



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Магнитогорский государственный технический университет им. Г.И.
Носова»



УТВЕРЖДАЮ
Директор ИИС
Ю.В. Сомова

03.02.2025 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

DEVOPS

Направление подготовки (специальность)
01.03.02 Прикладная математика и информатика

Направленность (профиль/специализация) программы
Большие и открытые данные

Уровень высшего образования - бакалавриат

Форма обучения
очная

Институт/ факультет	Институт естествознания и стандартизации
Кафедра	Прикладной математики и информатики
Курс	1
Семестр	2

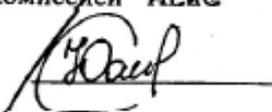
Магнитогорск
2025 год

Рабочая программа составлена на основе ФГОС ВО - бакалавриат по направлению подготовки 01.03.02 Прикладная математика и информатика (приказ Минобрнауки России от 10.01.2018 г. № 9)

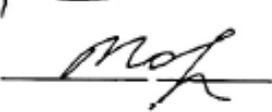
Рабочая программа рассмотрена и одобрена на заседании кафедры Прикладной математики и информатики
14.01.2025, протокол № 5

Зав. кафедрой  Ю.А. Извеков

Рабочая программа одобрена методической комиссией ИЕиС
03.02.2025 г. протокол № 3

Председатель  Ю.В. Сомова

Рабочая программа составлена:
доцент кафедры ПМИИ, канд. физ.-мат. наук

 О.А. Торшина

Рецензент:

зав. кафедрой Физики, канд. физ.-мат. наук  Д.М. Долгушин

Лист актуализации рабочей программы

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2026 - 2027 учебном году на заседании кафедры Прикладной математики и информатики

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ Ю.А. Извеков

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2027 - 2028 учебном году на заседании кафедры Прикладной математики и информатики

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ Ю.А. Извеков

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2028 - 2029 учебном году на заседании кафедры Прикладной математики и информатики

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ Ю.А. Извеков

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2029 - 2030 учебном году на заседании кафедры Прикладной математики и информатики

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ Ю.А. Извеков

1 Цели освоения дисциплины (модуля)

Целями освоения дисциплины «DevOps» является формирование у слушателей знаний и навыков по методологии

DevOps для активного взаимодействия специалистов по разработке со специалистами по

информационно-технологическому обслуживанию и взаимной интеграции их рабочих

процессов для обеспечения качества продукта. В процессе прохождения курса подробно

разбирается жизненный цикл (ЖЦ) программного обеспечения, роль DevOps-инженера в ЖЦ,

а также программные инструменты DevOps, такие, как Docker, Jenkins, Ansible, Kubernetes и

Prometheus.

2 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина DevOps входит в часть учебного плана формируемую участниками образовательных отношений образовательной программы.

Для изучения дисциплины необходимы знания (умения, владения), сформированные в результате изучения дисциплин/ практик:

Практикум на ЭВМ

Учебная - технологическая (проектно-технологическая) практика

Нейронные сети

Знания (умения, владения), полученные при изучении данной дисциплины будут необходимы для изучения дисциплин/практик:

Разработка интернет приложений

Большие и открытые данные

Базы данных

Объектно-ориентированное программирование

3 Компетенции обучающегося, формируемые в результате освоения дисциплины (модуля) и планируемые результаты обучения

В результате освоения дисциплины (модуля) «DevOps» обучающийся должен обладать следующими компетенциями:

Код индикатора	Индикатор достижения компетенции
ПК-1	Способен анализировать большие данные и проводить исследования с их технологиями
ПК-1.1	Выбирает методы и инструментальные средства для проведения аналитических работ с большими данными
ПК-1.2	Разрабатывает и оценивает модели больших данных
ПК-1.3	Организует контроль эффективности работы и предлагает решения руководителю (заказчику)

4. Структура, объём и содержание дисциплины (модуля)

Общая трудоемкость дисциплины составляет 3 зачетных единиц 108 академических часов, в том числе:

- контактная работа – 54,15 академических часов;
- аудиторная – 51 академический час;
- внеаудиторная – 3,15 академических часов;
- самостоятельная работа – 18,15 академических часов;
- в форме практической подготовки – 0 академических часов;
- подготовка к экзамену – 35,7 академических часов

Форма аттестации - экзамен

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в академических часах)			Самостоятельная работа студента	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код компетенции
		Лек.	лаб. зан.	практ. зан.				
1. Основы DevOps								
1.1 Основные определения	2	2	2		2			
1.2 Жизненный цикл ПО		2						
1.3 DevOps-инженер			2		2			
Итого по разделу		4	4		4			
2. Системы виртуализации и контейнеризации								
2.1 Классификация систем	2	2			2			
2.2 Введение в экосистему контейнеров на основе Docker		2	4					
2.3 Настройка рабочего окружения, подготовка и запуск Docker-контейнеров		2			2			
2.4 Системы непрерывной интеграции		2			2			
Итого по разделу		8	4		6			
3. Системы непрерывной интеграции								
3.1 Обзор систем непрерывной интеграции	2		2					
3.2 Основы автоматизации разработки, тестирования и доставки ПО с использованием Jenkins			2					
3.3 Настройка Jenkins, выбор и установка плагинов			2					
Итого по разделу			6					

4. : Системы управления конфигурацией. Инфраструктура как код								
4.1 Обзор систем управления конфигурацией	2		4					
4.2 Основы управления конфигурацией с использованием Ansible			2		2			
4.3 Основные встроенные модули Ansible		2	4					
Итого по разделу		2	10		2			
5. Системы оркестрации								
5.1 Обзор систем оркестрации	2	2						
5.2 Основы оркестрации с использованием Kubernetes			4		2			
5.3 Микросервисная архитектура		1	4					
Итого по разделу		3	8		2			
6. Системы мониторинга								
6.1 Поставка систем сбора метрик как сервис внутри проекта, разделение ответственности между разработкой и эксплуатацией	2		2					
6.2 Обзор актуальных систем мониторинга и логирования					4,15			
Итого по разделу			2		4,15			
Итого за семестр		17	34		18,15		экзамен	
Итого по дисциплине		17	34		18,15		экзамен	

5 Образовательные технологии

1. Традиционные технологии обучения, предполагающие передачу информации в готовом виде, формирование учебных умений по образцу: лекция-изложение, лекция-объяснение, практические (семинарские) занятия.

Использование традиционных технологий обеспечивает ориентирование студента в потоке информации, связанной с различными подходами к определению сущности, содержания, методов, форм развития и саморазвития личности; самоопределение в выборе оптимального пути и способов личностно-профессионального развития; систематизацию знаний, полученных студентами в процессе аудиторной и самостоятельной работы. Лекционные занятия проводятся с использованием мультимедийных средств. Лабораторные занятия обеспечивают развитие и закрепление умений и навыков определения целей и задач саморазвития, а также принятия наиболее эффективных решений по их реализации. Практические занятия проводятся в компьютерных классах вычислительного центра ФГБОУ ВО «МГТУ».

В ходе проведения лекционных занятий предусматривается: использование электронного демонстрационного материала по темам, требующим иллюстрации работы программных продуктов: MS Word, MS Excel.

В ходе проведения лабораторных занятий предусматривается использование средств вычислительной техники при выполнении индивидуальных заданий и тестирования.

2. Интерактивные формы обучения, предполагающие организацию обучения как продуктивной творческой деятельности в режиме взаимодействия студентов друг с другом и с преподавателем

Использование интерактивных образовательных технологий способствует повышению интереса и мотивации учащихся, активизации мыслительной деятельности и творческого потенциала студентов, делает более эффективным усвоение материала, позволяет индивидуализировать обучение и ввести экстренную коррекцию знаний.

В рамках дисциплины «DevOps» предусматриваются часы аудиторных занятий, проводимых в интерактивной форме.

При проведении лабораторных занятий используются групповая работа, технология коллективной творческой деятельности, технология сотрудничества, обсуждение проблемы в форме дискуссии, дебаты, круглый стол. Данные технологии обеспечивают высокий уровень усвоения студентами знаний, эффективное и успешное овладение умениями и навыками в предметной области, формируют познавательную потребность и необходимость дальнейшего самообразования, позволяют активизировать исследовательскую деятельность, обеспечивают эффективный контроль усвоения знаний.

3. Возможности образовательного портала ФГБОУ ВО «МГТУ» для предоставления студентам графика самостоятельной работы, расписания консультаций, заданий для самостоятельного выполнения и рекомендуемых тем для самостоятельного изучения.

Методика, предлагаемая для изучения курса «DevOps» ориентирована на лекции проблемно-информационного характера, семинарские занятия исследовательского типа и подготовку рефератов.

Используемые образовательные технологии позволяют активно применять в учебном процессе интерактивные формы проведения занятий (компьютерная симуляция, разбор конкретных ситуаций), что способствует формированию и развитию профессиональных навыков обучающихся. Применяемые в процессе изучения дисциплины поисковый и исследовательский методы в полной мере соответствуют требованиям ФГОС по реализации компетентного подхода.

6 Учебно-методическое обеспечение самостоятельной работы обучающихся

Представлено в приложении 1.

7 Оценочные средства для проведения промежуточной аттестации

Представлены в приложении 2.

8 Учебно-методическое и информационное обеспечение дисциплины

а) Основная литература:

1. Зыков, С. В. Архитектура информационных систем. Основы проектирования : учебник для вузов / С. В. Зыков. — Москва : Издательство Юрайт, 2025. — 260 с. — (Высшее образование). — ISBN 978-5-534-21538-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/575500> (дата обращения: 26.03.2025).

2. Щербак, А. В. Поддержка и тестирование программных модулей : учебник для среднего профессионального образования / А. В. Щербак. — Москва : Издательство Юрайт, 2025. — 145 с. — (Профессиональное образование). — ISBN 978-5-534-19290-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/580603> (дата обращения: 26.03.2025).

б) Дополнительная литература:

в) Методические указания:

г) Программное обеспечение и Интернет-ресурсы:

Программное обеспечение

Наименование ПО	№ договора	Срок действия лицензии
7Zip	свободно распространяемое ПО	бессрочно
Inkscape Project	свободно распространяемое ПО	бессрочно
Браузер Yandex	свободно распространяемое ПО	бессрочно
СУБД Ред База Данных	Сертификат №01-04\22 от 06.05.2022	06.05.2025

Профессиональные базы данных и информационные справочные системы

Название курса	Ссылка
Электронные ресурсы библиотеки МГТУ им. Г.И. Носова	https://host.megaprolib.net/M/P0109/Web
Российская Государственная библиотека. Каталоги	https://www.rsl.ru/ru/4readers/catalogues/

9 Материально-техническое обеспечение дисциплины (модуля)

Материально-техническое обеспечение дисциплины включает:

Материально-техническое обеспечение дисциплины включает:

Лекционная аудитория. Оснащение аудитории: доска, мультимедийные средства хранения, передачи и представления информации

Учебные аудитории для проведения лабораторных занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации: компьютерный класс. Оснащение аудитории: персональные компьютеры с пакетом MS Office и выходом в Интернет.

Помещения для самостоятельной работы обучающихся. Оснащение аудитории: персональные компьютеры с пакетом MS Office, выходом в Интернет и с доступом в электронную информационно-образовательную среду университета.

Помещение для хранения и профилактического обслуживания учебного оборудования. Оснащение аудитории: стеллажи для хранения учебно-наглядных пособий и учебно-методической документации.

Учебно-методическое обеспечение самостоятельной работы обучающихся

Контрольные задания и вопросы для оценки знаний и навыков слушателей задаются и выполняются в следующих областях: •

1. Жизненный цикл ПО. •
2. DevOps-инженер – роль в проекте разработки и внедрения ПО. •
3. Экосистема контейнеров на основе Docker. •
4. Автоматизация разработки, тестирования и доставки ПО с использованием Jenkins.
5. Основы управления конфигурацией с использованием Ansible. •
6. Основные встроенные модули Ansible.
7. Взаимодействие Ansible с Docker.
8. Основы оркестрации с использованием Kubernetes.
9. Микросервисная архитектура.
10. Взаимодействие Kubernetes с Docker.
11. Особенности сбора метрик с микросервисов и Docker-контейнеров.

Примеры заданий для самостоятельной работы

Цель: научиться подключаться к удаленному серверу по протоколу SSH. Научиться работать с базовыми командами и скриптами. Изучить основные инструменты для работы с файловой системой. Научиться использовать базовые инструменты обработки текста в Linux.

1. Запустить SSH-клиент. В случае использования графического клиента запустить программу. В случае использования терминальных клиентов — открыть терминал.

2. Подключиться по SSH к своей виртуальной машине. В графическом клиенте произвести необходимые настройки. В терминальном — выполнить команду подключения. Например, для openssh: `ssh username@ip_address -p port_number`

3. Создать пару SSH-ключей с опциями по умолчанию. Пользуясь разобранным в методических указаниях алгоритмом создать пару ключей, не изменяя значения по умолчанию.

4. Разместить пару ключей в нужных местах. Пользуясь разобранным в методических указаниях алгоритмом разместить пару ключей в нужных местах для использования в авторизации.

5. Отключиться от ОС сервера и попробовать авторизоваться по ключу.

6. Создать файл `script.sh` и сделать его исполняемым. С помощью команд «`touch`» и «`chmod +x`» создать в своей домашней директории файл скрипта и дать ему права на исполнение. 7. Открыть файл скрипта любым редактором. Выбор редактора индивидуален и зависит от преследуемых целей и личных предпочтений.

8. Придумать и написать алгоритм создания дерева каталогов. С помощью известных логических конструкций или команд написать в скрипте алгоритм создания дерева каталогов вида `/home/username/1/2/3`.

9. Запустить скрипт. Выполнить свой скрипт и получить нужный результат.

10. Просмотреть список монтируемых ФС. Посмотреть содержимое файла «`/etc/fstab`», сделать вывод о том, какие ФС монтируются при загрузке и какой тип они имеют, для чего используются.

11. Просмотреть утилизацию примонтированных ФС. С помощью команды «`df`» просмотреть общий и свободный объем ФС, примонтированных в данный момент к ОС.

12. Посчитать объем файлов в домашней директории. С помощью команды «`du`» посчитать объем всех файлов, расположенных в своей домашней директории.

13. Создать LVM с новым диском, подключенным к VM. Создать структуру как минимум с одним Physical Volume, Volume Group и Logical Volume. Создать на нем ФС `ext4`, которая будет монтироваться в директорию `/mnt/storage` при загрузке VM.

14. Описать все эти действия в скрипте, снабженном комментариями.

15. Поиск текста по регулярным выражениям. Найти в главном файле Syslog все сообщения за (определенный период времени).

16. Форматирование вывода. Вывести на экран название PV и соответствующих им VG (дополнительно: в формате "PV = VG"), используя команду вывода из состава LVM. Результат: подключились к своей VM, создали ключ и авторизовались по нему. Написали скрипт, создающий дерево каталогов. Посмотрели, какие ФС используются в ОС и узнали их объем. Посмотрели, как работают программы, рассмотрели базовые случаи.

7 Оценочные средства для проведения промежуточной аттестации

а) Планируемые результаты обучения и оценочные средства для проведения промежуточной аттестации:

Код индикатора	Индикатор достижения компетенции	Оценочные средства
ПК-1: Способен анализировать большие данные и проводить исследования с их технологиями		
ПК-1.	Выбирает методы и инструментальные средства для проведения аналитических работ с большими данными	<p>Примерные темы рефератов</p> <ol style="list-style-type: none"> 1. Применение ИИ в Devops 2. Chaos engineering 3. Системное и сетевое администрирование 4. Наблюдаемость и Operational intelligence 5. Cloud Native Engineering 6. DevOps практики и культура 7. Platform Engineering 8. Reliability Engineering 9. Безопасность, DevSecOps
ПК-1.2	Разрабатывает и оценивает модели больших данных	<p>Примерные практические задания</p> <ol style="list-style-type: none"> 1. Установить curl. 2. Проверить, что curl возвращает содержимое html-страницы произвольного сайта. 3. Установить nginx (или использовать ранее установленный nginx из прошлой лабораторной), запустить, убедиться, что содержимое тестовой страницы отображается в браузере, доступно по запросу curl. 4. Заменить страницу, которую отдает nginx по умолчанию или добавить новую на отдельный виртуальный хост в соответствии с примером в методическом указании. 5. Установить Docker с помощью curl, настроить его и проверить работу с помощью тестового образа. 6. Запустить nginx в контейнере, проверить его работу в соответствии с примером в методическом указании (через веб-браузер и с помощью curl). 7. Собрать модификацию образа nginx, заменив в нем страницу, отдаваемую по умолчанию, запустить контейнер с этим образом и проверить его работу через браузер или curl-запрос. 8. Создать новый проект docker-compose в соответствии с примером в методическом указании, собрать проект, убедиться, что сборка

		<p>проходит корректно. 9. Запустить проект docker-compose, убедиться, что в результате клиент получает от сервера html-файл. Проверить, что сервер также доступен через браузер. 10. Остановить сервер, заменить содержимое html-страницы на свое, собрать и запустить проект заново, убедиться, что по клиенту доступна новая страница.</p>
ПК-1.3	<p>Организует контроль эффективности работы и предлагает решения руководителю (заказчику)</p>	<p>Требования к оформлению отчета: 1. Титульный лист, оформленный согласно утвержденному образцу. 2. Цели выполняемой лабораторной работы. 3. Задание на лабораторную работу. 4. Описание процесса выполнения работы: для каждого действия, производимого в командной строке, в отчет следует включить: – краткое описание действия; – вводимая команда или команды; – реакция системы на ввод команд (если объем выводимых данных превышает несколько строк, всю информацию включить в отчет не следует). 5. Выводы.</p>

б) Порядок проведения промежуточной аттестации, показатели и критерии оценивания:

Промежуточная аттестация по дисциплине «Численные методы математической физики» включает теоретические вопросы, позволяющие оценить уровень усвоения обучающимися знаний, и практические задания, выявляющие степень сформированности умений и владений, проводится в форме зачета в 6 семестре и экзамена в 7 семестре.

Зачет по дисциплине проводится по результатам отчетности на лабораторных занятиях с опросом в устной форме по этапам выполнения и активного выступления в беседе-обсуждении на лекционных занятиях. Экзамен по результатам ответа на билет.

Показатели и критерии оценивания экзамена:

– на оценку **«отлично»** (5 баллов) – обучающийся демонстрирует высокий уровень сформированности компетенций, всестороннее, систематическое и глубокое знание учебного материала, свободно выполняет практические задания, свободно оперирует знаниями, умениями, применяет их в ситуациях повышенной сложности.

– на оценку **«хорошо»** (4 балла) – обучающийся демонстрирует средний уровень сформированности компетенций: основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.

– на оценку **«удовлетворительно»** (3 балла) – обучающийся демонстрирует пороговый уровень сформированности компетенций: в ходе контрольных мероприятий допускаются ошибки, проявляется отсутствие отдельных знаний, умений, навыков, обучающийся испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

– на оценку **«неудовлетворительно»** (2 балла) – обучающийся демонстрирует знания не более 20% теоретического материала, допускает существенные ошибки, не может показать интеллектуальные навыки решения простых задач.

– на оценку **«неудовлетворительно»** (1 балл) – обучающийся не может показать знания на уровне воспроизведения и объяснения информации, не может показать интеллектуальные навыки решения простых задач.

Перечень примерных вопросов к экзамену

1. Linux.
2. Networks.
3. Container orchestration.
4. Виртуализация и контейнеризация.
5. CI/CD, Clouds and Automation.
6. Monitoring/Logging.
7. Information Security.
8. Development, Databases.