

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**федеральное государственное автономное
образовательное учреждение высшего образования_
«Национальный исследовательский Нижегородский государственный университет
им. Н.И. Лобачевского»**

Институт информационных технологий, математики и механики

УТВЕРЖДЕНО

решением президиума Ученого совета ННГУ

протокол № 1 от 16.01.2024 г.

Рабочая программа дисциплины

Функциональное программирование

Уровень высшего образования

Магистратура

Направление подготовки / специальность

01.04.02 - Прикладная математика и информатика

Направленность образовательной программы

Компьютерные науки и приложения

Форма обучения

очная

г. Нижний Новгород

2024 год начала подготовки

1. Место дисциплины в структуре ОПОП

Дисциплина Б1.В.ДВ.04.01 Функциональное программирование относится к части, формируемой участниками образовательных отношений образовательной программы.

2. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями и индикаторами достижения компетенций)

| Формируемые компетенции (код, содержание компетенции) | Планируемые результаты обучения по дисциплине (модулю), в соответствии с индикатором достижения компетенции | | Наименование оценочного средства | |
|---|--|---|------------------------------------|------------------------------|
| | Индикатор достижения компетенции (код, содержание индикатора) | Результаты обучения по дисциплине | Для текущего контроля успеваемости | Для промежуточной аттестации |
| ПК-11: Способен разрабатывать и анализировать концептуальные и теоретические модели решаемых задач производственно-технологической деятельности | <p>ПК-11.1: Знает методы разработки и анализа концептуальных и теоретических моделей решаемых производственно-технологических задач</p> <p>ПК-11.2: Умеет применять методы разработки и анализа концептуальных и теоретических моделей решаемых производственно-технологических задач</p> <p>ПК-11.3: Имеет навыки применения методов разработки и анализа концептуальных и теоретических моделей решаемых производственно-технологических задач</p> | <p>ПК-11.1: Знать способы разработки ПО на языке Haskell. Знать синтаксис и основные конструкции языка Haskell.</p> <p>ПК-11.2: Уметь писать программы на языке Haskell, использующие хвостовую и нехвостовую рекурсию, функции высших порядков, ленивые вычисления, алгебраические типы данных, классы типов и монады. Уметь определять новые типы, классы и монады для структурирования программ.</p> <p>ПК-11.3: Владеть навыками чтения сообщения об ошибках и исправления ошибок, навыками чтения и составления документации. Владеть способами планирования времени для успешного выполнения еженедельных домашних заданий.</p> | Задания | Зачёт: Контрольная работа |
| ПК-4: Способен разрабатывать и анализировать концептуальные и теоретические модели решаемых | ПК-4.1: Знает методы разработки и анализа концептуальных и теоретических моделей решаемых научных проблем | ПК-4.1: Знать особенности, преимущества и недостатки парадигмы функционального программирования, ее место в | Задания | Зачёт: Контрольная работа |

| | | | | |
|-------------------------|--|---|--|--|
| научных проблем и задач | и задач ПК-4.2: Умеет применять методы разработки и анализа концептуальных и теоретических моделей решаемых научных проблем и задач ПК-4.3: Имеет навыки применения методов разработки и анализа концептуальных и теоретических моделей решаемых научных проблем и задач | современной программной инженерии. Знать основы лямбда-исчисления как математического основания функционального программирования. ПК-4.2: Уметь использовать свойства лямбда-исчисления для доказательства правильности программ. Уметь анализировать сложность программ. ПК-4.3: Владеть навыками мышления и разработки программ в функциональной парадигме. | | |
|-------------------------|--|---|--|--|

3. Структура и содержание дисциплины

3.1 Трудоемкость дисциплины

| | |
|--|--------------------------|
| | очная |
| Общая трудоемкость, з.е. | 3 |
| Часов по учебному плану | 108 |
| в том числе | |
| аудиторные занятия (контактная работа): | |
| - занятия лекционного типа | 16 |
| - занятия семинарского типа (практические занятия / лабораторные работы) | 16 |
| - КСР | 1 |
| самостоятельная работа | 75 |
| Промежуточная аттестация | 0 Зачёт |

3.2. Содержание дисциплины

(структурированное по темам (разделам) с указанием отведенного на них количества академических часов и виды учебных занятий)

| Наименование разделов и тем дисциплины | Всего (часы) | в том числе | | | Самостоятельная работа обучающегося, часы |
|--|-----------------|--|---|-------|--|
| | | Контактная работа (работа во взаимодействии с преподавателем), часы из них | | | |
| | | Занятия лекционного типа | Занятия семинарского типа (практические занятия/ лабора торные | Всего | |

| | | | работы), часы | | |
|--|-------------|-------------|---------------|-------------|-------------|
| | о ф о | о ф о | о ф о | о ф о | о ф о |
| 1. Основы программирования на Haskell. | 26 | 4 | 4 | 8 | 18 |
| 2. Лямбда-исчисление с типами. | 20 | 3 | 3 | 6 | 14 |
| 3. Определение новых типов. | 20 | 3 | 3 | 6 | 14 |
| 4. Монады и их использование. | 41 | 6 | 6 | 12 | 29 |
| Аттестация | 0 | | | | |
| КСР | 1 | | | 1 | |
| Итого | 108 | 16 | 16 | 33 | 75 |

Содержание разделов и тем дисциплины

1. Установка интерпретатора и работа с ним. Основные типы данных и их значения. Определение функций. Конструкции `let` и `where`. Ограничения. Сопоставление с образцом. Частично примененные функции. Сечения. Бинарные операторы. Модули. Рекурсия. Ленивые вычисления. Арифметические прогрессии. Генераторы списков. Двумерный синтаксис. Бесточечная запись. Функции высших порядков. Левая и правая свертки.
2. Лямбда-исчисление с простыми типами Редукции и конверсии Параметрический полиморфизм Определение типов подвыражений и нахождение типовых ошибок
3. Классы типов. Алгебраические типы. Синонимы типов (`type`) и обертки вокруг типов (`newtype`). Объявление классов типов и их экземпляров. Разностные списки. Класс типов `Monoid` и его элементы. Класс `Foldable`. Моноид эндоморфизмов.
4. Введение в монады на примере типов `Maybe`, `Either` и списков. Функция `return`. `do`-нотация. Объявление монады. Функция `fail`. Использование списочной монады для решения задач перебором. Примеры монад, имеющих вид пары. Монада `Writer`. Использование `Writer` для журналирования. Псевдослучайные числа и булевы значения. Генерирование псевдослучайных формул с явной передачей состояния. Тип `State`, объявление его монадой. Монада `State` в стандартной библиотеке. Модуль `System.Random`.

4. Учебно-методическое обеспечение самостоятельной работы обучающихся

Самостоятельная работа обучающихся включает в себя подготовку к контрольным вопросам и заданиям для текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведенным в п. 5.

Холомьев А. Учебник по Haskell. 2012.

Макеев Г.А. Основы функционального программирования на языке Haskell. УГАТУ, 2013.

5. Фонд оценочных средств для текущего контроля успеваемости и промежуточной аттестации по дисциплине (модулю)

5.1 Типовые задания, необходимые для оценки результатов обучения при проведении текущего контроля успеваемости с указанием критериев их оценивания:

5.1.1 Типовые задания (оценочное средство - Задания) для оценки сформированности компетенции ПК-11:

1. Используя генератор списков, напишите функцию `powerset :: [a] -> [[a]]`, которая возвращает список всех подсписков данного списка. Порядок элементов в возвращаемых списках неважен.
2. Напишите функцию `primes :: [Integer]`, которая вычисляет бесконечный список простых чисел с помощью алгоритма "Решето Эратосфена". Начинать следует с бесконечного списка нечетных чисел, больших 1. Для прореживания списка используйте генератор списков.
3. Объясните, почему `map (f . g) == map f . map g`.
4. Выразите функции `elem` и `filter` через `foldr`.

5.1.2 Типовые задания (оценочное средство - Задания) для оценки сформированности компетенции ПК-4:

1. Для каждого из следующих типов напишите какой-нибудь терм, имеющий этот тип.
`(a -> a) -> a -> a`
`(a -> b) -> a -> b`
`((a -> b) -> a) -> (a -> b) -> b`
2. Напишите функцию `composeM :: Monad m => (a -> m b) -> (b -> m c) -> a -> m c`, которая возвращает композицию двух монадных функций. В `Control.Monad` такая функция называется `(>=>)`.
3. Напишите функцию `knapsack`, которая решает задачу о рюкзаке полным перебором. Функция получает список `weights` весов предметов, список `values` их стоимостей (эти списки имеют одинаковую длину), а также максимальный вес `wl` (`weight limit`), который может вместить рюкзак. Функция возвращает пару `(v, lst)`, где `v` — это максимальная общая стоимость предметов, которые можно поместить в рюкзак, не нарушая ограничение по весу, а `lst` — список из 0 и 1, который показывает, какие предметы нужно взять (1 означает, что предмет берется, а 0 — что он исключается).

Критерии оценивания (оценочное средство - Задания)

| Оценка | Критерии оценивания |
|------------|---|
| зачтено | Задача решена полностью или решена основная часть задачи, или задача решена с недочетами. |
| не зачтено | Задача не решена или сделан только первый этап решения задачи. |

5.2. Описание шкал оценивания результатов обучения по дисциплине при промежуточной аттестации

Шкала оценивания сформированности компетенций

| Уровень сформированности компетенций (индикатора достижения компетенций) | плохо | неудовлетворительно | удовлетворительно | хорошо | очень хорошо | отлично | превосходно |
|--|---|--|--|---|---|---|--|
| | не зачтено | | зачтено | | | | |
| <u>Знания</u> | Отсутствие знаний теоретического материала. Невозможность оценить полноту знаний вследствие отказа обучающегося от ответа | Уровень знаний ниже минимальных требований. Имели место грубые ошибки | Минимально допустимый уровень знаний. Допущено много негрубых ошибок | Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько негрубых ошибок | Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько несущественных ошибок | Уровень знаний в объеме, соответствующем программе подготовки. Ошибок нет. | Уровень знаний в объеме, превышающем программу подготовки. |
| <u>Умения</u> | Отсутствие минимальных умений. Невозможность оценить наличие умений вследствие отказа обучающегося от ответа | При решении стандартных задач не продемонстрированы основные умения. Имели место грубые ошибки | Продemonстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном объеме | Продemonстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания в полном объеме, но некоторые с недочетами | Продemonстрированы все основные умения. Решены все основные задачи. Выполнены все задания в полном объеме, но некоторые с недочетами. | Продemonстрированы все основные умения. Решены все основные задачи с отдельными и несущественными недочетами, выполнены все задания в полном объеме | Продemonстрированы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме без недочетов |
| <u>Навыки</u> | Отсутствие базовых навыков. Невозможность оценить наличие навыков вследствие отказа обучающегося от ответа | При решении стандартных задач не продемонстрированы базовые навыки. Имели место грубые ошибки | Имеется минимальный набор навыков для решения стандартных задач с некоторыми недочетами | Продemonстрированы базовые навыки при решении стандартных задач с некоторыми недочетами | Продemonстрированы базовые навыки при решении стандартных задач без ошибок и недочетов | Продemonстрированы навыки при решении нестандартных задач без ошибок и недочетов | Продemonстрирован творческий подход к решению нестандартных задач |

Шкала оценивания при промежуточной аттестации

| Оценка | | Уровень подготовки |
|---------|-------------|--|
| зачтено | превосходно | Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «превосходно», продемонстрированы знания, умения, владения по соответствующим компетенциям на уровне выше предусмотренного программой |

| | | |
|-------------------|----------------------------|--|
| | отлично | Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично». |
| | очень хорошо | Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «очень хорошо» |
| | хорошо | Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «хорошо». |
| | удовлетворительно | Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «удовлетворительно», при этом хотя бы одна компетенция сформирована на уровне «удовлетворительно» |
| не зачтено | неудовлетворительно | Хотя бы одна компетенция сформирована на уровне «неудовлетворительно». |
| | плохо | Хотя бы одна компетенция сформирована на уровне «плохо» |

5.3 Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения на промежуточной аттестации с указанием критериев их оценивания:

5.3.1 Типовые задания (оценочное средство - Контрольная работа) для оценки сформированности компетенции ПК-11

Напишите функцию `hIndex :: [Int] -> Int`, которая вычисляет индекс Хирша автора. Входом к этой функции является список, где каждый элемент соответствует одной публикации. Значение элемента равно количеству ссылок в работах других людей на эту публикацию. Например, если автор написал три статьи, из которых первые две упоминались в одной статье каждая, а третья упоминалась в пяти статьях, то список будет `[1, 1, 5]`. Индекс Хирша равен максимальному числу `n`, такому что автор имеет по крайней мере `n` публикаций, на каждую из которых ссылались по крайней мере `n` раз.

5.3.2 Типовые задания (оценочное средство - Контрольная работа) для оценки сформированности компетенции ПК-4

Напишите функцию `concatS :: [ShowS] -> String`, которая возвращает конкатенацию всех строк, представленных разностными списками. Определение должно использовать `foldMap` и моноид эндоморфизмов.

Критерии оценивания (оценочное средство - Контрольная работа)

| Оценка | Критерии оценивания |
|------------|----------------------------|
| зачтено | Решено не менее 60% задач. |
| не зачтено | Решено менее 60% задач. |

6. Учебно-методическое и информационное обеспечение дисциплины (модуля)

Основная литература:

1. Душкин Р.В. Функциональное программирование на языке Haskell : монография / Душкин Р.В. - Москва : ДМК-пресс, 2016. - 608 с. - ISBN 978-5-97060-362-8., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=868796&idb=0>.

Дополнительная литература:

1. Изучай Haskell во имя добра! / Липовача М. - Москва : ДМК-пресс, 2012., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=645083&idb=0>.
2. Душкин Р.В. Справочник по языку Haskell : справочник / Душкин Р.В. - Москва : ДМК-пресс, 2016. - 540 с. - ISBN 978-5-97060-361-1., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=868761&idb=0>.

Программное обеспечение и Интернет-ресурсы (в соответствии с содержанием дисциплины):

Загрузка инструментов для работы с Haskell. URL: <https://www.haskell.org>.
Стандартная библиотека языка Haskell. URL: <https://hackage.haskell.org/package/base>.
Стандарт языка Haskell 2010. URL: <https://www.haskell.org/onlinereport/haskell2010>.
Сборник статей по различным вопросам языка Haskell. URL: <https://wiki.haskell.org/Haskell>.
Краткий обзор синтаксиса Haskell. URL: <http://www.cse.chalmers.se/edu/year/2014/course/TDA452/haskell-syntax.html>.

7. Материально-техническое обеспечение дисциплины (модуля)

Учебные аудитории для проведения учебных занятий, предусмотренных образовательной программой, оснащены мультимедийным оборудованием (проектор, экран), техническими средствами обучения, компьютерами.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечены доступом в электронную информационно-образовательную среду.

Программа составлена в соответствии с требованиями ОС ННГУ по направлению подготовки/специальности 01.04.02 - Прикладная математика и информатика.

Автор(ы): Макаров Евгений Маратович.

Программа одобрена на заседании методической комиссии от 13.12.2023, протокол № 3.