

МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»
(РУТ (МИИТ))



Рабочая программа дисциплины (модуля),
как компонент образовательной программы
специализированного высшего образования
по направлению подготовки
09.04.03 Прикладная информатика,
утвержденной директором РУТ (МИИТ)
Покусавым О.Н.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Технология разработки ПО

Направление подготовки: 09.04.03 Прикладная информатика

Направленность (профиль): IT-инженер ВСМ

Форма обучения: Очная

Рабочая программа дисциплины (модуля) в виде
электронного документа выгружена из единой
корпоративной информационной системы управления
университетом и соответствует оригиналу

Простая электронная подпись, выданная РУТ (МИИТ)
ID подписи: 2017
Подписал: заместитель директора Ефимова Ольга
Владимировна
Дата: 09.06.2026

1. Общие сведения о дисциплине (модуле).

Цели дисциплины:

- формирование у студентов системного представления о современных технологиях разработки программного обеспечения;
- освоение инструментов и подходов к разработке, сопровождению и автоматизации процессов жизненного цикла ПО;
- развитие практических навыков использования Linux, Git, Docker и брокеров сообщений.

Задачи дисциплины:

- изучить архитектуру и принципы работы современных операционных систем и инструментов для разработки ПО;
- овладеть навыками работы с системой контроля версий Git и платформой GitHub;
- освоить методы контейнеризации и автоматизации развёртывания приложений;
- научиться работать с брокерами сообщений (RabbitMQ, Kafka);
- изучить базовые подходы к DevOps и CI/CD;
- применить изученные технологии на практике в рамках комплексного проекта.

2. Планируемые результаты обучения по дисциплине (модулю).

Перечень формируемых результатов освоения образовательной программы (компетенций) в результате обучения по дисциплине (модулю):

ПК-2 - Способен создавать архитектуру решения для IT-инфраструктуры ВСМ;

ПК-3 - Способен создавать схемы IT-ландшафта ВСМ в рамках методологии разработки архитектуры ПО.

Обучение по дисциплине (модулю) предполагает, что по его результатам обучающийся будет:

Знать:

- жизненный цикл программного обеспечения и современные подходы к его реализации;
- принципы работы Linux, систем контроля версий, контейнеризации и брокеров сообщений;
- основы DevOps и непрерывной интеграции/развёртывания.

Уметь:

- использовать инструменты Linux, Git, Docker, RabbitMQ и Kafka для разработки ПО;

- организовывать взаимодействие между микросервисами;

- настраивать процессы автоматической сборки и тестирования.

Владеть:

- практическими навыками командной разработки;

- средствами обеспечения воспроизводимости и масштабируемости приложений;

- инструментами мониторинга и деплоя.

3. Объем дисциплины (модуля).

3.1. Общая трудоемкость дисциплины (модуля).

Общая трудоемкость дисциплины (модуля) составляет 6 з.е. (216 академических часа(ов)).

3.2. Объем дисциплины (модуля) в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении учебных занятий:

Тип учебных занятий	Количество часов	
	Всего	Семестр №1
Контактная работа при проведении учебных занятий (всего):	48	48
В том числе:		
Занятия лекционного типа	16	16
Занятия семинарского типа	32	32

3.3. Объем дисциплины (модуля) в форме самостоятельной работы обучающихся, а также в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении промежуточной аттестации составляет 168 академических часа (ов).

3.4. При обучении по индивидуальному учебному плану, в том числе при ускоренном обучении, объем дисциплины (модуля) может быть реализован полностью в форме самостоятельной работы обучающихся, а также в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении промежуточной аттестации.

4. Содержание дисциплины (модуля).

4.1. Занятия лекционного типа.

№ п/п	Тематика лекционных занятий / краткое содержание
1	Введение в технологии разработки программного обеспечения Рассматриваемые вопросы: <ul style="list-style-type: none">– основные этапы жизненного цикла ПО;– методологии разработки (Waterfall, Agile, DevOps);– обзор современных стеков технологий.
2	Архитектура и базовые команды Linux Рассматриваемые вопросы: <ul style="list-style-type: none">– структура ОС Linux;– файловая система и процессы;– командная строка и скрипты.
3	Работа в командной строке Linux Рассматриваемые вопросы: <ul style="list-style-type: none">– основные команды bash;– работа с файлами, правами доступа;– пакетный менеджмент.
4	Системы контроля версий: Git и GitHub Рассматриваемые вопросы: <ul style="list-style-type: none">– история изменений, фиксация, коммиты;– работа с ветками;– организация командной работы.
5	Продвинутые возможности Git Рассматриваемые вопросы: <ul style="list-style-type: none">– слияние, ребейз, работа с конфликтами;– submodules и Git hooks;– CI/CD-интеграции.
6	Контейнеризация приложений: Docker Рассматриваемые вопросы: <ul style="list-style-type: none">– контейнеры vs виртуальные машины;– образы, слои, Dockerfile;– docker-compose.
7	Брокеры сообщений: RabbitMQ и Kafka Рассматриваемые вопросы: <ul style="list-style-type: none">– роль брокеров в архитектуре ПО;– очереди сообщений и pub/sub;– настройка и примеры.
8	DevOps и автоматизация разработки Рассматриваемые вопросы: <ul style="list-style-type: none">– принципы DevOps-культуры;– инструменты CI/CD;– реализация пайплайнов.

4.2. Занятия семинарского типа.

Практические занятия

№ п/п	Тематика практических занятий/краткое содержание
1	Основы работы в Linux В результате выполнения практической работы студенты научатся использовать базовые команды Linux.
2	Работа с файловой системой и правами В результате выполнения практической работы студенты научатся управлять файлами и правами доступа.
3	Написание скриптов Bash В результате выполнения практической работы студенты создадут и запустят простые скрипты.
4	Установка и настройка Git В результате выполнения практической работы студенты установят Git и создадут локальный репозиторий.
5	Работа с удалёнными репозиториями В результате выполнения практической работы студенты научатся использовать GitHub.
6	Ветвление и слияние в Git В результате выполнения практической работы студенты освоят работу с ветками и разрешением конфликтов.
7	CI/CD-интеграция в Git В результате выполнения практической работы студенты подключат и запустят автоматическую сборку.
8	Установка и работа с Docker В результате выполнения практической работы студенты создадут и запустят контейнер с приложением.
9	Создание Dockerfile и сборка образов В результате выполнения практической работы студенты создадут Dockerfile и соберут собственный образ.
10	Docker Compose В результате выполнения практической работы студенты реализуют многоконтейнерное окружение.
11	Основы RabbitMQ В результате выполнения практической работы студенты поднимут сервер RabbitMQ и реализуют обмен сообщениями.
12	Основы Kafka В результате выполнения практической работы студенты запустят Kafka и протестируют передачу сообщений.
13	Интеграция брокера сообщений с приложением В результате выполнения практической работы студенты реализуют микросервисное взаимодействие через брокер.
14	Основы DevOps и пайплайны В результате выполнения практической работы студенты изучат структуру пайплайна и создадут шаблон.
15	Разработка проекта с использованием Git, Docker и брокера сообщений В результате выполнения практической работы студенты объединят изученные технологии в проект.
16	Итоговая защита проекта В результате выполнения практической работы студенты представят рабочее решение с автоматизацией разработки.

4.3. Самостоятельная работа обучающихся.

№ п/п	Вид самостоятельной работы
1	Подготовка к практическим занятиям
2	Работа с лекционным материалом
3	Самостоятельное изучение рекомендуемой литературы
4	Выполнение курсовой работы.
5	Подготовка к промежуточной аттестации.
6	Подготовка к текущему контролю.

4.4. Примерный перечень тем курсовых работ

- 1.Создание CI/CD пайплайна для Python-приложения.
- 2.Разработка микросервисного проекта с использованием Docker и Kafka.
- 3.Разработка и деплой REST API с контейнеризацией в Docker.
- 4.Сравнение производительности RabbitMQ и Kafka в распределённых системах.
- 5.Построение системы логирования и мониторинга контейнеров.
- 6.Организация взаимодействия сервисов через брокеры сообщений.
- 7.Разработка и автоматизация деплоя с помощью GitHub Actions.
- 8.Внедрение системы управления версиями в командной разработке.
- 9.Интеграция Docker и CI/CD в студенческом проекте.
- 10.Разработка DevOps-архитектуры для учебного микросервиса.

5. Перечень изданий, которые рекомендуется использовать при освоении дисциплины (модуля).

№ п/п	Библиографическое описание	Место доступа
1	Малахов, С. В. Принципы работы операционной системы Linux. Bash-скрипты : учебное пособие / С. В. Малахов, Д. О. Якупов. — Самара : ПГУТИ, 2024. — 134 с. — ISBN 978-5-907336-50-6	https://e.lanbook.com/book/463574
2	Курячий, Г. В. Операционная система Linux: Курс лекций : учебное пособие / Г. В. Курячий, К. А. Маслинский. — Москва : ДМК Пресс, 2010. — 348 с. — ISBN 978-5-94074-591-4.	https://e.lanbook.com/book/1202
3	Эделман, Д. Автоматизация программируемых сетей : руководство / Д. Эделман, С. С. Лоу, М. Осуолт ; перевод с английского А. В. Снастина. —	https://e.lanbook.com/book/123708

	Москва : ДМК Пресс, 2019. — 616 с. — ISBN 978-5-97060-699-5.	
4	Кочер, П. С. Микросервисы и контейнеры Docker : руководство / П. С. Кочер ; перевод с английского А. Н. Киселева. — Москва : ДМК Пресс, 2019. — 240 с. — ISBN 978-5-97060-739-8.	https://e.lanbook.com/book/123710
5	Баланов, А. Н. Построение микросервисной архитектуры и разработка высоконагруженных приложений : учебное пособие для вузов / А. Н. Баланов. — 2-е изд., стер. — Санкт-Петербург : Лань, 2025. — 244 с. — ISBN 978-5-507-52652-9.	https://e.lanbook.com/book/456920

6. Перечень современных профессиональных баз данных и информационных справочных систем, которые могут использоваться при освоении дисциплины (модуля).

Официальный сайт РУТ (МИИТ) (<https://www.miit.ru/>);

Официальный сайт Минтранса России (<https://mintrans.gov.ru/>);

Научно-техническая библиотека РУТ (МИИТ) (<http://library.miit.ru/>);

Информационный портал Научная электронная библиотека eLIBRARY.RU (www.elibrary.ru/);

Образовательная платформа «Открытое образование» (<https://openedu.ru/>);

Общие информационные, справочные и поисковые системы «Консультант Плюс», «Гарант»;

Электронно-библиотечная система издательства «Лань» (<http://e.lanbook.com/>);

Электронно-библиотечная система ibooks.ru (<http://ibooks.ru/>);

Электронно-библиотечная система «Академия» (<http://academia-moscow.ru/>);

Электронно-библиотечная система «BOOK.ru» (<http://www.book.ru/>);

Электронно-библиотечная система «ZNANIUM.COM» (<http://www.znanium.com/>)

7. Перечень лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства, необходимого для освоения дисциплины (модуля).

Microsoft Internet Explorer (или другой браузер)

Операционная система Microsoft Windows

Microsoft Office

Visual studio Code

8. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю).

Учебные аудитории для проведения учебных занятий, оснащенные компьютерной техникой и наборами демонстрационного оборудования.

9. Форма промежуточной аттестации:

Курсовая работа в 1 семестре.

Экзамен в 1 семестре.

10. Оценочные материалы.

Оценочные материалы, применяемые при проведении промежуточной аттестации, разрабатываются в соответствии с локальным нормативным актом РУТ (МИИТ).

Авторы:

руководитель образовательной
программы

А.А. Кочурков

Согласовано:

Заместитель директора

О.В. Ефимова

Председатель учебно-методической
комиссии

Д.В. Паринов