

Аннотация рабочей программы дисциплины

Направление подготовки: 02.03.02 «Фундаментальная информатика и информационные технологии»

Тип образовательной программы: академический бакалавриат

Направленность (профиль): Информационная сфера

Квалификация (степень) выпускника: бакалавр

Форма обучения: очная

1. Наименование дисциплины

Б1.В.ДВ.7.1 Функциональное программирование

2. Цели и задачи дисциплины (модуля):

Цель дисциплины - получение навыков использования функциональных языков программирования и приобретения сведений о теоретических основах и принципах реализации функциональных языков программирования.

Задачи дисциплины – изучение основных приёмов создания программ на функциональных языках программирования на примере языка Норе; изучение теоретических основ функционального программирования - λ -исчисления; изучение принципов реализации функциональных языков программирования – Eval/Apply интерпретатора и SECD машины.

3. Требования к результатам освоения дисциплины (модуля):

Процесс изучения дисциплины направлен на формирование следующих профессиональных компетенций: ОПК-2 – способность применять в профессиональной деятельности современные языки программирования и языки баз данных, методологии системной инженерии, системы автоматизации проектирования, электронные библиотеки и коллекции, сетевые технологии, библиотеки и пакеты программ, современные профессиональные стандарты информационных технологий, ОПК-3 – способность к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям; ПК-3 – способность использовать современные инструментальные и вычислительные средства.

В результате изучения дисциплины студент должен:

Знать: основные принципы применения функциональных языков программирования, теоретические основы функционального программирования и принципы реализации функциональных языков программирования.

Уметь: программировать на языке Норе, определять пользовательские типы данных и функции для работы с ними, определять и использовать функции высшего порядка, использовать приёмы программирования основанные на возможностях ленивой реализации языка программирования.

Владеть: иметь представление о принципах интерпретации функциональных программ и способах их создания.

4. Объем дисциплины (модуля) и виды учебной работы

Вид учебной работы	Всего часов / зачетных единиц	Семестры			
		6			
Аудиторные занятия (всего)	88	88			

В том числе:	-	-	-	-	-
Практические занятия (ПЗ)	40	40			
Лабораторные работы (ЛР)	40	40			
Самостоятельная работа (всего)	119	119			
В том числе:	-	-	-	-	-
<i>Другие виды самостоятельной работы</i>	119	119			
Вид промежуточной аттестации (экзамен)	45	45			
Контактная работа (всего)	100	100			
Общая трудоемкость	часы	252	252		
	зачетные единицы	7	7		

5. Краткая характеристика содержания учебной дисциплины

Раздел I. Программирование на функциональных языках на примере Норе

Тема 1. Принципы функционального программирования

Функциональные (аппликативные) языки программирования. Полные и частичные функции. Композиция функций. Встроенные функции. Функциональность (прозрачность по ссылкам). Конструкции языка НОРЕ. Базовые типы данных Норе. Рекурсия. Объявляемые инфиксные операторы. Квалифицированные выражения.

Структура программы на Паскале. Расширения в Object Pascal по сравнению с классическим Паскалем. Модули Object Pascal (unit), их структура и использование.

Тема 2. Типы данных Норе

Определяемые пользователем типы данных. Полиморфные типы данных. Списки. Именованные типы. Определение функций над типами данных. Сопоставление с образцом. Символ "безразлично". Тип данных tree – двоичное дерево. Стили программирования на функциональных языках. Доказательства по индукции свойств функций.

Тема 3. Функции высшего порядка

Функции высшего порядка. Функции map и reduce. Анонимные функции (λ -выражения).

Тема 4. Виды вычислений

Виды вычислений. Вызов по необходимости. Энергичные и ленивые вычисления. Строгость функции. Энергичная и ленивая семантика. "Бесконечные" структуры данных. Функция from. Вычисления с неизвестными.

Раздел II. Теоретические основы функционального программирования

Тема 5. λ -исчисление

λ -исчисление - нотация для определения функций. λ -выражения. Применение функции. Карринг. Свободные и связанные переменные. Виды λ -выражений. Подстановки. α -конверсия. β -конверсия. η -конверсия. Обобщенные конверсии.

Тема 6. Вывод в λ -исчислении

Эквивалентность λ -выражений. Тождественность и эквивалентность. Свойство Лейбница. Алфавитная эквивалентность. Прямые конверсии. Нормальная форма. Порядки редукции. Теорема Чёрча–Россера. Теорема стандартизации. Разделение переменных. Схемы редукции и механизмы вызова. Рекурсия в λ -исчислении. Y-комбинатор. Слабая заголовочная нормальная форма (СЗНФ).

Тема 7. Чистое λ -исчисление

Синтаксис чистого λ -исчисления. Булевы константы. Булевы операции. Списки в чистом λ -исчислении. Натуральные числа в чистом λ -исчислении. λ -исчисление де Брейна.

Тема 8. Комбинаторы

Понятие комбинатора. Аппликативные выражения. Комбинаторная логика. Трансляция λ -выражений в комбинаторную форму. Определение функции абстрагирования. Функция трансляции λ -выражений. Дополнительные комбинаторы, упрощающие выражения КЛ. Оптимизация функции абстрагирования. Сравнение КЛ-представлений и λ -представлений.

Раздел III. Реализация функциональных языков

Тема 9. Вывод типов

Проверка типов. Вывод типов. Задачи системы вывода типов. Родовые переменные типа. Правила системы вывода типов. Алгоритм проверки типа W.

Тема 10. Промежуточные формы

Промежуточные формы представления функциональной программы. Примитивы выбора. Правила трансляции в промежуточный код. Трансляция составных данных. Трансляция сопоставления с образцом. Генерация дерева сопоставления (метод Ханта). Неперекрывающиеся образцы. Специализация (частный случай) образца. Однозначность набора образцов. Поиск наиболее подходящего образца. Литеральные образцы.

Тема 11. Eval/Apply интерпретатор

АТД Контекст. Энергичный интерпретатор. Определение APPLY. Обработка условных выражений. Ленивый интерпретатор. Задержка (рецепт). Вызов по необходимости.

Тема 12. SECD-машина – реализация на основе стеков

SECD-машина. Вычисление следующего состояния SECD-машины в энергичном варианте. Работа с условными выражениями. Представление рекурсии. Ленивая SECD-машина.

6. Форма промежуточной аттестации:

экзамен

7. Разработчик аннотации

Зав. кафедрой информационных технологий А.Е. Хмельнов