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.О.36 «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 2022

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 составил(а):

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

Рабочая программа дисциплины «Функциональное и логическое программирование» утверждена на заседании кафедры Вычислительных технологий протокол № 9 «18 » мая 2022 г.

Заведующий кафедрой (разработчика) Вишняков Ю.М 

(фамилия, инициалы подпись

Утверждена на заседании учебно-методической комиссии факультета Компьютерных Технологий и Прикладной Математики
протокол № 6 от «25» мая 2022 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 обязательной части блока Б1 Дисциплины (модули) и является обязательной дисциплиной.

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». Знания, умения и навыки, полученные студентами в дисциплине «Функциональное и логическое программирование» являются обязательными для изучения следующих дисциплин «Модели интеллектуальных систем», «Верификация программных систем», «Паттерны программирования», «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:**

Код и наименование индикатора	Результаты обучения по дисциплине *(знает, умеет, владеет (навыки и/или опыт деятельности))*
ОПК-3. Способен к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям	
Формулировки индикаторов	
ОПК-3.1. Знает методы теории алгоритмов, методы системного и прикладного программирования, основные положения и концепции в области математических, информационных и имитационных моделей. ОПК-3.2. Умеет соотносить знания в области программирования, интерпретацию прочитанного, определять и создавать информационные ресурсы глобальных сетей, образовательного контента, средств тестирования систем. ОПК-3.3. Имеет практический опыт применения разработки программного обеспечения.	
ПК-2. Готовность к включению в профессиональное сообщество; способность 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 6 зач.ед. (216 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, ИЗ, КР	

		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, ИЗ, КР	
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, ИЗ, КР	

		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	ЛР2
3	1	Разработка рекурсивных предикатов работы со списками Черча	ЛР3
4	1	Применение пролога для решения логических и переборных задач	ЛР4
5	1	Защита индивидуальных заданий	ИЗ1-4
6	2	Чистое лямбда исчисление.	
7	2	Комбинаторы.	
8	2	Реализация арифметических и логических функций в чистом λ-исчислении	
9	2	теоретические основы лямбда исчисления и систем формального вывода	КР1
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	Разработка модуля работы с БД с применением функциональной ООП архитектуры	ЛР8
14	3	Защита индивидуальных заданий	ИЗ5-8
15	3	Защита индивидуальных заданий	ИЗ5-8
16	3	Написание функций высших порядков	КР2

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, ИЗ1-4
2	**Раздел 2.** Лямбда-исчисление.	Источники основной и дополнительной литературы
3	**Раздел 3.** Основы функционального программирования.	Источники основной и дополнительной литературы, ИЗ5-8

Индивидуальное задание 1. Разработка дерева семьи.

Задание 1. Построить дерево семьи в любом графическом редакторе. Составить базу данных Prolog (набор фактов) о поле всех членов семьи (к пример, man, woman). Написать несколько запросов к терминалу на проверку пола конкретных членов семьи. Построить предикаты men и women, которые выводят на экран всех мужчин и всех женщин соответственно.

Задание 2. Построить базу данных Prolog отношения «является родителем», пример доступен по ссылке выше. Провести несколько запросов к терминалу Prolog. Построить предикат, который children(X), который выводит всех детей X. Построить предикат mother(X, Y), который проверяет, является ли X матерью Y. Построить предикат, mother(X), который выводит маму X. Построить предикат brother(X, Y), который проверяет, является ли X братом Y.

Задание 3. Построить предикат brothers(X), который выводит всех братьев X. Построить предикат b_s(X,Y), который проверяет, являются ли X и Y родными братом и сестрой или братьями или сестрами. Построить предикат b_s(X), который выводит всех братьев или сестер X. Провести трассировку предикатов для трех разных запросов к терминалу.

Задание 4. Построить предикат grand_pa(X, Y), который проверяет, является ли X дедушкой Y. Дополнить базу фактов таким образом, чтобы учитывать, что у каждого ребенка 2 бабушки и 2 дедушки. Внести изменения в дерево семьи, добавить дерево в отчет. Построить предикат grand_pas(X), который выводит всех дедушек X. Построить предикат grand_pa_and_son(X,Y), который проверяет, являются ли X и Y дедушкой и внуком или внуком и дедушкой. Провести трассировку предикатов для трех разных запросов к терминалу.

Задание 5. Построить предикат, который проверяет, является ли X дядей Y. Построить предикат, который выводит всех дядей X. Провести трассировку последнего предиката для трех разных запросов к терминалу.

Задание 6. Модифицировать дерево семьи так, чтобы были учтены возможности : родителям быть в разводе, сводные братья и сестры, усыновление и удочерение.

Задание 7. Напишите Предикат, находящий всех сводных братьев и сестер введенного человека.

Задание 8. Продумать и реализовать три предиката работы с деревом семьи.

Индивидуальное задание 2. Построение комбинаторных объектов.

Задание 1. Дано множество {a,b,c,d,e,f}. Построить все слова длины 5, в которых ровно две буквы a. Вывод в файл.

Задание 2. Дано множество {a,b,c,d,e,f}. Построить все слова длины 5, в которых ровно 2 буквы a, остальные буквы не повторяются. Вывод в файл.

Задание 3. Дано множество {a,b,c,d,e,f}. Построить все слова длины 5, в которых ровно одна буква повторяется 2 раза, остальные буквы не повторяются. Вывод в файл. Дано множество {a,b,c,d,e,f}. Построить все слова длины 6, в которых ровно 2 буквы повторяются 2 раза, остальные буквы не повторяются. Вывод в файл.

Задание 4. Дано множество {a,b,c,d,e,f}. Построить все слова длины 7, в которых ровно 1 буква повторяются 2 раза, ровно одна буква повторяется 3 раза остальные буквы не повторяются. Вывод в файл.

Задание 5. Дано множество {a,b,c,d,e,f}. Построить все слова длины 9, в которых ровно 2 буквы повторяются 2 раза, ровно одна буква повторяется три раза, остальные буквы не повторяются. Вывод в файл.

Задание 6. Дано множество {a,b,c,d,e,f}. Построить все слова длины 4, в которых больше двух букв a. Вывод в файл.

Задание 7. Дано множество {a,b,c,d,e,f}. Построить все слова длины 7, в которых ровно 4 различных буквы. Минимизировать перебор*. Вывод в файл.

Задание 8. Составить предикат средствами SWI-Prolog, который составляет и выводит в файл все слова алфавита {a,b,c,d,e,f} длины n, в которых ровно две буквы повторяются по k раз, остальные буквы встречаются ровно 1 раз или не встречаются вообще.

Индивидуальное задание 3. Работа с графами.

Реализовать приложение средствами алгоритмического языка с визуальным интерфейсом общения с пользователем. Приложение обращается к пользователю, просит того выбрать одну из 5 задач, ввести исходные данные, после чего обращается к модулю, написанном средствами Swi-Prolog для решения задачи и отображает результат пользователю. При выборе задачи на графы необходимо визуально отобразить построенный пользователем граф и визуально отобразить результаты (паросочетание, максимальный поток или множество вершин).

Вариант № 1.

Дано дерево в виде матрицы смежности вершин. Построить код Прюфера.

Дан неориентированный граф, проверить, будет ли он связным.

Дан неориентированный связный граф. Найти Эйлеров цикл.

Дан неориентированный граф. Найти число внутренней устойчивости графа.

Дан неориентированный граф. Построить базу.

Индивидуальное задание 4. Разработка вопрос-ответной системы

Задание 1. Разработать структуру предметной области (персонажи книги, фильма, объекты недвижимости, виды животных, художники 19 века, политические деятели, студенты Вашего факультета, марки шампуней и т.д.). Построить список вопросов. Построить список объектов с указанием ответов на вопросы. Выделить объекты, известные системе и объекты неизвестные. В системе должно быть от 20 до 30 объектов, возможность дополнить до 40 объектов, то есть все 40 объектов должны иметь различные ответы на вопросы. Указать ответы на вопросы для Ваших объектов. Включить вопросы, объекты и ответы на вопросы в отчет. Отчет на этом завершен.

Задание 2. Разработать программу в Swi-Prolog, реализующую простейшую вопрос-ответную систему на основании анализа, выполненного в задании Приложение задает пользователю вопросы и пользователь на них отвечает, в результате чего программа формирует ответ (кто был загадан).

Задание 3*. Реализовать возможность выдачи ответа пользователю в случае неполного ответа на вопросы, например, лишь объект X имеет текущую картину ответа на 5 вопросов, значит 6 и 7 вопрос задавать необязательно.

Задание 4*. Реализовать возможность разных вопросов для разных объектов, например, 6 вопросов одинаковых, но, чтобы отличить объект 18 от объекта 19 нужен вопрос 7, а чтобы отличить объект 20 от объекта 21, нужен вопрос 8, а остальные объекты отличаются друг от друга на основании первых 6 вопросов. Структура должна быть

подобрана исходя из предметной области. Внести изменения в отчет с анализом и структурой предметной области.

Задание 5. Внести изменения, которые позволят выполнять следующее: Приложение задает пользователю вопросы и пользователь на них отвечает, в результате чего программа формирует ответ (кто был загадан). Для реализации обязательно воспользоваться механизмом динамической базы фактов. Если в базе данных нет информации о введенном объекте, программа предлагает пользователю ввести наименование объекта и сохранить его в базу фактов. При следующем запуске данный объект уже должен быть доступен для отгадывания.

Задание № 6*. Реализовать приложение средствами алгоритмического языка с визуальным интерфейсом общения с пользователем. Приложение обращается к пользователю, просит того ввести исходные данные, после чего обращается к модулю, написанном средствами Swi-Prolog для решения задачи и отображает результат пользователю, способно отобразить изображение, соответствующее отгаданному персонажу.

Задание 7. Внести изменения в базу фактов, подготовить примеры для дописывания в базу, внести коррективы в отчет. Построить презентацию. Защита происходит по презентации.

Индивидуальное задание 5.

Задание 1. «Работа с числами». Составить 3 метода для работы с цифрами или делителей числа на основании варианта на основе императивного подхода.

Вариант № 1.

Метод 1. Найти сумму простых делителей числа.

Метод 2. Найти количество нечетных цифр числа, больших 3.

Метод 3. Найти прозведение таких делителей числа, сумма цифр которых меньше, чем сумма цифр исходного числа.

Задание 2. «Работа с числами». Переписать задание 1 с использованием только хвостовой рекурсии и тела-выражения. Реализовать максимально возможное разделение на функции.

Задание 3. «Работа с числами». Модифицировать возможность пользователя выполнять одну из нескольких операций над числами, введя функцию op, возвращающую функцию с одним аргументом и одним значением.

Задание 4. Написать функцию обход делителей числа, которая выполняет операции над делителями числа, принимает три аргумента, число, функция (например, сумма, произведение, минимум, максимум) и инициализирующее значение. Функция должна иметь два Int аргумента и возвращать Int.

Задание 5. Написать функцию обход взаимно простых компонентов числа, которая выполняет операции над числами, взаимно простыми с данным, принимает три аргумента, число, функция (например, сумма, произведение, минимум, максимум, количество) и инициализирующее значение. Функция должна иметь два Int аргумента и возвращать Int.

Задание 6. Реализовать функцию обход числа с условием, которая выполняет операции над цифрами, если цифры удовлетворяют заданному условию. Аргументы функции: число, функция с двумя аргументами Int, возвращающая Int, инициализирующее заполнение, функция с одним аргументом Int, возвращающая true-false.

Задание 7. Проверить каждую из функций 4,5,6 на 3 своих примерах.

Задание 8. Переписать функции задания 1 с применением функций из 4,5,6.

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
7	Л	Компьютерные презентации и обсуждение	14
	ЛР	Разбор конкретных ситуаций (задач), тренинги по решению задач, компьютерные симуляции (программирование алгоритмов)	14
	КСР	Контрольная работа	4
Итого:			32

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 «Лямбда исчисление»
Вариант № 0.

1. Дан терм, найти свободные и связанные переменные.

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

2. Редуцировать терм аппликативным и нормальным порядком

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

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

3. Проверить, что данный терм является комбинатором неподвижной точки

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

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

Редуцировать термы: Add 5 2;
Inc 3; Dec 3
Mult 4 2;

4. Or (and true false) (and true true).
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:

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

Определение	Название
$\mathbf{I} \equiv \lambda x.x$	тождественный
$\mathbf{B} \equiv \lambda xyz.x(yz)$	композитор
$\mathbf{C} \equiv \lambda xyz.xzy$	пермутатор
$\mathbf{K} \equiv \lambda xy.x$	канцелятор
$\mathbf{S} \equiv \lambda xyz.xz(yz)$	коннектор
$\mathbf{W} \equiv \lambda xy.xyy$	дупликатор

Контрольная работа № 2
по дисциплине
«Функциональное и логическое программирование»
направления 02.03.02 ФИИТ ФКТиПМ
на тему «Построение рекурсивных функций и функций высших порядков»
Вариант № 1.

1. Реализовать функцию, вычисляющую количество делителей числа n с помощью рекурсии вверх или вних
2. 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. Реализовать с помощью этой функции функцию, которая строит список из сумм цифр положительных элементов списка. Для реализации воспользоваться хвостовой рекурсией и списками Черча.
3. 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].

Факультет компьютерных технологий и прикладной математики
Кафедра вычислительных технологий
02.03.02
Функциональное и логическое программирование
Теоретический коллоквиум на тему
Лямбда исчисление

Вариант № 1.

Вопрос 1. Сформулируйте понятия эквивалентности термов.

Вопрос 2. Числа Черча. Операция плюс 1.

Вопрос 3. Рекурсивные функции (на примере любой функции)

Контрольная работа № 3
по дисциплине
«Функциональное и логическое программирование»
направления 02.03.02 ФИИТ ФКТиПМ
на тему «Построение предикатов на языке SWI-Prolog»
Вариант № 1.

Построить предикат vz_prost(+X,+Y), проверяющий числа на взаимную простоту. (любой числовой алгоритм с применением рекурсии вверх или вниз).

1. Построить предикат max_list(+List), определяющий индекс элемента списка, имеющего максимальную сумму чисел, меньших элемента и взаимно-простых с ним. Для построения воспользоваться предикатом из предыдущей задачи. (Воспользоваться рекурсией вверх или вниз).
2. 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).
3. С помощью предиката из предыдущих задач построить предикат с одним обязательным аргументом – список, который выводит на экран все сочетания взаимно-простых пар чисел.
4. 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.
5. 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 № 3
по дисциплине
«Функциональное и логическое программирование»
направления 02.03.02 ФИИТ ФКТиПМ
на тему «Построение предикатов на языке SWI-Prolog»
Вариант № 2.

1. 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).
2. Построить предикат fib_list(+List,-Count), находящий количество таких пар в списке, что первый элемент – это номер числа Фибоначчи, а второй – число Фибоначчи.
3. Построить предикат razm(+List,K,-Razm), записывающий в Razm все возможные размещения по K элементов без повторений. (Возможна формулировка для любого комбинаторного объекта, размещения, перестановки, подмножества, с размещениями или без).
4. С помощью предиката из предыдущих задач построить предикат с одним обязательным аргументом – список, который выводит на экран все размещения по k элементов, которые содержат только элементы последовательности Фибоначчи в возрастающем порядке.
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).

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. Кубенский, А. А. Функциональное программирование : учебник и практикум для вузов / Кубенский А. А. - Москва : Юрайт, 2022. - 348 с. - URL: https://urait.ru/bcode/490015 (дата обращения: 05.05.2022). - Режим доступа: для авториз. пользователей. - ISBN 978-5-9916-9242-7. - Текст : электронный.

2. Ефимова, Е.А. Основы программирования на языке Visual Prolog / Е.А. Ефимова. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 266 с. - URL: https://biblioclub.ru/index.php?page=book&id=428996 (дата обращения: 20.04.2021). - Режим доступа: для авториз. пользователей. - Текст : электронный.

3. Жемеров Д., Исакова С. Kotlin в действии. / пер. с англ. Киселев А. Н. - М.: ДМК Пресс, 2018. - 402 с.: ил.

4. Официальная документация языка Kotlin https://github.com/JetBrains/kotlin-web-site

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

1. Scopus http://www.scopus.com/
2. ScienceDirect https://www.sciencedirect.com/
3. Журналы издательства Wiley https://onlinelibrary.wiley.com/
4. Научная электронная библиотека (НЭБ) http://www.elibrary.ru/
5. Полнотекстовые архивы ведущих западных научных журналов на Российской платформе научных журналов НЭИКОН http://archive.neicon.ru
6. Национальная электронная библиотека (доступ к Электронной библиотеке диссертаций Российской государственной библиотеки (РГБ) https://rusneb.ru/
7. Президентская библиотека им. Б.Н. Ельцина https://www.prlib.ru/
8. База данных CSD Кембриджского центра кристаллографических данных (CCDC) https://www.ccdc.cam.ac.uk/structures/
9. Springer Journals: https://link.springer.com/
10. Springer Journals Archive: https://link.springer.com/
11. Nature Journals: https://www.nature.com/
12. Springer Nature Protocols and Methods: https://experiments.springernature.com/sources/springer-protocols
13. Springer Materials: http://materials.springer.com/
14. Nano Database: https://nano.nature.com/
15. Springer eBooks (i.e. 2020 eBook collections): https://link.springer.com/
16. "Лекториум ТВ" http://www.lektorium.tv/
17. 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 электронной почты.
Использование электронных презентаций при проведении лекций и практических занятий.

7.2 Перечень необходимого программного обеспечения

1. Intellij IDEA + plugin Kotlin
2. JDK
3. SWI-prolog
4. MySQL

5. Excel
6. PowerPoint

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 4. MySQL
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 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)	1. Intellij IDEA + plugin Kotlin 2. JDK 3. SWI-prolog 4. MySQL