС
- Б1.В.ДВ.7.1 Теория передачи информации
- Б1.В.ДВ.10.1 Организация разработки программного обеспечения
- Б1.В.ДВ.10.2 Разработка IT-систем
- Б2.П.4 Преддипломная

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 обучающихся

Аудиторная работа

Цикл:	Б1 – профессиональный цикл	Семестр
Часть цикла:	Дисциплина по выбору	
№ дисциплины по учебному плану:	Б1.В.ДВ.8.1	
Часов (всего) по учебному плану:	216	7 семестр
	72	8 семестр
Трудоемкость в зачетных единицах (ЗЕТ)	6	7 семестр
	2	8 семестр
Лекции (ЗЕТ; часов)	1, 36	7 семестр
	0. 39,14	8 семестр
Лабораторные работы (ЗЕТ; часов)	1, 36	7 семестр
	0.39,14	8 семестр
Практические занятия (ЗЕТ; часов)	1, 36	7 семестр
	-	8 семестр
Объем самостоятельной работы по учебному плану (ЗЕТ; часов)	3,108	7 семестр
	1.2, 44	8 семестр
Зачет (в объеме самостоятельной работы ЗЭТ, часов)	0.5, 18	7 семестр
	0.5, 18	8 семестр

Самостоятельная работа студентов

Вид работ	Трудоемкость, ЗЕТ, час
Изучение материалов лекций (лк)	1.4, 50
Подготовка к практическим занятиям (пз)	0.7, 26
Подготовка к выполнению и защите лабораторной работы (лаб)	1.1, 40
Выполнение расчетно-графической работы (реферата)	
Выполнение курсового проекта (работы)	
Самостоятельное изучение дополнительных материалов дисциплины (СРС)	
Подготовка к контрольным работам	
Подготовка к тестированию	
Подготовка к зачетам	1, 36
Всего:	4.2, 152

4. Содержание дисциплины, структурированное по темам с указанием отведенного на них количества академических часов и видов учебных занятий.

№ п/ п	Разделы дисциплины	Семестр	Всего часов на раздел	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)				
				Лк	Лр	Пз	СРС	В т.ч. ин- тер- акт.
1	Раздел 1. Классификация языков и систем программирования	7	20	4	4	4	8	
2	Раздел 2. Понятие о базах знаний.	7	10	2	4		4	
3	Раздел 3. Модели баз знаний	7	20	4	4	4	8	
4	Раздел 4. Логические модели баз знаний.	7	42	8	8	8	18	
5	Раздел 5. Принцип резолюции.	7	10	2		4	4	
6	Раздел 6. Опровергающие деревья.	7	12	2	4		6	
7	Раздел 7. Понятие о логической программе.	7	32	4	8	4	16	
8	Раздел 8. Элементы языка Пролог.	7	40	8		8	24	
9	Раздел 9. Общение с ЭВМ на языке, близком к естественному.	7	30	2	4	4	20	
10	Раздел 10. Основы формализации языков функционального программирования.	8	20	6	4	-	10	
11	Раздел 11. Языки функционального программирования.	8	12	2	4	-	6	
12	Раздел 12. Параллельное программирование.	8	12	2	-	-	10	
13	Раздел 13. Концепция Microsoft.NET.	8	18	2	6	-	10	
14	Раздел 14. Синтаксис и семантика языков программирования.	8	10	2	-	-	8	
Зачеты с оценкой								
Всего по видам занятий			288	50	50	36	152	

Содержание дисциплины по видам учебных занятий

Раздел 1. Классификация языков и систем программирования.

Лекция 1. Общие сведения о парадигмах программирования. Эволюция языков программирования. Языки низкого уровня. Императивно – процедурная парадигма. Структурное программирование. Объектно – ориентированная технология программирования. (2 час.)

Лабораторная работа 1. Программирование алгебраических задач и задач на графовых моделях на языке Prolog. (4 час.).

Лекция 2. Декларативное (функциональное и логическое) программирование. Парадигма программирования на основе логики предикатов первого порядка. Парадигма программирования на основе лямбда - исчислений. Сферы применения декларативных языков. Языки параллельного программирования. Скриптовые языки. Компонентная технология. .Net технологии. Языки управления базами данных и разметки. (2 час.).

Практическое занятие 1.

Программирование линейных алгоритмов на низкоуровневом языке. (4 час.).

Самостоятельная работа.

Подготовка к лекциям (4 час.), лабораторной работе (2 час.) и практическому занятию (2 час.).

Раздел 2. Понятие о базах знаний.

Лекция 3. Математические основы логического программирования. Понятие о системах искусственного интеллекта. Экспертные системы. Понятие о базах знаний. Виды знаний. Формализованные системы описания и представления знаний. Языки представления знаний (2 час.).

Лабораторная работа 2.

Циклы и рекурсивные программы. Обработка списочных структур данных (4 час.).

Самостоятельная работа.

Подготовка к лекции (2 час.) и лабораторной работе (2 час.).

Раздел 3. Модели баз знаний.

Лекция 4. Семантические сети. Сферы применения семантических сетей для формализации знаний в логических программах. Фреймы. Протофреймы и экзофреймы. Классы фреймов. Возможности эквивалентных преобразований знаний, формализованных фреймами (2 час.).

Практическое занятие 2.

Программирование ветвлений на низкоуровневом языке. (4 час.).

Лекция 5. Системы продукций. Прямые и обратные системы продукции. Примеры применения. Разложимые системы продукций. Применение продукций в логических программах.

Реляционные модели. Основные определения. Реляционная алгебра.

Основные реляционные операции над отношениями (2 час.).

Лабораторная работа 3.

Работа с табличными данными и файлами. Работа с динамическими базами данных (4 час.).

Самостоятельная работа.

Подготовка к лекциям (4 час.), лабораторной работе (2 час.) и практическому занятию (2 час.).

Текущий контроль. Опрос при проведении допуска к лабораторной работе.

Раздел 4. Логические модели баз знаний.

Лекция 6. Логические модели. Логика высказываний. Язык логики высказываний. Интерпретации. Эквивалентные преобразования сложных высказываний. Правила преобразований.

Логическое следование в логике высказываний. Теорема и следствие о логическом следовании. Доказательство корректности выводов и правил выводов (2 час.).

Практическое занятие 3.

Программирование ветвлений на низкоуровневом языке. (4 час.).

Лекция 7. Исчисление высказываний. Логический вывод. Средства вывода в исчислении высказываний. Основные аксиомы и правила вывода в исчислении высказываний. Примеры доказательств теорем исчисления высказываний. Непротиворечивость и полнота исчисления высказываний. Разрешимость исчисления высказываний (2 час.).

Лабораторная работа 4.

Моделирование и расчеты электрических и логических схем. Разработка модульных программ с применением оконного интерфейса на языке Пролог. (4 час.).

Текущий контроль. Опрос при проведении допуска к лабораторной работе.

Лекция 8. Логика предикатов. Язык логики предикатов. Эквивалентные преобразования предложений логики предикатов. Связь с предложениями Хорна логических программ. Нормальная и предваренная формы. Сколемовские преобразования предложений, имеющих кванторы существования (2 часа).

Практическое занятие 4.

Программирование циклов на низкоуровневом языке. (4 час.).

Лекция 9. Исчисление предикатов. Средства вывода в исчислении предикатов. Основные аксиомы и правила вывода в исчислении предикатов. Непротиворечивость и полнота исчисления предикатов. Разрешимость исчисления предикатов (2 час.).

Лабораторная работа 5.

Реализация на Турбо Прологе алгоритмов сортировки (4 час.).

Самостоятельная работа.

Подготовка к лекциям (8 час.) и выполнению лабораторных работ (6 час.) и практическим занятиям (4 час.).

Раздел 5. Принцип резолюции.

Лекция 10. Метод резолюций. Резольвенты. Полнота и непротиворечивость метода. Понятие о дизъюнктах. Отличие от классических схем вывода. Демонстрация метода на примере дерева вывода. Резолюции в логике высказываний. Пример построения вывода. Резолюции в логике предикатов (2 час.).

Практическое занятие 5.

Программирование экранных операций на низкоуровневом языке. (4 час.).

Самостоятельная работа.

Подготовка к лекции (2 час.) и практическому занятию (2 час.).

Раздел 6. Опровергающие деревья.

Лекция 11. Построение опровергающих деревьев в логике высказываний. Доказательство непротиворечивости множества дизъюнктов. Замкнутые деревья (2 час.).

Лабораторная работа 6.

Разработка проектов, содержащих внешние модули, на языке Турбо Пролог (4 час.).

Самостоятельная работа.

Подготовка к лекции (2 час.) и выполнению лабораторной работы (4 час.).

Текущий контроль. Опрос при проведении допуска к лабораторной работе.

Раздел 7. Понятие о логической программе.

Лекция 12. Понятие о логической программе. Структура логической программы. Факты, правила, целевые утверждения. Универсальные правила. Алгоритм доказательства цели (2 час.).

Практическое занятие 6.

Программирование записей и структур на низкоуровневых языках (4 час.).

Лекция 13. Унификация. Подстановки. Композиция подстановок. Множество рассогласований. Стратегии унификации.

Пример дерева вывода для логической программы стратегией “вглубь”. Управление процессом решения в логической программе. Бэктрекинг. Отсечение. Повторение (2 час.).

Лабораторная работа 7.

Разработка клиент – серверной программы на высокоуровневом языке (например, Delphi 7) (8 час.).

Самостоятельная работа.

Подготовка к лекциям (4 час.) и выполнению лабораторной работы (8 час.) и практическому занятию (4).

Текущий контроль. Опрос при проведении допуска к лабораторной работе.

Раздел 8. Элементы языка Пролог.

Лекция 14. Структура программы в Турбо-Прологе. Структуры данных в Турбо-Прологе. Ввод и вывод данных. Декларативные и процедурные элементы Турбо-Пролога. Организация ветвлений и циклов в программе. Рекурсии (2 час.).

Лекция 15. Программирование отношений в Турбо-Прологе. Примеры программирования композиций отношений. Организация работы с файлами. Средства Турбо-Пролога для работы с файлами. Динамические базы данных. Стандартные предикаты для работы с ними (2 час.).

Практическое занятие 7.

Работа с подпрограммами на низкоуровневом языке (4 час.).

Лекция 16. Графические средства Турбо-пролога.

Модульное программирование. Организация меню (2 час.).

Практическое занятие 8

Программирование преобразований в различных системах счислений на низкоуровневом языке (4 час.).

Лекция 17 Программирование технических задач. Применение логических программ для решения технических задач. Моделирование и расчеты электрических цепей и логических схем на Прологе. Программирование проектов на Турбо Прологе (2 час.).

Самостоятельная работа.

Подготовка к лекциям (8 час.), зачету (8 час.) и практическим занятиям (8 час.).

Раздел 9. Общение с ЭВМ на языке, близком к естественному.

Лекция 18. Языки общения. Пример системы общения.

Структура лингвистического транслятора в системах общения.

Применение Пролога в системах общения (2 час.).

Защита лабораторных работ и практических занятий (8 час.).

Самостоятельная работа. Подготовка к зачету (10), лекции (2 час.) и к защитам лабораторных работ и практических занятий (8 час.).

Раздел 10. Основы формализации языков функционального программирования.

Лекция 19. Средства формализации языков функционального программирования. λ - исчисление как основа формализации языков функционального программирования. Средства формализации λ – исчисления. Аксиомы и правила вывода.

Лямбда – выражения. Лямбда – термы. Аппликация и абстракция. Свободные и связанные переменные. Комбинаторы.

Подстановки. Конверсия. Равенство и редукция лямбда – термов (2 час.)

Лекция 20. Представление данных в лямбда – исчислении. Булевские значения. Пары и кортежи. Натуральные числа. Рекурсивные функции и множества. Именованные выражения. Типизация (2 час.)

Лекция 21. Комбинаторная логика как абстракция формализации языков функционального программирования и абстрактных вычислительных машин. Категориальная абстрактная машина (2 час.)

Лабораторная работа 8.

Моделирование на языке высокого уровня контроллера флэш – памяти.(4 час.)

Самостоятельная работа. Подготовка к лекциям (6 час.) и лабораторной работе (4 час.).

Текущий контроль. Опрос при проведении допуска к лабораторной работе

Раздел 11. Языки функционального программирования.

Лекция 22. Характеристики языка функционального программирования Лисп. Структуры данных. Атомы. Списки. Выражения. Функции. Композиция функций. S – выражения. Рекурсии S – выражений. Каррирование. Правила записи программ на Лиспе (2 час.)

Лабораторная работа 9.

Разработка библиотеки .dll модулей для программы лабораторной работы 8(4 час.)

Самостоятельная работа. Подготовка к лекции (2 час.) и лабораторной работе (4 час.).

Раздел 12. Параллельное программирование.

Лекция 23. Парадигма параллельного программирования. Классы языков и моделей параллельного программирования. Модели параллелизма известных языков программирования. Средства моделирования распараллеливания. Обмен данными. Языки параллельного программирования (2 час.).

Самостоятельная работа. Подготовка к лекции (2 час.) и зачету (8 час.).

Раздел 13. Концепция Microsoft.NET.

Лекция 24. Основные возможности концепции Microsoft.NET. Поддержка многоязыковой среды разработки приложений. Ее возможности. Универсальный многоязыковый интерфейс. Интеграция с технологией Web – сервисов (2 час.).

Лабораторная работа 10.

Разработка сетевой клиент – серверной программы с шифрованием паролей и применением СУБД и SQL – запросов для работы с базой данных клиентов (6 час.).

Самостоятельная работа. Подготовка к лекции (2 час.), зачету (4 час.) и лабораторной работе (4 час.).

Текущий контроль. Опрос при проведении допуска к лабораторной работе

Раздел 14. Синтаксис и семантика языков программирования.

Лекция 25. Способы задания и нотации синтаксиса языков функционального программирования. Нотация Бэкуса – Наура (БНФ). Синтаксис комбинаторной логики. Компиляторы и интерпретаторы. Концептуальные подходы к моделированию семантики языков программирования..

Самостоятельная работа. Подготовка к зачету (6 час.) и лекции (2 час.).

5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

Самостоятельная работа студентов по дисциплине организуется в соответствии с «Положением об организации самостоятельной работы студентов», утвержденным заместителем директора филиала ФБГОУ ВО «НИУ «МЭИ» в г. Смоленске 04.02.2014 г.

Для обеспечения самостоятельной работы студентов разработаны:

- методические рекомендации по самостоятельной работе (Приложение Б1.В.ДВ.8.1(СРС));
- методические указания к лабораторным работам (Приложение Б1.В.ДВ.8.1 (лр));
- конспект лекций (Приложение Б1.В.ДВ.8.1 (лк)).

6. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

6.1 Перечень компетенций с указанием этапов их формирования

При освоении дисциплины формируются следующие компетенции:

ОПК- 1 - способностью устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем.

ОПК-2 - способностью осваивать методики использования программных средств для решения практических задач

ПК -2 способностью разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования

ПК-3 - способностью обосновывать принимаемые проектные решения, осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности.

Указанные компетенции формируются в соответствии со следующими этапами:

1. Формирование и развитие теоретических знаний, предусмотренных указанными компетенциями (лекционные занятия, самостоятельная работа студентов).
2. Приобретение и развитие практических умений, предусмотренных компетенциями (практические занятия, самостоятельная работа студентов).
3. Закрепление теоретических знаний, умений и практических навыков, предусмотренных компетенциями, в ходе решения конкретных технических задач на практических занятиях, успешной сдачей экзамена и получения зачета.

6.2 Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описания шкал оценивания

Сформированность компетенции в рамках освоения данной дисциплины оценивается по трехуровневой шкале:

- **пороговый** уровень является обязательным для всех обучающихся по завершении освоения дисциплины;
- **продвинутый** уровень характеризуется превышением минимальных характеристик сформированности компетенции по завершении освоения дисциплины;
- **эталонный** уровень характеризуется максимально возможной выраженностью компетенции и является важным качественным ориентиром для самосовершенствования.

При достаточном качестве освоения более 80% приведенных знаний, умений и навыков преподаватель оценивает освоение данной компетенции в рамках настоящей дисциплины на эталонном уровне, при освоении более 60% приведенных знаний, умений и навыков – на продвинутом, при освоении более 40% приведенных знаний, умений и навыков – на пороговом уровне.

В противном случае компетенция в рамках настоящей дисциплины считается неосвоенной.

Уровень сформированности каждой компетенции на различных этапах ее формирования в процессе освоения данной дисциплины оценивается в ходе текущего контроля успеваемости и представлен различными видами оценочных средств.

Общая оценка сформированности компетенций определяется на этапе промежуточной аттестации.

Формой промежуточной аттестации по данной дисциплине является экзамен на 4 курсе.

Экзамен и зачет с оценкой оцениваются по принятой в НИУ «МЭИ» четырехбальной системе: "отлично", "хорошо", "удовлетворительно", "неудовлетворительно".

Оценка «удовлетворительно» означает, что все компетенции, закрепленные за дисциплиной, освоены на уровне не ниже порогового.

Оценка «хорошо» означает, что все компетенции, закрепленные за дисциплиной, освоены на уровне не ниже продвинутого.

Оценка «отлично» означает, что все компетенции, закрепленные за дисциплиной, освоены на эталонном уровне.

Критерии оценивания для экзамена (в соответствии с инструктивным письмом НИУ МЭИ от 14 мая 2012 года № 21-23:

Оценки «отлично» заслуживает студент, обнаруживший всестороннее, систематическое и глубокое знание материалов изученной дисциплины, умение свободно выполнять задания, предусмотренные программой, усвоивший основную и знакомый с дополнительной литературой, рекомендованной рабочей программой дисциплины; проявивший творческие способности в понимании, изложении и использовании материалов изученной дисциплины, безупречно ответившему не только на вопросы билета, но и на дополнительные вопросы в рамках рабочей программы дисциплины, правильно выполнившему практические задание

Оценки «хорошо» заслуживает студент, обнаруживший полное знание материала изученной дисциплины, успешно выполняющий предусмотренные задания, усвоивший основную литературу, рекомендованную рабочей программой дисциплины; показавшему систематический характер знаний по дисциплине, ответившему на все вопросы билета, правильно выполнившему практические задание, но допустившему при этом не принципиальные ошибки.

Оценки «удовлетворительно» заслуживает студент, обнаруживший знание материала изученной дисциплины в объеме, необходимом для дальнейшей учебы и предстоящей работы по профессии, справляющийся с выполнением заданий, знакомы с основной литературой, рекомендованной рабочей программой дисциплины; допустившим погрешность в ответе на теоретические вопросы и/или при выполнении практических заданий, но обладающий необходимыми знаниями для их устранения под руководством преподавателя, либо неправильно выполнившему практическое задание, но по указанию преподавателя выполнившим другие практические задания из того же раздела дисциплины.

Оценка «неудовлетворительно» выставляется студенту, обнаружившему серьезные пробелы в знаниях основного материала изученной дисциплины, допустившему принципиальные ошибки в выполнении заданий, не ответившему на все вопросы билета и дополнительные вопросы и неправильно выполнившему практическое задание (неправильное выполнение только практического задания не является однозначной причиной для выставления оценки «неудовлетворительно»).

Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение по образовательной программе без дополнительных занятий по соответствующей дисциплине (формирования и развития компетенций, закреплённых за данной дисциплиной). Оценка «неудовлетворительно» выставляется также, если студент: после начала экзамена отказался его сдавать или нарушил правила сдачи экзамена (списывал, подсказывал, обманом пытался получить более высокую оценку и т.д.).

В зачетную книжку студента выносятся оценки за зачет в седьмом и восьмом семестрах.

В выпуску к диплому выносится оценка на зачете в восьмом семестре.

6.3 Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Примерные задания для контрольных работ

ТЕСТ № 1

1. Упростить выражение: $\overline{A \vee A} \leftrightarrow \overline{B \vee C} \rightarrow D \rightarrow B \vee A \vee \overline{A} \vee B$

2. Доказать или опровергнуть правила вывода:

а) $\frac{A, B, (A \rightarrow B) \rightarrow C}{C \vee D}$ б) $\frac{A \vee B \rightarrow \overline{C} \vee \overline{D}, A, B}{\overline{C} \vee \overline{D}}$

3. Привести к предваренной нормальной форме выражение:

$$\overline{\forall c T(c, t) \leftrightarrow \exists x P(x)} \Rightarrow \overline{\forall y R(x, y) \vee \exists z Q(z)}$$

4. Проверить, правильно ли сделан вывод:

Заработная плата возрастет, если будет инфляция. Если будет инфляция, увеличится стоимость жизни. Заработная плата возрастает. Следовательно, увеличится стоимость жизни.

ТЕСТ № 2

1. Найти композицию унификаций:

$$\theta_1 = \{ f(x)/y, f(y)/x, x/a, y/b \};$$

$$\theta_2 = \{ y/m, x/m, y/x_1, z_1/x \};$$

$$\theta_3 = \{ y/x_1, m/n_1, z/x \}.$$

2. а). Построить дерево опровержений для совокупности предложений, предварительно преобразовав их в дизъюнкты:

$$S = \{ a \rightarrow \overline{x}, x \leftrightarrow b, b \rightarrow a \rightarrow c \};$$

б). Построить таблицу истинностных значений для конъюнкции предложений S. Сравнить результаты с пунктом а).

3. Построить дерево унификаций и вывода цели для процедуры:

$\text{join}([], L, L)$
 $\text{join}([X|L_1], L_2, [X|L_3]) \leftarrow \text{join}(L_1, L_2, L_3)$
 Цель: $\text{join}([1, 2, 3], [4, 5, 6], L)$

4. Найти все следствия из посылок:

$$a \leftrightarrow y \rightarrow z; y \rightarrow \bar{z}; y \vee a$$

Вопросы по формированию и развитию теоретических знаний, предусмотренных компетенциями, выносимые на зачет с оценкой в 7 семестре

1. Общие сведения о парадигмах программирования. Эволюция языков программирования. Языки низкого уровня. Императивно – процедурная парадигма.
2. Структурное программирование. Объектно – ориентированная технология программирования. (2 час.)
3. Декларативное программирование. Парадигма программирования на основе логики предикатов первого порядка. Парадигма программирования на основе лямбда - исчислений. Сферы применения декларативных языков.
4. Языки параллельного программирования. Скриптовые языки. Компонентная технология. .Net технологии. Языки управления базами данных и разметки. (2 час.).
5. Понятие об искусственном интеллекте и экспертных системах. Понятие интеллектуальной задачи. Примеры сфер применения искусственного интеллекта.
6. Понятие о базах знаний. Средства описания и представления знаний. Формы представления знаний.
7. Семантические сети. Явные и неявные зависимости в сетях. Пример применения семантических сетей для логических программ.
8. Фреймы. Протофреймы и экзофреймы. Классы фреймов. Примеры описания задач с помощью фреймов.
9. Системы продукций. Прямые и обратные системы продукции. Примеры применения. Разложимые системы продукций.
10. Реляционные модели. Основные понятия и операции реляционной алгебры. Возможности выводимости и истинностно – ложной интерпретации.
11. Логические модели. Логика высказываний. Интерпретации формул логики высказываний. Основные эквивалентности в логике высказываний. Эквивалентные преобразования сложных высказываний. Правила преобразований.
12. Логическое следование в логике высказываний. Теорема и следствие. Пример применения теоремы.
13. Вывод следствий из посылок. Пример.
14. Нахождение недостающих посылок при известном следовании. Пример.
15. Доказательство корректности выводов из посылок и правил выводов. Пример.
16. Исчисление высказываний. Средства вывода в исчислении высказываний. Понятие логического вывода. Примеры вывода.
17. Логика предикатов. Понятие предиката. Язык логики предикатов. Понятие формул со свойством чистоты предметных переменных. Связь с предложениями Хорна логических программ. Основные эквивалентности формул логики предикатов.
18. Эквивалентные преобразования предложений логики предикатов Нормальная и приведенная формы формул логики предикатов. Примеры приведения.
19. Исчисление предикатов. Система аксиом и правила вывода в исчислении предикатов.
20. Метод резолюций. Резолютивный вывод. Полнота метода. Отличие от классических схем вывода. Понятие о дизъюнктах.
21. Резолюции в логике высказываний. Приведение к дизъюнктам произвольных формул логики высказываний. Пример построения вывода.

22. Резолюции в логике предикатов. Приведение к дизъюнктам произвольных формул логики предикатов. Пример выводимости.
23. Построение опровергающих деревьев в логике высказываний. Доказательство противоречивости множества дизъюнктов. Замкнутые деревья. Пример построения деревьев.
24. Понятие о логической программе. Структура логической программы. Факты, правила, целевые утверждения. Алгоритм доказательства цели.
25. Унификация. Подстановки. Композиция подстановок. Множество рассогласований.
25. Стратегии унификации. Пример дерева вывода стратегией "вглубь".
26. Управление процессом решения в логической программе. Бэктрекинг. Отсечение. Повторение.
27. Структура программы в Турбо-Прологе. Структуры данных в Турбо-Прологе. Ввод и вывод данных.
28. Декларативные и процедурные элементы Турбо-Пролога. Примеры применения в логической программе.
29. Организация ветвлений и циклов в программе. Рекурсии. Прямая и обратная рекурсии.
30. Обработка отношений в Турбо-Прологе. Примеры программирования композиций отношений.
31. Организация работы с файлами. Средства Турбо-Пролога для работы с файлами.
32. Динамические базы данных. Стандартные предикаты для работы с ними. Пример применения.
33. Графические средства Турбо Пролога. Примеры применения.
34. Модульное программирование. Организация меню. Пример.
35. Применение логических программ для решения технических задач.
36. Моделирование логических схем на Турбо Прологе. Пример.
37. Разработка многомодульных проектов на Турбо Прологе. Пример.
38. Общение с ЭВМ на языке, близком к естественному. Проблемы общения. Языки в системах общения.
39. Структура лингвистического транслятора для общения на языке, близком к естественному.

Вопросы по формированию и развитию теоретических знаний, предусмотренных компетенциями, выносимые на зачет с оценкой в 8 семестре

1. Средства формализации языков функционального программирования.
2. λ - исчисление как основа формализации языков функционального программирования. Средства формализации λ – исчисления. Аксиомы и правила вывода. Лямбда – выражения. Лямбда – термы. Аппликация и абстракция. Свободные и связанные переменные.
3. Комбинаторы. Подстановки. Конверсия. Равенство и редукция лямбда – термов.
4. Представление данных в лямбда – исчислении. Булевские значения. Пары и кортежи. Натуральные числа. Рекурсивные функции и множества. Именованные выражения. Типизация.
5. Комбинаторная логика как абстракция формализации языков функционального программирования и абстрактных вычислительных машин.
6. Концепция абстрактной машины. Категориальная абстрактная машина.
7. Характеристики языка функционального программирования Лисп. Структуры данных. Атомы. Списки. Выражения.
8. Функции. Композиция функций. S – выражения. Рекурсии S – выражений. Каррирование. Правила записи программ на Лиспе.
9. Парадигма параллельного программирования. Классы языков и моделей параллельного программирования.

10. Модели параллелизма известных языков программирования. Применение сетевых граф - схем для моделирования распараллеливания. Обмен данными.
11. Языки параллельного программирования.
12. Основные возможности концепции Microsoft.NET. Поддержка многоязыковой среды разработки приложений. Ее возможности.
13. Универсальный многоязыковый интерфейс. Интеграция с технологией Web – сервисов (2 час.).
14. Способы задания и нотации синтаксиса языков функционального программирования. Нотация Бэкуса – Наура (БНФ). Синтаксис комбинаторной логики.
15. Компиляторы и интерпретаторы. Концептуальные подходы к моделированию семантики языков программирования.

6.4 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Процедуры оценивания знаний, умений, навыков, характеризующих этапы формирования компетенций, изложены в:

- методические рекомендации по самостоятельной работе (Приложение Б1.В.ДВ.8.1 (СРС));
- методические указания к лабораторным работам (Приложение Б1.В.ДВ.8.1 (лр));
- конспект лекций (Приложение Б1.В.ДВ. 8.1 (лк)).

7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины «Основные парадигмы программирования».

Основная литература

1. Прыкина Е.Н. Основы логического программирования в среде Турбо Пролог: учебное пособие для вузов. Кемерово: КемГУКИ, 2006 г. – 68с.
<http://biblioclub.ru/index.php?page=book&id=227891&sr=1>, свободный доступ.
2. Шрайдер П.А. Основы программирования на языке Пролог. М.: Интернет - Университет Информационных технологий, 2005, -175 с.
<http://biblioclub.ru/index.php?page=book&id=233214&sr=1>, свободный доступ.
3. Воеводин В.В. , Воеводин Вл.В Параллельные вычисления.-СПб.: БХВ Петербург, 2002.- 608с.
<http://biblioclub.ru/index.php?page=book&id=229317&sr=1> , свободный доступ.

Дополнительная литература

1. Набебин А.А. Логика и Пролог в дискретной математике. Учебное пособие для вузов. М.: изд. МЭИ. 1994, - 251 с.
2. Братко И. Программирование на языке Пролог для искусственного интеллекта. М., Мир, 1990, -559 с.
3. Цуканова Н.И., Дмитриева Т.А. Логическое программирование на языке Visual Prolog. - М.: Горячая Линия - Телеком, 2008. - 144 С.
4. Городняя Л.В. Парадигмы программирования. – М.: Интернет- Университет Информационных технологий, 2007, -: <http://www.intuit.ru>.
5. XLISP Home Page – <http://xlisp.org>

6. Association of Lisp Users - <http://www.lisp.org>
7. LispWorks – <http://www.lispworks.com>

8. Перечень ресурсов информационно – телекоммуникационной сети «Интернет» необходимых для освоения дисциплины

1. <http://e.lanbook.com/> – электронно-библиотечная система «Лань».
2. <http://www.biblioclub.ru> – «университетская библиотека онлайн».
3. <http://www.elibrary.ru> – научная электронная библиотека.
4. <http://www.opac.mpei.ru> – электронная библиотека МЭИ.

9. Методические указания для обучающихся по освоению дисциплины

Успешное изучение курса требует посещения лекций, активной работы на практических занятиях, выполнения всех учебных заданий преподавателя, ознакомления с основной и дополнительной литературой.

Во время **лекции** студент должен вести краткий конспект.

Работа с конспектом лекций предполагает просмотр конспекта в тот же день после занятий. При этом необходимо пометить материалы конспекта, которые вызывают затруднения для понимания. При этом обучающийся должен стараться найти ответы на затруднительные вопросы, используя рекомендуемую литературу. Если ему самостоятельно не удалось разобраться в материале, необходимо сформулировать вопросы и обратиться за помощью к преподавателю на консультации или ближайшей лекции.

Обучающемуся необходимо регулярно отводить время для повторения пройденного материала, проверяя свои знания, умения и навыки по контрольным вопросам.

Лабораторные работы составляют важную часть профессиональной подготовки студентов. Основная цель их проведения - формирование у студентов аналитического, творческого мышления путем приобретения практических навыков.

Методические указания к лабораторным работам по дисциплине наряду с рабочей программой и графиком учебного процесса относятся к методическим документам, определяющим уровень организации и качества образовательного процесса.

Содержание лабораторных работ фиксируется в РПД настоящей программы.

Лабораторные работы выполняют следующие задачи:

стимулируют регулярное изучение рекомендуемой литературы, а также внимательное отношение к лекционному курсу;

закрепляют знания, полученные в процессе лекционного обучения и самостоятельной работы над литературой;

расширяют объем профессионально значимых знаний, умений, навыков;

позволяют проверить правильность ранее полученных знаний;

прививают навыки самостоятельного мышления, устного выступления;

способствуют свободному оперированию терминологией;

предоставляют преподавателю возможность систематически контролировать уровень самостоятельной работы студентов.

При подготовке к лабораторным работам необходимо просмотреть конспекты лекций и методические указания, рекомендованную литературу по данной теме; подготовиться к ответу на контрольные вопросы.

В ходе выполнения индивидуального задания лабораторных работ студент готовит отчет о работе (в программе *MS Word* или любом другом текстовом редакторе). В отчет заносятся результаты выполнения каждого пункта задания. Примерный образец оформления отчета имеется у преподавателя.

За 10 мин. до окончания занятия преподаватель проверяет объём выполненной на занятии работы и отмечает результат в рабочем журнале.

Оставшиеся невыполненными пункты задания лабораторных работ студент обязан доделать самостоятельно.

После проверки отчета преподаватель может проводить устный или письменный опрос студентов для контроля усвоения ими основных теоретических и практических знаний по теме занятия. По результатам проверки отчета и опроса выставляется отметка о выполнении лабораторной работы.

Самостоятельная работа студентов (СРС) по дисциплине играет важную роль в ходе всего учебного процесса. Методические материалы и рекомендации для обеспечения СРС готовятся преподавателем и являются неотъемлемой частью программы.

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем (при необходимости)

При проведении лекционных занятий предусматривается использование систем мультимедиа.

Лабораторные работы и практические занятия проводятся в компьютерном классе с использованием программных продуктов.

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

Для обеспечения данной дисциплины необходимо: лекционная аудитория и компьютерный класс, соответствующие действующим санитарным и противопожарным нормам, оборудованный вычислительными средствами (ПЭВМ).

Лекции по данной дисциплине проводятся в аудитории, оснащенной мультимедийной техникой (проектор, экран, компьютер/ноутбук).

Лабораторные работы и практические занятия по данной дисциплине проводятся в компьютерном классе.

Для проведения практических занятий в компьютерном классе необходимо устанавливать пакеты NASM, Delphi 7.

Для проведения лабораторных занятий в компьютерном классе используются пакеты Visual Prolog 6.0.

Автор
канд. техн. наук, доцент

Н.И. Сухачев

Зав. кафедрой ВТ
д-р техн. наук, профессор

А.С. Федулов

Программа одобрена на заседании кафедры 31 августа 2016 года, протокол № 01.

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Но- мер изме- не- ния	Номера страниц				Всего стра- ниц в доку- менте	Наименование и № документа, вводящего изменения	Подпись, Ф.И.О. внесшего измене- ния в данный эк- земпляр	Дата внесения из- менения в данный эк- земпляр	Дата введения изменения
	изме- не- ний	заме- не- ний	но- вых	анну- ли- ро- ван- ных					
1	2	3	4	5	6	7	8	9	10