

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Владимирский государственный университет  
имени Александра Григорьевича и Николая Григорьевича Столетовых»  
(ВлГУ)

Институт прикладной математики, физики и информатики  
(Наименование института)



УТВЕРЖДАЮ:  
Директор института

\_\_\_\_\_ К.С. Хорьков

« 30 » 08 2021 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**Декларативное программирование**

(наименование дисциплины)

**направление подготовки / специальность**

02.03.03 Математическое обеспечение и администрирование информационных систем  
(код и наименование направления подготовки (специальности))

**направленность (профиль) подготовки**

Проектирование и защита информационных систем и баз данных  
(направленность (профиль) подготовки)

г. Владимир  
2021

## 1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Цель освоения дисциплины – изучение декларативных языков программирования и математических принципов, лежащих в основе функциональных и логических языков.

Задачи:

- изучение резолюций в языке Prolog, языка Haskell;
- формирование умений в разработке рекурсивных программ, составлении рекурсивных функций, выполнении бета-редукций;
- формирование навыков в обработке списков, использовании лямбда-исчисления, в работе с комбинаторной логикой, решении задач методом резолюций;
- освоение процесса построения декларативных программ;
- выработка навыков практического применения полученных знаний в разнообразных видах самостоятельной работы, предусмотренной программой подготовки и рабочим учебным планом.

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Декларативное программирование» относится к части, формируемой участниками образовательных отношений, блока 1 дисциплин (модулей) учебного плана.

## 3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОПОП (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине, в соответствии с индикатором достижения компетенции		Наименование оценочного средства
	Индикатор достижения компетенции (код, содержание индикатора)	Результаты обучения по дисциплине	
ПК-1. Способен проектировать и реализовывать программное обеспечение в соответствии с требованиями	<p>ПК-1.1. Знает методологии разработки программного обеспечения и технологии программирования, языки формализации функциональных спецификаций, принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения, основные концепции и атрибуты качества программного обеспечения.</p> <p>ПК-1.2. Умеет проводить анализ исполнения требований, вырабатывать варианты их реализации, проводить оценку и обоснование рекомендуемых решений, выбирать и использовать средства и варианты реализации программного обеспечения.</p> <p>ПК-1.3. Владеет навыками оценки возможностей, времени и трудоемкости реализации требований к программному обеспечению, разработки и согласования технических спецификаций на программное обеспечение, формирования и предоставления отчетности в соответствии с установленными регламентами, проектирования структур данных, баз данных, программных интерфейсов.</p>	<p>Знать: синтаксис языков логического и функционального программирования, отличия императивного и декларативного программирования, математические аспекты функционального и логического программирования;</p> <p>Уметь: преобразовывать циклы в рекурсию, составлять рекурсивные функции, реализовывать вычислительный процесс в декларативном стиле, выполнять альфа-конверсию и бета-редукцию, производить операции каррирования;</p> <p>Владеть: навыками обработки списков, построения <math>\lambda</math>-выражений, построения рекурсивных функций, навыками реализации функций высших порядков, записи дизъюнкторов в Prolog, обработки списковых структур в Prolog.</p>	<p>Отчёты по лабораторным работам</p> <p>Контрольные вопросы к лабораторным работам</p> <p>Контрольные вопросы к рейтинг-контролю и промежуточной аттестации</p>

#### 4. ОБЪЕМ И СТРУКТУРА ДИСЦИПЛИНЫ

Трудоемкость дисциплины составляет 4 зачетных единиц, 144 часов

##### Тематический план

форма обучения – очная

№ п/п	Наименование тем и/или разделов/тем дисциплины	Семестр	Неделя семестра	Контактная работа обучающихся с педагогическим работником				Самостоятельная работа	Формы текущего контроля успеваемости, форма промежуточной аттестации (по семестрам)
				Лекции	Практические занятия	Лабораторные работы	в форме практической подготовки		
1.	Императивное и декларативное программирование	8	1	2	—	—	4	3	
2.	Математические основы функционального программирования	8	1-3	10	—	—	2	8	Рейтинг-контроль №1
3.	Функциональные языки	8	1-4	4	—	12	3	12	
4.	Математические основы логического программирования	8	5-6	8	—	6	2	10	Рейтинг-контроль №2
5.	Логические языки	8	7-8	8	—	6	3	12	
6.	Внедрение принципов декларативного программирования в современные языки программирования	8	9	4	—	3	4		Рейтинг-контроль №3
Всего за 8 семестр:		-	-	36	-	27	18	45	экзамен, 36
Наличие в дисциплине КИ/КР		-	-	-	-	-	-	-	-
Итого по дисциплине		-	-	36	-	27	-	45	экзамен, 36

##### Содержание лекционных занятий по дисциплине

###### Раздел 1. Императивное и декларативное программирование

Введение в дисциплину. Понятие о функциональном программировании, его сравнение с императивным программированием и место в решении задач искусственного интеллекта.

###### Раздел 2 Математические основы функционального программирования

– Лямбда-исчисление Чёрча. Тезис Чёрча-Тьюринга. Алфавит, утверждения, аксиомы и правила вывода в лямбда-исчислении. Конвертируемость и редукция лямбда-выражений. Нормальная форма. Рекурсии в лямбда-исчислении. Чистое лямбда-исчисление.

– Принципы функционального программирования: Функции высших порядков. Рекурсивные функции. Статическое и динамическое связывание.

– Принципы функционального программирования: Отложенные вычисления. Понятие о строгих функциональных языках.

– Принципы функционального программирования: Карринг.

###### Раздел 3 Функциональные языки

– Языки функционального программирования: Основы языка Haskell. Символы, константы, логические значения. Списки и способы их записи. Кортежи.

– Базовые функции языка. Передача параметров. Область действия параметров.

– Определение функции в программе. Простая рекурсия. Работа со списками.

– Haskell как язык ООП: встроенные классы, примеры структурированных типов данных, инкапсуляция данных в модулях.

– Ввод и вывод информации: монады, интерактивный ввод/ вывод информации, работа с файлами.

###### Раздел 4 Математические основы логического программирования

– Основные конструкции рекурсивно-логического программирования.

- Простой абстрактный интерпретатор логических программ. Значение логической программы. Программирование баз данных.
- Рекурсивные правила. Связь логических программ с моделью реляционных баз данных.
- Рекурсивное программирование. Арифметика натуральных чисел. Списки. Правила построения рекурсивных программ.

#### Раздел 5 Логические языки

- Вычислительная модель Пролога. Программирование на Прологе.
- Циклы. Применение Пролога для решения задач искусственного интеллекта.

#### Раздел 6 Внедрение принципов декларативного программирования в современные языки программирования

- Декларативное программирование на C#.
- Декларативное программирование на Python.
- Декларативное программирование на JavaScript.

### Содержание лабораторных занятий по дисциплине

#### Раздел 3 Функциональные языки

- 1) Язык функционального программирования Haskell. Основные типы данных и конструкции языка.
- 2) Язык Haskell. Рекурсии в функциональном программировании. Сопоставление с образцом.
- 3) Язык Haskell. Функции высших порядков.
- 4) Язык Haskell. Конструкции let и where. Охраняющие условия. Полиморфизм.

#### Пользовательские типы данных.

- 5) Язык Haskell. Рекурсивные типы данных. Операторы.
- 6) Язык Haskell. Модули. Операции ввода-вывода.

#### Раздел 4 Математические основы логического программирования

- 7) Средства разработки логических программ. Предикаты для описания семейных связей.

#### Раздел 5 Логические языки

- 8) Структурирование данных и рекурсия.

- 9) Работа со списками.

#### Раздел 6 Внедрение принципов декларативного программирования в современные языки программирования

- 10) Изучение особенностей внедрения декларативного программирования в языки C#, Python, JavaScript.

## 5. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

### 5.1. Текущий контроль успеваемости

#### Вопросы для рейтинг-контроля №1.

Δ

1. Показать, что  $m + n = \lambda f x. m f (n f x)$ , если  $n = \lambda f x. f^n x$  и  $m, n \in N$ .
2. Свободные переменные.
3. Форма Бэкуса-Наура определения лямбда - терма.
4. Понятие нормальной формы  $\lambda$  - термов.
5. Пары и кортежи: определение и функции извлечения компонентов пары.
6. Постфиксная форма именования выражений (where)?
7. Синтаксическая эквивалентность термов с точностью до  $\alpha$  - конверсии. Как обозначается? Привести примеры.
8. Можно ли определить рекурсивные функции без именования? Каким образом?

9. Приведите причины, по которым следует придерживаться принципов строгой типизации.
10. Определите функцию cube, возводящую в куб числа типа Float.
11. Наибольший приоритет имеет функция или оператор ?:
12. Отличия списка от кортежа.
13. Что такое охранные выражения? Привести пример использования.
14. Дать рекурсивное определение функции 2n.
15. Укажите тип следующих величин: 3, even и even 3.
16. Если в объявлении функции f содержится три стрелочки, то f x будет содержать.
17. Назначение функции filter. Привести пример применения.
18. Дать определение композиции функций. Как обозначена операция композиции в Haskell.
19. Конструкторы создания списков.
20. Вычислите map (map (^2)) [[1,2],[3,4,5]].
21. Задайте бесконечный список натуральных чисел.
22. Объявить класс с методами сложения и взятия обратного элемента. Определить экземпляр класса для типа Int.
23. Объявление экземпляра класса instance означает
24. Назначение класса Eq. Какие методы определены в классе.
25. Ключевое слово для определения наследования.
26. Определить тип-синоним месяца года.
27. Написать программу, запрашивающую с клавиатуры ввод названия входного и выходного файла и осуществляющую копирование информации из одного в другой файл.

#### **Вопросы для рейтинг-контроля №2.**

1. Применение логического программирования в системах искусственного интеллекта (ИИ). Современные парадигмы решения задач искусственного интеллекта.
2. Роль логики в решении задач ИИ.
3. Понятие логической программы. Основные конструкции. Факты, правила, вопросы.
4. Логические переменные, подстановки и примеры.
5. Экзистенциальные вопросы. Универсальные факты.
6. Конъюнктивные вопросы и общие переменные. Правила. Процедуры.
7. Декларативный и процедурный подходы к разработке логических программ.
8. Простой абстрактный интерпретатор.
9. Значение логической программы.
10. Структурирование и абстрактные данные.
11. Рекурсивные правила.
12. Связь логического программирования с реляционной моделью данных.
13. Полнота и корректность логической программы.
14. Реализация арифметики натуральных чисел. Типовые условия.
15. Описание списков в логических программах.
16. Предикаты работы со списками.
17. Правила построения рекурсивных программ.

#### **Вопросы для рейтинг-контроля №3.**

1. Обзор языков декларативного программирования.
2. Внедрение принципов декларативного программирования в язык программирования C++.
3. Внедрение принципов декларативного программирования в язык программирования C#.
4. Внедрение принципов декларативного программирования в язык программирования Java.
5. Внедрение принципов декларативного программирования в язык программирования JavaScript.

6. Внедрение принципов декларативного программирования в язык программирования Python.

7. Внедрение принципов декларативного программирования в язык программирования Kotlin.

8. Внедрение принципов декларативного программирования в язык программирования Scala.

9. Внедрение принципов декларативного программирования в язык программирования F#.

10. Лямбда-выражения в .NET.

### **Б. Вопросы к экзамену.**

1. Понятие о функциональном программировании. Основные принципы ФП.

2. Сравнение ФП с императивным программированием.

3. Применение функционального программирования для решения задач.

4. ФП и системы искусственного интеллекта.

5. Классификация лямбда-термов. Форма Бэкуса-Наура.

6. Свободные и связанные переменные.

7. Конверсия и равенство лямбда-выражений.

8. Редукция лямбда-выражений и редукционные стратегии.

9. Комбинаторы. Комбинатор неподвижной точки.

10. Представление данных в лямбда-исчислении: булевские значения, условия, пары и кортежи.

11. Представление данных в лямбда-исчислении: натуральные числа и операции над ними.

12. Именованные выражения `let` и `where`.

13. Типизированное лямбда-исчисление. Типизация по Черчу и Карри.

14. Нормальная форма лямбда-выражений. Сильная нормализация.

15. Рекурсии в лямбда-исчислении.

16. Функции высших порядков.

17. Рекурсивные функции.

18. Статическое и динамическое связывание.

19. Отложенные вычисления.

20. Карринг.

21. Базовые типы языка Haskell.

22. Списки в языке Haskell.

23. Кортежи в языке Haskell.

24. Охранные выражения.

25. Сопоставление с образцом. Пример.

26. Определение рекурсивных функций. Примеры.

27. Операторы языка Haskell. Префиксная и инфиксная нотации. Приоритеты.

Ассоциативность.

28. Карринг в языке Haskell.

29. Функции высшего порядка: Функции на списках.

30. Функции высшего порядка: Лямбда функции.

31. Функции высшего порядка: Композиция функций.

32. Бесконечные списки. Принцип построения. Примеры.

33. Объектно-ориентированные возможности языка Haskell.

34. Встроенные классы.

35. Определение новых типов данных в языке Haskell.

36. Монады.

37. Ввод и вывод информации в языке Haskell. Файлы.

38. Основные конструкции рекурсивно-логического программирования: факты, вопросы, логические переменные, подстановки, примеры.

39. Основные конструкции рекурсивно-логического программирования: экзистенциальные вопросы, универсальные факты, конъюнктивные вопросы, правила логического вывода (совпадение, обобщение, конкретизация).

40. Правила в логической программе: определение, варианты интерпретации, процедуры. Обобщенный закон *modus ponens* и его применение при поиске ответов на вопросы к логической программе.

41. Простой абстрактный интерпретатор логических программ (для основных вопросов). Основная редукция и дерево вывода.

42. Значение логической программы.

43. Логические базы данных: принципы и примеры построения, структурированные и абстрактные данные.

44. Логические базы данных: связь логических программ с моделью реляционных баз данных.

45. Натуральные числа как рекурсивный тип термов: определение, примеры отношений.

46. Общие подходы и особенности построения рекурсивных программ.

47. Вычислительная модель Пролога. Сравнение Пролога с процедурными языками программирования.

48. Особенности программирования на чистом Прологе.

49. Внедрение особенностей декларативного программирования в языки программирования C#, Python.

50. Внедрение особенностей декларативного программирования в язык программирования JavaScript.

### **В. Самостоятельная работа**

1. Подготовка к текущему контролю и промежуточной аттестации.

2. Подготовка к практическим работам и оформление отчетов по результатам из выполнения. Контроль осуществляется на занятиях в виде устных ответов на вопросы преподавателя по содержанию отчета.

3. Работа с дополнительной литературой по вопросам, вынесенным на самостоятельное изучение. Контроль осуществляется на зачете.

Фонд оценочных материалов (ФОМ) для проведения аттестации уровня сформированности компетенций обучающихся по дисциплине оформляется отдельным документом.

## **6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

### **6.1. Книгообеспеченность**

Наименование литературы: автор, название, вид издания, издательство	Год издания	КНИГООБЕСПЕЧЕННОСТЬ
		Наличие в электронной библиотеке ВлГУ
Основная литература		
Козырева, Г. Ф. Функциональное и логическое программирование : учебно-методическое пособие / Г. Ф. Козырева. — Саратов : Ай Пи Эр Медиа, 2018. — 120 с. — ISBN 978-5-4486-0122-4.	2018	URL: <a href="http://www.iprbookshop.ru/71596.html">http://www.iprbookshop.ru/71596.html</a> . — Режим доступа: для авторизир. пользователей
Практикум по дисциплине Логическое и функциональное программирование / составители А. С. Доткулова. — М. : Московский технический университет связи и информатики, 2016. — 16 с. — ISBN 2227-8397.	2016	URL: <a href="http://www.iprbookshop.ru/61489.html">http://www.iprbookshop.ru/61489.html</a> . — Режим доступа: для авторизир. пользователей
Дополнительная литература		
Функциональное программирование на языке Haskell [Электронный ресурс] / Душкин Р.В. - М. : ДМК Пресс, 2008. -	2008	URL: <a href="http://www.studentlibrary.ru/book/ISBN5940743358.html">http://www.studentlibrary.ru/book/ISBN5940743358.html</a> . —

		Режим доступа: для авторизир. пользователей
Функциональное программирование на F# [Электронный ресурс] / Сошников Д.В. - М. : ДМК Пресс, 2011.	2011	URL: <a href="http://www.studentlibrary.ru/book/ISBN9785940746898.html">http://www.studentlibrary.ru/book/ISBN9785940746898.html</a> . — Режим доступа: для авторизир. пользователей

### 6.2. Периодические издания

1. Вестник компьютерных и информационных технологий
2. Вычислительные технологии
3. Автоматизация. Современные технологии

### 6.3. Интернет-ресурсы

1. Haskell – a purely functional language [Электронный ресурс].  
<http://www.haskell.org/haskellwiki/Haskell>.
2. Prolog Development Center //Режим доступа: <http://www.visual-prolog.com/>
3. Bibliographies on Logic Programming // Режим доступа:  
<http://liinwww.ira.uka.de/bibliography/LogicProgramming/>
4. Основы программирования на языке Пролог. НОУ ИНТУИТ. // Режим доступа:  
<http://www.intuit.ru/studies/courses/44/44/info>

## 7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Для реализации данной дисциплины имеются специальные помещения для проведения занятий лекционного типа, занятий практического/лабораторного типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы. Практические/лабораторные работы проводятся в компьютерных классах кафедры ФиПМ или ИВЦ ВлГУ.

Рабочую программу составил

доц. каф. ФИПМ А.Ю. Лексин  
(должность, ФИО, подпись)

Рецензент

Генеральный директор ООО «ФС Сервис» Д.С. Квасов  
(место работы, должность, ФИО, подпись)

Программа рассмотрена и одобрена на заседании кафедры ФИПМ  
Протокол №1 от 30.08.2021 года

Заведующий кафедрой \_\_\_\_\_ С.М. Аракелян  
(ФИО, подпись)

Рабочая программа рассмотрена и одобрена  
на заседании учебно-методической комиссии направления 02.03.03 Математическое  
обеспечение и администрирование информационных систем

Протокол №1 от 30.08.2021 года

Председатель комиссии \_\_\_\_\_ С.М. Аракелян  
(ФИО, должность, подпись)

### ЛИСТ ПЕРЕУТВЕРЖДЕНИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ

Рабочая программа одобрена на 20<sup>22</sup> / 20<sup>23</sup> учебный года

Протокол заседания кафедры № 1 от 30.08.2021 года

Заведующий кафедрой \_\_\_\_\_ С.М. Аракелян

Рабочая программа одобрена на 20 \_\_\_\_ / 20 \_\_\_\_ учебный года

Протокол заседания кафедры № \_\_\_\_ от \_\_\_\_ года

Заведующий кафедрой \_\_\_\_\_

Рабочая программа одобрена на 20 \_\_\_\_ / 20 \_\_\_\_ учебный года

Протокол заседания кафедры № \_\_\_\_ от \_\_\_\_ года

Заведующий кафедрой \_\_\_\_\_