МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ»
(ТУСУР)

УТВЕРЖДАЮ
Директор департамента образования
_____ П. Е. Троян
«___» _____ 20__ г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Функциональное и логическое программирование

Уровень образования: **высшее образование - бакалавриат**
Направление подготовки / специальность: **09.03.01 Информатика и вычислительная техника**
Направленность (профиль) / специализация: **Программное обеспечение средств вычислительной техники и автоматизированных систем**
Форма обучения: **заочная (в том числе с применением дистанционных образовательных технологий)**
Факультет: **ФДО, Факультет дистанционного обучения**
Кафедра: **АСУ, Кафедра автоматизированных систем управления**
Курс: **4**
Семестр: **7**
Учебный план набора 2018 года

Распределение рабочего времени

№	Виды учебной деятельности	7 семестр	Всего	Единицы
1	Самостоятельная работа под руководством преподавателя	16	16	часов
2	Лабораторные работы	8	8	часов
3	Контроль самостоятельной работы	4	4	часов
4	Всего контактной работы	28	28	часов
5	Самостоятельная работа	143	143	часов
6	Всего (без экзамена)	171	171	часов
7	Подготовка и сдача экзамена	9	9	часов
8	Общая трудоемкость	180	180	часов
			5.0	З.Е.

Контрольные работы: 7 семестр - 2
Экзамен: 7 семестр

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Шелупанов А.А.
Должность: Ректор
Дата подписания: 28.02.2018
Уникальный программный ключ:
c53e145e-8b20-45aa-9347-a5e4dbb90e8d

Томск 2018

Согласована на портале № 42593

Рассмотрена и одобрена на заседании кафедры
протокол № 6 от «17» 5 2018 г.

ЛИСТ СОГЛАСОВАНИЯ

Рабочая программа дисциплины составлена с учетом требований федерального государственного образовательного стандарта высшего образования (ФГОС ВО) по направлению подготовки (специальности) 09.03.01 Информатика и вычислительная техника, утвержденного 12.01.2016 года, рассмотрена и одобрена на заседании кафедры АСУ «___» _____ 20__ года, протокол №_____.

Разработчики:

доцент каф. ТЭО	_____	Ю. В. Морозова
ст. преподаватель каф. АСУ	_____	А. В. Ковшов

Заведующий обеспечивающей каф. АСУ	_____	А. М. Кориков

Рабочая программа дисциплины согласована с факультетом и выпускающей кафедрой:

Декан ФДО	_____	И. П. Черкашина
Заведующий выпускающей каф. АСУ	_____	А. М. Кориков

Эксперты:

Доцент кафедры технологий электронного обучения (ТЭО)	_____	Ю. В. Морозова
Доцент кафедры автоматизированных систем управления (АСУ)	_____	А. И. Исакова

Согласована на портале № 42593

1. Цели и задачи дисциплины

1.1. Цели дисциплины

Цель дисциплины - формирование у студентов профессиональных знаний и практических навыков по разработке и созданию моделей и систем с помощью языков функционального и логического программирования. Функциональный и логический стили программирования являются альтернативными способами решения задач по отношению к традиционным стилям программирования - процедурному и объектно-ориентированному. Его отличительными чертами являются “математичность” и “человечность”. В функциональном программировании единственным действием является вызов функции, в нем не существует прямого обращения к ячейкам памяти, операторов присваивания, циклов, блок-схем и передачи управления. Логическое программирование - это переход на еще более высокий уровень программирования. Применение его позволяет автоматически строить доказательства теорем и решать задачи искусственного интеллекта.

1.2. Задачи дисциплины

– Задачей дисциплины является изучение общих концепций и методов современного декларативного программирования и, в частности, таких его разновидностей, как функциональное и логическое программирование, позволяющих эффективно решать задачи, связанные с обработкой символьной информации, нетипизированных данных, построения систем поддержки принятия решения, искусственного интеллекта, а также экспертных систем

2. Место дисциплины в структуре ОПОП

Дисциплина «Функциональное и логическое программирование» (Б1.В.ОД.9) относится к блоку 1 (вариативная часть).

Предшествующими дисциплинами, формирующими начальные знания, являются: Дискретная математика, Математическая логика и теория алгоритмов, Программирование.

Последующими дисциплинами являются: Экспертные системы.

3. Требования к результатам освоения дисциплины

Процесс изучения дисциплины направлен на формирование следующих компетенций:

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

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

В результате изучения дисциплины обучающийся должен:

– **знать** области практического применения функциональных и логических языков; - особенности логического и функционального программирования; - языки функционального и логического программирования; - основные методы и средства эффективной разработки программного продукта; - типовые роли в процессе разработки программного обеспечения; - методологии разработки программного обеспечения; - математические основы лямбда-исчисления, предикатов первого порядка.

– **уметь** - использовать методы и технологии разработки для генерации исполняемого кода; - анализировать поставленные задачи, разрабатывать алгоритмы, представлять знания для решения поставленных задач; - разрабатывать модели различных классов систем с применением языков функционального и логического программирования; - программировать на языках Лисп и Пролог.

– **владеть** основными методологиями процессов разработки программного обеспечения; - математическим аппаратом, применяемым в функциональном и логическом программировании; - языками Лисп и Пролог для построения моделей искусственного интеллекта.

4. Объем дисциплины и виды учебной работы

Общая трудоемкость дисциплины составляет 5.0 зачетных единицы и представлена в таблице 4.1.

Таблица 4.1 – Трудоемкость дисциплины

Виды учебной деятельности	Всего часов	Семестры

Согласована на портале № 42593

		7 семестр
Контактная работа (всего)	28	28
Самостоятельная работа под руководством преподавателя (СРП)	16	16
Лабораторные работы	8	8
Контроль самостоятельной работы (КСР)	4	4
Самостоятельная работа (всего)	143	143
Подготовка к контрольным работам	16	16
Оформление отчетов по лабораторным работам	8	8
Подготовка к лабораторным работам	8	8
Самостоятельное изучение тем (вопросов) теоретической части курса	111	111
Всего (без экзамена)	171	171
Подготовка и сдача экзамена	9	9
Общая трудоемкость, ч	180	180
Зачетные Единицы	5.0	

5. Содержание дисциплины

5.1. Разделы дисциплины и виды занятий

Разделы дисциплины и виды занятий приведены в таблице 5.1.

Таблица 5.1 – Разделы дисциплины и виды занятий

Названия разделов дисциплины	СРП, ч	Лаб. раб., ч	КСР, ч	Сам. раб., ч	Всего часов (без экзамена)	Формируемые компетенции
7 семестр						
1 Концепция функционального программирования и определение функций	2	0	4	12	14	ОПК-2, ПК-3
2 Основа языка ЛИСП	2	4		25	31	ОПК-2, ПК-3
3 Рекурсия	1	0		8	9	ОПК-2, ПК-3
4 Функции высокого порядков	1	0		6	7	ОПК-2, ПК-3
5 Концепция логического программирования	2	0		16	18	ОПК-2, ПК-3
6 Введение в Пролог	4	4		34	42	ОПК-2, ПК-3
7 Структуры данных	2	0		14	16	ОПК-2, ПК-3
8 Управление повторением в Прологе	1	0		16	17	ОПК-2, ПК-3
9 Внелогические предикаты Пролога	1	0		12	13	ОПК-2, ПК-3
Итого за семестр	16	8	4	143	171	
Итого	16	8	4	143	171	

Согласована на портале № 42593

5.2. Содержание разделов дисциплины (самостоятельная работа под руководством преподавателя)

Содержание разделов дисциплин (самостоятельная работа под руководством преподавателя) приведено в таблице 5.2.

Таблица 5.2 – Содержание разделов дисциплин (самостоятельная работа под руководством преподавателя)

Названия разделов	Содержание разделов дисциплины (самостоятельная работа под руководством преподавателя)	Трудоемкость, ч	Формируемые компетенции
7 семестр			
1 Концепция функционального программирования и определение функций	Концепция и особенности функционального программирования. Свойства функциональных языков. Вычислимые выражения. Понятие функции, префиксная нотация. Вычисление лямбда-выражений.	2	ОПК-2, ПК-3
	Итого	2	
2 Основа языка ЛИСП	Основные особенности Лиспа, достоинства языка.Элементарные понятия языка Лисп: атомы и списки. Программа на языке Лисп. Определение функций в Лиспе. Базовые функции языка, предикаты. Внутреннее представление списков. Вспомогательные функции над списками.Глобальные и локальные переменные. Изменение значений переменных. Диалоговый режим работы. Функции вводавывода. Разрушающие функции. Обратная блокировка.Циклы и блочные функции. Обработка текстовых данных.	2	ОПК-2, ПК-3
	Итого	2	
3 Рекурсия	Понятие рекурсии. Правила записи рекурсивной функции. Терминальная ветвь, рекурсивная ветвь. Прямая и косвенная рекурсия. Рекурсия с несколькими терминальными ветвями, рекурсивными ветвями.	1	ОПК-2, ПК-3
	Итого	1	
4 Функции высокого порядков	Функции высших порядков. Различие между данными и функциями. Функционалы.Работа с графами и деревьями: представление, обработка, поиск пути на графе.	1	ОПК-2, ПК-3
	Итого	1	
5 Концепция логического программирования	Концепция и особенности логического программирования. Основы языка Пролог: термы, факты, предикаты. Программа на языке Пролог. Переменные и константы. Сложные термы: структуры, списки.	2	ОПК-2, ПК-3
	Итого	2	

Согласована на портале № 42593

6 Введение в Пролог	Объекты данных. Сопоставление: унификация термов. Декларативный смысл пролог-программ. Процедурная семантика. Порядок предложений и целей. Взаимосвязь между Прологом и логикой. Поиск решения на Прологе, понятие резольвенты, завершение поиска.	4	ОПК-2, ПК-3
	Итого	4	
7 Структуры данных	Предикат унификации. Арифметические выражения. Примеры программ с числами. Дифференцирование. Списки. Синтаксис и семантика списков. Некоторые предикаты для списков. Структуры	2	ОПК-2, ПК-3
	Итого	2	
8 Управление повторением в Прологе	Отсечение. Определение отсечения. Примеры программ с отсечением. Отрицание как неудача. Трудности с отсечением и отрицанием. Рекурсия	1	ОПК-2, ПК-3
	Итого	1	
9 Внелогические предикаты Пролога	Анализ и синтез термов. Проверка типа термов. Создание и декомпозиция термов. Ввод и вывод. Метапрограммирование. Операции с базой данных	1	ОПК-2, ПК-3
	Итого	1	
Итого за семестр		16	

5.3. Разделы дисциплины и междисциплинарные связи с обеспечивающими (предыдущими) и обеспечиваемыми (последующими) дисциплинами

Разделы дисциплины и междисциплинарные связи с обеспечивающими (предыдущими) и обеспечиваемыми (последующими) дисциплинами представлены в таблице 5.3.

Таблица 5.3 – Разделы дисциплины и междисциплинарные связи

Наименование дисциплин	№ разделов данной дисциплины, для которых необходимо изучение обеспечивающих и обеспечиваемых дисциплин								
	1	2	3	4	5	6	7	8	9
Предшествующие дисциплины									
1 Дискретная математика	+				+				
2 Математическая логика и теория алгоритмов	+				+	+			+
3 Программирование	+	+	+	+	+	+	+	+	+
Последующие дисциплины									
1 Экспертные системы	+					+	+	+	+

5.4. Соответствие компетенций, формируемых при изучении дисциплины, и видов занятий

Соответствие компетенций, формируемых при изучении дисциплины, и видов занятий представлено в таблице 5.4.

Таблица 5.4 – Соответствие компетенций, формируемых при изучении дисциплины, и видов занятий

Компетен	Виды занятий	Формы контроля

Согласована на портале № 42593

ции	СРП	Лаб. раб.	КСР	Сам. раб.	
ОПК-2	+	+	+	+	Контрольная работа, Экзамен, Проверка контрольных работ, Отчет по лабораторной работе, Тест
ПК-3	+	+	+	+	Контрольная работа, Экзамен, Проверка контрольных работ, Отчет по лабораторной работе, Тест

6. Интерактивные методы и формы организации обучения

Не предусмотрено РУП.

7. Лабораторные работы

Наименование лабораторных работ приведено в таблице 7.1.

Таблица 7.1 – Наименование лабораторных работ

Названия разделов	Наименование лабораторных работ	Трудоемкость, ч	Формируемые компетенции
7 семестр			
2 Основа языка ЛИСП	Лабораторная работа "Основы языка Лисп"	4	ОПК-2, ПК-3
	Итого	4	
6 Введение в Пролог	Лабораторная работа "Основы языка Пролог. Создание простейших функций"	4	ОПК-2, ПК-3
	Итого	4	
Итого за семестр		8	

8. Контроль самостоятельной работы

Виды контроля самостоятельной работы приведены в таблице 8.1.

Таблица 8.1 – Виды контроля самостоятельной работы

№	Вид контроля самостоятельной работы	Трудоемкость (час.)	Формируемые компетенции
7 семестр			
1	Контрольная работа	2	ОПК-2, ПК-3
2	Контрольная работа	2	ОПК-2, ПК-3
Итого		4	

9. Самостоятельная работа

Виды самостоятельной работы, трудоемкость и формируемые компетенции представлены в таблице 9.1.

Таблица 9.1 – Виды самостоятельной работы, трудоемкость и формируемые компетенции

Названия разделов	Виды самостоятельной работы	Трудоемкость, ч	Формируемые компетенции	Формы контроля
7 семестр				
1 Концепция функционального программирования и определение функций	Самостоятельное изучение тем (вопросов) теоретической части курса	12	ОПК-2, ПК-3	Тест, Экзамен
	Итого	12		

Согласована на портале № 42593

2 Основа языка ЛИСП	Самостоятельное изучение тем (вопросов) теоретической части курса	13	ОПК-2, ПК-3	Контрольная работа, Отчет по лабораторной работе, Тест, Экзамен
	Подготовка к лабораторным работам	4		
	Оформление отчетов по лабораторным работам	4		
	Подготовка к контрольным работам	4		
	Итого	25		
3 Рекурсия	Самостоятельное изучение тем (вопросов) теоретической части курса	8	ОПК-2, ПК-3	Тест, Экзамен
	Итого	8		
4 Функции высокого порядков	Самостоятельное изучение тем (вопросов) теоретической части курса	6	ОПК-2, ПК-3	Контрольная работа, Тест, Экзамен
	Итого	6		
5 Концепция логического программирования	Самостоятельное изучение тем (вопросов) теоретической части курса	16	ОПК-2, ПК-3	Тест, Экзамен
	Итого	16		
6 Введение в Пролог	Самостоятельное изучение тем (вопросов) теоретической части курса	18	ОПК-2, ПК-3	Контрольная работа, Отчет по лабораторной работе, Тест, Экзамен
	Подготовка к лабораторным работам	4		
	Оформление отчетов по лабораторным работам	4		
	Подготовка к контрольным работам	8		
	Итого	34		
7 Структуры данных	Самостоятельное изучение тем (вопросов) теоретической части курса	10	ОПК-2, ПК-3	Контрольная работа, Тест, Экзамен
	Подготовка к контрольным работам	4		
	Итого	14		
8 Управление повторением в Прологе	Самостоятельное изучение тем (вопросов) теоретической части курса	16	ПК-3	Тест, Экзамен
	Итого	16		
9 Внелогические предикаты Пролога	Самостоятельное изучение тем (вопросов) теоретической части курса	12	ОПК-2, ПК-3	Тест, Экзамен

Согласована на портале № 42593

	Итого	12		
	Выполнение контрольной работы	4	ОПК-2, ПК-3	Контрольная работа
Итого за семестр		143		
	Подготовка и сдача экзамена	9		Экзамен
Итого		152		

10. Контроль самостоятельной работы (курсовой проект / курсовая работа)

Не предусмотрено РУП.

11. Рейтинговая система для оценки успеваемости обучающихся

Рейтинговая система не используется.

12. Учебно-методическое и информационное обеспечение дисциплины

12.1. Основная литература

1. Зюзьков В.М. Функциональное программирование [Электронный ресурс]: Учебное пособие. — Томск Томский межвузовский центр дистанционного образования, 2005. Доступ из личного кабинета студента. - Режим доступа: https://study.tusur.ru/study/library/ (дата обращения: 10.08.2018).
2. Зюзьков В.М. Логическое программирование [Электронный ресурс]: Учебное пособие. — Томск Томский межвузовский центр дистанционного образования, 2005. Доступ из личного кабинета студента. - Режим доступа: https://study.tusur.ru/study/library/ (дата обращения: 10.08.2018).

12.2. Дополнительная литература

1. Функциональное программирование и интеллектуальные системы [Электронный ресурс]: Учебное пособие / Салмина Н. Ю. - 2016. Доступ из личного кабинета студента. - Режим доступа: https://study.tusur.ru/study/library/ (дата обращения: 10.08.2018).

12.3. Учебно-методические пособия

12.3.1. Обязательные учебно-методические пособия

1. Зюзьков В. М. Логическое и функциональное программирование [Электронный ресурс]: Учебно-методическое пособие. - Томск Томский межвузовский центр дистанционного образования, 2000. Доступ из личного кабинета студента. - Режим доступа: https://study.tusur.ru/study/library/ (дата обращения: 10.08.2018).
2. Зюзьков В.М. Функциональное и логическое программирование : электронный курс / В. М. Зюзьков. – Томск ТУСУР, ФДО, 2018. Доступ из личного кабинета студента.
3. Зюзьков В.М. Функциональное и логическое программирование [Электронный ресурс]: методические указания по организации самостоятельной работы для студентов заочной формы обучения технических направлений, обучающихся с применением дистанционных образовательных технологий / В.М. Зюзьков. – Томск ФДО, ТУСУР, 2018. Доступ из личного кабинета студента. - Режим доступа: https://study.tusur.ru/study/library/ (дата обращения: 10.08.2018).

12.3.2. Учебно-методические пособия для лиц с ограниченными возможностями здоровья и инвалидов

Учебно-методические материалы для самостоятельной и аудиторной работы обучающихся из числа лиц с ограниченными возможностями здоровья и инвалидов предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации.

Для лиц с нарушениями зрения:

- в форме электронного документа;
- в печатной форме увеличенным шрифтом.

Для лиц с нарушениями слуха:

- в форме электронного документа;
- в печатной форме.

Для лиц с нарушениями опорно-двигательного аппарата:

Согласована на портале № 42593

– в форме электронного документа;
– в печатной форме.

12.4. Профессиональные базы данных и информационные справочные системы

1. ЭБС «Лань»: www.e.lanbook.com (доступ из личного кабинета студента по ссылке http://lanbook.fdo.tusur.ru).
2. http://fprog.ru/ - Журнал «Практика функционального программирования»
3. http://alexott.net/ru/fp/books/ - Обзор литературы о функциональном программировании.
4. http://fprog.ru/2009/issue1/alex-ott-literature-overview/ - Обзор литературы о логическом программировании.
5. http://www.osp.ru – Издательство «Открытые системы»
6. http://www.cnews.ru – Издание о высоких технологиях
7. http://habrahabr.ru/ - Сообщество IT-профессионалов

13. Материально-техническое обеспечение дисциплины и требуемое программное обеспечение

13.1. Общие требования к материально-техническому и программному обеспечению дисциплины

13.1.1. Материально-техническое и программное обеспечение дисциплины

Кабинет для самостоятельной работы студентов

учебная аудитория для проведения занятий лабораторного типа, помещение для проведения групповых и индивидуальных консультаций, помещение для проведения текущего контроля и промежуточной аттестации, помещение для самостоятельной работы

634034, Томская область, г. Томск, Вершинина улица, д. 74, 207 ауд.

Описание имеющегося оборудования:

- Коммутатор MicroTeak;
- Компьютер PENTIUM D 945 (3 шт.);
- Компьютер GELERON D 331 (2 шт.);
- Комплект специализированной учебной мебели;
- Рабочее место преподавателя.

Программное обеспечение:

– 7-zip
– Google Chrome
– Kaspersky Endpoint Security 10 для Windows
– Microsoft Windows
– OpenOffice
– SWI-Prolog (с возможностью удаленного доступа)
– XLisp (с возможностью удаленного доступа)

13.1.2. Материально-техническое и программное обеспечение для лабораторных работ

Кабинет для самостоятельной работы студентов

учебная аудитория для проведения занятий лабораторного типа, помещение для проведения групповых и индивидуальных консультаций, помещение для проведения текущего контроля и промежуточной аттестации, помещение для самостоятельной работы

634034, Томская область, г. Томск, Вершинина улица, д. 74, 207 ауд.

Описание имеющегося оборудования:

- Коммутатор MicroTeak;
- Компьютер PENTIUM D 945 (3 шт.);
- Компьютер GELERON D 331 (2 шт.);
- Комплект специализированной учебной мебели;
- Рабочее место преподавателя.

Программное обеспечение:

– 7-zip

Согласована на портале № 42593

– Google Chrome
– Microsoft Windows
– OpenOffice
– SWI-Prolog (с возможностью удаленного доступа)
– XLisp (с возможностью удаленного доступа)

13.1.3. Материально-техническое и программное обеспечение для самостоятельной работы

Для самостоятельной работы используются учебные аудитории (компьютерные классы), расположенные по адресам:

- 634050, Томская область, г. Томск, Ленина проспект, д. 40, 233 ауд.;
- 634045, Томская область, г. Томск, ул. Красноармейская, д. 146, 201 ауд.;
- 634034, Томская область, г. Томск, Вершинина улица, д. 47, 126 ауд.;
- 634034, Томская область, г. Томск, Вершинина улица, д. 74, 207 ауд.

Состав оборудования:

- учебная мебель;
- компьютеры класса не ниже ПЭВМ INTEL Celeron D336 2.8ГГц. - 5 шт.;
- компьютеры подключены к сети «Интернет» и обеспечивают доступ в электронную информационно-образовательную среду университета.

Перечень программного обеспечения:

- Microsoft Windows;
- OpenOffice;
- Kaspersky Endpoint Security 10 для Windows;
- 7-Zip;
- Google Chrome.

13.2. Материально-техническое обеспечение дисциплины для лиц с ограниченными возможностями здоровья и инвалидов

Освоение дисциплины лицами с ограниченными возможностями здоровья и инвалидами осуществляется с использованием средств обучения общего и специального назначения.

При занятиях с обучающимися **с нарушениями слуха** предусмотрено использование звукоусиливающей аппаратуры, мультимедийных средств и других технических средств приема/передачи учебной информации в доступных формах, мобильной системы преподавания для обучающихся с инвалидностью, портативной индукционной системы. Учебная аудитория, в которой занимаются обучающиеся с нарушением слуха, оборудована компьютерной техникой, аудиотехникой, видеотехникой, электронной доской, мультимедийной системой.

При занятиях с обучающимися **с нарушениями зрениями** предусмотрено использование в лекционных и учебных аудиториях возможности просмотра удаленных объектов (например, текста на доске или слайда на экране) при помощи видеоувеличителей для комфортного просмотра.

При занятиях с обучающимися **с нарушениями опорно-двигательного аппарата** используются альтернативные устройства ввода информации и другие технические средства приема/передачи учебной информации в доступных формах, мобильной системы обучения для людей с инвалидностью.

14. Оценочные материалы и методические рекомендации по организации изучения дисциплины

14.1. Содержание оценочных материалов и методические рекомендации

Для оценки степени сформированности и уровня освоения закрепленных за дисциплиной компетенций используются оценочные материалы в составе:

14.1.1. Тестовые задания

1. Одним из основных методов в функциональном программировании является суперпозиция функций. Рассматриваются суперпозиции функций CAR и CDR. Дан список (setq x `(a s (d) f

Согласована на портале № 42593

g)). Что вернет функция (caddr x) ?

1) ((d) f g)

2) (a s)

3) s

4) (d)

2. Что будет получено в результате вызова следующей суперпозиции базовых функций языка Лисп: (cons (car `(1 2 3)) `(+ 2 6)) ?

1) (1 2 3 8)

2) (1 + 2 6)

3) (1 8)

4) (1 . 8)

3. Для разветвления вычислений в функциональном языке Лисп используется условное предложение COND. Задан список (setq x `((1) (2) 3 (4))). Что будет получено в результате работы следующего выражения

(cond ((null x) 0) ((atom (car x)) 1) ((eq (cadr x) `(2)) 2) (t 3)) ?

1) 0

2) 1

3) 2

4) 3

4. В основе всех функциональных языков лежит лямбда-исчисление в том смысле, что все функциональные программы можно преобразовать в лямбда-выражение. Что выдаст следующее лямбда-выражение, описанное на языке Лисп?

((lambda (x y) (cond ((zerop x) (* y y)) ((< x 0) (+ y y)) (t (+ x y)))) (+ -10 2) (+ 2 10))

1) 24

2) 4

3) Nil

4) 144

5. Программа на функциональном языке Лисп представляет собой последовательность вычислимых выражений. Что будет выдано программой в результате следующей последовательности вызова вычислимых выражений?

> (setq x 10)

10

> (defun f (x y) (+ (* x x) y))

F

> (f 2 3)

?

1) 103

2) 10

3) 7

4) error

6. Механизм рекурсивного вызова является одним из основных принципов функционального программирования. Что выполняет следующая рекурсивная функция, аргументом которой является список?

(defun q (z) (cond ((null z) nil)

(t (append [q (cdr z)] [list (car z)]))))

1) переставляет последний элемент списка в начало;

2) меняет первый и последний элемент списка местами;

3) переставляет первый элемент списка в конец списка;

4) переставляет элементы списка в обратном порядке.

7. Любая рекурсивная функция должна иметь терминальные ветви (определяющие правило останова) и рекурсивные ветви. Какое количество терминальных ветвей содержит следующая рекурсивная функция?

(defun q (z) (cond ((null z) nil)

((null (cdr z)) 0)

Согласована на портале № 42593

((not (numberp (car z))) nil)
(t (+ [*(car z)(cadr z)] [q (cddr z)]))))
1) 1
2) 2
3) 3
4) 4

8. Функции, которые не формируют новые списки, а изменяют структуру существующих списков, называются разрушающими. Чему будет равен Y в результате следующей последовательности вызова вычислимых выражений с использованием разрушающей функции?

(setq x `(2 3))
(setq y (cons 1 x))
(rplaca x 7)
y - ?
1) (1 2 3)
2) (1 7)
3) 7
4) (1 7 3)

9. Любой функциональный язык содержит функционалы: функции, имеющие аргументы, значением которых являются функции. Что будет получено в результате работы следующего функционала?

(mapcar `length `((1 2 3)(a s d f)(4 5)))
1) (3 4 2)
2) (3 2 1)
3) (2 4 3)
4) Nil

10. Использование механизма циклов вместо рекурсии позволяет экономить память и строить, зачастую, более эффективные программы. Что будет получено в результате работы следующего вычислимого выражения с использованием цикла?

(let ((x 0)(y nil))(loop (setq x (+ 1 x))(setq y (cons x y))(cond((= x 5)(return y)))))
1) 5
2) nil
3) (5 4 3 2 1)
4) (1 2 3 4 5)

11. Применение функционалов в программе основано на том, что программы и данные в языках функционального программирования представляются одинаково. Определена функция SUM, аргументом которой является список, а результат работы – сумма элементов списка. Что будет получено в результате работы следующего функционала, где функция SUM рассматривается как аргумент другой функции?

(maplist `sum `(1 2 3 4 5))
1) 15
2) (15 14 12 9 5)
3) (5 9 12 14 15)
4) (1 2 3 4 5)

12. Основная структура данных в языках функционального программирования – списки. По сути, любая функция на языке Лисп является функцией обработки списков. Какой список свойств получится в результате выполнения следующей последовательности вычислимых выражений?

(setf (get `as `v4) `(4))
(setf (get `as `v2) `(2))
(setf (get `as `v3) `(3))
(setf (get `as `v1) `(1))
1) (v4 4 v3 3 v2 2 v1 1)
2) (v1 1 v3 3 v2 2 v4 4)
3) (v4 (4) v2 (2) v3 (3) v1 (1))
4) (v1 (1) v3 (3) v2 (2) v4 (4))

Согласована на портале № 42593

13. Особый вид списков, используемых в языке Лисп – ассоциативные списки, элементами которых являются точечные пары. Какой ассоциативный список получится в результате работы следующей функции:

(pairlis `((a) (b) (c)) `(1 2 3) ())?

1) ((c . 3)(b . 2)(a . 1))

2) (c 3 b 2 a 1)

3) ((a . 1)(b . 2)(c . 3))

4) ((a 1)(b 2)(c 3))

14. Фундаментальной операцией над объектами в логическом программировании является механизм унификации – сопоставление термов и переменных. Проанализируйте, унифицируемы ли следующие предикаты?

Pred1 (X, Y, Y) ? Pred1 (5, 10, 12)

1) Да

2) Нет: переменные и константы не сравнимы

3) Нет: в предикате не может быть две одинаковые переменные

4) Нет, одна переменная не может иметь два разных значения

15. Задана следующая последовательность предикатов.

F(0,1).

F(1,2).

F(2,3).

F(3,4).

Pr(X,Y) :- f(X,Z), f(Z,Y).

Чему будут равны значения переменных A и Z в результате следующего вопроса:

? – pr(0,A), pr(A,Z).

1) A=1, Z=2

2) A=1, Z=3

3) A=3, Z=4

4) Нет решения

16. Приведенная ниже процедура описывает следующее знание: «Мэри любит всех животных, кроме змей».

Love (mary, X) :- snake(X), fail.

Love (mary, X) :- animal(X), not(snake(X)).

Перепишите процедуру, убрав отрицание, используя при необходимости механизм отсечения.

1) Love (mary, X) :- snake(X),fail,!.

Love (mary, X) :- animal(X).

2) Love (mary, X) :- !,snake(X),fail.

Love (mary, X) :- animal(X).

3) Love (mary, X) :- snake(X),!,fail.

Love (mary, X) :- animal(X).

4) Love (mary, X) :- snake(X),!.

Love (mary, X) :- animal(X).

17. Рекурсия – это способ задания функции путем определения каждого его значения в терминах ранее определенных значений. Рекурсивный механизм является мощнейшим инструментом построения программ в логическом программировании. Что выполняет следующая рекурсивная процедура, аргументом которой является список?

a ([ ], 1).

a ([X | XT], P) :- a (XT, P1), P=P1*X.

1) перемножает элементы списка, стоящие на нечетных позициях;

2) перемножает элементы списка, стоящие на четных позициях;

3) перемножает элементы списка, не равные нулю;

4) перемножает элементы списка.

18. Задана следующая рекурсивная процедура:

Q ([ ], 0).

Согласована на портале № 42593

Q ([X], X).
Q ([X, _| Y], N) :- Q (Y, N1), N=N1+X.
Чему будет равно X в результате следующего вопроса?
?- q ([3,1,5,2,6], X).
1) 14
2) 5
3) 8
4) 3

19. Язык логического программирования Пролог создан для задач анализа и понимания естественного языка. Как можно определить предикат внук(X,Y) через предикат родитель(X,Y), основываясь на понятиях родственных отношений?
1) Внук(X,Y):-родитель(X,Y),родитель(X,Y).
2) Внук(X,Y):-родитель(X,Z),родитель(Z,Y).
3) Внук(X,Y):-родитель(X,X),родитель(Y,Y).
4) Внук(X,Y):-родитель(Z,X),родитель(Z,Y).

20. Работа программы в логическом программировании основана, прежде всего, на обработке существующих фактов. Дана база фактов: животное(<наименование>,<ареал>,<количество_особей>). Как будет выглядеть предикат, формирующий список, элементами которого являются <количество_особей> по всем животным?
1) findall(животное(_,_,X)).
2) findsll(X,животное(_,_,X),X).
3) findall (X,животное(_,_,X),L).
4) findall(X,животное(_,_,L),[X|L]).

14.1.2. Экзаменационные тесты

Приведены примеры типовых заданий из банка экзаменационных тестов, составленных по пройденным разделам дисциплины.

Вопрос 1

Лисп. Какая ошибка в определении функции, проверяющей, является ли данный список одноуровневым?

(defun f (s)
(if (not (atom (car s))) nil (f (cdr s))))
1) без ошибок;
2) перепутаны случаи "то" и "иначе" в условной функции;
3) надо пользоваться предикатом or;
4) нет окончания рекурсии.

Вопрос 2

Лисп. Сколько элементов самого верхнего уровня в следующих списках:
1) ((1 2 3));
2) ((a b) c (d (e)));
3) (a ((())) nil nil);
4) ((((a (b (c d) e) f) g) h ((i (j) k) l) m) n)

Вопрос 3

Лисп. Каково общее число подсписков в подсписках (т. е. списки уровня 3; исходный список имеет уровень 1) в следующих списках:
1) (6 (3 6 (7 (4 5) 8)) (9 3));
2) (((a 9 ((b 7) c)) (d 5)));
3) (quote (s (3 5) (7 (9 8))));
4) (d (q 2 7 4) (+ 1 (* 7 (+ 3 (- 2)))))?

Вопрос 4

Лисп. Какие из следующих утверждений верны?
1. Язык XLisp - функциональный язык только с ленивыми вычислениями.

Согласована на портале № 42593

2. Язык XLisp - функциональный язык только с энергичными вычислениями.
3. Язык XLisp - функциональный язык с энергичными и ленивыми вычислениями.
4. Язык C++ - функциональный язык.

Вопрос 5
Лисп. Даны определения функций
(defun twice (f)
(function (lambda (x) (funcall f (funcall f x)))))
(defun do (x) (funcall (twice 'list) x))
Чему равно значение (do '0) ? Введите символьное выражение без пробелов.

Вопрос 6
Лисп. Дано определения функций
(defun many (f x)
(mapcar (function (lambda (g) (funcall g x))) f))
(defun f1 (x) (+ x x))
(defun f2 (x) (* x x))
Чему равно значение (length (many '(f1 f2) 1)) ?

Вопрос 7
Лисп. Дано определение функции
(defun create (x y)
(eval (cons 'defun (cons x (cdr y)))))
Вызов этой функции приводит к определению некоторой новой функции f.
(create 'f '(lambda (x) (* x x)))
Чему равно значение (f 2)?

Вопрос 8
Лисп. Дано определение функции
(defun factor (n)
(if (< n 2) '(1) (append (factor (- n 1)) (list '* n))))
Чему равно значение (length (factor 3)) ?

Вопрос 9
Лисп. Каково значение следующего выражения
(eval (cons (quote >) (cons 5 (list ((lambda (x y) (- x y)) 3 7))))) ?
Введите значение маленькими буквами.

Вопрос 10
Лисп. Дано определение функции
(defun f (x s)
(if (= x (car s)) 1 (+ 1 (f x (cdr s)))))
Чему равно значение (f 2 '(1 3 2)) ?

Вопрос 11
Дана цель для интерпретатора Пролога
?- not parent(X,pat).
на которую был получен отрицательный ответ. Какое из следующих трех предложений правильно передает логический смысл этого ответа:
1) "Не существует родитель у Pat."
2) "Не все X являются родителями Pat."
3) "Есть родители, но не у Pat."
Варианты ответов:
1) первое предложение;

Согласована на портале № 42593

2) второе предложение;
3) третье предложение;
4) все три предложения не передают логический смысл ответа.

Вопрос 12

Пусть дано отношение 'родитель' на Прологе. Определим отношение "X - родственник Y" следующим образом :

'родственник'(X,Y):-'родитель'(X,Y).
'родственник'(X,Y): - 'родитель'(Y,X).
'родственник'(X,Y):-'родственник'(X,Z),'родственник'(Z,Y).

Какие из следующих утверждений верны?

1. Одно из первых двух правил лишнее.
2. Третье правило потенциально опасное - оно может привести в некоторых запросах к бесконечной рекурсии.
3. Любой запрос к данной программе приводит к конечной работе интерпретатора Пролога.
4. Правила неправильно задают отношение 'родственник'.

Вопрос 13

Пролог. Определим предикат length для вычисления длины списка:

```
length([], 0).
length([_|T], N) :- length(T, N1), N is 1+N1.
```

Если во втором правиле в его теле поменять две цели местами, то при вызове

```
?- length([1, 2, 3], N).
```

произойдет следующее:

1) интерпретатор не сможет вычислить цель, а сообщит о ошибке;
2) N получит значение равное 3;
3) цель успешно вычислится, но N в качестве значения получит не число;
4) интерпретатор ответит: No.

Вопрос 14

Пролог. Последовательность чисел Фибоначчи имеет вид

1, 1, 2, 3, 5, 8, 13,…

Каждый член последовательности, за исключением первых двух, представляет собой сумму предыдущих двух членов. Какой метод программирования позволяет написать предикат fib(N, F), эффективно (т. е. линейно по времени) вычисляющий N-ое число Фибоначчи F?

1) использование запоминающих функций;
2) такого метода нет, рекурсия всегда не эффективна;
3) использование отсечений;
4) изменение порядка целей и предложений;
5) использование накапливающих параметров.

Вопрос 15

Пролог. Определите, будет ли каждая из следующих пар термов унифицируемой:

1) 'книга'('название'('Ферма животных'), 'автор'('Джордж Оруэлл')) и 'книга'(название'(T)', Avtor);
2) 'дата'('день недели'('среда'), 'число'(12), 'месяц'(M), 'год'(1986)) и 'дата'(W,D,X,Y);
3) 'праздник'('рождество', 'дата'('день'(25), 'месяц'('декабрь'), 'год'(Y))) и 'праздник'(H, 'дата'(D,M, 'год'(1986));
4) 'праздник'('Первомай'(1, 'май')) и 'праздник'('Первомай', 1, 'май').

Вопрос 16

Пролог. Какие следующие утверждения истинны?

1. Правило имеет заголовок (голову) и тело.
2. Унификация цели может вызвать необходимость унификации подцелей.

Согласована на портале № 42593

3. Пролог всегда совершает возврат для повторной унификации любой цели.

4. Пытаясь согласовать цель повторно, Пролог начинает поиск снова с начала программы.

5. Пользователь может заставить Пролог совершить возврат, отвергнув полученный ответ.

Вопрос 17

Пролог. К чему приведет следующий вызов предиката

?- max(4+7, 8*9, N).

если предикат max/3 определен следующим образом:

max(X,Y,X):-X>=Y.

max(X,Y,Y):-Y>X.

1) Пролог ответит Yes и выдаст N=72;

2) Пролог ответит Yes и выдаст N=8*9;

3) Пролог ответит No;

4) Пролог сообщит об ошибке в операциях сравнения.

Вопрос 18

В каком порядке Пролог ищет утверждения программы для унификации с целью?

1) в порядке размещения клауз (предложений) в тексте программы - сверху вниз;

2) сначала просматриваются факты в программе, потом - правила сверху вниз;

3) Пролог сам устанавливает порядок, исходя из эффективности программы.

Вопрос 19

Пролог. Какие следующие утверждения истинны?

1. В Прологе единственная структура данных - термы.

2. Структура характеризуется своим функтором. Не допускается использовать структуры с одинаковым функтором и разной местностью (арностью).

3. Мы задаем вопросы, используя термы в качестве целей.

4. Структура относится к рекурсивному типу данных.

5. Переменная - это "забронированное" место. Любой терм может заменить переменную, но для разных вхождений переменной в структуру не обязательна должна иметь место одна и та же замена.

6. Когда два терма унифицируются, переменные в них заменяются на некоторые значения.

Вопрос 20

Пролог. Определите, будет ли каждая из следующих пар термов унифицируемой:

1) 'книга'('название'('Ферма животных'), 'автор'('Джордж Оруэлл')) и 'книга'(название'(T)', Avtor);

2) 'дата'('день недели'('среда'), 'число'(12), 'месяц'(M), 'год'(1986)) и 'дата'(W,D,X,Y);

3) 'праздник'('рождество', 'дата'('день'(25), 'месяц'('декабрь'), 'год'(Y))) и 'праздник'(H, 'дата'(D,M, 'год'(1986));

4) 'праздник'('Первомай'(1, 'май')) и 'праздник'('Первомай', 1, 'май').

Введите через пробел номера пар (в порядке возрастания), для которых, вы считаете, проходит унификация.

14.1.3. Темы контрольных работ

Контрольная работа 1

Задание состоит из трех задач, в которых требуется составить программы на Лиспе. В первой задаче не требуется рекурсия, остальные две задачи требуют применения простой рекурсии. При составлении программ (если не оговорено противное) можно использовать все встроенные функции Лиспа. Тексты всех программ, если вы мыслите в духе функционального программирования, буквально состоят из нескольких строчек.

Отладку программ можно осуществлять с помощью функции трассировки (trace <имя функции>), трассировка функции отключается - (untrace <имя функции>).

Пример задания:

Задача 1

Согласована на портале № 42593

Пусть l1 и l2 -списки. Напишите функцию, которая возвращала бы t, если первые два элемента этих списков соответственно равны друг другу, и nil - в противном случае (например, если длина одного из списков меньше 2).

Задача 2

Напишите функцию, зависящую от двух аргументов x и u, удаляющую все вхождения x в список u на всех уровнях.

Задача 3

Сортировка слиянием. Даны два упорядоченных по возрастанию списка чисел x и y. Написать функцию (merge x y), которая в качестве значения выдает общий упорядоченный список элементов x и y. Например,

merge('(1 3 5 7 8) '(2 3 5 7)) => (1 2 3 3 5 5 7 7 8).

Контрольная работа 2

Задание состоит из двух задач, в которых требуется составить программы на Прологе для написания простых предикатов. При составлении программ (если не оговорено противное) можно использовать все встроенные

предикаты Пролога. Тексты всех программ, если вы мыслите в духе логического программирования, получаются небольшие. SWI-Prolog не имеет стандартного help'a для Windows, для этого используется предикат help. Вызов help(<имя предиката>) выдает на экран информацию об этом предикате. Вызов help(7) выдает на экран список всех встроенных предикатов с комментариями. Текстовый файл руководства по SWI-Prolog - pl\library\manual. Отладку предикатов можно осуществлять с помощью предиката трассировки trace(<имя предиката>), трассировка предиката отключается - trace(<имя предиката>, -all).

Пример задания:

Задача 1

Постройте предикат position_max(+L, -M, -N), который в списке L находит максимальное значение M и порядковый номер N этого значения.

Задача 2

Определите умножение целых чисел через сложение и вычитание.

14.1.4. Темы лабораторных работ

Лабораторная работа "Основы языка Лисп"

Лабораторная работа "Основы языка Пролог. Создание простейших функций"

14.1.5. Методические рекомендации

Учебный материал излагается в форме, предполагающей самостоятельное мышление студентов, самообразование. При этом самостоятельная работа студентов играет решающую роль в ходе всего учебного процесса.

Начать изучение дисциплины необходимо со знакомства с рабочей программой, списком учебно-методического и программного обеспечения. Самостоятельная работа студента включает работу с учебными материалами, выполнение контрольных мероприятий, предусмотренных учебным планом.

В процессе изучения дисциплины для лучшего освоения материала необходимо регулярно обращаться к рекомендуемой литературе и источникам, указанным в учебных материалах; пользоваться через кабинет студента на сайте Университета образовательными ресурсами электронно-библиотечной системы, а также общедоступными интернет-порталами, содержащими научно-популярные и специализированные материалы, посвященные различным аспектам учебной дисциплины.

При самостоятельном изучении тем следуйте рекомендациям:

- чтение или просмотр материала необходимо осуществлять медленно, выделяя основные идеи; на основании изученного составить тезисы. Освоив материал, попытаться соотнести теорию с примерами из практики;

- если в тексте встречаются термины, следует выяснить их значение для понимания дальнейшего материала;

- необходимо осмысливать прочитанное и изученное, отвечать на предложенные вопросы.

Студенты могут получать индивидуальные консультации с использованием средств теле-

Согласована на портале № 42593

коммуникации.

По дисциплине могут проводиться дополнительные занятия в форме вебинаров. Расписание вебинаров публикуется в кабинете студента на сайте Университета. Запись вебинара публикуется в электронном курсе по дисциплине.

14.2. Требования к оценочным материалам для лиц с ограниченными возможностями здоровья и инвалидов

Для лиц с ограниченными возможностями здоровья и инвалидов предусмотрены дополнительные оценочные материалы, перечень которых указан в таблице 14.

Таблица 14 – Дополнительные материалы оценивания для лиц с ограниченными возможностями здоровья и инвалидов

Категории обучающихся	Виды дополнительных оценочных материалов	Формы контроля и оценки результатов обучения
С нарушениями слуха	Тесты, письменные самостоятельные работы, вопросы к зачету, контрольные работы	Преимущественно письменная проверка
С нарушениями зрения	Собеседование по вопросам к зачету, опрос по терминам	Преимущественно устная проверка (индивидуально)
С нарушениями опорно-двигательного аппарата	Решение дистанционных тестов, контрольные работы, письменные самостоятельные работы, вопросы к зачету	Преимущественно дистанционными методами
С ограничениями по общемедицинским показаниям	Тесты, письменные самостоятельные работы, вопросы к зачету, контрольные работы, устные ответы	Преимущественно проверка методами исходя из состояния обучающегося на момент проверки

14.3. Методические рекомендации по оценочным материалам для лиц с ограниченными возможностями здоровья и инвалидов

Для лиц с ограниченными возможностями здоровья и инвалидов предусматривается доступная форма предоставления заданий оценочных средств, а именно:

- в печатной форме;
- в печатной форме с увеличенным шрифтом;
- в форме электронного документа;
- методом чтения ассистентом задания вслух;
- предоставление задания с использованием сурдоперевода.

Лицам с ограниченными возможностями здоровья и инвалидам увеличивается время на подготовку ответов на контрольные вопросы. Для таких обучающихся предусматривается доступная форма предоставления ответов на задания, а именно:

- письменно на бумаге;
- набор ответов на компьютере;
- набор ответов с использованием услуг ассистента;
- представление ответов устно.

Процедура оценивания результатов обучения лиц с ограниченными возможностями здоровья и инвалидов по дисциплине предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в форме электронного документа;
- в печатной форме увеличенным шрифтом.

Для лиц с нарушениями слуха:

- в форме электронного документа;
- в печатной форме.

Для лиц с нарушениями опорно-двигательного аппарата:

Согласована на портале № 42593

- в форме электронного документа;
- в печатной форме.

При необходимости для лиц с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения может проводиться в несколько этапов.

Согласована на портале № 42593