

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ

Проректор по учебной работе и
образовательным инновациям

 О.И.Чуприс
2019 г.

Регистрационный № УД- 7118 / уч.

ФУНКЦИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ

**Учебная программа учреждения высшего образования
по учебной дисциплине для специальности первой ступени высшего
образования:**

**1-31 03 07 Прикладная информатика (по направлениям)
направления специальности**

**1-31 03 07 - 01 Прикладная информатика (программное
обеспечение компьютерных систем)**

2019 г.

Учебная программа составлена на основе ОСВО 1-31 03 07-2013 и учебных планов G31-167/уч. от 30.05.2013, G31и-198/уч. от 30.05.2013.

СОСТАВИТЕЛЬ:

Курбацкий А.Н., профессор кафедры технологий программирования факультета прикладной математики и информатики Белорусского государственного университета, доктор технических наук;

Винник В.Ю., инженер-программист ООО «Солавиндз МСП Технолоджи», кандидат физико-математических наук.

РЕЦЕНЗЕНТ:

Дравица В.И., директор Государственного предприятия «Центр систем идентификации», канд. физ.-мат. наук

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

Кафедрой технологий программирования
(протокол № 12 от 16 мая 2019 г.).

Советом факультета прикладной математики и информатики
(протокол № 7 от 21 мая 2019 г.).

Зав. кафедрой,
профессор



А.Н.Курбацкий



ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Цели и задачи учебной дисциплины

Цель преподавания учебной дисциплины – овладение студентами бурно развивающимся и набирающим популярность стилем программирования, подготовка к его практическому использованию.

Особое внимание уделено общезначимым, независимым от конкретного языка, принципам, методам и понятиям ФП, что позволит студентам применять полученные знания при работе на множестве различных языков.

Задачи учебной дисциплины:

- дать студентам основу, необходимую для успешного усвоения перспективных технологий и методов программирования;
- дать студентам более глубокую и полную картину программирования за счёт альтернативной точки зрения на программы и процессы их выполнения;
- приобретенные знания позволяют понять современные тенденции в языках программирования, не относящихся к ФП (например, популярность Linq в платформе .Net, введение лямбда-выражений в язык C++ и т. д.);
- получить знания, необходимые им в дальнейшем для успешной работы в качестве разработчика программного обеспечения либо руководителя проектом в области ИКТ.

Место учебной дисциплины в системе подготовки специалиста с высшим образованием.

Учебная дисциплина относится к циклу дисциплин специализации 1-31 03 07-01 06 Программное обеспечение встроенных систем, компонента учреждения высшего образования.

Программа составлена с учетом **межпредметных связей** с учебными дисциплинами. Основой для изучения учебной дисциплины являются следующие учебные дисциплины первой степени высшего образования: «Программирование», «Дискретная математика и математическая логика», «Технологии программирования».

Учебная дисциплина «Функциональное программирование» знакомит студентов с парадигмой функционального программирования (ФП) в общем виде, её особенностями, выгодно отличающими её от императивной и объектно-ориентированной парадигм, языком Haskell как наиболее развитым и современным языком ФП, связанной с ФП парадигмой реактивного программирования, а также с математическими основаниями ФП: λ-исчислением и теорией категорий.

Методы, излагаемые в курсе, являются дополнением к углубленному изучению базовых дисциплин специализации. Так, на базе ФП строятся синтаксические анализаторы, средства обработки текстовых данных, а также в системы с повышенными требованиями к корректности.

Требования к компетенциям

Освоение учебной дисциплины «Функциональное программирование» должно обеспечить формирование следующих академических, социально-личностных и профессиональных компетенций:

академические компетенции:

- АК-1. Уметь применять базовые научно-теоретические знания для решения теоретических и практических задач.
- АК-2. Владеть системным и сравнительным анализом.
- АК-3. Владеть исследовательскими навыками.
- АК-5. Быть способным порождать новые идеи (обладать креативностью).
- АК-6. Владеть междисциплинарным подходом при решении проблем.
- АК-7. Иметь навыки, связанные с использованием технических устройств, управления информацией и работой с компьютером.
- АК-9. Уметь учиться, повышать свою квалификацию в течение всей жизни.

социально-личностные компетенции:

- СЛК-2. Быть способным к социальному взаимодействию.
- СЛК-3. Обладать способностью к межличностным коммуникациям.
- СЛК-5. Быть способным к критике и самокритике.
- СЛК-6. Уметь работать в команде.

профессиональные компетенции:

- ПК-1. Проектировать, разрабатывать и тестировать программное обеспечение различных видов.
- ПК-2. Разрабатывать техническую документацию на программное обеспечение.
- ПК-5. Проектировать, разрабатывать, внедрять и тестировать насыщенные Интернет приложения.

В результате освоения учебной дисциплины студент должен:

знать:

- главные характеристики парадигмы ФП, отличающие её от других распространённых парадигм;
- основные вехи истории ФП, особенности главных языков ФП (Lisp, F#, ML, Refal, Haskell);
- карринг (преобразование функции многих аргументов в функцию высшего порядка);
- ленивый порядок вычислений;
- рекурсивный подход к определению алгоритмов, особенности хвостовой рекурсии;
- операции свёртки и развёртки, их применение для решения прикладных задач;
- алгебраические типы данных, бесконечные структуры данных;

- понятие класса типов; классы типов в стандартной библиотеке языка Haskell;
- программирование в терминах функторов и аппликативных функторов;
- программирование с использованием монад;
- основы λ -исчисления и теории категорий, связанные с ФП.

уметь:

- осуществлять разработку программного продукта по заданной спецификации на языке ФП;
- преобразовывать постановку задачи к виду, удобному для реализации в функциональном стиле;
- производить декомпозицию задач в терминах функций, свёрток, отображений, функторов, монад и других понятий ФП;
- находить в стандартной библиотеке и сторонних модулях функции, пригодные для повторного использования в своих задачах.

владеть:

- компилятором GHC, диалоговым интерпретатором GHCi;
- системами управления проектами cabal и stack;
- репозиторием пакетов hackage;
- системой поиска функций по сигнатуре hooogle.

научно-исследовательская деятельность

- ПК-7. Применять профессиональные знания и навыки для проведения научных исследований в области прикладной информатики.
- ПК-9. Работать с научно-технической информацией с использованием современных информационных технологий.
- ПК-10. Формулировать выводы и рекомендации по применению результатов научно-исследовательской работы.

эксплуатационная деятельность

- ПК-12. На основе технической документации выполнять внедрение и сопровождение программного обеспечения, в том числе разработанного сторонними организациями.

экспертно-консультационная деятельность

- ПК-18. Оказывать консультации по вопросам работы программного обеспечения, в том числе разработанного сторонними организациями.
- ПК-21. Анализировать результаты работы установленного программного обеспечения и выработать предложения по улучшению качества его работы.
- ПК-23. Проводить обучение специалистов, занимающихся эксплуатацией программного обеспечения.

взаимодействовать со специалистами смежных профилей

- ПК-33. Осуществлять поиск, систематизацию и анализ информации по перспективам развития отрасли, инновационным технологиям, проектам и решениям.
- ПК-34. Определять цели инноваций и способы их достижения.

социально-личностные

компетенции

- СЛК-2. Быть способным к социальному взаимодействию.
- СЛК-3. Обладать способностью к межличностным коммуникациям.
- СЛК-5. Быть способным к критике и самокритике.
- СЛК-6. Уметь работать в команде.

Структура учебной дисциплины

Дисциплина изучается в седьмом семестре. Всего на изучение учебной дисциплины «Функциональное программирование» отведено:

– для очной формы получения высшего образования – 158 часов, в том числе 68 аудиторных часа, из них: лекции – 34 часа, лабораторные занятия – 30 часов, управляемая самостоятельная работа – 4 часа.

Трудоемкость учебной дисциплины составляет 4 зачетные единицы.

Форма текущей аттестации – зачет, экзамен.

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Тема 1. Предмет и задачи дисциплины

Понятие парадигмы программирования. Основные принципы императивной и объектно-ориентированной парадигм, их характерные недостатки и сложности. Необходимость в иной парадигме как средство преодоления этих трудностей. Принципы функциональной парадигмы. Обзор истории функциональных языков. Современное состояние ФП. Краткие сведения о математических основаниях ФП.

Тема 2. Элементарные основы ФП

Базовые синтаксические конструкции и семантические элементы языка Haskell: определение значения и функции, применение функции к аргументам. Начальные сведения о сопоставлении с образцом. Понятие о чистой функции. Простейшие средства ввода-вывода.

Тема 3. Рекурсия

Создание рекурсивных реализаций для арифметических алгоритмов. Понятие хвостовой рекурсии и её преимущества перед рекурсией общего вида. Преобразование рекурсивных алгоритмов к хвостовому виду с помощью аккумулятора.

Тема 4. Списки и функции над ними

Тип списка. Строки как списки символов. Генераторы списков. Создание функций над списками, сопоставление аргумента-списка с образцом. Функции head и tail, last и init, их реализация своими руками. Конкатенация. Функции elem, length, drop, sum, maximum. Функция fmap. Списки кортежей. Бесконечные списки.

Тема 5. Система типов языка Haskell

Базовые типы. Простейшие конструкторы типов: кортеж и список. Типы функций. Переменные-типы. Понятие о классе типов. Классы Eq, Ord, Show, Read, Enum, Bounded. Начальные сведения об автоматическом выводе типов.

Тема 6. Элементы средней сложности

Функции высшего порядка. Карринг. Частичное применение функции к аргументам. Сечение бинарной операции. Выражения where и let. Сопоставление с образцом. Выражения if, case, охраняемые выражения. Лямбда-функции. Композиция функций.

Тема 7. Модули в языке Haskell

Импорт модулей. Обзор часто используемых стандартных модулей. Создание собственных модулей.

Тема 8. Алгебраические типы

данных

Создание собственных алгебраических типов данных. Конструкторы значений. Альтернативы. Синтаксис записей. Параметры типов. Конструкторы типов. Рекурсивные типы. Синонимы типов. Включение своего типа данных в имеющийся класс. Создание собственных классов типов.

Тема 9. Функторы и аппликативные функторы

Список как пример функтора. Общее понятие о функторе. Трактовка функтора как контейнера, управляющего преобразованием своих элементов. Класс Functor. Законы функторов. Функторы Identity, Empty, Maybe, Either. Тип функции как функтор. Потребность в обобщении понятия функтора на функции многих аргументов. Понятие о применении завернутой функции к завернутому аргументу. Аппликативные функторы и класс Applicative. Примеры. Законы аппликативных функторов.

Тема 10. Монады

Ввод-вывод как пример монады, решение проблемы чистоты функций. Понятие о монаде в языке Haskell. Списки, функции, Maybe, Either, Reader, Writer, State как монады. Законы монад.

Тема 11. Теоретические основания ФП: теория категорий

Теория категорий. Мотивация для развития такой теории. Базовые определения. Категория множеств и типов. Эпиморфизмы и мономорфизмы. Дуальность. Изоморфизмы. Инициальные и терминальные объекты. Сумма и произведение в категории типов. Алгебра типов данных с категорной точки зрения.

Тема 12. Теория категорий: функторы

Общее определение функтора. Категория категорий. Эндофункторы. Степени. Типы функций с категорной точки зрения. Декартово замкнутые категории. Категории Клейсли. Монады.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов					УСР	Форма контроля знаний
		Лекции	Практические	Семинарские	Лабораторные	Иное		
1	Предмет и задачи дисциплины.	2						Устный опрос
2	Элементарные основы ФП	2			2			Устный опрос. Отчёт по лабораторной работе
3	Рекурсия	2			2			Коллективные обсуждения. Отчёт по лабораторной работе
4	Списки и функции над ними	4			2			Устный опрос. Отчёт по лабораторной работе
5	Система типов языка Haskell	2			2			Устный опрос. Отчёт по лабораторной работе
6	Элементы средней сложности	4			4	2		Устный опрос. Отчёт по лабораторной работе
7	Модули в языке Haskell	2			4			Устный опрос. Отчёт по лабораторной работе
8	Алгебраические типы данных	2			4			Устный опрос. Отчёт по лабораторной работе

--	--	--	--	--	--	--	--	--

9	Функторы и аппликативные функторы	4			2			Устный опрос. Отчёт по лабораторной работе
10	Монады	4			4	2		Устный опрос. Отчёт по лабораторной работе
11	Теоретические основания ФП: теория категорий	4			2			Устный опрос. Отчёт по лабораторной работе
12	Теория категорий: функторы	2			2			Проект. Отчёт по лабораторной работе
ИТОГО		34			30	4		

ИНФОРМАЦИОННО- МЕТОДИЧЕСКАЯ ЧАСТЬ

Перечень основной литературы

1. Холomorphic, А. Учебник по Haskell. //Личный сайт автора [Электронный документ]. – 2012. – Режим доступа: <https://anton-k.github.io/ru-haskell-book/files/ru-haskell-book.pdf>. – Дата доступа: 18.05.2019.
2. Липовача, М. Изучай Haskell во имя добра! – М.: ДМК Пресс, 2012. – 439 с.
Режим доступа: <http://learnyouahaskell.com> (Lipovača M. Learn you a Haskell for great good! A beginner's guide).
3. Milewski, В. Category Theory for Programmers. // Open domain [Электронный документ]. – 2017. – Режим доступа: <https://github.com/hmemcru/milewski-ctfp-pdf>. – Дата доступа: 12.04.2019.

Перечень дополнительной литературы

1. Lemmer, R. Haskell Design Patterns. / R. Lemmer. – Packt, 2015. – 166 p.
2. Bird R. Thinking Functionally in Haskell. / R. Bird. – Cambridge university, 2015. – 358 p.
3. O'Sullivan, В. Real Worls Haskell. / В. O'Sullivan, J. Goerzen, D. Stewart. – O'Reilly, 2009. – 712 p.

Перечень рекомендуемых средств диагностики и методика формирования итоговой оценки

Для аттестации обучающихся на соответствие их персональных достижений конечным требованиям образовательной программы создаются фонды оценочных средств, включающие типовые задания.

Для диагностики компетенций в рамках учебной дисциплины рекомендуется использовать следующие формы:

1. Устная форма: опросы, коллективные обсуждения.
2. Письменная форма: отчет по лабораторной работе, проект.

Формой текущей аттестации по дисциплине «Функциональное программирование» учебным планом предусмотрен зачет/экзамен.

При формировании итоговой оценки используется рейтинговая оценка знаний студента, дающая возможность проследить и оценить динамику процесса достижения целей обучения. Рейтинговая оценка предусматривает использование весовых коэффициентов для текущего контроля знаний и текущей аттестации студентов по дисциплине.

Примерные весовые коэффициенты, определяющие вклад текущего контроля знаний и текущей аттестации в рейтинговую оценку:

Формирование оценки за текущую успеваемость:

- ответы на лабораторных занятиях – 50 %;
- проект – 50 %.

Рейтинговая оценка по дисциплине рассчитывается на основе оценки текущей успеваемости и экзаменационной оценки с учетом их весовых коэффициентов. Вес оценка по текущей успеваемости составляет 40 %, экзаменационная оценка – 60 %.

Примерный перечень заданий для управляемой самостоятельной работы студентов

1. Реализация алгоритмов дискретной математики, синтаксического разбора, криптографии на языке Haskell (2 ч.).
2. Реализация на языке Haskell реактивного микросервиса (2 ч.).

Примерная тематика семинарских (практических, лабораторных) занятий

Лабораторная работа № 1. Знакомство с компилятором ghc, диалоговым интерпретатором ghci и др. инструментарием.

Лабораторная работа № 2.

Создание простейших функций. Каррированные функции, частичное применение. Сечение бинарных операций.

Лабораторная работа № 3. Создание рекурсивных функций. Хвостовая рекурсия. Преобразование рекурсии общевидового вида в хвостовую.

Лабораторная работа № 4. Работа со списками. Самостоятельная реализация основных функций над списками.

Лабораторная работа № 5. Операции свёртки и развёртки, их самостоятельная реализация и использование.

Лабораторная работа № 6. Создание более сложных функций над списками: сортировка, генерация всех перестановок и всех подсписков.

Лабораторная работа № 7. Работа с бесконечными списками.

Лабораторная работа № 8. Операция неподвижной точки. Сведение рекурсий к неподвижным точкам.

Лабораторная работа № 9. Алгебраические типы данных. Реализация двоичных деревьев и функций для их обработки.

Лабораторная работа № 10. Функторы и аппликативные функторы. Создание собственных функторных и аппликативных типов.

Лабораторная работа № 11. Монады: использование стандартных монад.

Лабораторная работа № 12. Монады: создание собственных монад.

Лабораторная работа № 13. Ввод-вывод, работа с консолью и файловой системой.

Лабораторная работа № 14. Изменяемое состояние и императивное программирование на языке Haskell.

Лабораторная работа № 15. Параллельное программирование на языке Haskell.

Рекомендуемый сквозной проект для самостоятельной тренировки: Сжатие данных методом Хаффмана.

Текущий контроль знаний проводится в соответствии с учебно-методической картой дисциплины.

Описание инновационных подходов и методов к преподаванию учебной дисциплины (эвристический, проективный, практико-ориентированный)

Использование эвристического подхода, который предусматривает творческую самореализацию обучающихся в процессе создания образовательных продуктов, демонстрацию многообразия решений большинства профессиональных задач.

Использование практико-ориентированного подхода, который предполагает освоение материала дисциплины через решения практических задач, приобретение навыков эффективного выполнения разных видов профессиональной деятельности.

На лекционных занятиях по дисциплине «Функциональное программирование» рекомендуется использовать элементы проблемного обучения.

На лабораторных занятиях рекомендуется отрабатывать изложенные на лекциях аспекты функционального программирования на конкретных примерах. Кроме этого рекомендуется развивать навыки командной работы при создании комплексного проекта.

Методические рекомендации по организации самостоятельной работы обучающихся

На занятиях по дисциплине специализации «Функциональное программирование» рекомендуется обратить особое внимание на современное состояние и перспективы развития функционального программирования и на проникновение его элементов в языки, относящиеся к другим парадигмам.

Условием для самостоятельной работы студентов являются наличие научно-методического обеспечения (перечни заданий и контрольных мероприятий УСР, учебная литература, мультимедийные и видеоматериалы), доступ к электронным информационным ресурсам, размещенных на сервере факультета.

Примерный перечень вопросов к экзамену/зачету

1. Понятие о функциональной парадигме, её отличие от прочих парадигм программирования.
2. Понятие о чистой функции и референциальной прозрачности.
3. Карринг, частичное применение и функции высшего порядка.
4. Рекурсия. Понятие о хвостовой рекурсии, метод приведения рекурсии общего вида к хвостовому.
5. Операция неподвижной точки.
6. Алгебраические типы данных.
7. Классы типов в языке Haskell.
8. Понятия функтора и аппликативного функтора. Примеры из стандартной библиотеки.
9. Понятие монады. Примеры из стандартной библиотеки.
10. Функция как функтор и монада.
11. Монада State.
12. Монада ST.
13. Средства многопоточного программирования в языке Haskell.
14. Определения категории, мономорфизма, эпиморфизма.
15. Функторы и эндоморфизмы. Вложения. Подкатегории.
16. Инициальный и терминальный объекты.
17. Категория типов. Произведения и копроизведения.

**ПРОТОКОЛ
СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ**

Название учебной дисциплины, с которой требуется согласование	Название кафедры	Предложения об изменениях в содержании учебной программы учреждения высшего образования по учебной дисциплине	Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола)
Практика применения современного C++	Технологий программирования	Нет	Оставить содержание учебной дисциплины без изменения, протокол № 12 от 16.05.2019 г.
Безопасность информационных систем	Технологий программирования	Нет	Оставить содержание учебной дисциплины без изменения, протокол № 12 от 16.05.2019 г.

**ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ ПО
ИЗУЧАЕМОЙ УЧЕБНОЙ ДИСЦИПЛИНЕ**

на ____ / ____ учебный год

№ п/п	Дополнения и изменения	Основание

Учебная программа пересмотрена и одобрена на заседании кафедры технологий программирования (протокол № ____ от _____ 201_ г.)

Заведующий кафедрой технологий программирования _____

УТВЕРЖДАЮ
Декан ФПМИ
