Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение   
высшего образования «Новосибирский национальный исследовательский  
государственный университет» (Новосибирский государственный университет, НГУ)

**Факультет информационных технологий**

СОГЛАСОВАНО

Декан ФИТ НГУ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.М. Лаврентьев

«03» июля 2019 г.

РАБОЧАЯ ПРОГРАММА Дисциплины

|  |
| --- |
| **Декларативное программирование** |

Направление подготовки: 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

Направленность (профиль): Компьютерные науки и системотехника

Форма обучения: очная

Год обучения: 1 семестр: 1, 2

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Вид деятельности** | **Семестр** | |
| **1** | **2** |
| **1** | Лекции, час. | 32 | 32 |
| **2** | Практические занятия, час. | 32 | 32 |
| **3** | Лабораторные занятия, час. |  |  |
| **4** | Занятий в контактной форме без учета промежуточной аттестации, час, из них | 66 | 66 |
| **5** | в электронной форме, час. |  |  |
| **6** | из них аудиторных занятий, час. | 64 | 64 |
| **7** | из них в активной и интерактивной форме, час. | 64 | 64 |
| **8** | консультаций, час. | 2 | 2 |
| **9** | Самостоятельная работа, час. | 40 | 76 |
| **10** | в том числе на выполнение письменных работ, час |  |  |
| **11** | Форма аттестации (экзамен, зачет, дифференцированный зачет), час | ДЗ 2 | ДЗ 2 |
| **12** | Всего зачетных единиц[[1]](#footnote-1) | 3 | 4 |

Новосибирск 2019

Рабочая программа дисциплины составлена на основании федерального государственного образовательного стандарта (ФГОС) высшего образования - бакалавриат по направлению подготовки 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА.

Федеральный государственный образовательный стандарт (ФГОС) высшего образования - бакалавриат по направлению подготовки 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА введен в действие приказом Минобрнауки от 19.09.2017 № 929.

Место дисциплины в структуре учебного плана: Блок 1 Дисциплины (модули); обязательная часть; обязательная дисциплина.

Рабочая программа дисциплины утверждена решением Ученого совета факультета информационных технологий от 02.07.2019, протокол № 75.

Программу разработал:

доцент кафедры систем информатики ФИТ,

кандидат физико-математических наук В.Н. Власов

доцент кафедры систем информатики ФИТ,

кандидат физико-математических наук Д.С. Мигинский

Заведующий кафедрой систем информатики ФИТ,

доктор физико-математических наук М.М. Лаврентьев

Ответственный за образовательную программу:

доцент кафедры систем информатики ФИТ,

кандидат физико-математических наук Д.С. Мигинский

**Аннотация к рабочей программе дисциплины   
«Декларативное программирование»**

Дисциплина «Декларативное программирование» реализуется в рамках образовательной программы высшего образования – программы бакалавриата 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА, направленность (профиль): Компьютерные науки и системотехника по очной форме обучения на русском языке.

**Место в образовательной программе:**

Дисциплина «Декларативное программирование» реализуется в первом и втором семестрах в рамках обязательной части дисциплин (модулей) Блока 1.

Освоение данной дисциплины необходимо для дисциплин: Модели вычислений, Объектно-ориентированное программирование, Проектирование программного обеспечения, Организация, поиск и защита информации.

Дисциплина «Декларативное программирование» направлена на формирование компетенции:

Способен разрабатывать алгоритмы и программы, пригодные для практического применения (ОПК-8), в части следующих индикаторов достижения компетенции:

ОПК-8.1 Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения

ОПК-8.2 Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы , интегрировать программные модули

ОПК-8.3 Владеть: языком программирования; навыками отладки и тестирования работоспособности программы

**Перечень основных разделов дисциплины:**

Дисциплина «Декларативное программирование» предусматривает проведение лекций и практических занятий.

Дисциплина «Декларативное программирование» имеет своей целью: ознакомить студентов с основными структурами и методами декларативного программирования.

Общий объем дисциплины – 7 зачетных единиц (252 часа)

**Основные разделы (темы) дисциплины:**

Парадигмы программирования. Основы языка Haskell, определения функций, сопоставление с образцом (pattern matching), ветвления. Императивное программирование на Haskell, его эмуляция. Изменяемые переменные и изменяемые состояния

Моноиды, в математике и в практике. Функторы и аппликативные функторы

Графы, деревья, обходы деревьев. Конечные автоматы в функциональном языке, связь с регулярными выражениями Грамматики, дерево синтаксического разбора, парсеры. Схема восходящего парсера. Парсинг сбалансированных скобок и булевых выражений.

Параллельное вычисление чисто функциональных программ. Конкурентность и связанные с этим проблемы.

Декларативное программирование вне функциональных языков программирования. Реляционные базы данных. SQL – декларативный язык управления и описания баз. История разработки.

Основы реляционной алгебры. Переименование. Декартово произведение отношений. Естественное соединение, θ-соединение. Внутреннее соединение. Внешнее соединение. Значение NULL.

Обзор проблематики баз данных. Эффективность. Альтернативные модели данных. Свойства ACID.

**Правила аттестации по дисциплине.**

Программой дисциплины предусмотрено проведение следующих видов контроля: текущий контроль успеваемости в форме портфолио (прием задач), промежуточный контроль в форме дифференцированного зачета (в 1 и во 2 семестре).

В состав портфолио входят задания.

В каждом семестре задания делятся на две части: основную и дополнительную.

Для того чтобы получить доступ к дополнительной части заданий, необходимо сдать задания основной части в установленный заранее срок (10я неделя семестра).

Решенные студентом задания должны быть оформлены в виде текста программы и показаны преподавателю.

Предполагается, что студент умеет запустить предоставленный код и ответить на вопросы касательно содержимого.

Таким образом, от студента требуется реализовать необходимую функциональность и ответить на связанные с его реализацией вопросы преподавателя. После этого задание считается сданным.

Отдельные оценки за каждое из заданий не предполагаются.

Основным критерием оценивания является количество сданных заданий.

Если студент сдаёт необходимое количество заданий из базовой части до 10 недели, он получает возможность решать задачи из дополнительной части и претендовать на оценку хорошо и отлично.

Таким образом, сдача основной части даёт оценку «удовлетворительно».

Если студент не сдаёт основную часть заданий вовремя, он не получает возможности претендовать на высокую оценку.

Если студент не сдаёт основную часть в принципе, он не получает зачёта по предмету.

По результатам освоения дисциплины «Декларативное программирование» выставляется оценка «неудовлетворительно», «удовлетворительно», «хорошо», «отлично». Оценки «отлично», «хорошо», «удовлетворительно» означают успешное прохождение промежуточной аттестации

**Учебно-методическое обеспечение дисциплины.**

Учебно-методические материалы по дисциплине «Декларативное программирование» выложены на странице курса в сети Интернет:

<https://classroom.google.com/u/1/c/MTc1MzgwNTg4NjBa>

**1. Внешние требования к дисциплине**

Таблица 1.1

|  |  |
| --- | --- |
| ***Компетенция* ОПК-8 Способен разрабатывать алгоритмы и программы, пригодные для практического применения, *в части следующих индикаторов достижения компетенции:*** | |
| **ОПК-8.1** | Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения |  |
| **ОПК-8.2** | Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули |  |
| **ОПК-8.3** | Владеть: языком программирования; навыками отладки и тестирования работоспособности программы |  |

**2. Требования к результатам освоения дисциплины**

Таблица 2.1

|  |  |  |  |
| --- | --- | --- | --- |
| **Результаты изучения дисциплины по уровням освоения (иметь представление, знать, уметь, владеть)** | **Формы организации занятий** | | |
| **Лекции** | **Практики / семинары** | **Самостоятельная работа** |
| ОПК-8.1 Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения | | | |
| 1. Знать принципы и концепции декларативного программирования в части как функционального так и логического подхода, а также основные структуры и методы связанные с декларативными языками. | + | + | + |
| ОПК-8.2 Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы , интегрировать программные модули | | | |
| 2. Уметь объяснить характер и порядок вычислений декларативной программы, следуя семантике и прагматике соответствующего языка и виртуальной машины, на которой он реализован | + | + | + |
| ОПК-8.3 Владеть: языком программирования; навыками отладки и тестирования работоспособности программы | | | |
| 3. Знать и понимать связи между базисными математическими понятиями (лямбда выражения, типы, классы) и конструкциями языков программирования | + | + | + |
| **4.** Уметь строить иерархические конструкции в языках функционального и логического программирования | + | + | + |

**3. Содержание и структура учебной дисциплины**

Таблица 3.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Темы лекций** | **Активные формы, час.** | **Часы** | **Ссылки на результаты обучения** | |
| **Семестр: 1** | | | | |
| Лекция 1. История Функционального программирования и его основные представители. | 2 | 2 | | 1, 2, 3, 4 |
| Лекция 2. Парадигмы программирования. | 2 | 2 | | 1, 2, 3, 4 |
| Лекция 3. Основы языка Haskell, определения функций, сопоставление с образцом (pattern matching), ветвления. Параметрический полиморфизм | 2 | 2 | | 1, 2, 3, 4 |
| Лекция 4. Карринг и лямбда-абстракция, замыкания (обзор реализации в др. языках программирования), Операторы композиции функции и применения | 2 | 2 | | 1, 2, 3, 4 |
| Лекция 5. Система типов в Haskell, кортежи, списки, множества. Базовые функции для работы с ними | 2 | 2 | | 1, 2, 3, 4 |
| Лекция 6. Свертки, их варианты (foldl, foldr) и их использование для обработки списков и обобщенных структур. | 2 | 2 | | 1, 2, 3, 4 |
| Лекция 7. Система типов в Haskell и др. языках программирования. Виды типизации: полная, сильная, статическая, явная | 2 | 2 | | 1, 2, 3, 4 |
| Лекция 8. Классы типов в Haskell, абстрактные типы, ad-hoc-полиморфизм. Предопределенные классы типов в Haskell | 2 | 2 | | 1, 2, 3, 4 |
| Лекция 9. Структуры данных в Haskell. Тип данных «Maybe», «Either» и их возможности. Рекурсивные типы – списки, деревья. Массивы в функциональном программировании | 2 | 2 | | 1, 2, 3, 4 |
| Лекция 10. Представление текста в Haskell и его обработка. Регулярные выражения в Haskell | 2 | 2 | | 1, 2, 3, 4 |
| Лекция 11. Структура программ, точка входа. Стандартный ввод-вывод. Проекты, запуск и компиляция. Обработка текстовых файлов. Общее представление о работе с бинарными файлами | 2 | 2 | | 1, 2, 3, 4 |
| Лекция 12. Императивное программирование на Haskell, его эмуляция. Изменяемые переменные и изменяемые состояния | 2 | 2 | | 1, 2, 3, 4 |
| Лекция 13. Моноиды, в математике и в практике. Функторы и аппликативные функторы | 2 | 2 | | 1, 2, 3, 4 |
| Лекция 14. Монады, монадные значения и монадная функция. Класс «Monad». Законы монад | 2 | 2 | | 1, 2, 3, 4 |
| Лекция 15. Конкретные монады и их приложения (Maybe, Either, Writer, Монады State и Cont) | 2 | 2 | | 1, 2, 3, 4 |
| Лекция 16. Трансформеры, монадные утилиты, MonadPlus | 2 | 2 | | 1, 2, 3, 4 |
| **Итого** | **32** | **32** | |  |
| **Семестр: 2** | | | | |
| Лекция 1. Графы, деревья, обходы деревьев. Конечные автоматы в функциональном языке, связь с регулярными выражениями | 2 | 2 | | 1, 2, 3, 4 |
| Лекция 2. Грамматики, дерево синтаксического разбора, парсеры. Схема восходящего парсера. Парсинг сбалансированных скобок и булевых выражений. | 2 | 2 | | 1, 2, 3, 4 |
| Лекция 3. Шаблоны (Template) в Haskell как возможности метапрограммирования | 2 | 2 | | 1, 2, 3, 4 |
| Лекция 4. Расширения системы типов Haskell. Многопараметрические классы типов и функциональные зависимости. Семейства типов. GADT | 2 | 2 | | 1, 2, 3, 4 |
| Лекция 5. Вывод типов, модель типизации Хиндли-Милнера. Дальнейшее развитие возможностей системы типов: Idris, Agda, Clean, Curry | 2 | 2 | | 1, 2, 3, 4 |
| Лекция 6. Линзы и варианты ООП в Haskell | 2 | 2 | | 1, 2, 3, 4 |
| Лекция 7. Параллельное вычисление чисто функциональных программ. Конкурентность и связанные с этим проблемы. | 2 | 2 | | 1, 2, 3, 4 |
| Лекция 8. Декларативное программирование вне функциональных языков программирования. Реляционные базы данных. SQL – декларативный язык управления и описания баз. История разработки. | 2 | 2 | | 1, 2, 3, 4 |
| Лекция 9. Основы языка SQL: оператор SELECT. Кортежи и типы. Типы в языке SQL. Унарные реляционные операции: проекция, выборка. Основные предикаты над числами и строками. Отношения, как множества. Операции над множествами. Уникальная выборка (DISTINCT), сортировка (ORDER BY), окна, агрегирующие операции. | 2 | 2 | | 1, 2, 3, 4 |
| Лекция 10. Основы реляционной алгебры. Переименование. Декартово произведение отношений. Естественное соединение, θ-соединение. Внутреннее соединение. Внешнее соединение. Значение NULL. Практика: Выборка из нескольких таблиц. Внутреннее и внешнее соединение. | 2 | 2 | | 1, 2, 3, 4 |
| Лекция 11. Агрегация и группировка (GROUP BY). Агрегирующие операции и условия при группировке (HAVING). Суперключ. Возможный, первичный, внешний ключи. | 2 | 2 | | 1, 2, 3, 4 |
| Лекция 12. Понятие о нормализации отношений. Нормальные формы (1-5). Функциональные, множественные зависимости, зависимости соединения. Элементы проектирования баз данных. ER-модель. Множественность связей, 1-1, 1-n, n-n. Связи типа 0..1-n, внешнее соединение. Представление деревьев и графов. | 2 | 2 | | 1, 2, 3, 4 |
| Лекция 13. Вложенные запросы. Каскадная агрегация. Эквивалентные запросы и переписывание запроса. | 2 | 2 | | 1, 2, 3, 4 |
| Лекции 14-15. Эффективность запросов. Основные алгоритмы вычисления соединений. Понятие индекса. Оптимизация плана запроса: алгебраические свойства реляционных операций, отложенные вычисления. | 4 | 4 | | 1, 2, 3, 4 |
| Лекция 16. Обзор проблематики баз данных. Эффективность. Альтернативные модели данных. Свойства ACID. | 2 | 2 | | 1, 2, 3, 4 |
| **Итого** | **32** | **32** | |  |

Таблица 3.2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Темы практических занятий** | **Активные формы, час.** | | **Часы** | **Ссылки на результаты обучения** | **Учебная деятельность** |  |
| **Семестр: 1** | | | | | | |
| Тема 1. Занятия 1-4 | 8 | | 8 | 1, 2, 3, 4 | Основы языка Haskell, типизация данных и функций, возможности ghci  Сопоставление с образцом (pattern matching).  Создание простых вычислительных функций  Решение задач и обсуждение вопросов связанных с теорией. |  |
| Тема 2. Занятия 5-10 | 12 | | 12 | 1, 2, 3, 4 | Рекурсия и ко-рекурсия, Хвостовая рекурсия, представление циклов в рекурсивном виде.  Обобщенные функции и параметрический полиморфизм.  Решение задач и обсуждение вопросов связанных с теорией. |  |
| Тема 3. Занятия 11-16 | 12 | | 12 | 1, 2, 3, 4 | Обработка списков, свёртки, множества.  Бесконечные структуры данных.  Решение задач и обсуждение вопросов связанных с теорией |  |
| **Итого** | **32** | | **32** |  |  |
| **Семестр: 2** | | | | | |  |
| Тема 1. Занятия 1-4 | 6 | | 6 | 1, 2, 3, 4 | Типы и классы типов.  Обработка текстовых файлов  Решение задач и обсуждение вопросов связанных с теорией. |  |
| Тема 2. Занятия 5-8 | | 8 | 8 | 1, 2, 3, 4 | Монады и функторы, использование возможностей утилит.  Решение задач и обсуждение вопросов связанных с теорией. |  |
| Тема 3 Занятия 9-14 | | 10 | 10 | 1, 2, 3, 4 | Реализация запросов с применением основных элементов SQL: проекция, селекция, соединения, группировка и агрегация.  Вложенные запросы. |  |
| Тема 4 Занятия 15-16 | | 8 | 8 | 1,2,3,4 | Эффективность запросов. Переписывание запросов. Реализация планов запросов на структурах данных в Haskell. |  |
| **Итого** | | **32** | **32** |  |  |

**4. Самостоятельная работа бакалавров**

**Таблица 4.1**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **№** | **Виды самостоятельной работы** | | **Ссылки на результаты обучения** | | **Часы на выполнение** | **Часы на консультации** | |
| **Семестр: 1** | | | | | | | |
| 1 | Изучение разделов дисциплины по учебной литературе, в том числе вопросов, не освещаемых на лекциях | | 1, 2, 3, 4 | | 10 |  | |
| Изучение предлагаемых теоретических разделов в соответствии с настоящей Программой. Учебно-методические материалы по дисциплине «Декларативное программирование» выложены на странице курса в сети Интернет | | | | | | |
| 2 | Подготовка к практическим занятиям, к текущему контролю знаний и промежуточной аттестации | | 1, 2, 3, 4 | | 20 | 2 | |
| Разбор решенных задач, самостоятельное решение задач | | | | | | |
| 3 | Подготовка к дифференцированному зачету | |  | | 10 |  | |
| Повторение теоретического материала по вопросам, совпадающим с темами лекций | | | | | | |
| Итого | |  | | **40** | **2** | |
| **Семестр: 2** | | | | | | | |
| 1 | Изучение разделов дисциплины по учебной литературе, в том числе вопросов, не освещаемых на лекциях | | 1, 2, 3, 4 | | 10 |  | |
| Изучение предлагаемых теоретических разделов в соответствии с настоящей Программой. Учебно-методические материалы по дисциплине «Декларативное программирование» выложены на странице курса в сети Интернет | | | | | | |
| 2 | Подготовка к практическим занятиям, к текущему контролю знаний и промежуточной аттестации | | 1, 2, 3, 4 | | 56 | 2 | |
| Разбор решенных задач, самостоятельное решение задач | | | | | | |
| 3 | Подготовка к дифференцированному зачету | | 1, 2, 3, 4 | | 10 |  | |
| Повторение теоретического материала по вопросам, совпадающим с темами лекций | | | | | | |
| **Итого** |  | | **76** | | | **2** |

**5. Образовательные технологии**

В ходе реализации учебного процесса по дисциплине проводятся лекционные и семинарские занятия. Темы, рассматриваемые на лекциях и изучаемые самостоятельно, закрепляются на практических занятиях, по вопросам, вызывающим затруднения, проводятся консультации.

В ходе реализации учебного процесса по дисциплине применяются лекционные и практические занятия, а также применяются следующие интерактивные формы обучения (таблица 5.1).

Таблица 5.1

|  |  |  |  |
| --- | --- | --- | --- |
| **1** | | Лекция в форме дискуссии | ОПК-8 |
| **Формируемые умения:** знать основные приемы функционального программирования и построения структурированных запросов. | | | |
| **Краткое описани е применения:** Обсуждение, в контексте изученной теории, практического применения различных аспектовдекларативного программирования | | | |
| **2** | Портфолио | | ОПК-8 |
| **Формируемые умения:** владеть основными приемами функционального программирования и построения структурированных запросов. | | | |
| **Краткое описание применения:** бакалавры ведут портфолио (оценки за выполненные задания), которое является основой для проведения аттестации по дисциплине | | | |

Для организации и контроля самостоятельной работы бакалавров, а также проведения консультаций применяются информационно-коммуникационные технологии (таблица 5.2).

Таблица 5.2

|  |  |
| --- | --- |
| Информирование | Адрес почты – сообщается бакалаврам на первом занятии. |
| Консультирование | Адрес почты – сообщается бакалаврам на первом занятии. |
| Контроль | <https://classroom.google.com/u/1/c/MTc1MzgwNTg4NjBa> |
| Размещение учебных материалов | <https://classroom.google.com/u/1/c/MTc1MzgwNTg4NjBa> |

**6. Правила аттестации бакалавров по учебной дисциплине**

По дисциплине «Декларативное программирование» проводится текущая и промежуточная аттестация (итоговая по дисциплине).

Промежуточная аттестация по дисциплине производится в форме дифференцированного зачета.

Текущая аттестация проходит в форме портфолио, в состав которого входят задания.

В каждом семестре задания делятся на две части: основную и дополнительную.

Для того чтобы получить доступ к дополнительной части заданий, необходимо сдать задания основной части в установленный заранее срок (10я неделя семестра).

Решенные студентом задания должны быть оформлены в виде текста программы и показаны преподавателю.

Предполагается, что студент умеет запустить предоставленный код и ответить на вопросы касательно содержимого.

Таким образом, от студента требуется реализовать необходимую функциональность и ответить на связанные с его реализацией вопросы преподавателя. После этого задание считается сданным.

Отдельные оценки за каждое из заданий не предполагаются.

Основным критерием оценивания является количество сданных заданий.

Если студент сдаёт необходимое количество заданий из базовой части до 10 недели, он получает возможность решать задачи из дополнительной части и претендовать на оценку хорошо и отлично.

Таким образом, сдача основной части даёт оценку «удовлетворительно».

Если студент не сдаёт основную часть заданий вовремя, он не получает возможности претендовать на высокую оценку.

Если студент не сдаёт основную часть в принципе, он не получает зачёта по предмету.

По результатам освоения дисциплины «Декларативное программирование» выставляется оценка «неудовлетворительно», «удовлетворительно», «хорошо», «отлично». Оценки «отлично», «хорошо», «удовлетворительно» означают успешное прохождение промежуточной аттестации

В таблице 6.1 представлено соответствие форм аттестации заявляемым требованиям к результатам освоения дисциплины.

Таблица 6.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Коды компетенций ФГОС** | **Результаты обучения** | **Формы аттестации** | | | |
| **семестр 1** | | **семестр 2** | |
| портфолио | дифзачет | портфолио | дифзачет |
| **ОПК-8** | **ОПК-8.1**  Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения | **+** | **+** | **+** | **+** |
| **ОПК-8.2** Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули | **+** | **+** | **+** | **+** |
| **ОПК-8.3** Владеть: языком программирования; навыками отладки и тестирования работоспособности программы | + | + | + | + |

Требования к структуре и содержанию портфолио, оценочные средства, а также критерии оценки сформированности компетенций и освоения дисциплины в целом, представлены в Фонде оценочных средств, являющемся приложением 1 к настоящей рабочей программе дисциплины.

**7. Литература**

1. Касьянов, Виктор Николаевич (1948-). Практикум по программированию : учебное пособие : [для студентов вузов] / В.Н. Касьянов, Е.В. Касьянова ; М-во образования и науки РФ, Новосиб. гос. ун-т, Мех.-мат. фак., Каф. программирования. Новосибирск : Редакционно-издательский центр НГУ, 2013. 197 с. : ил. ; 20 см. URL: <http://e-lib.nsu.ru/dsweb/Get/Resource-949/page001.pdf>. ISBN 978-5-4437-0167-7.
2. Лавров, И.А. Задачи по теории множеств, математической логике и теории алгоритмов / И.А. Лавров, Л.Л. Максимова. - 5-е изд., исправл. - Москва : Физматлит, 2002. - 258 с. - ISBN 5-9221-0026-2 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=75576>.
3. Когабаев, Нурлан Талгатович. Лекции по теории алгоритмов [Текст: электронный ресурс] : учебное пособие : [для студентов 1 курса Мех.-мат. фак. НГУ] / Н.Т. Когабаев ; Федер. агентство по образованию, Новосиб. гос. ун-т, Мех.-мат. фак. (Новосибирск : РИЦ НГУ, 2016) . URL: <http://e-lib.nsu.ru/dsweb/Get/Resource-1276/page00000.pdf>.
4. Власов, Владимир Николаевич. Конструирование дискретных математических объектов в парадигме функционального программирования [Текст: электронный ресурс] : учебно-методическое пособие : [для студентов вузов] / В.Н. Власов ; М-во образования и науки РФ, Новосиб. гос. ун-т, Мех.-мат. фак., Каф. дискрет. математики и информатики. (Новосибирск : РИЦ НГУ, 2016) . URL: <http://e-lib.nsu.ru/dsweb/Get/Resource-368/page001.pdf>.

*Интернет-ресурсы*

*Таблица 7.1*

|  |  |  |
| --- | --- | --- |
| № п/п | Наименование Интернет-ресурса | Краткое описание |
|  | Власов В.Н. Haskell: введение в функциональное программирование <https://wiki.nsunc.com/haskell> | Авторский курс (2011-2018 гг.), читаемый на базе кафедры информатики СУНЦ НГУ |
|  | Власов В.Н. Дискретная математика для computer science [Электронный ресурс] : электронный учебно-методический комплекс; Новосиб. гос. ун-т. - Новосибирск, [2018]  <https://classroom.google.com/c/MTgwMDQwMjIzNzNa> | Авторский курс, читаемый на базе СИ ФИТ НГУ |
|  | http://habrahabr.ru | Электронное издание, посвящённое программированию и связанным областям |
|  | http://intuit.ru | Онлайн курсы на русском языке |
|  | http://coursera.org | Онлайн курсы на английском языке |
|  | <http://projecteuler.net> | Сборник задач для самостоятельного освоения языков программирования. |

**8. Учебно-методическое и программное обеспечение дисциплины**

**8.1. Учебно-методическое обеспечение**

учебные и учебно-методические материалы дисциплины

Власов В.Н. Дискретная математика для computer science [Электронный ресурс] : электронный учебно-методический комплекс / В.Н. Власов; Новосиб. гос. ун-т. - Новосибирск, [2018]

<https://classroom.google.com/c/MTgwMDQwMjIzNzNa>.

Власов В.Н. Haskell: введение в функциональное программирование [Электронный ресурс] : электронный учебно-методический комплекс / В.Н. Власов; Новосиб. гос. ун-т. - Новосибирск, [2011-2018] <https://wiki.nsunc.com/haskell>.

**8.2.** **Программное обеспечение**

Для обеспечения реализации дисциплины используется стандартный комплект программного обеспечения (ПО), включающий регулярно обновляемое лицензионное ПО Windows и MS Office.

Специализированное программное обеспечение для изучения дисциплины не требуется.

**9. Профессиональные базы данных и информационные справочные системы**

1. Полнотекстовые журналы Springer Journals за 1997-2015 г., электронные книги (2005-2016 гг.), коллекция научных биомедицинских и биологических протоколов SpringerProtocols, коллекция научных материалов в области физических наук и инжиниринга SpringerMaterials, реферативная БД по чистой и прикладной математике zbMATH.
2. Электронная библиотека диссертаций Российской государственной библиотеки (ЭБД РГБ)
3. Электронные ресурсы Web of Science Core Collection (Thomson Reuters Scientific LLC.), Journal Citation Reports + ESI
4. БД Scopus (Elsevier)

**10. Материально-техническое обеспечение**

Таблица 10.1

|  |  |  |
| --- | --- | --- |
| **№** | **Наименование** | **Назначение** |
| 1 | Презентационное оборудование (мультимедиа-проектор, экран, компьютер для управления) | Для проведения лекционных занятий |
| 2 | Компьютерный класс (с выходом в Internet) | Для проведения практических занятий и организации самостоятельной работы обучающихся |

Материально-техническое обеспечение образовательного процесса по дисциплине для обучающихся из числа лиц с ограниченными возможностями здоровья осуществляется согласно «Порядку организации и осуществления образовательной деятельности по образовательным программам для инвалидов и лиц с ограниченными возможностями здоровья в Новосибирском государственном университете».

**Лист актуализации рабочей программы дисциплины  
«Декларативное программирование»**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Характеристика внесенных изменений (с указанием пунктов документа) | Дата и №  протокола Ученого совета ФИТ | Подпись  ответственного |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. С учетом выделенных часов на промежуточную аттестацию [↑](#footnote-ref-1)