

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Алтайский государственный технический университет им. И.И. Ползунова»

**СОГЛАСОВАНО**

Декан ФИТ  
Авдеев

А.С.

## **Рабочая программа дисциплины**

Код и наименование дисциплины: Б1.В.ДВ.2.1 «Функциональные языки разработки распределенных систем»

Код и наименование направления подготовки (специальности): 09.04.04

Программная инженерия

Направленность (профиль, специализация): Разработка программно-информационных систем

Статус дисциплины: элективные дисциплины (модули)

Форма обучения: очная

Статус	Должность	И.О. Фамилия
Разработал	доцент	С.М. Старолетов
Согласовал	Зав. кафедрой «ПМ»	Е.Г. Боровцов
	руководитель направленности (профиля) программы	С.М. Старолетов

г. Барнаул

## 1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с индикаторами достижения компетенций

Компетенция	Содержание компетенции	Индикатор	Содержание индикатора
ПК-1	Владение методами программной реализации распределенных информационных систем	ПК-1.1	Осуществляет выбор методов программной реализации распределенных информационных систем
		ПК-1.2	Создает программное обеспечение распределенных информационных систем
ПК-5	Понимание существующих подходов к верификации моделей программного обеспечения	ПК-5.1	Анализирует существующие подходы к верификации моделей программного обеспечения
		ПК-5.2	Применяет в профессиональной деятельности существующие подходы к верификации моделей программного обеспечения
ПК-10	Владение навыками организации промышленного тестирования создаваемого программного обеспечения	ПК-10.1	Анализирует и выбирает методы тестирования создаваемого программного обеспечения
		ПК-10.2	Осуществляет тестирование создаваемого программного обеспечения

## 2. Место дисциплины в структуре образовательной программы

Дисциплины (практики), предшествующие изучению дисциплины, результаты освоения которых необходимы для освоения данной дисциплины.	Программирование параллельных процессов
Дисциплины (практики), для которых результаты освоения данной дисциплины будут необходимы, как входные знания, умения и владения для их изучения.	Преддипломная практика

## 3. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающегося с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающегося

Общий объем дисциплины в з.е. /час: 4 / 144

Форма промежуточной аттестации: Экзамен

Форма обучения	Виды занятий, их трудоемкость (час.)				Объем контактной работы обучающегося с преподавателем (час)
	Лекции	Лабораторные работы	Практические занятия	Самостоятельная работа	
очная	16	32	0	96	57

**4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий**

**Форма обучения: очная**

**Семестр: 2**

**Лекционные занятия (16ч.)**

- 1. Введение. Обзор функционального программирования и функциональных языков {дискуссия} (2ч.)[1,4,6,7,8,10,11,12,13]** Методы программной реализации распределенных информационных систем. Программное обеспечение распределенных информационных систем. Существующие подходы к верификации моделей программного обеспечения. Организация промышленного тестирования создаваемого программного обеспечения. Выбор методов тестирования создаваемого программного обеспечения. Вычисление в терминах функций. Императивное и функциональное программирование. Функциональные особенности нефункциональных языков. Применение функционального программирования к решению высокопроизводительных задач. Обзор особенностей современных функциональных языков OCaml, F#, Scala и др.
- 2. Основы языка Erlang(2ч.)[1,2,8,14,15]** Общие сведения об Erlang. Установка Erlang. IDE для Erlang. Работа в интерпретаторе. Основы языка Erlang. Переменные. Грамматика. Списки. Функции. Атомы. Кортежи. Функторы. Рекурсия. Функциональный стиль. Модульное тестирование и документирование. Пример выполнения задания: умножение матриц.
- 3. Многопроцессорное и распределённое взаимодействие на Erlang(2ч.)[1,2,5,14,15]** Акторная парадигма. Общая характеристика процессов в Erlang. Создание процесса. Отправка и получение сообщений. Связи процессов и слежение за состоянием процесса. Словари процессов. Принципы параллельного программирования на Erlang. Анализ работы параллельных процессов в Erlang. Распределённое программирование. Моделирование работы протоколов на Erlang
- 4. Взаимодействие Erlang с Java {метод кейсов} (2ч.)[1,2,13,14]** JInterface - возможности. Подключение JInterface в проект на Java. Узел. Процесс. Типы данных. Пример работы (пинг-понг). Удалённый вызов функций из модулей Erlang. Возможное применение интерфейса Java-Erlang.
- 5. Верификация акторных распределённых систем(2ч.)[1]** Программа для проверки. Язык Promela и LTL-требования. Установка необходимых средств разработки. Реализация демо-модели. Верификация демо-модели. Верификация модели протокола PoET.
- 6. ОTR фреймворк {дискуссия} (2ч.)[1,2,14,15]** Проблемы логической

организации Erlang кода. Проблема написания отказоустойчивых приложений. Проблема горячей замены кода приложения без его остановки. Проблема развёртывания Erlang приложений. OTP фреймворк и шаблоны поведения. Application, supervisor, gen\_server и пример построения сервера. OTP gen\_event и логирование. Конечные автоматы в Erlang и OTP gen\_fsm. Организация горячей замены кода стандартными средствами и в OTP. Установка приложений OTP

7. Базовые сведения о Haskell(2ч.)[1,3,4,16,18] Общие сведения о Haskell. Установка Haskell. IDE для Haskell. Hello world на Haskell + базовые вещи о монадах. Функции и сопоставление в Haskell. Модульное тестирование. Списки и pattern matching в Haskell. Кортежи и pattern matching в Haskell. Собственные типы данных и операции, а также тестирование с quickCheck Ленивость в Haskell.

8. Параллелизм и межпроцессорное взаимодействие в Haskell(2ч.)[1,3,5,17,18] Сборка программы с включением параллельных инструкций. Создание параллельно выполняющихся действий. 0 транзакционной памяти.

Взаимодействие через каналы Control.Concurrent.Chan. Межпроцессорное взаимодействие в Haskell с помощью Cloud Haskell. Установка Cloud Haskell. Ping-pong на Cloud Haskell. Пример реализации распределённой взаимодействующей системы на Cloud Haskell. Генерация серверов на Erlang и Haskell и клиентов к ним по стандартизированному описанию.

### Лабораторные работы (32ч.)

1. Обзор функционального программирования и функциональных языков {использование общественных ресурсов} (4ч.)[1,4,6,7,10,11,12,13] Подготовить презентацию про один из функциональных языков по распределению преподавателя и его особенности. Оценить его применимость с точки зрения параллельного и распределённого программирования

2. Основы языка Erlang(4ч.)[1,2,4,14,15] Решение задачи по рекурсии и спискам.

Для выполнения задания необходимо продумать математическую постановку задачи, сделать рекурсивное решение на языке Erlang, при этом код должен быть функциональным и по- возможности, не содержать последовательного выполнения операторов. Должны быть написаны модульные тесты, покрывающие все основные случаи. Все промежуточные функции должны быть написаны самостоятельно

3. Многопроцессорное и распределённое взаимодействие на Erlang(4ч.)[1,2,5,9,14,15] Решение задания по параллельным процессам и распределённости в Erlang.

1. Выбрать задачу реального мира из нескольких (5-6) взаимодействующих процессов.

2. Промоделировать её на языке UML (диаграмма Activity).

3. Реализовать работу системы через сообщения и вывод состояния.

4. Запустить систему на нескольких работающих узлах
5. Установить возможность распараллеливания задания из прошлого раздела. Определить производительность (время решения) на больших исходных данных с использованием последовательного и параллельного подходов.
4. Взаимодействие Erlang с Java(4ч.)[1,2,14] Выполнение задания по интеграции с Java.  
Реализовать логику, аналогичную предыдущему заданию с использованием ввода/вывода в графические окна и JInterface на языках Java/Kotlin. Рекомендуется реализовать часть логики в виде JVM программы, которая взаимодействует с другими частями распределённой системы на Erlang через RPC и передачу/получение/регистрацию Pid
5. Верификация акторных распределённых систем(4ч.)[1,2,5] Реализовать модель для системы взаимодействующих процессов, реализованной ранее на языке Erlang, на языке Promela. Определить требования её корректной работы.
  2. Провести симуляцию и верификацию модели.
  3. При необходимости внести ошибки и продемонстрировать ошибочные последовательности.
6. OTP фреймворк {разработка проекта} (4ч.)[1,2,14,15]
  1. Портировать распределённую систему с акторами на OTP. Для этого логику программы создать при помощи gen\_server (возможно, с использованием gen\_fsm). Заменить все операции передачи сообщений на другие узлы на gen\_server:call. Заменить ввод-вывод на логгинг с помощью gen\_event. Распределённая система должна запускаться через запуск приложений OTP как каждое в своей консоли.
  2. Переделать логику работы, чтобы обработка запросов для одного или нескольких объектов по логике работы велась на одном из свободных рабочих узлов внутри приложения. Логика обработки запроса должна предусматривать сложные сценарии обработки (например, с использованием задания на рекурсию на больших данных) с потенциальной возможностью падения узла и большую загрузку процессора.
  3. Продемонстрировать работу распределённой системы при остановке узлов в процессе работы или падения сети или обработчика на переданных некорректных данных. Система должна переносить такие варианты корректно.
  4. Выделить важный функционал во взаимодействующем приложении с прошлого задания. Обеспечить возможность его горячей замены без остановки приложения и без побочных эффектов для его работы.
7. Базовые сведения о Haskell(4ч.)[1,3,16,18] Реализовать предложенный в первой работе для Erlang алгоритм решения рекурсивной задачи, но на языке Haskell. Выявить схожие черты и особенности реализации алгоритма для каждого из языков.
8. Параллелизм и межпроцессорное взаимодействие в Haskell {разработка проекта} (4ч.)[1,3,16,17,18]
  1. Реализовать параллельную систему,

спроектированную в лабораторной работе с процессами Haskell, при помощи каналов и с помощью Cloud Haskell.

2. Сгенерировать простой код серверов на Erlang и Haskell и разобраться в этом коде.

3. Реализовать гетерогенную систему на двух функциональных языках, взаимодействующую через REST-API.

#### Самостоятельная работа (96ч.)

1. Изучение особенностей одного из современных языков и подготовка презентации перед другими студентами {анализ казусов} (60ч.) [1,4,6,7,8,10,11,12,13]

2. Подготовка к экзамену (36ч.) [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18]

#### 5. Перечень учебно-методического обеспечения самостоятельной работы обучающихся по дисциплине

Для каждого обучающегося обеспечен индивидуальный неограниченный доступ к электронно-библиотечным системам: Лань, Университетская библиотека он-лайн, электронной библиотеке АлтГТУ и к электронной информационно-образовательной среде:

1. Старолетов С.М. Функциональные языки разработки распределенных систем [Электронный ресурс]: Учебное пособие.– Электрон. дан.– Барнаул: АлтГТУ, 2020.– Режим доступа: [http://elib.altstu.ru/eum/download/pm/Staroletov\\_FunLang\\_up.pdf](http://elib.altstu.ru/eum/download/pm/Staroletov_FunLang_up.pdf), авторизованный

#### 6. Перечень учебной литературы

##### 6.1. Основная литература

2. Чезарини, Ф. Проектирование масштабируемых систем с помощью Erlang/OTP / Ф. Чезарини, С. Виноски ; перевод с английского Д. В. Литовченко. – Москва : ДМК Пресс, 2017. – 486 с. – ISBN 978-5-97060-212-6. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/111436> (дата обращения: 23.06.2021). – Режим доступа: для авториз. пользователей.

3. Курт, У. Програмируй на Haskell : руководство / У. Курт ; перевод с английского Я. О. Касюевича. – Москва : ДМК Пресс, 2019. – 648 с. – ISBN 978-5-97060-694-0. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/123706> (дата обращения: 23.06.2021). – Режим доступа: для авториз. пользователей.

## 6.2. Дополнительная литература

4. Рубио-Санчес, М. Введение в рекурсивное программирование : руководство / М. Рубио-Санчес ; перевод с английского Е. А. Борисова. – Москва : ДМК Пресс, 2019. – 436 с. – ISBN 978-5-97060-703-9. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/131727> (дата обращения: 17.06.2021). – Режим доступа: для авториз. пользователей.

5. Крючкова Е. Н., Старолетов С.М. Программирование параллельных процессов [Электронный ресурс]: Учебное пособие.– Электрон. дан.– Барнаул: АлтГТУ, 2020.– Режим доступа: [http://elib.altstu.ru/eum/download/pm/Staroletov\\_ProgParProc\\_up.pdf](http://elib.altstu.ru/eum/download/pm/Staroletov_ProgParProc_up.pdf), авторизованный

6. Чукич, И. Функциональное программирование на C++ : учебное пособие / И. Чукич ; перевод с английского В. Ю. Винника, А. Н. Киселева. – Москва : ДМК Пресс, 2020. – 360 с. – ISBN 978-5-97060-781-7. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/140597> (дата обращения: 23.06.2021). – Режим доступа: для авториз. пользователей.

7. Душкин, Р. В. Квантовые вычисления и функциональное программирование / Р. В. Душкин. – Москва : ДМК Пресс, 2015. – 232 с. – ISBN 978-5-97060-275-1. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/97340> (дата обращения: 23.06.2021). – Режим доступа: для авториз. пользователей.

8. Авдошин, С. М. Дискретная математика. Формально-логические системы и языки / С. М. Авдошин, А. А. Набебин. – Москва : ДМК Пресс, 2018. – 390 с. – ISBN 978-5-97060-622-3. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/100912> (дата обращения: 23.06.2021). – Режим доступа: для авториз. пользователей.

9. Фред, Х. Изучай Erlang во имя добра! : справочник / Х. Фред ; перевод с английского Д. Литовченко. – Москва : ДМК Пресс, 2015. – 688 с. – ISBN 978-5-97060-086-3. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/93267> (дата обращения: 23.06.2021). – Режим доступа: для авториз. пользователей.

10. Юрич, С. Elixir в действии : руководство / С. Юрич ; перевод с английского Н. А. Нестеровой. – Москва : ДМК Пресс, 2020. – 376 с. – ISBN 978-5-97060-773-2. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/140587> (дата обращения: 23.06.2021). – Режим доступа: для авториз. пользователей.

11. Мински, Я. Программирование на языке Ocaml / Я. Мински, А. Мадхавапедди, Д. Хикки ; перевод с английского А. Н. Киселева. – Москва : ДМК Пресс, 2014. – 536 с. – ISBN 978-5-97060-102-0. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/82820> (дата обращения: 23.06.2021). – Режим

доступа: для авториз. пользователей.

12. Сошников, Д. В. Функциональное программирование на F# / Д. В. Сошников. – Москва : ДМК Пресс, 2011. – 192 с. – ISBN 978-5-94074-689-8. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/1274> (дата обращения: 23.06.2021). – Режим доступа: для авториз. пользователей.

13. Уорбэртон, Р. Лямбда-выражения в Java 8. Функциональное программирование – в массы / Р. Уорбэртон ; перевод с английского А. А. Слинкина. – Москва : ДМК Пресс, 2014. – 192 с. – ISBN 978-5-94074-919-6. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/93568> (дата обращения: 23.06.2021). – Режим доступа: для авториз. пользователей.

## 7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

14. Erlang documentation [Электронный ресурс]. – URL: <https://www.erlang.org/docs>

15. Armstrong J., Virding R., Wikstrom C., Williams R. Concurrent programming in ERLANG [Электронный ресурс]. – Prentice Hall International (UK) Ltd., 2015. URL: <https://erlang.org/download/erlang-book-part1.pdf>

16. Haskell/Category theory [Электронный ресурс]. – URL: [https://en.wikibooks.org/wiki/Haskell/Category\\_theory](https://en.wikibooks.org/wiki/Haskell/Category_theory)

17. Cloud Haskell [Электронный ресурс]. – URL: <https://haskell-distributed.github.io/documentation.html>

18. GHC Users' guide [Электронный ресурс]. – URL: [https://downloads.haskell.org/ghc/latest/docs/html/users\\_guide/](https://downloads.haskell.org/ghc/latest/docs/html/users_guide/)

## 8. Фонд оценочных материалов для проведения текущего контроля успеваемости и промежуточной аттестации

Содержание промежуточной аттестации раскрывается в комплекте контролирующих материалов, предназначенных для проверки соответствия уровня подготовки по дисциплине требованиям ФГОС, которые хранятся на кафедре-разработчике РПД в печатном виде и в ЭИОС.

## 9. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем

Для успешного освоения дисциплины используются ресурсы электронной информационно-образовательной среды, образовательные интернет-порталы, глобальная компьютерная сеть Интернет. В процессе изучения дисциплины происходит интерактивное

взаимодействие обучающегося с преподавателем через личный кабинет студента.

<b>№пп</b>	<b>Используемое программное обеспечение</b>
1	Eclipse IDE
2	Git
3	LibreOffice
4	Linux
5	Windows
6	Антивирус Kaspersky

<b>№пп</b>	<b>Используемые профессиональные базы данных и информационные справочные системы</b>
1	Национальная электронная библиотека (НЭБ) – свободный доступ читателей к фондам российских библиотек. Содержит коллекции оцифрованных документов (как открытого доступа, так и ограниченных авторским правом), а также каталог изданий, хранящихся в библиотеках России. ( <a href="http://нэб.рф/">http://нэб.рф/</a> )

#### **10. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине**

<b>Наименование специальных помещений и помещений для самостоятельной работы</b>
учебные аудитории для проведения учебных занятий
помещения для самостоятельной работы

Материально-техническое обеспечение и организация образовательного процесса по дисциплине для инвалидов и лиц с ограниченными возможностями здоровья осуществляется в соответствии с «Положением об обучении инвалидов и лиц с ограниченными возможностями здоровья».