

## АННОТАЦИЯ К РАБОЧЕЙ ПРОГРАММЕ ДИСЦИПЛИНЫ «Функциональные языки разработки распределенных систем»

по основной профессиональной образовательной программе по направлению подготовки  
09.04.04 «Программная инженерия» (уровень магистратуры)

**Направленность (профиль):** Разработка программно-информационных систем

**Трудоемкость дисциплины** – 4 з.е. (144 часов)

**Форма промежуточной аттестации** – Экзамен.

**В результате освоения дисциплины обучающийся должен обладать следующими компетенциями:**

- - ПК-2: Владение методами программной реализации распределенных информационных систем;
- - ПК-6: Владение навыками организации промышленного тестирования создаваемого программного обеспечения;
- - ПК-11: Понимание существующих подходов к верификации моделей программного обеспечения;

**Содержание дисциплины:**

Дисциплина «Функциональные языки разработки распределенных систем» включает в себя следующие разделы:

**Форма обучения очная. Семестр 2.**

**1. Функциональные языки и методы реализации распределенных систем: обзор ключевых особенностей.** Понятие распределенной системы и функционального языка. Модели взаимодействий в распределенных системах. Современные функциональные языки (обзор). Применение функциональности к многопоточности и распределенному взаимодействию. Понятие акторной парадигмы программирования.

**2. Ключевые синтаксический и семантические особенности функциональных языков в применении к реализации распределенных задач. Язык Erlang. Промышленное тестирование программ на Erlang.** Понятие рекурсии. Реализация математических функций через рекурсию. Понятие хвостовой рекурсии. Понятия сопоставления (template matching). Рекурсивная работа со списками. Язык Erlang, машина BEAM и OTP (обзор). Реализация всего вышеперечисленного на языке Erlang. Применение подхода модульного тестирования.

**3. Процессо-ориентированное программирование. Акторные системы на Erlang.** Ожидание и отправка сообщений. Понятие почтового ящика (messagebox). Обработка сообщений сопоставлением. Реализация протокола или распределенной системы реального мира через передачу предопределенных сообщений..

**4. Распределенное программирование на функциональном языке Erlang. Подходы к моделированию и верификации корректной последовательности работы распределенной системы.** Отправка и получение сообщений на различные узлы. Создание распределенной системы, работающей на разных узлах с синхронизацией. Способы репликации решения. Использование языка Promela для создания модели акторной системы, а также ее верификация согласно заданным LTL-требованиям..

**5. Распределенное взаимодействие с кодом, реализованным на других языках. OTP фреймворк для создания отказоустойчивых распределенных программ.** Обмен через сокеты. Преимущества реализации серверов на Erlang. Взаимодействие с кодом на Java через Erlang-Java Interface. Сравнение императивной и функциональной парадигмы на этом примере. OTP фреймворк. Реализация скелета OTP приложения, включающего сервер, супервизор, логгер и клиента к нему..

**6. Введение в язык Haskell. Последовательный и ленивый код. Монады..** Описание ключевых особенностей языка. Абстракция последовательных вычислений. Чистые функции. Работа с бесконечными структурами данных на основе ленивости. Проблемы автоматического распараллеливания кода..

**7. Параллельное и распределенное программирование на Haskell. Генерация серверов на Erlang и Haskell по REST-описанию.** Программирование с помощью каналов и Cloud Haskell. Преимущества и недостатки с другими решениями в императивных языках и OTP. Генерация кода

серверов по их описанию и применение для создания серверов общего назначения и микросервисов.

Разработал:  
доцент  
кафедры ПМ  
Проверил:  
Декан ФИТ



С.М. Старолетов

А.С. Авдеев