Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Кубанский государственный университет»

Факультет компьютерных технологий и прикладной математики Кафедра вычислительных технологий

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Б1.В.ДВ.03.01 «Функциональное и логическое программирование»

Направление подготовки/специальность 02.03.02 **Фундаментальная информатика и информационные технологии**
(код и наименование направления подготовки/специальности)

Направленность (профиль) /специализация
Математическое и программное обеспечение компьютерных технологий

Программа подготовки *академический бакалавриат*

Форма обучения *очная*

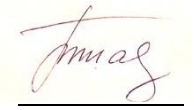
Квалификация выпускника *бакалавр*

Краснодар 2024

Рабочая программа дисциплины «Функциональное и логическое программирование» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии

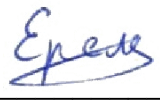
Программу составили:

Жук Арсений Сергеевич, ст. преподаватель _____
Ф.И.О. , должность, ученая степень, ученое звание подпись



Приходько Татьяна Александровна, к.т.н., доцент, доцент _____
Ф.И.О. , должность, ученая степень, ученое звание подпись

Рабочая программа дисциплины «Функциональное и логическое программирование» утверждена на заседании кафедры Вычислительных технологий протокол № 7 «03 » мая 2024 г.



ИО заведующий кафедрой (разработчика) Ерёмин А.А. _____
(фамилия, инициалы подпись

Утверждена на заседании учебно-методической комиссии факультета Компьютерных Технологий и Прикладной Математики
протокол № 3 от «21» мая 2024 г

Председатель УМК факультета Коваленко А.В.
фамилия, инициалы подпись

Рецензенты:

Гаркуша О.В., доцент кафедры информационных технологий ФБГОУ ВО «Кубанский государственный университет», кандидат физико-математических наук.

Схаляхо Ч.А., доцент КВВУ им.С.М.Штеменко, к.ф.-м.н., доцент

1. Цели и задачи освоения дисциплины

1.1 Цель освоения дисциплины

Учебная дисциплина «Функциональное и логическое программирование» предназначена для изучения декларативной парадигмы программирования и её места в современной коммерческой и научной разработке.

Целью преподавания и изучения дисциплины «Функциональное и логическое программирование» является знакомство студентов с понятием парадигма программирования, изучение принципов работы в декларативном стиле, определение круга задач, решаемых модулями, написанными в императивной или декларативной парадигме, получение практических навыков писать читаемый код в функциональном или логическом стиле на актуальных языках программирования с применением современных платформ и фреймворков.

1.2 Задачи дисциплины

В результате освоения данной компетенции студент должен:

знать фундаментальные концепции написания программ в декларативном стиле, математические принципы лямбда исчисления, принципы функционального программирования, принципы логических переборных языков программирования.

уметь реализовывать модули анализа данных на основе функциональных интерфейсов, строить чистые функции высших порядков, реализовывать системы формального вывода и переборные алгоритмы средствами логического программирования, внедрять их в комплексные программные решения.

владеть навыками определения парадигмы, подходящей для решения конкретной задачи, навыками написания модулей работы с внешними системами (размеченные файлы, базы данных, потоки ввода) средствами языков функциональной и логической парадигмы программирования

1.3. Место дисциплины (модуля) в структуре образовательной программы

Курс «Функциональное и логическое программирование» относится к части блока Б1 Дисциплины (модули), части, формируемой участниками образовательных отношений и является дисциплиной по выбору.

Для изучения дисциплины студент должен владеть знаниями, умениями и навыками полученными на дисциплинах «Дискретная математика», «Комбинаторный анализ», «Конструирование алгоритмов и структур данных», «Теория алгоритмов и вычислительных процессов», «Управление информацией», «Анализ и проектирование информационных систем», «Интерпретируемые языки программирования», «Паттерны программирования». Знания, умения и навыки, полученные студентами в дисциплине «Функциональное и логическое программирование» являются обязательными для изучения следующих дисциплин «Модели интеллектуальных систем», «Верификация программных систем», «Программирования для мобильных платформ».

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

Изучение данной учебной дисциплины направлено на формирование у обучающихся следующих **компетенций:**

Результаты обучения (знания, умения, опыт, компетенции):

Код и наименование индикатора	Результаты обучения по дисциплине *(знает, умеет, владеет (навыки и/или опыт деятельности))*
ПК-1. Способен понимать и применять в научно-исследовательской и прикладной деятельности современный математический аппарат, основные законы естествознания, современные языки программирования и программное обеспечение; операционные системы и сетевые технологии	
Формулировки индикаторов	
ПК-1.1. Знает основы научно- исследовательской деятельности в области информационных технологий, имеет научные знания в теории информационных систем. ПК-1.2. Умеет применять полученные знания в области фундаментальных научных основ теории информации и решать стандартные задачи в собственной научно-исследовательской деятельности. ПК-1.3. Имеет практический опыт научно- исследовательской деятельности в области информационных технологий.	
ПК-2. Способен проводить под научным руководством локальные исследования на основе существующих методов в конкретной области профессиональной деятельности	
Формулировки индикаторов	
ПК-2.1. Знает принципы построения научной работы, методы сбора и анализа полученного материала, способы аргументации владеет навыками подготовки научных обзоров, публикаций, рефератов и библиографий по тематике проводимых исследований на русском и английском языке. ПК-2.2. Умеет решать научные задачи в связи с поставленной целью и в соответствии с выбранной методикой. ПК-2.3. Имеет практический опыт выступлений и научной аргументации при анализе объекта научной и профессиональной деятельности.	

Результаты обучения по дисциплине достигаются в рамках осуществления всех видов контактной и самостоятельной работы обучающихся в соответствии с утвержденным учебным планом.

Индикаторы достижения компетенций считаются сформированными при достижении соответствующих им результатов обучения.

2. Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоемкость дисциплины составляет 4 зач.ед. (144 часа), их распределение по видам работ представлено в таблице *(для студентов ОФО)*

Вид учебной работы	Всего часов	Семестры (часы)			
		6			
Контактная работа в том числе:	72,3	72,3			
Аудиторные занятия (всего):	64	64			
В том числе:					
Занятия лекционного типа	32	32			
Занятия семинарского типа (семинары, практ. занятия)					
Лабораторные занятия	32	32			
Иная контрольная работа					
Контроль самостоятельной работы	8	8			
Промежуточная аттестация (ИКР)	0,3	0,3			
Самостоятельная работа (всего)	36	36			

В том числе:						
Курсовая работа						
Самостоятельное изучение разделов, самоподготовка (проработка и повторение лекционного материала и материала учебников и учебных пособий,	12	12				
Подготовка к лабораторным и практическим занятиям.)	12	12				
Подготовка к текущему контролю	12	12				
Контроль:	35.7	35.7				
Подготовка к экзамену:	35.7	35.7				
Общая трудоемкость час	144	144				
в т.ч. контактная работа	72,3	72,3				
зач. ед.	4	4				

2.2 Структура дисциплины:

Распределение видов учебной работы и их трудоемкости по разделам дисциплины. Разделы дисциплины, изучаемые в _6__семестре *(очная форма)*

3

№	Наименование разделов	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа
			Л	КСР	ЛР	СРС
1	2	3	4	5	6	7
1	**Раздел 1.** Основы логического программирования		10		10	12
2	**Раздел 2.** Лямбда-исчисление.		6		8	8
3	**Раздел 3.** Основы функционального программирования.		16		14	16
	Итого по разделам дисциплины		32		32	36
	Контроль самостоятельной работы(КСР)	0,3				
	Промежуточная аттестация (ИКР)	8				
	Подготовка к экзамену	35.7				
	Итого по дисциплине:	144				

4

4.2 Содержание разделов дисциплины:
2.3.1 Занятия лекционного типа

№ раз-дела	Наименование раздела	Содержание раздела	Форма текущего контроля	Разработано с участием представителей работодателей
1	2	3	4	5
1	**Раздел 1.** Основы логического программирования	Парадигмы программирования. Императивное программирование. Декларативное программирование. Особенности логического программирования и языка Prolog. Высказывания Общие принципы работы с SWI-Prolog Предикаты и правила Числовые задачи. Рекурсия. Построение дерева вывода.Унификация,	ЛР, КР, ЭП	

		бектрекинг и отсечение. Рекурсия вверх и вниз. Хвостовая рекурсия . Списки Черча. Общий подход. Реализация стандартных задач на списках Встроенные предикаты SWI-Prolog работы со списками Форматы представления строк. Работа с файлами. Динамические и статические факты и правила. Встроенные предикаты построения новых фактов или высказываний. Решение прикладных задач комбинаторики и теории графов средствами SWI-Prolog.		
2	**Раздел 2.** Лямбда-исчисление.	Принципы и достоинства функционального программирования. Чистая функция, функция высших порядков, неизменяемость данных, карринг, замыкание, неименованные функции. Принципы лямбда нотации. Понятие лямбда терма Свободные и связанные переменные. Подстановка и преобразования Лямбда исчисление. Эквивалентность лямбда выражений. Экстенсиональность Редукция Теорема Чёрча-Россера Комбинаторы. Лямбда исчисление как язык программирования. Представление данных в лямбда-исчислении Рекурсивные функции Let-выражения Достижение уровня полноценного языка программирования Списки Чёрча Типизированное лямбда исчисление Полиморфизм	ЛР, КР, ЭП	
3	**Раздел 3.** Основы функционального программирования.	Знакомство с Kotlin. Язык. Философия и инструментарий. Основные элементы: переменные и функции. Классы и свойства. Представление и обработка выбора: перечисления и конструкция «when» Итерации: циклы «while» и «for» Исключения в Kotlin Создание коллекций в Kotlin Определение и вызов функций Упрощение вызова функций Добавление методов в сторонние классы: функции-расширения и свойства-расширения Работа с коллекциями: переменное число аргументов, инфиксная форма записи вызова и поддержка в библиотеке Работа со строками и регулярными выражениями Классы, объекты и интерфейсы локальные функции и расширения Создание иерархий классов Объявление классов с нетривиальными конструкторами или свойствами Методы, сгенерированные компилятором: классы данных и делегирование Лямбда-выражения Совместное объявление класса и экземпляра Лямбда-выражения Функциональный API для работы с коллекциями Последовательности Система типов Лямбда-выражения с получателями Значение null Базовые типы Массивы и коллекции Перегрузка операторов Перегрузка арифметических операторов Перегрузка операторов сравнения Коллекции и диапазоны Мультидекларации и функции Component Функции высшего порядка Делегирование свойств Объявление функций высших порядков Встраиваемые функции Порядок выполнения функций высших порядков Обобщенные типы Параметры обобщенных типов	ЛР, КР, ЭП	

		Стирание и овеществление параметров типов Обобщенные типы и подтипы Объявление и применение аннотаций Рефлексия: интроспекция объектов Kotlin во время выполнения Понятие предметно-ориентированного языка Внутренние предметно-ориентированные языки Структура предметно-ориентированных языков Создание разметки HTML с помощью внутреннего DSL Создание структурированных API: лямбда-выражения с получателями в DSL Гибкое вложение блоков с использованием соглашения «invoke» Предметно-ориентированные языки Kotlin на практик Сборка кода на Kotlin с помощью Gradle Сборка проектов на Kotlin с помощью Maven Сериализация JSON Клиенты HTTP Доступ к базам данных		

2.3.2. Занятия семинарского типа

Занятия семинарского типа – не предусмотрены.

2.3.3. Лабораторные занятия

№ работы	№ раздела дисциплины	**Наименование лабораторных работ**	Форма текущего контроля
1	1	Введение в пролог, знакомство с логическим подходом на основе создания древовидных структур данных	ЛР1
2	1	Разработка рекурсивных предикатов для реализации числовых алгоритмов, работы со списками Черча, логических задач	ЛР2
3	1	Применение пролога для решения переборных задач комбинаторики и теории графов	ЛР3
4	1	Защита индивидуальных заданий	-
5	1	Контрольная работа № 1. Написание предикатов	КР1
6	2	Чистое лямбда исчисление.	-
7	2	Комбинаторы.	ЛР4
8	2	Реализация арифметических и логических функций в чистом λ-исчислении	-
9	2	Контрольная работа № 2. теоретические основы лямбда исчисления и систем формального вывода	КР2
10	3	Написание функций высших порядков в kotlin	ЛР5
11	3	Коллекции и высшие функции работы с ними	ЛР6
12	3	Анализ десериализованных данных с помощью механизмов функционального ООП	ЛР7
13	3	Защита индивидуальных заданий	-
14	3	Защита индивидуальных заданий	-
15	3	Защита индивидуальных заданий	-
16	3	Контрольная работа № 3. Написание функций высших порядков	КР3

2.3.3 Примерная тематика курсовых работ (проектов)

Учебным планом не предусмотрены.

2.3.4 Расчетно-графические задания

Учебным планом не предусмотрены.

2.4 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

№	Вид СРС	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
1	2	3
1	**Раздел 1.** Основы логического программирования	Источники основной и дополнительной литературы, ЭП, логическая часть
2	**Раздел 2.** Лямбда-исчисление.	Источники основной и дополнительной литературы
3	**Раздел 3.** Основы функционального программирования.	Источники основной и дополнительной литературы, ЭП, функциональный модуль

Требования к ЭП:

Олимпиадные задания.

Необходимо решить 5 олимпиадных задач.
Для каждой задачи необходимо:

FR<№задачи>.1. составить эффективный алгоритм с выполнением требований по времени и памяти на языке java;

FR<№задачи>.2. составить подробное описание решения;

FR<№задачи>.3. подготовить 10 нетривиальных тестов с конкретными исходными данными;

FR<№задачи>.4. провести тестирование;

FR<№задачи>.5. записать результаты тестирования, указав время работы и использованную память, ответы должны совпасть с тест кейсами;

FR<№задачи>.6. составить решение в логическом стиле на языке prolog;

FR<№задачи>.7. составить подробное описание решения;

FR<№задачи>.8. провести тестирование на подготовленных данных, ответы должны совпасть с тест кейсами;

FR<№задачи>.9. составить решение полностью в функциональном стиле без использования циклов и переменных на языке Kotlin;

FR<№задачи>.10. составить подробное описание решения;

FR<№задачи>.11. провести тестирование на подготовленных данных, ответы должны совпасть с тест кейсами.

Нефункциональные требования – решения должны быть на общем для команды репозитории, тесты там же, должна быть видна история коммитов, задача в один коммит не принимается.

Требования по оцениванию каждого из параметров:

- логическая часть,
- императивная часть,
- функциональная часть,
- интеграция в готовое решение.

Императивная часть

На оценку + необходимо выполнить требования:

FR1.1.
FR1.3.
FR1.4.
FR2.1.
FR2.3.
FR2.4.
FR2.5.

На оценку Удовлетворительно необходимо выполнить требования:

требования на +
FR3.1.
FR3.3.
FR3.4.

На оценку Хорошо необходимо выполнить требования:

требования на удовлетворительно
FR4.1.
FR4.3.
FR4.4.

На оценку Отлично необходимо выполнить требования:

требования на хорошо
FR5.1.
FR5.3.
FR5.4.

Логическая часть.

На оценку + необходимо выполнить требования:

требования на + императивной части
FR1.6.
FR1.8.
FR2.6.
FR2.8.

На оценку Удовлетворительно необходимо выполнить требования:

требования на + логической части
требования на Удовлетворительно императивной части
FR3.6.
FR3.8.

На оценку Хорошо необходимо выполнить требования:

требования на удовлетворительно логической части
требования на хорошо императивной части
FR4.6.
FR4.8.

На оценку Отлично необходимо выполнить требования:
требования на хорошо логической части
требования на отлично императивной части
FR5.6.
FR5.8.

Функциональная часть.

На оценку + необходимо выполнить требования:
требования на + императивной части
FR1.9.
FR1.11.
FR2.9.
FR2.11.

На оценку Удовлетворительно необходимо выполнить требования:
требования на + функциональной части
требования на Удовлетворительно императивной части
FR3.9.
FR3.11.

На оценку Хорошо необходимо выполнить требования:
требования на удовлетворительно функциональной части
требования на хорошо императивной части
FR4.9.
FR4.11.

На оценку Отлично необходимо выполнить требования:
требования на хорошо функциональной части
требования на отлично императивной части
FR5.9.
FR5.11.

Интеграционная часть.

Для олимпиадных задач интеграционная часть заключается в подготовке подробных описаний решений с объяснениями и структурой программы

После того, как задача корректно решена в любом стиле, можно оформлять документацию.

Документация должна быть согласована с комиссией преподавателей дисциплины.

Документация оформляется в week, отметка о согласовании получается там же.

Всего необходимо выполнить 20 требований:

FR1.2.
FR1.5.
FR1.7.
FR1.10.

FR2.2.
FR2.5.
FR2.7.
FR2.10.
FR3.2.
FR3.5.
FR3.7.
FR3.10.
FR4.2.
FR4.5.
FR4.7.
FR4.10.
FR5.2.
FR5.5.
FR5.7.
FR5.10.
Оценка + выставляется в случае выполненных 8 требований
Оценка 3 выставляется в случае выполненных 12 требований
Оценка 4 выставляется в случае выполненных 16 требований
Оценка 5 выставляется в случае выполненных 20 требований

Требования к докладу.
Время на доклад 15 минут.
Во время доклада отобразить – гит с историей выполнения, согласованную документацию, постановку решенных задач, несколько нетривиальных тест кейсов, краткое пояснение задумки в решении.
Время на вопросы к докладу – 5 минут.

***Акинатор*:**

Требования к акинатору.

Должна быть выбрана конкретная область для объектов, количество объектов не менее 150 штук, количество вопросов – не менее 8.

FR1.1. Подготовлено приложение Prolog Акинатор, вызываемый из терминала Prolog, позволяющий угадывать персонажа и добавлять персонажа, если не существует объекта, имеющего полученную последовательность ответов на вопросы. Реализована возможность выдачи ответа пользователю в случае неполного ответа на вопросы, например, если лишь объект X имеет текущую картину ответа на 5 вопросов, значит 6 и 7 вопрос задавать необязательно. Реализована возможность разных вопросов для разных объектов, например, 6 вопросов одинаковых, но, чтобы отличить объект 18 от объекта19 нужен вопрос 7, а чтобы отличить объект20 от объекта21, нужен вопрос 8, а остальные объекты отличаются друг от друга на основании первых 6 вопросов. Структура должна быть подобрана исходя из предметной области.

FR1.2. Продумать и построить http сервис,

- получающий ответ на текущий вопрос и возвращающий следующий вопрос или объект или результат об отсутствии объекта;
- добавляющий новый объект в случае отсутствия такового.

FR1.3. Модифицировать сервис, реализовав следующую возможность:

пусть есть два объекта X1 и X2, имеющих одинаковые ответы на все заданные вопросы, и пусть X1 уже в базе а X2 – нет. Если пользователь загадал X2, дать ему возможность добавить X2 в базу, дать ему возможность добавить вопрос, по которому можно отличить X1 от X2 с вариантами ответа.

FR1.4. Придумать и реализовать дополнительную фичу в rest сервисе.

FR2.1. Построить БД, содержащую информацию о пользователях и их сеансах игры в акинатор, в частности – загаданные персонажи, ответы на вопросы, добавленные персонажи.

FR2.2. Построить Rest сервис java spring, позволяющий добавлять пользователей, информацию о сеансах игры, включая – дату время игры, загаданных персонажей, ответы на вопросы, добавленных персонажей.

FR2.3. Модифицировать сервис и БД, добавив возможность добавлять вопросы и ответы на вопросы, сохраняя информацию о пользователе, добавившем вопрос и ответ, согласовав данную модификацию с требованиями FR1.3.

FR2.4. Поддержать сохранение информации о функционале, реализованном в FR1.4.

FR3.1. Построить два содержательных имеющих смысл запроса к БД из FR2.1 на языке kotlin с использованием функционального интерфейса о работе пользователей с акинатором.

FR3.2. Построить rest сервис с использованием kotlin spring, позволяющий выполнять построенные ранее два запроса.

FR3.3. Построить два дополнительных нетривиальных запроса с использованием информации из FR2.3.

FR3.4. Построить дополнительный запрос с использованием информации из FR2.4.

FR4.1. Построить Java swing desktop приложение, реализующее игру в

акинатор, с использованием функционала, выполненного в FR1.1, FR2.1, FR3.1. Обязательна авторизация пользователя и сохранение всей информации о сеансе игры. Важно, любая собираемая и отображаемая статистика должна быть результатами ваших запросов из FR3.1, запросы – отдельно, отображаемая статистика – отдельно – такая ситуация запрещена.

FR4.2. Построить web приложение, реализующее игру в акинатор с использованием http сервисов из FR1.2, FR2.2, FR3.2. Обязательна авторизация пользователя и сохранение всей информации о сеансе игры. Важно, любая собираемая и отображаемая статистика должна быть результатами ваших запросов из FR3.1, запросы – отдельно, отображаемая статистика – отдельно – такая ситуация запрещена.

FR4.3. Реализовать возможность добавления вопросов в соответствии с FR1.3., FR2.3., FR3.3.

FR4.4. Реализовать фичу после FR1.4, FR2.4, FR3.4

FR4.5. Отобразить полное дерево предметной области с ответами на вопросы и объектами (или сохраняя изображение с крупным разрешением и возможностью масштабирования или отображая рисунок в приложении с возможностью масштабирования и пролистывания в масштабе).

Требования к ведению проекта.

Все сервисы должны быть подняты на ресурсах лаборатории КубГУ.

Все изменения сохраняются только на гитлабе лаборатории КубГУ. Реализация любого требования – больше одного коммита. У любого участника команды – больше одного коммита. На http сервисы прописаны автотесты и проведены автотесты, сохранены отчёты. На каждый сервис подготовлен swagger и доступен на ресурсах лаборатории КубГУ.

Логическая часть

На оценку + необходимо выполнить требования:

FR1.1.

На оценку Удовлетворительно необходимо выполнить требования:

FR1.2.

На оценку Хорошо необходимо выполнить требования:

требования на удовлетворительно логической части

или FR1.3. или FR1.4.

На оценку Отлично необходимо выполнить требования:

требования на удовлетворительно логической части

FR1.3.

FR1.4.

Императивная часть

На оценку + необходимо выполнить требования:

требования на + логической части

FR2.1.

На оценку Удовлетворительно необходимо выполнить требования:

требования на удовлетворительно логической части

FR2.2.

На оценку Хорошо необходимо выполнить требования:

требования на хорошо логической части

требования на удовлетворительно императивной части

или FR2.3. или FR2.4.

На оценку Отлично необходимо выполнить требования:

требования на отлично логической части

требования на удовлетворительно императивной части

FR2.3.

FR3.4.

Функциональная часть

На оценку + необходимо выполнить требования:

требования на + императивной части

FR3.1.

На оценку Удовлетворительно необходимо выполнить требования:

требования на удовлетворительно императивной части

FR3.2.

На оценку Хорошо необходимо выполнить требования:

требования на хорошо императивной части

требования на удовлетворительно функциональной части

или FR3.3. или FR3.4.

На оценку Отлично необходимо выполнить требования:

требования на отлично императивной части

требования на удовлетворительно функциональной части

FR3.3.

FR3.4.

Интеграционная часть

На оценку + необходимо выполнить

требования на + логической, императивной и функциональной части

FR4.1

На оценку Удовлетворительно необходимо выполнить

требования на удовлетворительно логической, императивной и функциональной части

FR4.2

Или

На оценку Удовлетворительно необходимо выполнить

требования на + интеграционной части

FR4.5

На оценку Хорошо необходимо выполнить

требования на хорошо логической, императивной и функциональной части

FR4.3 или FR4.4.

или

На оценку Хорошо необходимо выполнить
требования на удовлетворительно интеграционной части
FR4.5

На оценку Отлично необходимо выполнить
требования на отлично логической, императивной и функциональной части
FR4.3
FR4.4

Или
На оценку Отлично необходимо выполнить
требования на хорошо интеграционной части
FR4.5

Требования к докладу

Время на доклад 15 минут.
Во время доклада отобразить – гит с историей выполнения, продемонстрировать работу ПО, продемонстрировать реализацию каждого из требований.
Прислать перед выступлением ссылки на все сервисы, web приложения и swagger каждого сервиса.
Время на вопросы к докладу – 5 минут.

***Логическая игра*:**

Нельзя – крестики нолики 3 на 3.

FR1.1. Подготовлено приложение Prolog, вызываемое из терминала Prolog, позволяющее играть в игру с компьютером.

FR1.2. Модифицировать приложение, реализовав возможность установить 2 уровня сложности.

FR1.3. Продумать и построить http сервис:

- выполняющий действие – сделай ход;
- проверяющий ситуацию на победу игрока или бота.

FR1.4. Модифицировать сервис, реализовав возможность установить 2 уровня сложности.

FR1.5. Модифицировать сервис, реализовав возможность установить 4 уровня сложности.

FR1.6. Модифицируйте сервис, придумав свою собственную фичу в ведении игры ботом.

FR1.7. Модифицируйте сервис, придумав способ определения того, кто ближе к победе по текущему состоянию игры.

FR2.1. Построить БД, содержащую информацию о пользователях и их сеансах игры, в частности – победы, поражения,

FR2.2. Модифицировать БД, сохранив информацию об уровнях сложности игры, обязательно после FR1.2.

FR2.3. Построить Rest сервис java spring, позволяющий добавлять пользователей, информацию о сеансах игры, включая – дату время игры, победы, поражения, текущий сеанс игры, получать текущий сеанс игры, обязательно после FR1.3.

FR2.4. Модифицировать сервис и БД, добавив возможность сохранять информацию об уровне сложности, обязательно после FR1.4 или FR1.5.

FR2.5. Модифицировать сервис и БД, добавив возможность сохранять и получать информацию об нескольких сеансах игры одним пользователем.

FR2.6. Модифицировать сервис и БД, реализовав фичу из 2.6.

FR2.7. Модифицировать сервис и БД, добавив возможность предварительного завершения игры и сохранения информации об этом, обязательно после FR1.7.

FR3.1. Построить два содержательных имеющих смысл запроса к БД из FR2.1 на языке kotlin с использованием функционального интерфейса о работе пользователей с игрой, только после реализованной БД из 2.1.

FR3.2. Построить rest сервис с использованием kotlin spring, позволяющий выполнять построенных ранее два запроса.

FR3.3. Построить два дополнительных нетривиальных запроса из реализованного функционала 2.4 – 2.7.

FR3.4. Построить дополнительный сложный запрос из реализованного функционала 2.4 – 2.7.

FR4.1. Построить Java swing desktop приложение, реализующее игру, с использованием функционала, выполненного в FR1.1, FR2.1, FR3.1. Обязательна

авторизация пользователя и сохранение всей информации о сеансе игры. Важно, любая собираемая и отображаемая статистика должна быть результатами ваших запросов из FR3.1, запросы – отдельно, отображаемая статистика – отдельно – такая ситуация запрещена.

FR4.2. Построить Java swing desktop приложение, реализующее игру, с использованием функционала, выполненного в FR1.2, FR2.2, FR3.1. Обязательна авторизация пользователя и сохранение всей информации о сеансе игры. Важно, любая собираемая и отображаемая статистика должна быть результатами ваших запросов из FR3.1, запросы – отдельно, отображаемая статистика – отдельно – такая ситуация запрещена.

FR4.3. Построить web приложение, реализующее игру в с использованием http сервисов из FR1.3, FR2.3, FR3.2. Обязательна авторизация пользователя и сохранение всей информации о сеансе игры, сохранение и продолжение игры пользователем. Важно, любая собираемая и отображаемая статистика должна быть результатами ваших запросов из FR3.2, запросы – отдельно, отображаемая статистика – отдельно – такая ситуация запрещена.

FR4.4. Модифицировать web приложение, реализовав уровни сложности, обязательно вместе (FR1.4 или FR1.5) и (FR2.4).

FR4.5. Модифицировать web приложение, реализовав возможность ведения нескольких игр, обязательно вместе с FR2.5

FR4.6. Модифицировать web приложение, реализовав свою фичу, обязательно вместе с FR1.6 и FR2.6.

FR4.7. Модифицировать web приложение, реализовав возможность предварительного завершения игры, обязательно вместе с FR1.7 и FR2.7.

FR4.8. Отобразить результаты 3.3 в web приложении.

FR4.9. Отобразить результаты 3.4 в web приложении.

Требования к ведению проекта.

Все сервисы должны быть подняты на ресурсах лаборатории КубГУ.

Все изменения сохраняются только на гитлабе лаборатории КубГУ. Реализация любого требования – больше одного коммита. У любого участника команды – больше одного коммита. На http сервисы прописаны автотесты и проведены автотесты, сохранены отчёты. На каждый сервис подготовлен swagger и доступен на ресурсах лаборатории КубГУ.

Логическая часть

На оценку + необходимо выполнить требования:

FR1.1.

На оценку Удовлетворительно необходимо выполнить требования:

FR1.2. или FR1.3

На оценку Хорошо необходимо выполнить требования:

FR1.3

FR1.4 или FR1.6 или FR1.7

На оценку Отлично необходимо выполнить требования:

FR1.3
FR1.5

Или
На оценку Отлично необходимо выполнить требования:
FR1.3
FR1.4
FR1.6
или
FR1.3
FR1.4
FR1.7
или
FR1.3
FR1.6
FR1.7

Императивная часть
На оценку + необходимо выполнить требования:
FR1.1
FR2.1.
На оценку Удовлетворительно необходимо выполнить требования:
FR2.2.
или
На оценку Удовлетворительно необходимо выполнить требования:
FR2.3.

На оценку Хорошо необходимо выполнить требования:
FR2.3
FR2.4 или FR2.5 или FR2.6 или FR2.7

На оценку отлично необходимо выполнить
FR2.3
Любые два из требований FR2.4 или FR2.5 или FR2.6 или FR2.7

Функциональная часть
На оценку + необходимо выполнить требования:
FR3.1.
На оценку Удовлетворительно необходимо выполнить требования:
требования на удовлетворительно императивной части
FR3.2.
На оценку Хорошо необходимо выполнить требования:
требования на хорошо императивной части
требования на удовлетворительно функциональной части
FR3.3.
На оценку Отлично необходимо выполнить требования:

требования на отлично императивной части
требования на хорошо функциональной части
FR3.4.

Интеграционная часть

На оценку + необходимо выполнить
требования на + логической, императивной и функциональной части
FR4.1

На оценку Удовлетворительно необходимо выполнить
требования на удовлетворительно логической, императивной части,
требования на + функциональной части
FR4.2

Или

На оценку Удовлетворительно необходимо выполнить
FR4.3

На оценку Хорошо необходимо выполнить
FR4.3
FR4.8
FR4.4 или FR4.5 или FR4.6 или FR4.7

На оценку Отлично необходимо выполнить
FR4.3
FR4.8
FR4.9
Любые два из требований FR2.4 или FR2.5 или FR2.6 или FR2.7

Требования к докладу

Время на доклад 15 минут.
Во время доклада отобразить – гит с историей выполнения, продемонстрировать работу ПО, продемонстрировать реализацию каждого из требований.
Прислать перед выступлением ссылки на все сервисы, web приложения и swagger каждого сервиса.
Время на вопросы к докладу – 5 минут.

БД:

Требования к БД.

FR2.1. Подготовить БД из 3-4 таблиц в 3НФ на любую предметную область, заполнено миниму по 20 записей в КАЖДОЙ таблице.

FR1.1. Реализовать prolog БД, дублирующую БД из FR1.1.

FR2.2. Построить java spring REST сервис, реализующий CRUD для каждой из таблиц.

FR1.2. Построить http prolog сервис, реализующий CRUD для каждой из таблиц.

FR3.1. Построить сложный запрос с использованием всех таблиц с применением функционального интерфейса kotlin.

FR3.2. Построить kotlin spring REST сервис, реализующий этот запрос.

FR4.1. Построить java swing приложение реализующее CRUD на 2.1 и 1.1, с возможностью выбора, где работать, в БД или прологе, и отобразить результаты 3.1.

FR4.2. Построить web приложение реализующее CRUD на 2.2 и 1.2, с возможностью выбора, где работать, в БД или прологе, и отобразить результаты 3.2.

FR4.3. Модифицировать web приложение и сервисы так, чтобы была возможность синхронизации из prolog в БД.

FR4.4. Модифицировать web приложение и сервисы так, чтобы была возможность синхронизации из БД в prolog.

FR3.3. Модифицировать kotlin spring REST сервис: Построить 2 доп запроса с использованием всех таблиц с применением функционального интерфейса kotlin.

FR3.4. Модифицировать kotlin spring REST сервис: Построить 2 доп запроса с использованием всех таблиц с применением функционального интерфейса kotlin.

FR4.6. Отобразить результаты 3.3 в web приложении.

FR4.7. Отобразить результаты 3.4 в web приложении.

Требования к ведению проекта.

Все сервисы должны быть подняты на ресурсах лаборатории КубГУ.

Все изменения сохраняются только на гитлабе лаборатории КубГУ. Реализация любого требования – больше одного коммита. У любого участника команды – больше одного коммита. На http сервисы прописаны автотесты и проведены автотесты, сохранены отчёты. На каждый сервис подготовлен swagger и доступен на ресурсах лаборатории КубГУ.

Логическая часть

На оценку + необходимо выполнить требования:

FR1.1.

На оценку Удовлетворительно необходимо выполнить требования:

FR1.2.

На оценку Хорошо необходимо выполнить требования:

FR4.3 или FR4.4

На оценку Отлично необходимо выполнить требования:
FR4.3
FR4.4

Императивная часть

На оценку + необходимо выполнить требования:
FR2.1.
На оценку Удовлетворительно необходимо выполнить требования:
FR2.2.
На оценку Хорошо необходимо выполнить требования:
FR4.3 или FR4.4

На оценку Отлично необходимо выполнить требования:
FR4.3
FR4.4

Функциональная часть

На оценку + необходимо выполнить требования:
FR3.1.
На оценку Удовлетворительно необходимо выполнить требования:
FR3.2.
На оценку Хорошо необходимо выполнить требования:
FR3.3.
На оценку Отлично необходимо выполнить требования:
FR3.4.

Интеграционная часть

На оценку + необходимо выполнить
FR4.1

На оценку Удовлетворительно необходимо выполнить
FR4.2

На оценку Хорошо необходимо выполнить
FR4.2
Любые два из требований FR4.3 или FR4.4 или FR4.5 или FR4.6

На оценку Отлично необходимо выполнить
FR4.2
FR4.3
FR4.4.
FR4.5.
FR4.6

Требования к докладу

Время на доклад 15 минут.

Во время доклада отобразить – гит с историей выполнения, продемонстрировать работу ПО, продемонстрировать реализацию каждого из требований.

Прислать перед выступлением ссылки на все сервисы, web приложения и swagger каждого сервиса.

Время на вопросы к докладу – 5 минут.

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

Для лиц с нарушениями зрения:

– в печатной форме увеличенным шрифтом,
– в форме электронного документа, Для лиц с нарушениями слуха:
– в печатной форме,
– в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

– в печатной форме,
– в форме электронного документа,

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

3. Образовательные технологии

Семестр	Вид занятия (Л, ПР, ЛР)	Используемые интерактивные образовательные технологии	Количество часов
6	Л	Компьютерные презентации и обсуждение	32
	ЛР	Разбор конкретных ситуаций (задач), тренинги по решению задач, компьютерные симуляции (программирование алгоритмов)	32
	КСР	Контрольная работа	8
Итого:			72

4. Оценочные средства для текущего контроля успеваемости и промежуточной аттестации

4.1 Фонд оценочных средств для проведения текущего контроля

Фонд оценочных средств дисциплины состоит из средств текущего контроля выполнения задач на лабораторных работах, контрольных работ и средств итоговой аттестации (экзамен в 6 семестре).

Контрольная работа № 1
по дисциплине
«Функциональное и логическое программирование»
направления 02.03.02 ФИИТ ФКТиПМ
на тему «Построение предикатов на языке SWI-Prolog»
Вариант № 1.

- Общая структура фактов в прологе. Составные факты.

- На примере числовых алгоритмов объясните смысл рекурсии вверх и рекурсии вниз в прологе.
- Построить предикат maxList(+List, Ind), определяющий индекс элемента списка, имеющего максимальную сумму цифр. (Воспользоваться рекурсией вверх или вниз).
- Построить предикат comb(+List,K,-Sochet), записывающий в Sochet все возможные сочетания по K элементов. (Возможна формулировка для любого комбинаторного объекта, размещения, перестановки, подмножества, с размещениями или без).
- Построить предикат, который выводит на экран все слова длины 6 над алфавитом [a,b,c,d,e,f], в которых три буквы a, две буквы b.
- Граф задан списком вершин и списком ребер, где каждое ребро – список двух вершин. [a,b,c,d,e], [[a,b],[a,e],[b,c],[c,d],[d,e],[c,e],[a,c]].

  Зная, что таким образом задан ОРИЕНТИРОВАННЫЙ граф, найти эйлеров цикл. (или любые задачи на графы из лекции)

Контрольная работа № 1
по дисциплине
«Функциональное и логическое программирование»
направления 02.03.02 ФИИТ ФКТиПМ
на тему «Построение предикатов на языке SWI-Prolog»
Вариант № 2.

1. Бектрекинг. Маркеры. Оператор отсечения.
2. Опишите принцип работы со списками Черча в Swi-prolog. Покажите реализации встроенных предикатов работы со списками на основе механизма унификации(append, reverse nth0).
3. Построить предикат fib(+N,?A), проверяющий является ли число A числом Фибоначчи с номером N или находящий число с таким номером и записывающим его в A. (любой числовой алгоритм с применением рекурсии вверх или вниз).
4. Построить предикат razm(+List,K,-Razm), записывающий в Razm все возможные размещения по K элементов без повторений. Построить алгоритм на языке Java, выводящий на экран все размещения заданного списка по k элементов без повторений – (в случае, если задача 3 – простая, в задаче 4 включается требование реализации одной и той же задачи в императивном и логическом стиле).
5. Построить предикат, который выводит на экран все слова длины 6, в которых первые три буквы любые из [a,b,c,d,e] без повторов, остальные буквы [v,w,x,y,z] возможно с повторами.
6. Граф задан списком вершин и списком ребер, где каждое ребро – список двух вершин. [a,b,c,d,e], [[a,b],[a,e],[b,c],[c,d],[d,e],[c,e],[a,c]].

   Зная, что таким образом задан НЕОРИЕНТИРОВАННЫЙ граф, найти гамильтонов цикл. (или любые задачи на графы из лекции).

Контрольная работа № 2
по дисциплине
«Функциональное и логическое программирование»
направления 02.03.02 ФИИТ ФКТиПМ
на тему «Лямбда исчисление»
Вариант № 0.

1. Дайте определение редукции лямбда термов
2. Типизация по Чёрчу.
3. Редуцировать терм аппликативным и нормальным порядком

$$(\lambda xuz.x(\lambda x.zu))((\lambda x.u(xx))(\lambda x.u(xx)))$$

4. Проверить, что данный терм является комбинатором неподвижной точки

$$\mathbf{SI}(\mathbf{SII}(\mathbf{B}(\mathbf{SI})(\mathbf{SII})))$$

5. Составить комбинаторы, позволяющие реализовать следующие функции

   1. $x_1 \vee x_2 \vee x_3$;
   2. $x_1x_2 \vee x_1x_3 \vee x_2x_3$;

3. $n-2$;
4. n^3;

6. Построить комбинатор F, обладающий следующим свойством для произвольных λ-термов X, Y, Z:

$$FXYZ = F(FX)(FXY)(FXYZ)$$

Контрольная работа № 2
по дисциплине
«Функциональное и логическое программирование»
направления 02.03.02 ФИИТ ФКТиПМ
на тему «Лямбда исчисление»
Вариант № 0.

7. Сформулируйте и докажите лемму о комбинаторах I, K, S
8. Let выражения
9. Дан терм, найти свободные и связанные переменные.

$$\big((\lambda\, u.\, (\lambda\, x.\, uxx)(\lambda\, x.\, uxy)\big)(\lambda\, y.\, y)$$

10. Редуцировать терм аппликативным и нормальным порядком

$$\mathbf{CI(SB)\big(SB(KI)\big)\,\big(SB\big(SB(KI)\big)\big)}$$

11. Проверить, что данный терм является комбинатором неподвижной точки

$$(\lambda\, xyz.\, y(xyy))\,(\lambda\, xyz.\, y(xyy))\,(\lambda\, xyz.\, y(xyy))$$

12. Редуцировать термы: Add 5 2; Inc 3; Dec 3; Mult 4 2; Or (and true false) (and true true).

Контрольная работа № 3
по дисциплине
«Функциональное и логическое программирование»
направления 02.03.02 ФИИТ ФКТиПМ
на тему «Построение рекурсивных функций и функций высших порядков»
Вариант № 0.

13. Объясните смысл термина функции высших порядков. Приведите примеры.
14. Ассоциативные массивы. Создания и принципы работы.
15. Реализовать функцию, вычисляющую количество делителей числа n с помощью рекурсии вверх или вних
16. Построить функцию, которая принимает три аргумента
    - список
    - предикат p : int -> bool
    - функцию f : int -> int

    И возвращает список, состоящий из значений f от тех элементов, которые удовлетворяют предикату p.
17. Реализовать с помощью функции из задания 4 функцию, которая принимает список и возвращает новый список, состоящий из количеств делителей простых положительных элементов списка.
18. Для введенного списка построить новый список, который получен из начального упорядочиванием по количеству встречаемости элемента,

    То есть из списка [5,6,2,2,3,3,3,5,5,5] необходимо получить список [5,5,5,5,3,3,3,2,2,6].

4.2 Фонд оценочных средств для проведения промежуточной аттестации

Лямбда-исчисление

1. Опишите основные принципы декларативного программирования и его отличия от императивного. Опишите основные принципы функционального программирования и вытекающие из них преимущества и недостатки
2. Опишите понятия высшая функция, чистая функция, каррирование и ленивые вычисления. Опишите математические предположения, которые привели к лямбда исчислению и объясните формат записи лямбда выражений.
3. Дайте определение лямбда терма. Сформулируйте понятия абстракции и аппликации. Опишите соглашения о возможности опускать скобки, принятые в лямбда выражении. На примере лямбда термов опишите принцип каррирования и его работы.
4. Дайте определения свободных и связанных переменных в лямбда термах.
5. Дайте определения редукции лямбда термов. Опишите стратегии редукции лямбда термов.
6. Дайте понятия подстановки и преобразования. Сформулируйте понятия эквивалентности.
7. Сформулируйте теорему Черча-Россера, сформулируйте и докажите следствия из теоремы Черча-Россера.
8. Сформулируйте и докажите лемму о комбинаторах I, K, S, Докажите, что любой терм представим в виде комбинаторов S K
9. Понятие и виды комбинаторов. Редукция комбинаторов. Приведите примеры, покажите связь с лямбда исчислением.
10. Числа Черча. Операция плюс 1. Арифметические операции над числами Черча + * ^, -1.
11. Булевы константы и оператор if, Реализация булевых операций.
12. Кортежи. Каррирование. Их связь в лямбда-исчислении.
13. Комбинатор неподвижной точки. Рекурсивные функции (на примере любой функции).
14. Let выражения. Реализация списков Черча.
15. Полнота лямбда исчисления по Тьюрингу
16. Типизация по Черчу.
17. Типизация по Карри. Полиморфизм в типизации по Карри
18. Сформулируйте и докажите леммы и теорему о сохранении типов в типизации по Карри.
19. Сформулируйте недостатки математических моделей типизации лямбда по Карри и Черчу. Сформулируйте теорему о сильной нормализации
20. Покажите способ реализации типизации для генераторов неподвижной точки и рекурсии.

Типовые формулировки задач

№ 1. Дан терм, представленный в виде аппликации и абстракции лямбда-термов. Определить свободные и связанные переменные, провести редукцию с помощью двух возможных стратегий.

№ 2. Дан терм, представленный в виде аппликации комбинаторов, провести редукцию двумя способами.

№ 3. Дан терм, проверить, является ли он комбинатором неподвижной точки.

№ 4. Составить комбинаторы, позволяющие реализовать указанные функции (числовые или алгебры логики)

№ 5. Провести редукцию арифметических и логических выражений.

Функциональное программирование на примере языка kotlin.

1. Знакомство с Kotlin. Язык. Философия и инструментарий.
2. Основные элементы: переменные и функции. Классы и свойства.
3. Представление и обработка выбора: перечисления и конструкция «when»

4. Итерации: циклы «while» и «for»
5. Исключения в Kotlin
6. Создание коллекций в Kotlin
7. Определение и вызов функций
8. Упрощение вызова функций
9. Добавление методов в сторонние классы: функции-расширения и свойства-расширения
10. Работа с коллекциями: переменное число аргументов, инфиксная форма записи вызова и поддержка в библиотеке
11. Работа со строками и регулярными выражениями
12. Классы, объекты и интерфейсы локальные функции и расширения
13. Создание иерархий классов
14. Объявление классов с нетривиальными конструкторами или свойствами
15. Методы, сгенерированные компилятором: классы данных и делегирование
16. Лямбда-выражения
17. Совместное объявление класса и экземпляра
18. Лямбда-выражения Функциональный API для работы с коллекциями
19. Последовательности
20. Система типов
21. Лямбда-выражения с получателями
22. Значение null Базовые типы
23. Массивы и коллекции Перегрузка операторов Перегрузка арифметических операторов
24. Перегрузка операторов сравнения
25. Коллекции и диапазоны
26. Мультидекларации и функции
27. Component
28. Функции высшего порядка
29. Делегирование свойств
30. Объявление функций высших порядков
31. Встраиваемые функции
32. Порядок выполнения функций высших порядков
33. Обобщенные типы
34. Параметры обобщенных типов
35. Стирание и овеществление параметров типов
36. Обобщенные типы и подтипы
37. Объявление и применение аннотаций
38. Рефлексия: интроспекция объектов Kotlin во время выполнения
39. Понятие предметно-ориентированного языка
40. Внутренние предметно-ориентированные языки
41. Структура предметно-ориентированных языков
42. Создание разметки HTML с помощью внутреннего DSL
43. Создание структурированных API: лямбда-выражения с получателями в DSL
44. Гибкое вложение блоков с использованием соглашения «invoke»
45. Предметно-ориентированные языки Kotlin на практик
46. Сборка кода на Kotlin с помощью Gradle
47. Сборка проектов на Kotlin с помощью
48. Maven Сериализация JSON
49. Клиенты HTTP
50. Доступ к базам данных

Типовые формулировки задач

1. В файле приведён фрагмент базы данных «Продукты» о поставках товаров в магазины районов города. База данных состоит из трёх таблиц.

Таблица «Движение товаров» содержит записи о поставках товаров в магазины в течение первой декады июня 202 г., а также информацию о проданных товарах. Поле *Тип*

операции содержит значение *Поступление* или *Продажа*, а в соответствующее поле *Количество упаковок, шт.* занесена информация о том, сколько упаковок товара поступило в магазин или было продано в течение дня. Заголовок таблицы имеет следующий вид.

ID операции	Дата	ID магазина	Артикул	Тип операции	Количество упаковок, шт.	Цена, руб./шт.

Таблица «Товар» содержит информацию об основных характеристиках каждого товара. Заголовок таблицы имеет следующий вид.

Артикул	Отдел	Наименование	Ед. изм.	Количество в упаковке	Поставщик

Таблица «Магазин» содержит информацию о местонахождении магазинов. Заголовок таблицы имеет следующий вид.

ID магазина	Район	Адрес

На рисунке приведена схема указанной базы данных.

Прочитайте информацию из Excel файла и определите на сколько увеличилось количество упаковок яиц диетических, имеющихся в наличии в магазинах Заречного района за период с 1 по 10 июня.

Для записи каждой таблицы построить класс, выбрать коллекцию для работы, решить задачу с помощью применения функциональных интерфейсов встроенных функций выбранной коллекции.

№ 2. Откройте файл электронной таблицы, содержащей в каждой строке пять натуральных чисел. Определите количество строк таблицы, в которых квадрат суммы максимального и минимального чисел в строке больше суммы квадратов трёх оставшихся.

№ 3. Построить обобщенный класс сортированное двоичное дерево. Построить конструктор класса, принимающий список. Построить метод, возвращающий отсортированный список. Построить методы, добавляющие элементы в дерево.

№ 4. Реализовать функцию, которая по трем спискам составляет список, состоящий из кортежей длины 3, где каждый кортеж (ai,bi,ci) с номером I получен следующим образом:
Ai – I по убыванию элемент первого списка
Bi – I по возрастанию суммы цифр элемент второго списка
Ci - I по убыванию количества делителей элемент третьего списка
Все элементы одного списка различны.
Если в списках B или C два элемента имеют одинаковый коэффициент, большим считается элемент, больший по абсолютной величине.

Решить данные задачи с применением возможностей класса list

№ 5. Дан файл, вывести в отдельный файл строки, состоящие из слов, не повторяющихся в исходном файле.

Логическое программирование на примере языка Swi-Prolog

1. Опишите понятие и структуру фактов в языке Пролог. Расскройте основные возможные типы, опишите понятие атом. Расскажите принцип работы терминальной машины Swi-Prolog, объясните каким образом происходит обработка вопросов.
2. Опишите смысл термина унификация, приведите показательные примеры. Объясните, как задаются предикаты, что такое правила и каким образом происходит работа с ними.
3. На примере числовых алгоритмов объясните смысл рекурсии вверх и рекурсии вниз в прологе.
4. Раскройте на примерах понятие backtracking, оператор отсечения и смысл его применения.
5. Опишите принцип работы со списками Черча в Swi-prolog. Покажите реализации встроенных предикатов работы со списками на основе механизма унификации(append, reverse nth0).
6. Объясните принцип работы со строками. Покажите на примерах основные встроенные предикаты работы со строками.
7. Покажите, каким образом происходит построение стандартных комбинаторных объектов средствами Swi-prolog.
8. Покажите основные принципы реализации переборных алгоритмов на графах средствами пролога.
9. Раскройте понятия статические и динамические факты. Поясните на примерах принцип работы с динамическими фактами.
10. Объясните принцип работы предикатов var, nonvar, atom, atomic, name, functor, arg, repeat.

Типовые формулировки задач

1. Построить предикат, позволяющий получить подмножество исходного множества. На его основе решить задачу компоновки рюкзака минимального веса. Дан рюкзак с заданным объемом V. Дан набор объектов [[3,4], [5,6], [7,8], [4,5]], где первое число объем объекта, второе число вес объекта. Сначала решить задачу нахождения подмножества объектов, сумма объемов которых равна объему рюкзака. Далее найти такое подмножество объектов, сумма объектов которых равна объему рюкзака, а сумма весов объекта минимальна возможна.
2. Построить предикат, позволяющий получить сочетание по k элементов из исходного списка. Далее построить предикат, позволяющий построить размещение без повторений по m элементов. На основе построенных предикатов построить все слова длины 10, в которых 3 буквы a, еще одна буква встречается 3 раза, остальные буквы не повторяются. В алфавите 6 букв [abcdef]. Рассчитать количество таких слов, проверить, совпадает ли количество слов в итоговом файле с рассчетным.
3. Написать предикат, который по заданному графу возвращает произвольное максимальное паросочетание. Модифицировать предикат так, чтобы он выявлял наибольшее паросочетание.
4. Дан файл, вывести в отдельный файл строки, состоящие из слов, не повторяющихся в исходном файле.
5. Номер 3797 обладает интересным свойством. Будучи простым, можно непрерывно удалять цифры слева направо и оставаться простыми на каждом этапе: 3797, 797, 97 и 7. Аналогично мы можем работать справа налево: 3797, 379, 37 и 3.

Найдите сумму простых чисел, меньших 1000000 которые можно обрезать слева направо и справа налево.

ПРИМЕЧАНИЕ. 2, 3, 5 и 7 не считаются усеченными простыми числами.

Критерии оценивания к экзамену

Оценка «отлично»: точные формулировки теорем и правильные объяснения принципов работы технологий программирования; точные решения поставленных задач разработки в соответсвии с парадигмой, точное выявление места применимости парадигм.

Оценка «хорошо»: при ответе на один вопрос даны точные формулировки теорем и правильные объяснения принципов работы технологий программирования; точные решения поставленных задач разработки в соответсвии с парадигмой, точное выявление места применимости парадигм; при ответе на второй вопрос имеются неточности формулировки теорем или пробелы в объяснении принципов работы технологий программирования; точное решение задач, но не оптимальное с точки зрения выбранной парадигмы.

Оценка «удовлетворительно»: при ответе на оба вопроса имеются неточности формулировки теорем или пробелы в объяснении принципов работы технологий программирования; задача решена правильно, но необходимы улучшения структуры кода.

Оценка «неудовлетворительно»: отсутствует ответ хотя бы на один из вопросов или имеются существенные неточности формулировки теорем или в объяснении принципов работы технологий программирования, задача не решена или решена неверно.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

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

Для лиц с нарушениями зрения:

– в печатной форме увеличенным шрифтом,

– в форме электронного

документа.Для лиц с

нарушениями слуха:

– в печатной форме,

– в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

– в печатной форме,

– в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

5. Перечень основной и дополнительной учебной литературы,

необходимой для освоения дисциплины

5.1 Основная литература:

1. Рубио-Санчес, М. Введение в рекурсивное программирование : учебное пособие : [16+] / М. Рубио-Санчес ; пер. с англ. Е. В. Борисова. – Москва : ДМК Пресс, 2019. – 437 с. : ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=596896 (дата обращения: 29.05.2024). – ISBN 978-5-97060-703-9. – Текст : электронный.

2. Жемеров, Д. Kotlin в действии : практическое пособие : [16+] / Д. Жемеров, С. Исакова. – Москва : ДМК Пресс, 2018. – 402 с. : ил., табл. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=578703 (дата обращения: 29.05.2024). – ISBN 978-5-97060-497-7. – Текст : электронный.

3. Цуканова, Н. И. Технология разработки экспертных систем на языке Visual Prolog 7.5 : учебное пособие / Н. И. Цуканова, К. А. Майков. – Москва : Курс, [2023]. – 250 с. : ил., табл. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=708089 (дата обращения: 29.05.2024). – Библиогр.: с. 205-206. – ISBN 978-5-906923-40-0. – Текст : электронный

4. Боровская, Е. В. Основы искусственного интеллекта : учебное пособие : [16+] / Е. В. Боровская, Н. А. Давыдова. – 4-е изд., электрон. – Москва : Лаборатория знаний, 2020. – 130 с. : схем. – (Педагогическое образование). – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=595419 (дата обращения: 29.05.2024). – Библиогр. в кн. – ISBN 978-5-00101-908-4. – Текст : электронный

5.2 Дополнительная литература:

1.Клоксин, Уильям. Программирование на языке Пролог : / У. Клоксин, К. Меллиш ; пер.: А. В. Горбунов, М. М. Комаров ; ред. пер.: А. К. Платонов, Ю. М. Лазутин. - М. : Мир, 1987. - 336 с. : ил. - (Математическое обеспечение ЭВМ). - Предм. указ.: с. 335-336.

2. Большакова Елена Игоревна, Груздева Надежда Валерьевна Программирование на языке Пролог: Учебное пособие. – М.: Издательский отдел факультета ВМК МГУ имени М.В.Ломоносова (лицензия ИД № 05899 от 24.09.2001); МАКС Пресс, 2013 – 112 с.

3. Филд А... Функциональное программирование. (Functional Programming) [Djv-13.6M] Авторы: А. Филд, П. Харрисон. Перевод с английского М.В. Горбатовой, А.А. Рябинина, В.Л. Торхова, М.В. Федорова под редакцией В.А. Горбатова. Научное издание. (Москва: Издательство «Мир». Редакция литературы по информатике, 1993)

4. John Harrison, Introduction to Functional Programming http://www.cl.cam.ac.uk/teaching/Lectures/funprog-jrh-1996/ (русский перевод: http://code.google.com/p/funprog-ru/)

5. Малышев Д.С., Мокеев Д.Б. Некоторые элементы неклассических логик и лямбда-исчисления: учебно-методическое пособие. — [электронный ресурс] — Нижний Новгород: Нижегородский госуниверситет, 2017 — 32 с.

6. Официальный сайт языка Котлин https://kotlinlang.org/

7. Руководство по языку Kotlin https://metanit.com/kotlin/tutorial/

8. Руководство по языку Kotlin https://kotlinlang.ru/

9. Гриффитс Дон, Гриффитс Дэвид Head First. Kotlin. — СПб.: Питер, 2020. — 464 с.: ил. — (Серия «Head First O’Reilly»). ISBN 978-5-4461-1335-4.

5.3. Интернет-ресурсы, в том числе современные профессиональные базы данных и информационные справочные системы

Электронно-библиотечные системы (ЭБС):

1. ЭБС «ЮРАЙТ» https://urait.ru/
2. ЭБС «УНИВЕРСИТЕТСКАЯ БИБЛИОТЕКА ОНЛАЙН» http://www.biblioclub.ru/
3. ЭБС «BOOK.ru» https://www.book.ru
4. ЭБС «ZNANIUM.COM» www.znanium.com
5. ЭБС «ЛАНЬ» https://e.lanbook.com

Профессиональные базы данных

7. Scopus http://www.scopus.com/
8. ScienceDirect https://www.sciencedirect.com/
9. Журналы издательства Wiley https://onlinelibrary.wiley.com/
10. Научная электронная библиотека (НЭБ) http://www.elibrary.ru/
11. Полнотекстовые архивы ведущих западных научных журналов на Российской платформе научных журналов НЭИКОН http://archive.neicon.ru
12. Национальная электронная библиотека (доступ к Электронной библиотеке диссертаций Российской государственной библиотеки (РГБ) https://rusneb.ru/
13. Президентская библиотека им. Б.Н. Ельцина https://www.prlib.ru/
14. База данных CSD Кембриджского центра кристаллографических данных (CCDC) https://www.ccdc.cam.ac.uk/structures/
15. Springer Journals: https://link.springer.com/
16. Springer Journals Archive: https://link.springer.com/
17. Nature Journals: https://www.nature.com/
18. Springer Nature Protocols and Methods: https://experiments.springernature.com/sources/springer-protocols
19. Springer Materials: http://materials.springer.com/
20. Nano Database: https://nano.nature.com/
21. Springer eBooks (i.e. 2020 eBook collections): https://link.springer.com/
22. "Лекториум ТВ" http://www.lektorium.tv/
23. Университетская информационная система РОССИЯ http://uisrussia.msu.ru

Информационные справочные системы

1. Консультант Плюс - справочная правовая система (доступ по локальной сети с компьютеров библиотеки)

Ресурсы свободного доступа

1. КиберЛенинка http://cyberleninka.ru/;
2. Американская патентная база данных http://www.uspto.gov/patft/
3. Министерство науки и высшего образования Российской Федерации https://www.minobrnauki.gov.ru/;
4. Федеральный портал "Российское образование" http://www.edu.ru/;
5. Информационная система "Единое окно доступа к образовательным ресурсам" http://window.edu.ru/;
6. Единая коллекция цифровых образовательных ресурсов http://school-collection.edu.ru/ .
7. Проект Государственного института русского языка имени А.С. Пушкина "Образование на русском" https://pushkininstitute.ru/;
8. Справочно-информационный портал "Русский язык" http://gramota.ru/;
9. Служба тематических толковых словарей http://www.glossary.ru/;
10. Словари и энциклопедии http://dic.academic.ru/;
11. Образовательный портал "Учеба" http://www.ucheba.com/;
12. Законопроект "Об образовании в Российской Федерации". Вопросы и ответы http://xn--273--84d1f.xn--p1ai/voprosy_i_otvety

Собственные электронные образовательные и информационные ресурсы КубГУ

1. Электронный каталог Научной библиотеки КубГУ http://megapro.kubsu.ru/MegaPro/Web
2. Электронная библиотека трудов ученых КубГУ http://megapro.kubsu.ru/MegaPro/UserEntry?Action=ToDb&idb=6
3. Среда модульного динамического обучения http://moodle.kubsu.ru
4. База учебных планов, учебно-методических комплексов, публикаций и конференций http://infoneeds.kubsu.ru/
5. Библиотека информационных ресурсов кафедры информационных образовательных технологий http://mschool.kubsu.ru;
6. Электронный архив документов КубГУ http://docspace.kubsu.ru/
7. Электронные образовательные ресурсы кафедры информационных систем и технологий в образовании КубГУ и научно-методического журнала "ШКОЛЬНЫЕ ГОДЫ" http://icdau.kubsu.ru/

6. Методические указания для обучающихся по освоению дисциплины

По курсу предусмотрено проведение лекционных занятий, на которых дается основной систематизированный материал, лабораторных работ, контрольных работ, выполнение индивидуальных заданий зачета и экзамена.

Важнейшим этапом курса является самостоятельная работа по дисциплине с использованием указанных литературных источников и методических указаний автора курса. Стоит отметить, что в рамках самостоятельной работы происходит разработка согласно Agile методологии и выполнение спринтов к четко обозначенным срокам.

Виды и формы СР, сроки выполнения, формы контроля приведены выше в данном документе.

Для лучшего освоения дисциплины при защите ЛР студент должен ответить на несколько вопросов из лекционной части курса.

В освоении дисциплины инвалидами и лицами с ограниченными возможностями здоровья большое значение имеет индивидуальная учебная работа (консультации) – дополнительное разъяснение учебного материала.

Индивидуальные консультации по предмету являются важным фактором, способствующим индивидуализации обучения и установлению воспитательного контакта между преподавателем и обучающимся инвалидом или лицом с ограниченными возможностями здоровья.

7. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине

7.1 Перечень информационных технологий

Проверка домашних заданий и консультирование посредством системы контроля версий github и системы управления проектами week.
Использование электронных презентаций при проведении лекций и практических занятий.

7.2 Перечень необходимого программного обеспечения

19. Intellij IDEA + plugin Kotlin
20. JDK
21. SWI-prolog
22. гит-ядро

23. Excel

7.3 Перечень информационных справочных систем:

1. Электронная библиотечная система eLIBRARY.RU (http://www.elibrary.ru)/

8. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Наименование специальных помещений	Оснащенность специальных помещений	Перечень лицензионного программного обеспечения
Учебные аудитории для проведения занятий лекционного типа	Мебель: учебная мебель Технические средства обучения: экран, проектор, компьютер	PowerPoint. ауд. 129, 131, А305.
Учебные аудитории для проведения занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации	Мебель: учебная мебель Технические средства обучения: экран, проектор, компьютер	Аудитория, (кабинет) – компьютерный класс 1. Intellij IDEA + plugin Kotlin 2. JDK 3. SWI-prolog
Учебные аудитории для проведения лабораторных работ. Лаборатория…	Мебель: учебная мебель Технические средства обучения: компьютер	Лаборатория, укомплектованная специализированными техническими средствами обучения – компьютерный класс, с возможностью подключения к сети «Интернет», программой экранного увеличения и обеспеченный доступом в электронную информационно-образовательную среду университета (лаб. 102-106.).

Для самостоятельной работы обучающихся предусмотрены помещения, укомплектованные специализированной мебелью, оснащенные компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду университета.

Наименование помещений для самостоятельной работы обучающихся	Оснащенность помещений для самостоятельной работы обучающихся	Перечень лицензионного программного обеспечения
Помещение для самостоятельной работы обучающихся (читальный зал Научной библиотеки)	Мебель: учебная мебель Комплект специализированной мебели: компьютерные столы Оборудование: компьютерная техника с подключением к информационно-коммуникационной сети «Интернет» и доступом в электронную информационно-образовательную среду образовательной организации, веб-камеры, коммуникационное оборудование, обеспечивающее доступ к сети интернет (проводное соединение и беспроводное соединение по технологии Wi-Fi)	
Помещение для самостоятельной работы обучающихся (ауд._____)	Мебель: учебная мебель Комплект специализированной мебели: компьютерные столы	1. Intellij IDEA + plugin Kotlin 2. JDK 3. SWI-prolog

	Оборудование: компьютерная техника с подключением к информационно-коммуникационной сети «Интернет» и доступом в электронную информационно-образовательную среду образовательной организации, веб-камеры, коммуникационное оборудование, обеспечивающее доступ к сети интернет (проводное соединение и беспроводное соединение по технологии Wi-Fi)	4. github