

MINISTRY OF SCIENCE AND HIGHER EDUCATION OF THE RUSSIAN FEDERATION

**Federal State Autonomous Educational Institution of Higher Education
«National Research Lobachevsky State University of Nizhny Novgorod»**

Институт информационных технологий, математики и механики

УТВЕРЖДЕНО
решением Ученого совета ННГУ
протокол № 10 от 02.12.2024 г.

Working programme of the discipline

Functional programming

Higher education level

Master degree

Area of study / speciality

02.04.02 - Fundamental Informatics and Information Technology

Focus /specialization of the study programme

Artificial Intelligence and Data Analysis

Mode of study

full-time

Nizhny Novgorod

Year of commencement of studies 2025

1. Место дисциплины в структуре ОПОП

Дисциплина Б1.О.06 Функциональное программирование относится к обязательной части 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-3: Способен эксплуатировать, разрабатывать и развивать системное и прикладное программное обеспечение, новые информационные технологии на основе анализа современного состояния науки и информационных технологий, и управлять такой эксплуатацией и разработкой в области профессиональной деятельности	<p>ПК-3.1: Знает основы ИТ и имеет навыки анализа современного состояния науки и информационных технологий в области профессиональной деятельности</p> <p>ПК-3.2: Умеет применять навыки проектирования и разработки и развития ИТ-решений на основе анализа современного состояния науки и информационных технологий в области профессиональной деятельности</p> <p>ПК-3.3: Имеет практический опыт управления разработкой и развитием ИТ-решений на основе анализа современного состояния науки и информационных технологий в области профессиональной деятельности</p>	<p>ПК-3.1: Знать особенности, преимущества и недостатки парадигмы функционального программирования, ее место в современной программной инженерии. Знать способы разработки ПО на языке Haskell. Знать синтаксис и основные конструкции языка Haskell.</p> <p>ПК-3.2: 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. Уметь анализировать сложность программ.</p> <p>ПК-3.3: Владеть навыками чтения сообщения об ошибках и исправления ошибок, навыками чтения и составления документации. Владеть способами планирования времени для успешного выполнения еженедельных домашних</p>	Тест Задания	Зачёт: Контрольная работа

		заданий.		
--	--	----------	--	--

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			
		Контактная работа (работа во взаимодействии с преподавателем), часы из них			Самостоятельная работа обучающегося, часы
		Занятия лекционного типа	Занятия семинарского типа (практические занятия/лабораторные работы), часы	Всего	
0 Ф 0	0 Ф 0	0 Ф 0	0 Ф 0	0 Ф 0	
1. Основы программирования на Haskell / Introduction to Haskell programming.	54	8	8	16	38
2. Лямбда-исчисление с типами / Lambda-calculus with types.	27	4	4	8	19
3. Определение новых типов и классов типов / Declaration of new types and type classes.	26	4	4	8	18
Аттестация	0				
КСР	1			1	
Итого	108	16	16	33	75

Contents of sections and topics of the discipline

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. Бинарные операторы и их сечения. Рекурсия. Ленивые вычисления. Арифметические прогрессии. Генераторы списков. Двумерный синтаксис. Бесточечная запись. Функции высших порядков. Левая и правая свертки.

Installing and using the interpreter. The main data types and their values. Declaring functions. Constructions let and where. Guards. Pattern matching. Partially applied functions. Binary operators and their sections. Recursion. Laziness. Arithmetic progressions. List comprehension. Two-dimensional syntax. Point-free notation. Higher-order functions. Left and right folds.

2. Лямбда-исчисление с простыми типами. Редукции и конверсии. Параметрический полиморфизм. Нахождение типовых ошибок.

Lambda-calculus with simple types. Reductions and conversions. Parametric polymorphism. Finding type errors.

3. Классы типов. Алгебраические типы. Синонимы типов (type) и обертки вокруг типов (newtype).

Объявление классов типов и их экземпляров. Класс типов Monoid и его элементы. Класс Foldable.

Type classes. Algebraic types. Type synonyms (type) and type wrappers (newtype). Declaring class types and their instances. Type class Monoid and its elements. Type class Foldable.

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.

Online Haskell course, part 1: <https://haskell.mooc.fi/part1>

Online Haskell course, part 2: <https://haskell.mooc.fi/part2>

5. Assessment tools for ongoing monitoring of learning progress and interim certification in the discipline (module)

5.1 Model assignments required for assessment of learning outcomes during the ongoing monitoring of learning progress with the criteria for their assessment:

5.1.1 Model assignments (assessment tool - Test) to assess the development of the competency ПК-3:

1. How many values does `f x = [x,x]` return?

- (a) Zero
- (b) One
- (c) Two

2. Why does the expression `Nothing 1` cause a type error?

- (a) Because `Nothing` takes no arguments
- (b) Because `Nothing` returns nothing
- (c) Because `Nothing` is a constructor

3. What is the type of the function `f x y = if x && y then Right x else Left "foo"`?

- (a) `Bool -> Bool -> Either Bool String`

- (b) String -> String -> Either String String
- (c) Bool -> Bool -> Either String Bool

Assessment criteria (assessment tool — Test)

Grade	Assessment criteria
pass	Владение основным и дополнительным материалом достаточное или с незначительными ошибками и погрешностями. Knowledge of the basic and additional material is sufficient or with minor errors and mistakes.
fail	Владение материалом, необходимым по данному предмету, недостаточно. Knowledge of the material required for this subject is insufficient.

5.1.2 Model assignments (assessment tool - Assignments) to assess the development of the competency ПК-3:

 -- Exercise 1: implement using recursion a function sumTo such that
 -- sumTo n
 -- computes the sum 1+2+...+n

sumTo :: Integer -> Integer
 sumTo = undefined

 -- Exercise 2: a variant of safe division. you should use
 -- Either to return a string error message.
 --
 -- Examples:
 -- eitherDiv 4 2 ==> Right 2
 -- eitherDiv 4 0 ==> Left "4/0"

eitherDiv :: Integer -> Integer -> Either String Integer
 eitherDiv x y = undefined

 -- Exercise 3: recall the withdraw example from the course material. Write a
 -- similar function, transfer, that transfers money from one account
 -- to another.
 --
 -- However, the function should not perform the transfer if
 -- * the from account doesn't exist,
 -- * the to account doesn't exist,
 -- * the sum is negative,
 -- * or the from account doesn't have enough money.
 --
 -- Hint: there are many ways to implement this logic. Map.member or

```
-- Map.notMember might help.
--
-- Examples:
-- let bank = Map.fromList [("Bob",100),("Mike",50)]
-- transfer "Bob" "Mike" 20 bank
-- ==> fromList [("Bob",80),("Mike",70)]
-- transfer "Bob" "Mike" 120 bank
-- ==> fromList [("Bob",100),("Mike",50)]
-- transfer "Bob" "Lisa" 20 bank
-- ==> fromList [("Bob",100),("Mike",50)]
-- transfer "Lisa" "Mike" 20 bank
-- ==> fromList [("Bob",100),("Mike",50)]
```

```
transfer :: String -> String -> Int -> Map.Map String Int -> Map.Map String Int
transfer from to amount bank = undefined
```

Assessment criteria (assessment tool — Assignments)

Grade	Assessment criteria
pass	Задача решена полностью или решена основная часть задачи, или задача решена с недочетами. The problem is solved completely, or the main part of the problem is solved, or the problem is solved with some shortcomings.
fail	Задача не решена или сделан только первый этап решения задачи. The problem has not been solved or only the first stage of solving the problem has been completed.

5.2. Description of scales for assessing learning outcomes in the discipline during interim certification

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	При решении	Продемонс	Продемонс	Продемонс	Продемонс	Продемонстр

	минимальных умений. Невозможность оценить наличие умений вследствие отказа обучающегося от ответа	стандартных задач не продемонстрированы основные умения. Имели место грубые ошибки	трированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном объеме	трированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания в полном объеме, но некоторые с недочетами	трированы все основные умения. Решены все основные задачи. Выполнены все задания в полном объеме, но некоторые с недочетами	трированы все основные умения. Решены все основные задачи с отдельными и несущественными недочетами, выполнены все задания в полном объеме	трированы 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 и недочетами	Продемонстрированы базовые навыки при решении стандартных задач с некоторыми и недочетами	Продемонстрированы базовые навыки при решении стандартных задач без ошибок и недочетов	Продемонстрированы навыки при решении нестандартных задач без ошибок и недочетов	Продемонстрирован творческий подход к решению нестандартных задач

Scale of assessment for interim certification

Grade		Assessment criteria
pass	outstanding	All the competencies (parts of competencies) to be developed within the discipline have been developed at a level no lower than "outstanding", the knowledge and skills for the relevant competencies have been demonstrated at a level higher than the one set out in the programme.
	excellent	All the competencies (parts of competencies) to be developed within the discipline have been developed at a level no lower than "excellent",
	very good	All the competencies (parts of competencies) to be developed within the discipline have been developed at a level no lower than "very good",
	good	All the competencies (parts of competencies) to be developed within the discipline have been developed at a level no lower than "good",
	satisfactory	All the competencies (parts of competencies) to be developed within the discipline have been developed at a level no lower than "satisfactory", with at least one competency developed at the "satisfactory" level.
fail	unsatisfactory	At least one competency has been developed at the "unsatisfactory" level.
	poor	At least one competency has been developed at the "poor" level.

5.3 Model control assignments or other materials required to assess learning outcomes during the interim certification with the criteria for their assessment:

5.3.1 Model assignments (assessment tool - Control work) to assess the development of the competency ПК-3

```
-- Given numbers start, count and end, build a list that starts  
-- with count copies of start and ends with end.  
--  
-- Use recursion and the : operator to build the list.  
--  
-- Examples:  
-- buildList 1 5 2 ==> [1,1,1,1,1,2]  
-- buildList 7 0 3 ==> [3]
```

```
buildList :: Int -> Int -> Int -> [Int]  
buildList start count end = undefined
```

Assessment criteria (assessment tool — Control work)

Grade	Assessment criteria
pass	Все задачи решены полностью или решение задач обосновано, но допущены незначительные ошибки. All problems are solved completely or the solution to the tasks is justified, but minor errors were made.
fail	Задачи не решены или сделан только первый этап в решении задач. The problems are not solved or only the first stage has been completed.

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

Основная литература:

1. Липовача М. Изучай Haskell во имя добра! : монография / Липовача М. - Москва : ДМК-пресс, 2023. - 492 с. - ISBN 978-5-89818-338-7., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=878899&idb=0>.

Дополнительная 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>.

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

The Haskell Language. URL: <https://www.haskell.org>.

Haskell standard library. URL: <https://hackage.haskell.org/package/base>.

Haskell 2010 language standard. URL: <https://www.haskell.org/onlinereport/haskell2010>.

Haskell Wiki. URL: <https://wiki.haskell.org/Haskell>.

Tour of the Haskell Syntax. 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 02.04.02 - Fundamental Informatics and Information Technology.

Авторы: Макаров Евгений Маратович.

Программа одобрена на заседании методической комиссии от 02.12.2024, протокол № 5.