МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»
(РУТ (МИИТ)

Рабочая программа дисциплины (модуля),
как компонент образовательной программы
базового высшего образования
по направлению подготовки
09.03.01 Информатика и вычислительная техника,
утвержденной директором ФГАОУ ВО РУТ
(МИИТ) Париновым Д.В.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Мониторинг и наблюдаемость ПО

Направление подготовки: 09.03.01 Информатика и вычислительная техника

Направленность (профиль): Технологии разработки программного обеспечения

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 ФГАОУ ВО РУТ (МИИТ)
ID подписи: 937226
Подписал: директор центра Проневич Ольга Борисовна
Дата: 12.11.2026

1. Общие сведения о дисциплине (модуле).

Дисциплина посвящена мониторингу и наблюдаемости распределенных программных систем, развернутых в серверной среде. В ходе изучения рассматриваются эксплуатационная устойчивость, показатели качества сервиса, метрики, журналы событий, распределенные трассировки, сбор телеметрии из Java Spring-приложений, хранение временных рядов, построение панелей наблюдения, правила оповещения, анализ сбоев и подготовка технической документации. На лабораторных занятиях обучающиеся последовательно оснащают серверное приложение средствами наблюдения, собирают данные о работе сервисов, выявляют причины отказов и оформляют воспроизводимое портфолио эксплуатационного контроля.

Целью освоения дисциплины является формирование способности проектировать, внедрять, проверять и документировать средства мониторинга и наблюдаемости программных систем, обеспечивающие своевременное обнаружение отказов, анализ причин ухудшения качества сервиса и поддержку устойчивой эксплуатации распределенных приложений.

Для достижения поставленной цели в рамках дисциплины решается комплекс задач, направленных на формирование у обучающихся способности – анализировать эксплуатационные требования к программной системе, определять ключевые показатели качества сервиса, собирать метрики, журналы событий и трассировки, настраивать сбор телеметрии Java Spring-приложения, строить панели наблюдения, задавать правила оповещения, проводить нагрузочную проверку, анализировать сбои и готовить техническую документацию по сопровождению программного продукта.

2. Планируемые результаты обучения по дисциплине (модулю).

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

ДПК-БП-2 - Способен обеспечивать наблюдаемость, производительность, надёжность и эксплуатационную устойчивость распределённого бизнес-приложения;

ИП-2 - Способен оптимизировать производительность программного обеспечения;

ППК-Р6 - Способен участвовать в промышленной разработке программного обеспечения.

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

Знать:

- назначение мониторинга и наблюдаемости в жизненном цикле программной системы и сопровождении серверных приложений
- понятия эксплуатационной устойчивости, доступности, надежности, времени ответа, пропускной способности, ошибки и деградации сервиса
- принципы определения ключевых показателей качества сервиса, пороговых значений, контрольных целей и признаков нарушения нормальной работы
- виды телеметрических данных программной системы, включая метрики, журналы событий, трассировки и сведения о состоянии инфраструктуры
- модели сбора метрик, включая временной ряд, метку, счетчик, измеритель текущего значения, гистограмму и сводное распределение
- устройство Prometheus, включая цели сбора, опрос конечных точек, хранение временных рядов, язык запросов PromQL и правила оповещения
- назначение Grafana при построении панелей наблюдения, графиков, таблиц, переменных, аннотаций и представлений для разных ролей сопровождения
- возможности Spring Boot Actuator и Micrometer при публикации метрик Java Spring-приложения
- принципы структурированного ведения журналов событий, уровни важности, контекст запроса, идентификатор корреляции и требования к отсутствию чувствительных данных
- назначение Loki при централизованном хранении и поиске журналов событий по меткам и содержанию сообщений
- принципы распределенной трассировки, включая трассу, отрезок выполнения, родительскую связь, контекст запроса и распространение идентификаторов между сервисами
- назначение OpenTelemetry при сборе и передаче телеметрии из приложения в хранилища метрик, журналов событий и трассировок
- назначение Tempo при хранении распределенных трассировок и переходе от метрики к журналу событий и трассе запроса
- типовые причины ухудшения качества сервиса, включая рост времени ответа, исчерпание ресурсов, ошибки зависимостей, накопление очередей и задержки сетевого взаимодействия
- методы нагрузочной проверки и проверки деградации, включая сценарий запроса, профиль нагрузки, контрольные метрики и сравнение состояния до и после изменения

- принципы построения правил оповещения, включая условие, длительность нарушения, важность, подавление повторов и маршрут уведомления ответственному исполнителю

- подходы к разбору отказа, включая временную линию события, гипотезы причины, сопоставление метрик, журналов событий и трассировок

- требования к технической документации по наблюдаемости, включая карту показателей, перечень источников телеметрии, панели наблюдения, правила оповещения и порядок действий при отказе

Уметь:

- уметь формализовать требования к наблюдаемости при помощи карты сервисов, показателей качества и контрольных целей в условиях распределенного Java Spring-приложения

- уметь подключать публикацию метрик при помощи Spring Boot Actuator и Micrometer в условиях серверного приложения с несколькими конечными точками

- уметь настраивать сбор метрик при помощи Prometheus в условиях периодического опроса приложения и хранения временных рядов

- уметь строить панели наблюдения при помощи Grafana в условиях отображения времени ответа, числа запросов, ошибок, ресурсов и состояния зависимостей

- уметь настраивать структурированные журналы событий при помощи Logback и идентификатора корреляции в условиях анализа прохождения пользовательского запроса

- уметь собирать и искать журналы событий при помощи Loki в условиях фильтрации по сервису, уровню важности и идентификатору запроса

- уметь подключать распределенную трассировку при помощи OpenTelemetry в условиях взаимодействия нескольких Java Spring-сервисов

- уметь анализировать трассировки при помощи Tempo в условиях поиска медленных участков, ошибок зависимостей и повторяющихся задержек

- уметь задавать правила оповещения при помощи Prometheus Alertmanager или Grafana в условиях нарушения порогов доступности, времени ответа и доли ошибок

- уметь проводить нагрузочную проверку при помощи k6 или Apache JMeter в условиях сравнения поведения системы при нормальной и повышенной нагрузке

- уметь выявлять причину деградации при помощи сопоставления метрик, журналов событий и трассировок в условиях отказа зависимости или роста задержки

- уметь готовить техническую документацию по наблюдаемости при помощи описания показателей, источников данных, панелей, правил оповещения и порядка разбора отказов

Владеть:

- навыком проектирования карты наблюдаемости распределенного серверного приложения
- навыком публикации и сбора метрик Java Spring-приложения средствами Spring Boot Actuator, Micrometer и Prometheus
- навыком построения панелей наблюдения и правил оповещения средствами Grafana
- навыком настройки структурированных журналов событий и централизованного поиска средствами Loki
- навыком подключения распределенной трассировки средствами OpenTelemetry и Tempo
- навыком проведения нагрузочной проверки и анализа деградации программной системы
- навыком сопоставления метрик, журналов событий и трассировок при разборе отказа
- навыком подготовки технической документации по эксплуатационному контролю программной системы

3. Объем дисциплины (модуля).

3.1. Общая трудоемкость дисциплины (модуля).

Общая трудоемкость дисциплины (модуля) составляет 3 з.е. (108 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 №7
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 60 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	Эксплуатационная устойчивость программных систем Рассматриваемые вопросы: - назначение мониторинга и наблюдаемости при сопровождении серверных приложений; - доступность, надежность, время ответа, пропускная способность и доля ошибок; - связь пользовательского качества сервиса с внутренним состоянием программной системы.
2	Телеметрические данные распределенного приложения Рассматриваемые вопросы: - метрики, журналы событий, трассировки и сведения о состоянии инфраструктуры; - идентификатор корреляции и связь событий одного запроса; - требования к полноте, точности, безопасности и стоимости хранения телеметрии.
3	Сбор метрик Java Spring-приложения Рассматриваемые вопросы: - возможности Spring Boot Actuator и Micrometer; - счетчики, измерители текущего значения, гистограммы и метки временных рядов; - публикация служебных и прикладных показателей через конечную точку приложения.
4	Хранение метрик и язык запросов Prometheus Рассматриваемые вопросы: - цели сбора, периодический опрос, временные ряды и метки; - запросы PromQL для времени ответа, числа запросов, ошибок и потребления ресурсов; - правила записи и подготовка агрегированных показателей для панелей наблюдения.
5	Панели наблюдения и правила оповещения Рассматриваемые вопросы: - построение графиков, таблиц, переменных и аннотаций в Grafana; - выбор показателей для разработчика, дежурного инженера и руководителя сервиса; - условия оповещения, длительность нарушения, важность и подавление повторов.
6	Журналы событий и распределенные трассировки Рассматриваемые вопросы: - структурированные журналы событий, уровни важности и защита чувствительных данных;

№ п/п	Тематика лекционных занятий / краткое содержание
	- централизованный поиск журналов событий в Loki; - трасса, отрезок выполнения, контекст запроса и хранение трассировок в Tempo.
7	Нагрузочная проверка и анализ деградации сервиса Рассматриваемые вопросы: - сценарий нагрузки, профиль запросов, контрольные метрики и ожидаемое поведение системы; - признаки деградации при росте времени ответа, ошибках зависимостей и исчерпании ресурсов; - сопоставление метрик, журналов событий и трассировок при поиске причины нарушения.
8	Разбор отказов и техническая документация наблюдаемости Рассматриваемые вопросы: - временная линия отказа, гипотезы причины и проверка по телеметрическим данным; - порядок действий при нарушении контрольных целей и восстановлении сервиса; - состав технической документации по показателям, панелям, правилам оповещения и известным ограничениям.

4.2. Занятия семинарского типа.

Лабораторные работы

№ п/п	Наименование лабораторных работ / краткое содержание
1	Подготовка серверного приложения для эксплуатационного контроля Студент запускает Java Spring-приложение с несколькими конечными точками и проверяет его работу контрольными запросами. Создается схема сервисов, зависимостей и пользовательских сценариев. Для каждого сценария фиксируются ожидаемое время ответа, допустимая доля ошибок и критичность нарушения.
2	Публикация метрик приложения Студент подключает Spring Boot Actuator и Micrometer к серверному приложению. Проверяется публикация показателей запросов, состояния виртуальной машины Java и пользовательских счетчиков. Итоговый набор метрик сопоставляется с картой пользовательских сценариев.
3	Сбор и запросы метрик в Prometheus Студент настраивает Prometheus на периодический опрос приложения. Для времени ответа, числа запросов, доли ошибок и потребления памяти составляются запросы PromQL. Результаты запросов проверяются при обычной работе и при искусственно созданной ошибке.
4	Построение панелей наблюдения в Grafana Студент подключает Grafana к хранилищу метрик и создает панель наблюдения для серверного приложения. На панель выносятся графики времени ответа, количества запросов, ошибок, загрузки процессора и памяти. Для панели задаются переменные, подписи и аннотации контрольных событий.
5	Настройка журналов событий и централизованного поиска Студент переводит журналы событий приложения к структурированному виду и добавляет идентификатор корреляции запроса. Настраивается передача журналов событий в Loki. Поиск выполняется по сервису, уровню важности, фрагменту сообщения и идентификатору корреляции.
6	Подключение распределенной трассировки Студент подключает OpenTelemetry к Java Spring-приложению и передает трассировки в Tempo. Для нескольких запросов проверяются отрезки выполнения, длительность операций и связь между сервисами. Медленный участок запроса выделяется по данным трассировки.
7	Правила оповещения и нагрузочная проверка Студент задает правила оповещения для роста времени ответа, увеличения доли ошибок и недоступности конечной точки. Затем выполняется нагрузочная проверка с помощью k6 или Apache

№ п/п	Наименование лабораторных работ / краткое содержание
	JMeter. По результатам фиксируются сработавшие правила, графики метрик и поведение приложения под нагрузкой.
8	Разбор отказа и оформление портфолио наблюдаемости Студент моделирует отказ зависимости, рост задержки или ошибку обработки запроса. Причина нарушения определяется через сопоставление метрик, журналов событий и трассировок. Итоговая документация включает карту показателей, панель наблюдения, правила оповещения, порядок разбора отказа и известные ограничения.

4.3. Самостоятельная работа обучающихся.

№ п/п	Вид самостоятельной работы
1	Изучение рекомендованной литературы.
2	Подготовка к лабораторным работам.
3	Подготовка к промежуточной аттестации.
4	Подготовка к текущему контролю.

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

№ п/п	Библиографическое описание	Место доступа
1	Программные инструменты обработки и визуализации данных. Elasticsearch, Logstash, Kibana, Grafana, Prometheus : учебное пособие / И. В. Никифоров, О. А. Юсупова, Н. В. Воинов [и др.] ; под редакцией И. В. Никифорова [и др.]. — Санкт-Петербург : СПбГПУ, 2023. — 140 с. — ISBN 978-5-7422-8075-0. — Текст : электронный	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/430157 (дата обращения: 22.06.2026)
2	Баланов, А. Н. DevOps «под ключ». Интеграция и автоматизация : учебное пособие / А. Н. Баланов. — Москва : Интермедиатор, 2025. — 114 с. — ISBN 978-5-91349-139-8. — Текст : электронный	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/518293 (дата обращения: 22.06.2026)
3	Аймен, Э. А. Облачные микросервисы в Kubernetes : руководство / Э. А. Аймен ; пер. с англ. В. С. Яценкова. — Москва : ДМК Пресс, 2025. — 276 с. — ISBN 978-5-93700-324-9. — Текст : электронный	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/514873 (дата обращения: 22.06.2026)
4	Баланов, А. Н. Построение микросервисной архитектуры и разработка высоконагруженных приложений : учебное пособие для вузов / А. Н. Баланов. — Санкт-Петербург : Лань, 2024. — 244	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/394538 (дата обращения: 22.06.2026)

	с. — ISBN 978-5-507-48747-9. — Текст : электронный	
5	Стин, в. М. Распределенные системы : сборник научных трудов / в. М. Стин, Э. С. Таненбаум ; перевод с английского В. А. Яроцкого. — Москва : ДМК Пресс, 2021. — 584 с. — ISBN 978-5-97060-708-4. — Текст : электронный	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/240992 (дата обращения: 22.06.2026)
6	Хабаров, А. Н. Разработка программных приложений : учебник / А. Н. Хабаров, А. Н. Ермакова. — Ставрополь : СтГАУ, 2025. — 208 с. — Текст : электронный	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/510223 (дата обращения: 22.06.2026)
7	Рудаков, Н. В. Эксплуатация, сопровождение и обслуживание информационных систем : учебное пособие / Н. В. Рудаков. — Иваново : ИГЭУ, 2023. — 160 с. — Текст : электронный	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/369743 (дата обращения: 22.06.2026)

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

ЭБС Лань – https://e.lanbook.com/.

Образовательная платформа Юрайт – https://urait.ru/.

Единый реестр российских программ для ЭВМ и баз данных – https://reestr.digital.gov.ru/reestr/.

Профессиональные стандарты и квалификации, справочная информация КонсультантПлюс – https://www.consultant.ru/document/cons_doc_LAW_157436/.

Документация Java SE – https://docs.oracle.com/en/java/javase/21/docs/api/index.html.

Документация Spring Boot Actuator – https://docs.spring.io/spring-boot/reference/actuator/.

Документация Micrometer – https://docs.micrometer.io/.

Документация Prometheus – https://prometheus.io/docs/.

Документация Grafana – https://grafana.com/docs/grafana/.

Документация Loki – https://grafana.com/docs/loki/.

Документация Tempo – https://grafana.com/docs/tempo/.

Документация OpenTelemetry – https://opentelemetry.io/docs/.

Документация k6 – https://grafana.com/docs/k6/.

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

Операционные системы – Astra Linux, ALT Linux, РЕД ОС, Debian GNU/Linux.

Платформа разработки – OpenJDK, Spring Boot, Maven.

Средства наблюдаемости – Micrometer, Prometheus, Grafana, Loki, Tempo, OpenTelemetry.

Нагрузочная проверка и сопровождение – k6, Apache JMeter, Git.

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.

Для лабораторных занятий – наличие персональных компьютеров вычислительного класса.

9. Форма промежуточной аттестации:

Экзамен в 7 семестре.

10. Оценочные материалы.

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

Авторы:

старший преподаватель кафедры
«Цифровые технологии управления
транспортными процессами»
Института железнодорожного
транспорта Е.А. Заманов

Согласовано:

Заведующий кафедрой ЦТУТП В.Е. Нутович

Директор ЦС IT - программ ВИШ О.Б. Проневич

Председатель учебно-методической
комиссии Н.А. Андриянова