

Министерство науки и высшего образования Российской Федерации  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Институт прикладной математики и компьютерных наук

УТВЕРЖДЕНО:  
Директор  
А. В. Замятин

Рабочая программа дисциплины

**Архитектура программных систем**

по направлению подготовки

**01.04.02 Прикладная математика и информатика**

Направленность (профиль) подготовки:  
**Информационная безопасность**

Форма обучения  
**Очная**

Квалификация  
**Магистр**

Год приема  
**2024**

СОГЛАСОВАНО:  
Руководитель ОП  
А.Ю. Матророва

Председатель УМК  
С.П. Сущенко

Томск – 2024

## **1. Цель и планируемые результаты освоения дисциплины**

Целью освоения дисциплины является формирование следующих компетенций:

ПК-1 Способен формализовать требования к программному обеспечению, спроектировать программное обеспечение, написать программный код, а также проверить работоспособность программного обеспечения и исправить дефекты.

Результатами освоения дисциплины являются следующие индикаторы достижения компетенций:

ИПК-1.1 Осуществляет анализ требований к программному обеспечению, построение формальной модели, проверку работоспособности программного обеспечения и исправление дефектов.

ИПК-1.2 Осуществляет разработку технических спецификаций на программные компоненты и их взаимодействие, разработку процедур проверки работоспособности и измерения характеристик программного обеспечения, разработку тестовых наборов данных.

ИПК-1.3 Осуществляет проектирование программного обеспечения, работу с системой контроля версий, рефакторинг и оптимизацию программного кода.

## **2. Задачи освоения дисциплины**

– Изучить основы архитектурного проектирования, изучение подходов к разработке сложных программных систем, методов организации устойчивого процесса разработки

– Изучить основные понятия и процессы, связанные с разработкой архитектур программных систем

– Освоить инструменты разработки программных систем

– Научиться применять типовые шаблоны архитектурного проектирования.

## **3. Место дисциплины в структуре образовательной программы**

Дисциплина относится к Блоку 1 «Дисциплины (модули)».

Дисциплина относится к части образовательной программы, формируемой участниками образовательных отношений, предлагается обучающимся на выбор. Дисциплина входит в модуль «Введение в моделирование систем искусственного интеллекта».

## **4. Семестр(ы) освоения и форма(ы) промежуточной аттестации по дисциплине**

Второй семестр, зачет

## **5. Входные требования для освоения дисциплины**

Для успешного освоения дисциплины требуются результаты обучения по следующим дисциплинам: «Введение в программную инженерию». Владение одним из распространённых языков программирования. Предпочтительно Java или Python.

## **6. Язык реализации**

Русский

## **7. Объем дисциплины**

Общая трудоемкость дисциплины составляет 3 з.е., 108 часов, из которых:

-лекции: 16 ч.

-лабораторные: 16 ч.

Объем самостоятельной работы студента определен учебным планом.

## **8. Содержание дисциплины, структурированное по темам**

Тема 1. История развития архитектур.

Ранний этап и мейнфреймы.

Эпоха клиент-серверных систем.

Переход к сервисной и микросервисной архитектурам.

Роль одноранговой архитектуры.

Тема 2. Распределённые и одноранговые системы.

общая характеристика распределённых систем, их сильные и слабые стороны.

Apache Hadoop и Apache Spark.

Общая характеристика одноранговых систем.

Bitcoin, DHT и другие примеры.

Теорема CAP.

Тема 3. Сервис-ориентированная архитектура.

общая характеристика сервис-ориентированной архитектуры.

Микросервисная архитектура.

Технологии микросервисной архитектуры.

Брокеры сообщений для микросервисной архитектуры.

Тема 4. Архитектуры операционных систем.

Требования к архитектуре и компоненты операционных систем.

Монолитная, микроядерная, гибридная и эгзоядерная архитектуры.

Примеры операционных систем для различных архитектур.

## **9. Текущий контроль по дисциплине**

Текущий контроль по дисциплине проводится путем контроля посещаемости, проведения контрольных работ, проверки лабораторных работ. По итогу лабораторной работы студентом оформляется отчёт и защищается перед преподавателем. По результатам выполнения работ и итоговой защиты, проставляется аттестация.

Оценочные материалы текущего контроля размещены на сайте ТГУ в разделе «Информация об образовательной программе» - <https://www.tsu.ru/sveden/education/eduop/>.

## **10. Порядок проведения и критерии оценивания промежуточной аттестации**

Промежуточная аттестация проводится в форме зачета во втором семестре. Студент получает зачет в случае, если он сдал все лабораторные работы и прошел итоговое тестирование, проверяющие компетенции ИПК-1.1, ИПК-1.2, ИПК-1.3.

Оценочные материалы для проведения промежуточной аттестации размещены на сайте ТГУ в разделе «Информация об образовательной программе» - <https://www.tsu.ru/sveden/education/eduop/>.

## **11. Учебно-методическое обеспечение**

а) Электронный учебный курс по дисциплине среде электронного обучения iDO - <https://lms.tsu.ru/course/view.php?id=32475>

б) Оценочные материалы текущего контроля и промежуточной аттестации по дисциплине.

## **12. Перечень учебной литературы и ресурсов сети Интернет**

а) основная литература:

– Мартин, Роберт Чистый код: создание, анализ и рефакторинг / Роберт Мартин. — М. : Питер, 2018. — 464 с.

- Мартин, Роберт Чистая архитектура. Искусство разработки программного обеспечения / Роберт Мартин. — М. : Питер, 2018. — 352 с.
- Макконнелл, Стив Совершенный код / Стив Макконнелл. — М. : БХВ-Петербург, 2017. — 896 с.
- Fairbanks, G. H. Just Enough Software Architecture: A Risk-Driven Approach / G. H. Fairbanks. — 1. — Marshall & Brainerd, 2010. — 376 с.
- Хант, Эндрю Программист-прагматик / Эндрю Хант, Дэвид Томас. — М. : Вильямс, 2020. — 368 с.
- Симан, Марк Внедрение зависимостей на платформе .NET / Марк Симан, Стивен ван Дерсен. — М. : Питер, 2021. — 608 с.

б) дополнительная литература:

- Соммервилл И. Инженерия программного обеспечения. – М.: Издательский дом «Вильямс», 2002. 624 с.

[URL://https://studizba.com/files/show/djvu/3009-1-i-sommervill--inzheneriya-programmnogo.html](https://studizba.com/files/show/djvu/3009-1-i-sommervill--inzheneriya-programmnogo.html)

- Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. — СПб.: Питер, 2002. 492 с.

[URL://https://ru.z-library.rs/book/571750/58d3ac/Унифицированный-процесс-разработки-программного-обеспечения.html](https://ru.z-library.rs/book/571750/58d3ac/Унифицированный-процесс-разработки-программного-обеспечения.html)

- Поллис Г., Огастин Л., Лоу К., Мадхар Д. Разработка программных проектов: на основе Rational Unified Process (RUP) – М.: ООО «Бином-Пресс», 2005.

[URL://https://cat.gpntb.ru/?id=EC/ShowFull&irbDb=ESVODT&bid=6acf4042ffce6f354d68d7fd130926b7](https://cat.gpntb.ru/?id=EC/ShowFull&irbDb=ESVODT&bid=6acf4042ffce6f354d68d7fd130926b7)

- Левинсон Дж. Тестирование ПО с помощью Visual Studio 2010. – Эком, 2012. 336с.

[URL://https://www.labirint.ru/books/315271/](https://www.labirint.ru/books/315271/)

- Ауэр К., Миллер Р. Экстремальное программирование: постановка процесса. С первых шагов и до победного конца. – СПб.: Питер, 2004.

[URL://https://search.rsl.ru/ru/record/01002092742](https://search.rsl.ru/ru/record/01002092742)

- Бек К. Экстремальное программирование: разработка через тестирование. – СПб.: Питер, 2017. 291с.

[URL://https://library.eol.pw/Разработка%20ПО/Кент%20Бек%20-%20Экстремальное%20программирование.pdf](https://library.eol.pw/Разработка%20ПО/Кент%20Бек%20-%20Экстремальное%20программирование.pdf)

- Ларман К. Применение UML и шаблонов проектирования. (второе издание) – М.: Вильямс, 2002. Или 3-е издание, 2019.

[URL://https://search.rsl.ru/ru/record/01010829682](https://search.rsl.ru/ru/record/01010829682)

- Арлоу Дж., Нейштадт А. UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование, 2-е изд. – СПб.: Символ-Плюс, 2008. 624 с.

[URL://https://k0d.cc/storage/books/UML/uml\\_2\\_i\\_unifitsirovannyi\\_protsses\\_2-e\\_izd.pdf](https://k0d.cc/storage/books/UML/uml_2_i_unifitsirovannyi_protsses_2-e_izd.pdf)

в) ресурсы сети Интернет:

- Программная инженерия. Анализ, моделирование, проектирование

URL: <https://stepik.org/175415>

- Основы Agile

URL: <https://stepik.org/183476>

- Современный Agile для Разработчиков [и Менеджеров]

URL: <https://stepik.org/203018>

### 13. Перечень информационных технологий

- а) лицензионное и свободно распространяемое программное обеспечение:

- MS Windows или Linux; IDE JetBrains Rider, Git, Jenkins.
- публично доступные облачные технологии (Google Docs, Яндекс диск и т.п.).

б) информационные справочные системы:

- Электронный каталог Научной библиотеки ТГУ – <http://chamo.lib.tsu.ru/search/query?locale=ru&theme=system>
- Электронная библиотека (репозиторий) ТГУ – <http://vital.lib.tsu.ru/vital/access/manager/Index>
- ЭБС Лань – <http://e.lanbook.com/>
- ЭБС Консультант студента – <http://www.studentlibrary.ru/>
- Образовательная платформа Юрайт – <https://urait.ru/>
- ЭБС ZNANIUM.com – <https://znanium.com/>
- ЭБС IPRbooks – <http://www.iprbookshop.ru/>

в) профессиональные базы данных (*при наличии*):

- Университетская информационная система РОССИЯ – <https://uisrussia.msu.ru/>
- Единая межведомственная информационно-статистическая система (ЕМИСС) – <https://www.fedstat.ru/>

#### **14. Материально-техническое обеспечение**

Аудитории для проведения занятий лекционного типа.

Аудитории для проведения занятий семинарского типа, индивидуальных и групповых консультаций, текущего контроля и промежуточной аттестации.

Помещения для самостоятельной работы, оснащенные компьютерной техникой и доступом к сети Интернет, в электронную информационно-образовательную среду и к информационным справочным системам.

#### **15. Информация о разработчиках**

Пожидаев Михаил Сергеевич, канд. техн. наук, доцент кафедры теоретических основ информатики.