

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«ТЮМЕНСКИЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ

Заведующий кафедрой ИСТ

_____ Данилов О. Ф.

« _____ » _____ 20__ г.

РАБОЧАЯ ПРОГРАММА

дисциплина: **Функционально-логическое программирование**

направление подготовки: **09.03.04 Программная инженерия**

направленность (профиль): **Разработка программно-информационных систем**

форма обучения: **очная**

Рабочая программа рассмотрена на заседании кафедры интеллектуальных систем и технологий для направления 09.03.04 Программная инженерия направленность (профиль) «Разработка программно-информационных систем»

1. Цели и задачи дисциплины

Целью изучения дисциплины является освоение фундаментальных принципов логического и функционального программирования на примере языков Prolog и Haskell.

Задачи дисциплины заключаются в:

- изучении символьных вычислений, задач на графах, функциональных структур данных Окасаки;
- приобретении навыков ввода вывода с использованием монад, работы с сетевыми протоколами;
- приобретении навыков многопоточного программирования;
- изучении арифметических выражений с использованием монад;
- реализации сбалансированных древовидных структур данных на функциональных языках;
- приобретении навыков ввода вывода в Haskell;
- реализации алгоритмов сортировки на функциональных языках.

2. Место дисциплины в структуре ОПОП

Дисциплина относится к дисциплинам части учебного плана, формируемой участниками образовательных отношений.

Необходимыми условиями для освоения дисциплины являются:

- знание основы программирования, алгоритмы и структуры данных, основы дискретной математики, существующие технологии программирования.
- умения разрабатывать алгоритмы и реализовывать их в виде программы, решать задачи из различных разделов дискретной математики и математической логики, строить модели объектов и понятий.
- владение математическим аппаратом дискретной математики и математической логики, навыком разработки алгоритмов и реализации их в виде программы.

Содержание дисциплины является логическим продолжением содержания дисциплин «Дискретная математика», «Программирование», «Теоретическая и прикладная информатика», «Технологии программирования» и помогает в подготовке выпускной квалификационной работы.

3. Результаты обучения по дисциплине

Процесс изучения дисциплины направлен на формирование следующих компетенций:

Таблица 3.1

Код и наименование компетенции	Код и наименование индикатора достижения компетенции (ИДК)	Код и наименование результата обучения по дисциплине
ПКС-1 – Способность проводить анализ требований к программному обеспечению, выполнять работы по проектированию программного обеспечения	ПКС-1.2. Разрабатывает технические спецификации на программные компоненты и их взаимодействие.	Знать (З1) основные элементы языков Prolog и Haskell, принципы функционального и логического программирования
		Уметь (У1) использовать терминологию, основные элементы и принципы функционального и логического программирования при разработке технических спецификаций на программные компоненты и их взаимодействие
		Владеть (В1) навыком разработки технических спецификаций на

		программные компоненты и их взаимодействие, реализованные на функционально-логических языках программирования.
	ПКС-1.3. Проектирует программное обеспечение.	Знать (З2) основные методы и средства эффективной разработки программного продукта Уметь (У2) осуществлять разработку программного обеспечения на языках Prolog и Haskell Владеть (В2) основными методологиями процессов разработки программного обеспечения
ПКС 4 – Способность осуществлять оценку и выбор варианта архитектуры программного средства, а также осуществлять контроль его реализации	ПКС-4.1. Оценивает и выбирает шаблоны проектирования для каждого слоя или компонента программного средства.	Знать (З3) шаблоны проектирования слоев и компонентов программного продукта на языках Prolog и Haskell
		Уметь (У3) применять шаблоны проектирования слоев и компонентов программного продукта на языках Prolog и Haskell
		Владеть (В3) методологией проектирования программного обеспечения с помощью шаблонов на языках Prolog и Haskell
	ПКС-4.3. Осуществляет координацию процессов создания и сборки программного средства из компонентов.	Знать (З4) принципы сборки программных средств из компонентов на языках Prolog и Haskell Уметь (У4) реализовывать процесс разработки и сборки программного средства из компонент с помощью языков Prolog и Haskell Владеть (В4) навыками координации процессов создания и сборки программных средств из компонентов на языках Prolog и Haskell
ПКС 5 – Способность выполнять работы по разработке и интеграции программных модулей и компонент системного, инструментального и пользовательского программного обеспечения	ПКС-5.2. Разрабатывает средства, модули и компоненты ПО и осуществляет их интеграцию.	Знать (З5) языки функционального и логического программирования
		Уметь (У5) разрабатывать модели различных классов систем с применением языков функционального и логического программирования
		Владеть (В5) математическим аппаратом, применяемым в функциональном и логическом программировании

4. Объем дисциплины

Общий объем дисциплины составляет 5 зачетных единиц, 180 часов.

Таблица 4.1.

Форма обучения	Курс/ семестр	Аудиторные занятия/контактная работа, час.			Самостоятельная работа, час.	Контроль, час	Форма промежуточной аттестации
		Лекции	Практические занятия	Лабораторные занятия			
очная	4/7	28	-	28	88	36	экзамен, КР

5. Структура и содержание дисциплины

5.1. Структура дисциплины: очная форма обучения (ОФО)

Таблица 5.1.1

№ п/п	Структура дисциплины		Аудиторные занятия, час.			СРС, час.	Всего, час.	Код ИДК	Оценочные средства
	Номер раздела	Наименование раздела	Л.	Пр.	Лаб.				
1	1	Введение	4	-	-	8	12	ПКС-1.1, ПКС-1.3, ПКС-4.1, ПКС-4.3, ПКС-5.2	Устный опрос
2	2	Основы логической парадигмы	5	-	5	8	18	ПКС-1.1, ПКС-1.3, ПКС-4.1, ПКС-4.3, ПКС-5.2	Защита лабораторных работ
3	3	Язык программирования Prolog.	5	-	6	9	20	ПКС-1.1, ПКС-1.3, ПКС-4.1, ПКС-4.3, ПКС-5.2	Защита лабораторных работ
4	4	Экспертные системы	4	-	5	9	18	ПКС-1.1, ПКС-1.3, ПКС-4.1, ПКС-4.3, ПКС-5.2	Защита лабораторных работ
5	5	Основы функциональной парадигмы	5	-	6	9	20	ПКС-1.1, ПКС-1.3, ПКС-4.1, ПКС-4.3, ПКС-5.2	Защита лабораторных работ
6	6	Язык программирования Haskell	5	-	6	9	20	ПКС-1.1, ПКС-1.3, ПКС-4.1, ПКС-4.3, ПКС-5.2	Защита лабораторных работ Тестирование
10	экзамен		-	-	-	36	36	ПКС-1.1, ПКС-1.3, ПКС-4.1, ПКС-4.3, ПКС-5.2	Вопросы к экзамену
11	курсовая работа		-	-	-	36	36	ПКС-1.1, ПКС-1.3, ПКС-4.1, ПКС-4.3, ПКС-5.2	Защита курсовой работы
Итого:			28	-	28	124	180	-	-

5.2. Содержание дисциплины.

5.2.1. Содержание разделов дисциплины (дидактические единицы).

Раздел 1. "Введение". Предмет дисциплины, ее объем, содержание и связь с другими дисциплинами учебного плана. Роль дисциплины в подготовке специалистов, цели и задачи дисциплины. Обзор литературы по курсу. Сравнительный анализ императивной и декларативной парадигм программирования. Ретроспектива и перспектива развития функционального и логического программирования.

Раздел 2. "Основы логической парадигмы". Исчисление предикатов как язык представления знаний. Виды импликации. Логика предикатов первого порядка, хорновские дизъюнкты. Вычислительная модель. Анализ структуры термов. Принцип Робинсона.

Раздел 3. "Язык программирования Prolog". Основные элементы языка. Алфавит языка. Термы. Виды термов: константы, переменные, структуры. Литеры и их типы. Интерпретация литер. Операторы. Свойства операторов (позиция, приоритет, ассоциативность). Инфиксные,

префиксные, постфиксные операторы. Факты. Правила. Запись фактов и правил. Предикаты. Цели, конъюнкция целей. Общая схема доказательства целевого утверждения. Арифметика в языке Prolog. Встроенные предикаты для сравнения чисел. Вычисление арифметических выражений. Предикат is и его варианты в различных версиях. Примеры программ с выполнением арифметических операций. Согласование целевых утверждений. Доказательство целевых утверждений при использовании механизма возврата. Правила установления соответствия. Недетерминизм первого и второго рода. Понятие свободной и связанной переменной. Примеры программ с использованием механизма возврата. Операционная модель вычисления Prolog-программ. Рекурсивное представление данных и программ. Рекурсивные функции. Структуры и деревья. Список как частный вид структуры. Формы записи списков. Работа со списками. Граничные условия и способы использования рекурсии. Примеры программ с рекурсивными определениями. Сортировка списков. Отсечение и способы его использования в языке. Причины использования отсечения. Предикат cut. Диаграмма работы программы с использованием отсечения. Общие случаи использования отсечения. Проблемы, связанные с использованием отсечения. Развитие отсечения в отсечение-отрезок в некоторых вариантах. Ввод и вывод. Примеры программ с использованием ввода и вывода. Встроенные предикаты ввода-вывода.

Раздел 4. "Экспертные системы". Динамические предикаты. Добавление и исключение утверждений, классификация термов, изменение и анализ утверждений, работа со структурами произвольного вида, воздействие на процесс возврата, реализация сложных способов выражения целевых утверждений, объявление операторов, обработка файлов, наблюдение за выполнением программы на Prolog'e. Примеры использования встроенных предикатов. Декларативная и операционная семантика. Интерпретация. Программирование второго порядка. Множественные выражения. Применение Prolog'a для построения баз знаний и экспертных систем. Встроенные средства для анализа текстов на естественных языках.

Раздел 5. "Основы функциональной парадигмы". Исторические предпосылки функционального программирования. Рекурсивные функции и лямбда-исчисление А. Черча. Комбинаторная логика Х. Карри. Программирование в функциональных обозначениях Л. В. Канторовича. Язык LISP и работы Дж. Маккарти. Другие функциональные языки. Применение функционального программирования.

Раздел 6. "Язык программирования Haskell". Строго функциональный язык. Символьные данные: лямбда-выражения и представление данных. Элементарные селекторы и конструкторы лямбда-выражений. Элементарные предикаты и арифметика. Рекурсивные функции: разбор случаев, рекурсивные определения, выбор подфункций. Примеры обработки списков. Дополнительные возможности. Приемы программирования. Аккумуляторы. Локальные определения. Функционалы и другие функции высших порядков. Примеры простых функциональных программ. Значения и типы. Полиморфные типы. Типы, определяемые пользователем. Бинарные конструкторы данных. Рекурсивные типы. Синонимы типов. Встроенные типы. Генераторы списков и арифметические последовательности. Строки. Функции. Инфиксные операции. Секции. Декларация пользовательских операций. Бесконечные структуры данных и ленивые вычисления. Сопоставление с образцом. As-образцы. Универсальные образцы. Семантика сопоставления с образцом. Выражение case. Ленивые образцы. Лексическая видимость и вложенные формы. Let-выражения. Предложение where. Двумерный синтаксис. Классы типов и перегрузка. Наследование. Множественное наследование. Типы высшего порядка. Контроль корректности типов. Описание newtype. Метки полей. Строгие конструкторы данных. Понятие монады. Система ввода-вывода. Базисные операции ввода-вывода. Сетевое программирование.

5.2.2. Содержание дисциплины по видам учебных занятий.

Лекционные занятия

Таблица 5.2.1

№ п/п	Номер раздела дисциплины	Объем, час.			Тема лекции
		ОФО	ЗФО	ОЗФО	

1	1	4	-	-	Введение
2	2	5	-	-	Основы логической парадигмы
3	3	5	-	-	Язык программирования Prolog.
4	4	4	-	-	Экспертные системы
5	5	5	-	-	Основы функциональной парадигмы
6	6	5	-	-	Язык программирования Haskell
Итого:		28	-	-	-

Практические занятия

Практические занятия учебным планом не предусмотрены

Лабораторные работы

Таблица 5.2.2

№ п/п	Номер раздела дисциплины	Объем, час.			Тема лабораторной работы
		ОФО	ЗФО	ОЗФО	
1	2	5	-	-	Основы логической парадигмы
2	3	6	-	-	Язык программирования Prolog.
3	4	5	-	-	Экспертные системы
4	5	6	-	-	Основы функциональной парадигмы
5	6	6	-	-	Язык программирования Haskell
Итого:		28	-	-	-

Самостоятельная работа студента

Таблица 5.2.4

№ п/п	Номер раздела дисциплины	Объем, час.			Тема	Вид СРС
		ОФО	ЗФО	ОФО		
1	1	8	-	-	Введение	Подготовка к устному опросу
2	2	8	-	-	Основы логической парадигмы	Подготовка к защите лабораторной работы
3	3	9	-	-	Язык программирования Prolog.	Подготовка к защите лабораторной работы
4	4	9	-	-	Экспертные системы	Подготовка к защите лабораторной работы
5	5	9	-	-	Основы функциональной парадигмы	Подготовка к защите лабораторной работы
6	6	9	-	-	Язык программирования Haskell	Подготовка к защите лабораторной работы
7	1-6	36	-	-	-	Подготовка к тестированию
8	1-6	36	-	-	-	Подготовка к защите курсовой работы
Итого:		100	-	-	-	-

5.2.3. Преподавание дисциплины ведется с применением следующих видов образовательных технологий: исследовательские методы обучения (лекции), работа в малых группах (лабораторные занятия).

6. Тематика курсовых работ

Тематика курсовой работы определяется преподавателем, осуществляющим руководство курсовой работой. Обучающийся выбирает тему проекта в соответствии со своими интересами и сообщает об этом преподавателю.

Примерные темы курсовых работ.

1. Разработка справочно-информационной системы «Справочная система авиакомпании».
2. Разработка справочно-информационной системы «Справочная система железнодорожной компании».
3. Разработка информационной системы анализа успеваемости студентов.
4. Разработка программы по обработке естественного языка.
5. Компьютерная графика в программах на языке Prolog.
6. Разработка прототипа экспертной системы по классификации объектов некоторой предметной области.
7. Разработка прототипа экспертной системы по ремонту/настройке устройств некоторой предметной области.
8. Пролог-реализация алгоритмов сортировки данных
9. Пролог-реализация психологических тестов
10. Пролог-реализация поиска кратчайшего маршрута
11. Пролог-реализация поиска пути в лабиринте.

7. Контрольные работы

Контрольные работы учебным планом не предусмотрены

8. Оценка результатов освоения учебной деятельности

8.1. Критерии оценивания степени полноты и качества освоения компетенций в соответствии с планируемыми результатами обучения приведены в Приложении 1.

8.2. Рейтинговая система оценивания степени полноты и качества освоения компетенций обучающихся очной формы обучения представлена в таблице 8.1 и таблице 8.2.

Таблица 8.1

№ п/п	Виды мероприятий в рамках текущего контроля	Количество баллов
1 текущая аттестация		
1	Устный опрос	15
2	Защита лабораторной работы №1, №2	30
ИТОГО за первую текущую аттестацию		45
2 текущая аттестация		
1	Защита лабораторных работ №3, №4 и №5	45
2	Тестирование	10
ИТОГО за вторую текущую аттестацию		55
ВСЕГО		100

Таблица 8.2

№ п/п	Виды мероприятий в рамках текущего контроля	Количество баллов
1 текущая аттестация		
1	Подготовка программного кода курсовой работы	50
ИТОГО за первую текущую аттестацию		50
1	Подготовка пояснительной записки КР	30
2	Защита работы. Ответы на вопросы	20
ИТОГО за вторую текущую аттестацию		50
ВСЕГО		100

9. Учебно-методическое и информационное обеспечение дисциплины

9.1 Перечень рекомендуемой литературы представлен в Приложении 2.

9.2 Современные профессиональные базы данных и информационные справочные системы:

- Электронный каталог/Электронная библиотека ТИУ <http://webirbis.tsogu.ru/>;
- Цифровой образовательный ресурс – библиотечная система IPR SMART — <https://www.iprbookshop.ru/>;
- Электронно-библиотечная система «Консультант студента» www.studentlibrary.ru;
- Электронно-библиотечная система «ЛАНЬ» https://e.lanbook.com;
- Образовательная платформа ЮРАЙТ www.urait.ru;
- Научная электронная библиотека ELIBRARY.RU http://www.elibrary.ru;
- Библиотеки нефтяных вузов России:
 - Электронная нефтегазовая библиотека РГУ нефти и газа им. Губкина <http://elib.gubkin.ru/>;
 - Электронная библиотека Уфимского государственного нефтяного технического университета <http://bibl.rusoil.net/>;
 - Библиотечно-информационный комплекс Ухтинского государственного технического университета УГТУ <http://lib.ugtu.net/books>;
- Электронная справочная система нормативно-технической документации «Технорматив»;
- ЭКБСОН – информационная система доступа к электронным каталогам библиотек сферы образования и науки.

9.3 Лицензионное и свободно распространяемое программное обеспечение, в т.ч. отечественного производства.

- Microsoft Windows – операционная система.
- Microsoft Office Professional Plus – набор офисных приложений.
- GNU Prolog (Свободно-распространяемое ПО) – язык и система логического программирования.
- Visual Studio Community (Свободно-распространяемое ПО) – интегрированная среда разработки.

10. Материально-техническое обеспечение дисциплины

Помещения для проведения всех видов работы, предусмотренных учебным планом, укомплектованы необходимым оборудованием и техническими средствами обучения.

Таблица 10.1

Обеспеченность материально-технических условий реализации ОПОП ВО

№ п/п	Наименование учебных предметов, курсов, дисциплин, практики, иных видов учебной деятельности, предусмотренных учебным планом образовательной программы	Наименование помещений для проведения всех видов учебной деятельности, предусмотренной учебным планом, в том числе помещения для самостоятельной работы, с указанием перечня основного оборудования, учебно-наглядных пособий	Адрес (местоположение) помещений для проведения всех видов учебной деятельности, предусмотренной учебным планом (в случае реализации образовательной программы в сетевой форме дополнительно указывается наименование организации, с которой заключен договор)
1	2	3	4
1	Функционально-логическое программирование	Учебная аудитория для проведения занятий лекционного типа; групповых и индивидуальных консультаций; текущего контроля и промежуточной аттестации. Основное оборудование: столы – 46 шт., стулья – 92 шт, доска аудиторная – 1 шт., моноблок – 1 шт., проектор – 1 шт.,	625001, г. Тюмень, ул. Луначарского, д. 2, корп. 1

		проекционный экран – 1 шт., акустическая система (колонки) -2 шт.	
		Учебная аудитория для проведения занятий лекционного типа; групповых и индивидуальных консультаций; текущего контроля и промежуточной аттестации. Основное оборудование: столы – 46 шт., стулья – 92 шт, доска аудиторная – 1 шт., моноблок – 1 шт., проектор – 1 шт., проекционный экран – 1 шт., акустическая система (комплект) -1 шт.	625001, г. Тюмень, ул. Луначарского, д. 4
		Учебная аудитория для проведения занятий лекционного типа; групповых и индивидуальных консультаций; текущего контроля и промежуточной аттестации, , курсового проектирования. Основное оборудование: столы – 52 шт., стулья – 52 шт, доска аудиторная – 1 шт., моноблок – 1 шт., проектор – 1 шт., проекционный экран – 1 шт., акустическая система (колонки) -2 шт., микрофон - 1 шт.	625001, Тюменская область, г. Тюмень, ул. Луначарского, д.4
		Помещение для самостоятельной работы обучающихся с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Учебная мебель: столы – 10 шт., стулья – 15 шт., доска аудиторная – 1 шт., моноблок – 5 шт.,	625001, Тюменская область, г. Тюмень, ул. Луначарского, д.2, корп.1
		Помещение для самостоятельной работы обучающихся с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Учебная мебель: столы – 15 шт., стулья – 25 шт., доска аудиторная – 1 шт., моноблок – 5 шт., проектор - 1 шт., экран - 1 шт.,	625001, Тюменская область, г. Тюмень, ул. Луначарского, д.2, корп.1

11. Методические указания по организации СРС

Самостоятельная работа является одной из важнейших форм изучения любой дисциплины. Она позволяет систематизировать и углубить теоретические знания, закрепить умения и навыки, способствует развитию умений пользоваться научной и учебно-методической литературой. Познавательная деятельность в процессе самостоятельной работы требует от студента высокого уровня активности и самоорганизованности.

В учебном процессе выделяют два вида самостоятельной работы: аудиторная и внеаудиторная.

Аудиторная самостоятельная работа по дисциплине выполняется на учебных занятиях под непосредственным руководством преподавателя и по его заданию.

Внеаудиторная самостоятельная работа студентов представляет собой логическое продолжение аудиторных занятий. Затраты времени на выполнение этой работы регламентируются рабочим учебным планом. Режим работы выбирает сам обучающийся в зависимости от своих способностей и конкретных условий.

Самостоятельная работа может осуществляться индивидуально или группами студентов в зависимости от цели, объема, конкретной тематики самостоятельной работы, уровня сложности, уровня умений студентов.

Самостоятельная работа включает в себя работу с конспектом лекций, изучение и конспектирование рекомендуемой литературы, изучение мультимедиапрезентаций, расположенных в

свободном доступе, решение ситуационных (профессиональных) задач, проектирование и моделирование разных видов и компонентов профессиональной деятельности, научно-исследовательскую работу и др.

Контроль результатов внеаудиторной самостоятельной работы студентов может осуществляться в пределах времени, отведенного на обязательные учебные занятия по дисциплине и внеаудиторную самостоятельную работу студентов по дисциплине, может проходить в письменной, устной или смешанной форме.

Планируемые результаты обучения для формирования компетенции и критерии их оценивания

Дисциплина: **Функционально-логическое программирование**

Код, направление подготовки: **09.03.04 Программная инженерия**

Направленность (профиль): **Разработка программно-информационных систем**

Код компетенции	Код и наименование результата обучения по дисциплине	Критерии оценивания результатов обучения			
		1-2	3	4	5
ПКС-1	Знать (З1) основные элементы языков Prolog и Haskell, принципы функционального и логического программирования	Неудовлетворительно знает основные элементы языков Prolog и Haskell, принципы функционального и логического программирования	Удовлетворительно знает способы основные элементы языков Prolog и Haskell, принципы функционального и логического программирования	Хорошо знает основные элементы языков Prolog и Haskell, принципы функционального и логического программирования	Отлично знает основные элементы языков Prolog и Haskell, принципы функционального и логического программирования
	Уметь (У1) использовать терминологию, основные элементы и принципы функционального и логического программирования при разработке технических спецификацию на программные компоненты и их взаимодействие	Неудовлетворительно умеет использовать терминологию, основные элементы и принципы функционального и логического программирования при разработке технических спецификацию на программные компоненты и их взаимодействие	Удовлетворительно умеет использовать терминологию, основные элементы и принципы функционального и логического программирования при разработке технических спецификацию на программные компоненты и их взаимодействие	Хорошо умеет использовать терминологию, основные элементы и принципы функционального и логического программирования при разработке технических спецификацию на программные компоненты и их взаимодействие	Отлично умеет использовать терминологию, основные элементы и принципы функционального и логического программирования при разработке технических спецификацию на программные компоненты и их взаимодействие

Код компетенции	Код и наименование результата обучения по дисциплине	Критерии оценивания результатов обучения			
		1-2	3	4	5
	Владеть (В1) навыком разработки технических спецификаций на программные компоненты и их взаимодействие, реализованные на функционально-логических языках программирования.	Неудовлетворительно владеет навыком разработки технических спецификаций на программные компоненты и их взаимодействие, реализованные на функционально-логических языках программирования.	Удовлетворительно владеет навыком разработки технических спецификаций на программные компоненты и их взаимодействие, реализованные на функционально-логических языках программирования.	Хорошо владеет навыком разработки технических спецификаций на программные компоненты и их взаимодействие, реализованные на функционально-логических языках программирования.	Отлично владеет навыком разработки технических спецификаций на программные компоненты и их взаимодействие, реализованные на функционально-логических языках программирования.
	Знать (З2) основные методы и средства эффективной разработки программного продукта	Неудовлетворительно знает основные методы и средства эффективной разработки программного продукта	Удовлетворительно знает основные методы и средства эффективной разработки программного продукта	Хорошо знает основные методы и средства эффективной разработки программного продукта	Отлично знает основные методы и средства эффективной разработки программного продукта
	Уметь (У2) осуществлять разработку программного обеспечения на языках Prolog и Haskell	Неудовлетворительно умеет осуществлять разработку программного обеспечения на языках Prolog и Haskell	Удовлетворительно умеет осуществлять разработку программного обеспечения на языках Prolog и Haskell	Хорошо умеет осуществлять разработку программного обеспечения на языках Prolog и Haskell	Отлично умеет осуществлять разработку программного обеспечения на языках Prolog и Haskell
	Владеть (В2) основными методологиями процессов разработки программного обеспечения	Неудовлетворительно владеет основными методологиями процессов разработки программного обеспечения	Удовлетворительно владеет основными методологиями процессов разработки программного обеспечения	Хорошо владеет основными методологиями процессов разработки программного обеспечения	Отлично владеет основными методологиями процессов разработки программного обеспечения
ПКС - 4	Знать (З3) шаблоны проектирования слоев и компонентов программного продукта на языках Prolog и Haskell	Неудовлетворительно знает шаблоны проектирования слоев и компонентов программного продукта на языках Prolog и Haskell	Удовлетворительно знает шаблоны проектирования слоев и компонентов программного продукта на языках Prolog и Haskell	Хорошо знает шаблоны проектирования слоев и компонентов программного продукта на языках Prolog и Haskell	Отлично знает шаблоны проектирования слоев и компонентов программного продукта на языках Prolog и Haskell

Код компетенции	Код и наименование результата обучения по дисциплине	Критерии оценивания результатов обучения			
		1-2	3	4	5
	Уметь (У3) применять шаблоны проектирования слоев и компонентов программного продукта на языках Prolog и Haskell	Неудовлетворительно умеет применять шаблоны проектирования слоев и компонентов программного продукта на языках Prolog и Haskell	Удовлетворительно умеет применять шаблоны проектирования слоев и компонентов программного продукта на языках Prolog и Haskell	Хорошо умеет применять шаблоны проектирования слоев и компонентов программного продукта на языках Prolog и Haskell	Отлично умеет применять шаблоны проектирования слоев и компонентов программного продукта на языках Prolog и Haskell
	Владеть (В3) методологией проектирования программного обеспечения с помощью шаблонов на языках Prolog и Haskell	Неудовлетворительно владеет методологией проектирования программного обеспечения с помощью шаблонов на языках Prolog и Haskell	Удовлетворительно владеет методологией проектирования программного обеспечения с помощью шаблонов на языках Prolog и Haskell	Хорошо владеет методологией проектирования программного обеспечения с помощью шаблонов на языках Prolog и Haskell	Отлично владеет методологией проектирования программного обеспечения с помощью шаблонов на языках Prolog и Haskell
	Знать (З4) принципы сборки программных средств из компонентов на языках Prolog и Haskell	Неудовлетворительно знает принципы сборки программных средств из компонентов на языках Prolog и Haskell	Удовлетворительно знает принципы сборки программных средств из компонентов на языках Prolog и Haskell	Хорошо знает принципы сборки программных средств из компонентов на языках Prolog и Haskell	Отлично знает принципы сборки программных средств из компонентов на языках Prolog и Haskell
	Уметь (У4) реализовывать процесс разработки и сборки программного средства из компонент с помощью языков Prolog и Haskell	Неудовлетворительно умеет реализовывать процесс разработки и сборки программного средства из компонент с помощью языков Prolog и Haskell	Удовлетворительно умеет реализовывать процесс разработки и сборки программного средства из компонент с помощью языков Prolog и Haskell	Хорошо умеет реализовывать процесс разработки и сборки программного средства из компонент с помощью языков Prolog и Haskell	Отлично умеет реализовывать процесс разработки и сборки программного средства из компонент с помощью языков Prolog и Haskell
	Владеть (В4) навыками координации процессов создания и сборки программных средств из компонентов на языках Prolog и Haskell	Неудовлетворительно владеет навыками координации процессов создания и сборки программных средств из компонентов на языках Prolog и Haskell	Удовлетворительно владеет навыками координации процессов создания и сборки программных средств из компонентов на языках Prolog и Haskell	Хорошо владеет навыками координации процессов создания и сборки программных средств из компонентов на языках Prolog и Haskell	Отлично владеет навыками координации процессов создания и сборки программных средств из компонентов на языках Prolog и Haskell
ПКС - 5	Знать (З5) языки функционального и логического программирования	Неудовлетворительно знает языки функционального и логического программирования	Удовлетворительно знает языки функционального и логического программирования	Хорошо знает языки функционального и логического программирования	Отлично знает языки функционального и логического программирования

Код компетенции	Код и наименование результата обучения по дисциплине	Критерии оценивания результатов обучения			
		1-2	3	4	5
	Уметь (У5) разрабатывать модели различных классов систем с применением языков функционального и логического программирования	Неудовлетворительно умеет разрабатывать модели различных классов систем с применением языков функционального и логического программирования	Удовлетворительно умеет разрабатывать модели различных классов систем с применением языков функционального и логического программирования	Хорошо умеет разрабатывать модели различных классов систем с применением языков функционального и логического программирования	Отлично умеет разрабатывать модели различных классов систем с применением языков функционального и логического программирования
	Владеть (В5) математическим аппаратом, применяемым в функциональном и логическом программировании	Неудовлетворительно владеет математическим аппаратом, применяемым в функциональном и логическом программировании	Удовлетворительно владеет математическим аппаратом, применяемым в функциональном и логическом программировании	Хорошо владеет математическим аппаратом, применяемым в функциональном и логическом программировании	Отлично владеет математическим аппаратом, применяемым в функциональном и логическом программировании

КАРТА

обеспеченности дисциплины учебной и учебно-методической литературой

Дисциплина: **Функционально-логическое программирование**Код, направление подготовки: **09.03.04 Программная инженерия**Направленность (профиль): **Разработка программно-информационных систем**

№ п/п	Название учебного, учебно-методического издания, автор, издательство, вид издания, год издания	Количество экземпляров в БИК	Контингент обучающихся, использующих указанную литературу	Обеспеченность обучающихся литературой, %	Наличие электронного варианта в ЭБС (+/-)
1	Конева, С. И. Функциональное программирование: учебное пособие. Ч. 1 / С. И. Конева. – Ростов-на-Дону: Северо-Кавказский филиал Московского технического университета связи и информатики, 2018. – 53 с. - ISBN 2227-8397 – Текст: электронный // ЭБС IRP BOOKS [сайт]. - URL: http://www.iprbookshop.ru/89511.html .	ЭР*	30	БИК	ЭБС «IRP BOOKS»
2	Кубенский, А. А. Функциональное программирование: учебник и практику для вузов / А. А. Кубенский. – Москва: Юрайт, 2023. – 348 с. – (Высшее образование). - ISBN 978-5-9916-9242-7 – Текст: электронный // ЭБС "Юрайт" [сайт]. - URL: https://urait.ru/bcode/511994	ЭР*	30	БИК	ЭБС «Юрайт»
3	Товбис, Е. М. Логическое программирование в SWI-prolog: учебное пособие / Е. М. Товбис, С. П. Якимов. – Красноярск: СибГУ им. академика М.Ф. Решетнёва, 2022. – 86 с. – Текст: электронный // ЭБС «Лань» [сайт]. - URL: https://e.lanbook.com/book/330134	ЭР*	30	БИК	ЭБС «Лань»

ЭР* – электронный ресурс для автор. пользователей доступен через Электронный каталог/Электронную библиотеку ТИУ <http://webirbis.tsogu.ru/>