

**Федеральное государственное автономное образовательное  
учреждение высшего образования  
«Московский физико-технический институт  
(национальный исследовательский университет)»**

**УТВЕРЖДЕНО**

**Директор физтех-школы  
прикладной математики и  
информатики**

**А.М. Райгородский**

	<b>Рабочая программа дисциплины (модуля)</b>
<b>по дисциплине:</b>	Промышленное программирование и DevOps
<b>по направлению:</b>	Прикладная математика и информатика
<b>профиль подготовки:</b>	AI360: Передовые методы искусственного интеллекта Физтех-школа Прикладной Математики и Информатики кафедра алгоритмов и технологий программирования
<b>курс:</b>	2
<b>квалификация:</b>	бакалавр

Семестр, формы промежуточной аттестации: 4 (весенний) - Дифференцированный зачет

Аудиторных часов: 60 всего, в том числе:

лекции: 30 час.

семинары: 30 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 30 час.

Всего часов: 90, всего зач. ед.: 2

Программу составил: О.Н. Ивченко, заведующий кафедрой

Программа обсуждена на заседании кафедры алгоритмов и технологий программирования 12.02.2024

## Аннотация

Дисциплина "Промышленное программирование и DevOps" представляет собой комплексный курс, охватывающий современные методы разработки программного обеспечения и принципы управления инфраструктурой. В ходе обучения студенты изучают основные концепции DevOps, включая непрерывную поставку, автоматизацию процессов развертывания и управления конфигурациями. Курс также охватывает методы мониторинга производительности, управления микросервисами, безопасностью и управлением секретами в контексте DevOps.

Студенты получают практические навыки работы с современными инструментами и технологиями, используемыми в индустрии для ускорения процессов разработки, улучшения качества и надежности программного обеспечения, а также для обеспечения безопасности и масштабируемости инфраструктуры. Курс также подчеркивает важность сотрудничества между разработчиками и операционными специалистами для достижения общих целей разработки и эксплуатации программных продуктов.

### 1. Цели и задачи

#### Цель дисциплины

Обучить студентов практическим навыкам разработки и внедрения программного обеспечения в промышленном масштабе с использованием методологий DevOps. Развить у студентов понимание всех этапов жизненного цикла программного обеспечения от разработки до развертывания и поддержки. Способствовать развитию у студентов компетенций для эффективной работы в командах разработчиков и операторов и успешного внедрения DevOps практик в организациях.

#### Задачи дисциплины

- развить умение разрабатывать и внедрять программное обеспечение с использованием современных технологий и инструментов;
- приобретение навыков работы с различными языками программирования и фреймворками.

### 2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.1 Способен применять современные вычислительную технику и сервисы сети Интернет в области (сфере) профессиональной деятельности
	ОПК-2.2 Знает и умеет применять численные математические методы и прикладное программное обеспечение для решения научных задач в профессиональной области
	ОПК-2.3 Знает основные требования информационной безопасности
ОПК-3 Способен составлять и оформлять научные и (или) технические (технологические, инновационные) отчеты (публикации, проекты)	ОПК-3.2 Владеет на практике методологией составления научно-технических отчетов (проектов)
	ОПК-3.1 Знает основные правила оформления научных публикаций и научно-технической документации, в том числе с использованием прикладного программного обеспечения
	ОПК-3.3 Владеет методами визуального и графического представления результатов научной (научно-технической, инновационной технологической) деятельности в виде отчетов, научных публикаций
	ПК-2.1 Знает принципы построения научной работы, методы сбора и анализа полученного материала, способы аргументации

ПК-2 Способен самостоятельно или в качестве члена (руководителя) малого коллектива организовывать и проводить научные исследования и их апробацию

ПК-2.2 Способен планировать и проводить научные исследования самостоятельно или в качестве члена (руководителя) малого научного коллектива

ПК-2.3 Способен проводить апробацию результатов научно-исследовательской работы посредством публикации научных статей и участия в конференциях

### 3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- принципы промышленного программирования и DevOps;
- основные инструменты и технологии DevOps.

уметь:

- разрабатывать и внедрять программное обеспечение с использованием DevOps практик;
- настраивать CI/CD пайплайны и управлять контейнерами;
- использовать облачные технологии в DevOps;
- эффективно работать в команде разработчиков и операторов.

владеть:

- практическими навыками программирования и DevOps;
- навыками работы с инструментами DevOps;
- пониманием влияния DevOps на бизнес и стратегические цели организации.

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

#### 4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Контейнеризация и оркестрация	6	8		3
2	Непрерывная интеграция и непрерывная поставка (CI/CD)	3	3		3
3	Инфраструктура как код (Infrastructure as Code)	5	5		4
4	Мониторинг и логирование	8	6		10
5	Безопасность и DevSecOps	8	8		10
Итого часов		30	30		30
Подготовка к экзамену		0 час.			
Общая трудоёмкость		90 час., 2 зач.ед.			

#### 4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

Семестр: 4 (Весенний)

##### 1. Контейнеризация и оркестрация

Docker: основные принципы, создание контейнеров, управление контейнерами. Kubernetes: управление контейнеризированными приложениями, масштабирование, отказоустойчивость.

##### 2. Непрерывная интеграция и непрерывная поставка (CI/CD)

CI/CD пайплайны: автоматизация сборки, тестирования и развертывания приложений.  
Инструменты CI/CD: Jenkins, GitLab CI/CD, CircleCI, Travis CI.

### 3. Инфраструктура как код (Infrastructure as Code)

Основные принципы IaC: автоматизация управления инфраструктурой через код.  
Инструменты IaC: Terraform, Ansible, Chef, Puppet.

### 4. Мониторинг и логирование

Мониторинг производительности приложений и инфраструктуры. Системы логирования: ELK Stack (Elasticsearch, Logstash, Kibana), Splunk, Prometheus.

### 5. Безопасность и DevSecOps

Принципы безопасной разработки и интеграции безопасности в процесс разработки.  
Инструменты для обеспечения безопасности в DevOps: OWASP, белый ящик тестирование, инструменты сканирования уязвимостей.

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

Необходимое оборудование для лекций: компьютер и мультимедийное оборудование (проектор, звуковая система).

## **6. Перечень рекомендуемой литературы**

Основная литература

1. Компьютерные сети, Computer networks, Электронная версия печатной публикации / . — Санкт-Петербург, Питер, 2021

Дополнительная литература

## **7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)**

Не используются

## **8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)**

На лекциях используется компьютер и мультимедийное оборудование (проектор, звуковая система),

## **9. Методические указания для обучающихся по освоению дисциплины (модуля)**

Студент, изучающий дисциплину, должен с одной стороны, овладеть общим понятийным аппаратом, а с другой стороны, должен научиться применять теоретические знания на практике. Успешное освоение дисциплины требует:

- посещения студентом всех видов аудиторных занятий;
- ведения конспекта в ходе лекционных занятий;
- качественной самостоятельной подготовки к практическим занятиям, активной работы на них;
- активной самостоятельной и аудиторной работы студента;
- своевременной сдачи преподавателю заданий по аудиторным видам работ.

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

**по направлению:** Прикладная математика и информатика  
**профиль подготовки:** АІ360: Передовые методы искусственного интеллекта  
Физтех-школа Прикладной Математики и Информатики  
кафедра алгоритмов и технологий программирования  
**курс:** 2  
**квалификация:** бакалавр

Семестр, формы промежуточной аттестации: 4 (весенний) - Дифференцированный зачет

**Разработчик:** О.Н. Ивченко, заведующий кафедрой

## 1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.1 Способен применять современные вычислительную технику и сервисы сети Интернет в области (сфере) профессиональной деятельности
	ОПК-2.2 Знает и умеет применять численные математические методы и прикладное программное обеспечение для решения научных задач в профессиональной области
	ОПК-2.3 Знает основные требования информационной безопасности
ОПК-3 Способен составлять и оформлять научные и (или) технические (технологические, инновационные) отчеты (публикации, проекты)	ОПК-3.2 Владеет на практике методологией составления научно-технических отчетов (проектов)
	ОПК-3.1 Знает основные правила оформления научных публикаций и научно-технической документации, в том числе с использованием прикладного программного обеспечения
	ОПК-3.3 Владеет методами визуального и графического представления результатов научной (научно-технической, инновационной технологической) деятельности в виде отчетов, научных публикаций
ПК-2 Способен самостоятельно или в качестве члена (руководителя) малого коллектива организовывать и проводить научные исследования и их апробацию	ПК-2.1 Знает принципы построения научной работы, методы сбора и анализа полученного материала, способы аргументации
	ПК-2.2 Способен планировать и проводить научные исследования самостоятельно или в качестве члена (руководителя) малого научного коллектива
	ПК-2.3 Способен проводить апробацию результатов научно-исследовательской работы посредством публикации научных статей и участия в конференциях

## 2. Показатели оценивания компетенций

В результате изучения дисциплины «Промышленное программирование и DevOps» обучающийся должен:

### знать:

- принципы промышленного программирования и DevOps;
- основные инструменты и технологии DevOps.

### уметь:

- разрабатывать и внедрять программное обеспечение с использованием DevOps практик;
- настраивать CI/CD пайплайны и управлять контейнерами;
- использовать облачные технологии в DevOps;
- эффективно работать в команде разработчиков и операторов.

### владеть:

- практическими навыками программирования и DevOps;
- навыками работы с инструментами DevOps;
- пониманием влияния DevOps на бизнес и стратегические цели организации.

## 3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

1. Что такое DevOps и какие основные принципы лежат в его основе?
2. Какие преимущества предоставляет контейнеризация для процесса разработки и развертывания приложений?
3. Какие инструменты используются для автоматизации процесса непрерывной интеграции (CI) и непрерывной поставки (CD)?

4. Что такое инфраструктура как код (IaC) и какие преимущества она предоставляет для управления инфраструктурой?
5. Какие методы мониторинга и логирования используются для отслеживания производительности приложений и инфраструктуры?
6. Какие принципы безопасной разработки входят в понятие DevSecOps и как они интегрируются в процесс разработки?
7. Какие основные задачи выполняют системы управления версиями кода (VCS) в процессе разработки программного обеспечения?
8. Какие принципы лежат в основе микросервисной архитектуры и как она отличается от монолитных приложений?
9. Какие проблемы решает процесс автоматизации тестирования в рамках DevOps-подхода к разработке?
10. Какие методы и инструменты используются для обеспечения отказоустойчивости и масштабируемости приложений в среде DevOps?

#### **4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся**

1. Какие основные принципы лежат в основе непрерывной поставки (Continuous Delivery) и как они отличаются от непрерывной интеграции (Continuous Integration)?
2. Какие роли и обязанности присутствуют в команде DevOps, и как они сотрудничают для достижения общих целей?
3. Какие технологии и практики используются для автоматизации процесса развертывания (deployment) приложений?
4. Каковы основные преимущества использования облачных платформ (например, AWS, Azure, Google Cloud) в контексте DevOps?
5. Какие методы и инструменты используются для управления конфигурациями и версионирования конфигураций в DevOps?
6. Что такое мониторинг производительности (performance monitoring) и какие инструменты помогают в его реализации в DevOps-контексте?
7. Какие принципы и методы включает в себя концепция "GitOps" и как она сопоставляется с традиционными методами управления инфраструктурой?
8. Какие факторы необходимо учитывать при проектировании и развертывании микросервисов в архитектуре приложения?
9. Какие решения и инструменты используются для управления секретами (secrets management) в среде DevOps?
10. Какие основные меры безопасности и контроля доступа важны для инфраструктуры, используемой в DevOps?

#### **Критерии оценивания**

Оценка "Отлично" (10) - полностью и вовремя решены все задачи без ошибок. Продemonстрирован грамотный подход к решению задач, реализованы оптимальные алгоритмы, код оформлен в едином удобочитаемом стиле.

Оценка "Отлично" (9) - полностью и вовремя решены все задачи без ошибок. Продemonстрирован грамотный подход к решению задач, реализованы оптимальные алгоритмы.

Оценка "Отлично" (8) - полностью и вовремя решены все задачи без ошибок. Продemonстрирован грамотный подход к решению задач.

Оценка "Хорошо" (7) - полностью решены все задачи. Допущены несущественные ошибки.

Оценка "Хорошо" (6) - полностью решено большинство задач. В некоторых задачах допущены и не исправлены ошибки, либо некоторые задачи решены частично.

Оценка "Хорошо" (5) - полностью решено две трети задач. В некоторых задачах допущены и не исправлены ошибки, либо некоторые задачи решены частично.

Оценка "Удовлетворительно" (4) - полностью решено более половины задач. В остальных задачах допущены и не исправлены ошибки, либо некоторые задачи решены частично.

Оценка "Удовлетворительно" (3) - полностью решено более половины задач.

Оценка "Неудовлетворительно" (2) - решено менее половины задач.

Оценка "Неудовлетворительно" (1) - не решено ни одной задачи.

## **5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности**

Дифференцированный зачет может проводиться по итогам текущей успеваемости и сдачи заданий и других видов работ, предусмотренных программой дисциплины и (или) путем организации специального опроса, проводимого в устной и (или) письменной форме.

При проведении устного дифференцированного зачета обучающемуся предоставляется 30 минут на подготовку. Опрос обучающегося не должен превышать одного астрономического часа.

Во время проведения дифференцированного зачета обучающиеся могут пользоваться программой дисциплины, а также справочной литературой, конспектами лекций или другими материалами.