

**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**

	Рабочая программа дисциплины (модуля)
по дисциплине:	Основы непрерывной интеграции. Devops
по направлению:	Прикладная математика и информатика
профиль подготовки:	Информатика
	Физтех-школа Прикладной Математики и Информатики кафедра алгоритмов и технологий программирования
курс:	2
квалификация:	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: 75 час.

Всего часов: 135, всего зач. ед.: 3

Программу составил: О.Н. Ивченко, заведующий кафедрой

Программа обсуждена на заседании кафедры алгоритмов и технологий программирования 15.01.2024

Аннотация

Целью обучения 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.

1. Цели и задачи

Цель дисциплины

- 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.

Задачи дисциплины

- изучение жизненного цикла (ЖЦ) программного обеспечения;
- изучить роль DevOps-инженера в ЖЦ;
- разобрать программные инструменты DevOps.

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

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны знать:

- понимание серверной и сетевой инфраструктуры;
- базовые навыки работы с операционными системами;
- базовые знания о базах данных;
- базовые навыки программирования;
- опыт работы с интерфейсами командной строки;
- понимание серверной и сетевой инфраструктуры.

уметь:

- применять нормативно-технические документы (стандарты и регламенты), определяющие требования к проектной и технической документации;
- использовать выбранную среду программирования для разработки процедур интеграции программных модулей.

владеть:

- навыками оценки результатов выполнения назначенных заданий на разработку процедур интеграции, сборку, подключение к внешней среде, проверку работоспособности выпусков программного продукта;
- навыками контроля и оценки качества разработанной проектной и технической документации.

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	Освоение технологии Gitflow	10	10		25
2	Освоение технологии Docker	10	10		25
3	Освоение технологии Gitlab CI	10	10		25
Итого часов		30	30		75
Подготовка к экзамену		0 час.			
Общая трудоёмкость		135 час., 3 зач.ед.			

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

Семестр: 4 (Весенний)

1. Освоение технологии Gitflow

Gitflow - это совместная модель ветвления, которая пытается использовать мощность, скорость и простоту ветвления Git.

2. Освоение технологии Docker

Docker - это проект с открытым исходным кодом для автоматизации развертывания приложений в виде переносимых автономных контейнеров, выполняемых в облаке или локальной среде.

3. Освоение технологии Gitlab CI

Gitlab CI - это сервис, основанный на SAAS — одной из форм облачных вычислений, где вы без труда сможете разместить свои Git-репозиторий, отслеживать возможные проблемы и писать wiki с помощью языка разметки markdown. Gitlab CI также позволяет вам настраивать непрерывную интеграцию с использованием любого из образов Docker.

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

Компьютеры либо ноутбуки с доступом в интернет, камерой и микрофоном. Практические занятия (семинары) и самостоятельная работа студентов должна проводиться в специализированных аудиториях с комплектом мультимедийного оборудования и/или доской для записей материалов.

6.Перечень рекомендуемой литературы

Основная литература

1. Введение в программирование , учебное пособие / И. Ю. Баженова, В. А. Сухомлин. — Москва, ИНТУИТ, 2016.— URL: <https://e.lanbook.com/book/100695> (дата обращения: 30.12.2020). - Полный текст (Режим доступа : из сети МФТИ / Удаленный доступ)
2. Программирование на C++ [Электронный ресурс] / Н. Дейл, Ч. Уимз, М. Хедингтон. — М., ДМК Пресс, 2007.— URL: <https://e.lanbook.com/book/1219> (дата обращения: 26.01.2021). - Полный текст (Режим доступа : из сети МФТИ / Удаленный доступ)

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)

В качестве программных средств будут использоваться редакторы исходного кода (Vim и другие) или интегрированные среды разработки (IntelliJ IDEA, Visual Studio и другие).

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

Успешное освоение курса требует напряжённой самостоятельной работы студента. В программе курса приведено минимально необходимое время для работы студента над темой.

Самостоятельная работа включает в себя:

- проработку учебного материала (по конспектам лекций, учебной и научной литературе), подготовку ответов на вопросы, предназначенных для самостоятельного изучения, доказательство отдельных утверждений, свойств;
- выполнение индивидуальных домашних заданий и итогового проекта.

Промежуточный контроль знаний проводится в виде индивидуальных домашних работ в формате проектной работы.

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Прикладная математика и информатика
профиль подготовки:	Информатика 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

2. Показатели оценивания компетенций

В результате изучения дисциплины «Основы непрерывной интеграции. Devops» обучающийся должен:

знать:

- понимание серверной и сетевой инфраструктуры;
- базовые навыки работы с операционными системами;
- базовые знания о базах данных;
- базовые навыки программирования;
- опыт работы с интерфейсами командной строки;
- понимание серверной и сетевой инфраструктуры.

уметь:

- применять нормативно-технические документы (стандарты и регламенты), определяющие требования к проектной и технической документации;
- использовать выбранную среду программирования для разработки процедур интеграции программных модулей.

владеть:

- навыками оценки результатов выполнения назначенных заданий на разработку процедур интеграции, сборку, подключение к внешней среде, проверку работоспособности выпусков программного продукта;
- навыками контроля и оценки качества разработанной проектной и технической документации.

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

Перечень примерных (типовых) вопросов к текущему контролю:

1. Что такое DevOps?
2. Жизненный цикл ПО
3. DevOps-инженер – роль в проекте разработки и внедрения ПО
4. Обзор систем виртуализации и контейнеризации
5. Введение в экосистему контейнеров на основе Docker
6. Настройка рабочего окружения, подготовка и запуск Docker-контейнеров
7. Структура и содержание профилей информационных систем
8. Методологические основы проектирования информационных систем
9. Методология структурного анализа и проектирования информационных систем SADT
10. Основные понятия нотации IDEF0

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

Перечень вопросов к дифференцированному зачету:

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.

Пример билета:

Билет №1

1. Жизненный цикл ПО.
2. DevOps-инженер – роль в проекте разработки и внедрения ПО.

Критерии оценивания

отлично

10: всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений;

9: систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, правильное обоснование принятых решений;

8: глубокие знания учебной программы дисциплины и умение применять их на практике при решении конкретных задач, правильное обоснование принятых решений;

хорошо

7: твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;

6: знает материал, грамотно излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;

5: знает основной материал, грамотно излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач неточности;

удовлетворительно

4: фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;

3: характер знаний достаточен для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;

неудовлетворительно

2: не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет правильно использовать полученные знания при решении типовых практических задач.

1: не знает формулировок основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

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

Дифференцированный зачет выполняется в формате устного ответа на билет. На подготовку дается 30 минут.