

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

**УТВЕРЖДЕНО**

**Директор высшей школы  
программной инженерии  
А.В. Малеев**

	<b>Рабочая программа дисциплины (модуля)</b>
<b>по дисциплине:</b>	Современные методы DevOps
<b>по направлению:</b>	Программная инженерия
<b>профиль подготовки:</b>	Разработка программно-информационных систем высшая школа программной инженерии высшая школа программной инженерии МФТИ - Яндекс
<b>курс:</b>	3
<b>квалификация:</b>	бакалавр

Семестр, формы промежуточной аттестации: 5 (осенний) - Дифференцированный зачет

Аудиторных часов: 60 всего, в том числе:

лекции: 30 час.

семинары: 30 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 48 час.

Всего часов: 108, всего зач. ед.: 3

Программу составил: А.В. Созыкин, канд. техн. наук, доцент

Программа обсуждена на заседании высшей школы программной инженерии МФТИ - Яндекс 28.04.2023

## Аннотация

Этот курс посвящен основам разработки программного обеспечения. Правильный дизайн - важная часть любого проекта. Этот курс охватывает основы языка программирования Python, основные концепции и языковые конструкции. Наряду с этим этот курс предоставляет инструменты для использования языка программирования Python в сложных проектах. Вы получите представление о правильном дизайне кода, поддержании кодовой базы и интеграции ваших приложений с другими.

### 1. Цели и задачи

#### Цель дисциплины

- Научитесь писать эффективный и читаемый код.
- Изучите передовой опыт разработки программного обеспечения.
- Получите необходимый опыт работы с Python.
- Привыкайте к тестированию и документированию кода.
- Будьте готовы внедрить методы машинного обучения и глубокого обучения.

#### Задачи дисциплины

- Разработка программного обеспечения
- Python
- Тестирование
- Работа в разных средах.

### 2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
УК-2 Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	УК-2.1 Формулирует совокупность взаимосвязанных задач в рамках поставленной цели работы, обеспечивающих ее достижение. Определяет ожидаемые результаты решения поставленных задач
ОПК-5 Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем	ОПК-5.2 Способен оценивать концепции и атрибуты качества программного обеспечения (надежности, безопасности, удобства использования), в том числе роль людей, процессов, методов, инструментов и технологий обеспечения качества
ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.2 Умеет применять языки программирования для решения прикладных задач
	ОПК-6.3 Знает методы тестирования программного кода на ошибки и способен проводить тестирование на различных уровнях (модульное, интеграционное, системное)
	ОПК-6.4 Имеет навыки программирования и тестирования программных продуктов
ПК-3 Способен проектировать, разрабатывать, интегрировать, проверять на работоспособность программное обеспечение	ПК-3.1 Различает синтаксис языков программирования, особенности программирования на этих языках, стандартные библиотеки языков программирования
	ПК-3.2 Умеет выбирать языки программирования для написания программного кода с учетом технического задания
	ПК-3.3 Умеет излагать основные принципы построения и виды архитектуры программного обеспечения, методы и средства проектирования программного обеспечения, методологии разработки программного обеспечения и технологии программирования
ПК-6 Способен разрабатывать и внедрять	ПК-6.1 Знает, как создавать стандарты и методологии разработки программного обеспечения в организации

стандарты и процессы разработки,  
производить их мониторинг и обновления

ПК-6.3 Владеет навыками мониторинга и обновления  
стандартов с учетом изменяющихся требований и  
технологий

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

В результате освоения дисциплины обучающиеся должны

знать:

- представление о строении, функционировании зрительного анализатора;
- представление о психофизиологических и информационных моделях бинокулярного зрения;
- принципы функционирования видеоинтерфейса применительно к системам VR/AR.

уметь:

- принципы функционирования и методологию разработки распределенных систем применительно к задачам создания систем VR/AR;
- строение и принципы функционирования существующих и перспективных графических API.

владеть:

- методологией разработки ПМО всех звеньев систем VR/AR (включая графическое ядро, подсистемы управления виртуальной средой, видеоинтерфейс и др.);
- объектно-ориентированной методологией проектирования и разработки программного кода для всего спектра задач создания систем VR/AR.

### 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	Что такое DevOps и в чем его ценность	3	3		5
2	Системы контроля версий	3	3		5
3	Работа с облачными инфраструктурами	3	3		5
4	Управление конфигурациями	3	3		5
5	Система сборки	3	3		5
6	Локальная разработка и тестирование с Docker-Compose	3	3		5
7	Continuous Integration	3	3		6
8	Основы Kubernetes	4	4		6
9	DO-9. Helm, непрерывная доставка приложений в Kubernetes	5	5		6
Итого часов		30	30		48
Подготовка к экзамену		0 час.			
Общая трудоёмкость		108 час., 3 зач.ед.			

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

Семестр: 5 (Осенний)

1. Что такое DevOps и в чем его ценность

DevOps как философия и способ мышления. DevOps как подход к решению задач и организации процесса разработки и эксплуатации. Мифы о DevOps. Когда нужны практики DevOps. Когда DevOps не нужен. Профессия DevOps-инженер.

## 2. Системы контроля версий

Современные системы управления кодом. Git. Введение. Базовые команды Git. Магия Git flow. Полезные возможности Git.

## 3. Работа с облачными инфраструктурами

Знакомство с облачными инфраструктурами. AWS — Amazon Web Services. GCP — Google Cloud Platform. Microsoft Azure. Яндекс.Облако. MCS — Mail.ru Cloud Solutions. Другие провайдеры.

## 4. Управление конфигурациями

Предпосылки появления подхода IaC. Что такое Infrastructure as Code (IaC). Ansible. Ansible Playbook. Продвинутое написание плейбуков. Ansible-роли. Лучшие практики использования Ansible.Puppet.

## 5. Система сборки

Что такое Docker. Создание Docker-образа. Работа с сетью и дисковой подсистемой в Docker. Основные команды Docker: работа с образами и контейнерами. Где не стоит применять Docker. Экосистема Docker.

## 6. Локальная разработка и тестирование с Docker-Compose

Знакомство с Docker Compose. Формат конфигурационного файла и запуск Docker Compose. Особенности работы с сетью и дисковой подсистемой в Docker Compose. Лучшие практики Docker Compose. Тестирование.

## 7. Continuous Integration

Введение в CI. Основные понятия. Реализация CI. Обзор наиболее распространенных систем CI.

## 8. Основы Kubernetes

Введение в Kubernetes. Архитектура Kubernetes и основные компоненты. Установка K8S. Kubectl. Управление объектами Kubernetes. Манифесты. Работа с секретами.

## 9. DO-9. Helm, непрерывная доставка приложений в Kubernetes

Введение в Helm. Использование готовых Helm чартов. Написание Helm Chart. Лучшие практики. Helm в CI/CD.

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

Стандартная аудитория

## 6.Перечень рекомендуемой литературы

Основная литература

1. Технологии разработки программного обеспечения, Электрон. версия печ. публикации / . — Санкт-Петербург, Питер, 2012

Рекомендуемая литература для самостоятельного изучения:

1. Дэвис Дженнифер. Философия DevOps. Искусство управления IT - СПб.: Питер, 2017 - 416 с. - ISBN 978-5-4461-1141-1.

2. Хамбл, Ким, Дебуа: Руководство по DevOps. Как добиться гибкости, надежности и безопасности мирового уровня - М.:Манн, Иванов и Фербер, 2018 - 512 с. - ISBN 978-5-00100-750-0.

Дополнительная литература

Рекомендуемая литература для самостоятельного изучения:

1. Д. Ким, К. Бер, Д. Спаффорд. Проект «Феникс». Роман о том, как DevOps меняет бизнес к лучшему: пер. с англ. - М.: Бомбора, 2021 - 384 с. - ISBN 978-5-04-101855-9.

2. DevOps для IT-менеджеров. Концентрированное структурированное изложение передовых идей/Скрынник Олег - М.: ДМК Пресс, 2019 - 126 с. - ISBN 978-5-97060-692-6

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

<http://dm.fizteh.ru/>

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

Мультимедийные технологии можно использовать на лекциях и практических занятиях, в том числе на презентациях.

## **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;
- подготовку к практическим занятиям, выполнение нескольких индивидуальных домашних заданий.

Промежуточный контроль знаний проводится в виде письменных опросов по теории. Кроме этого в ходе освоения курса студент должен выполнить проект, содержащий несколько взаимосвязанных заданий с их последующей защитой.

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

<b>по направлению:</b>	Программная инженерия
<b>профиль подготовки:</b>	Разработка программно-информационных систем высшая школа программной инженерии МФТИ - Яндекс высшая школа программной инженерии
<b>курс:</b>	3
<b>квалификация:</b>	бакалавр

Семестр, формы промежуточной аттестации: 5 (осенний) - Дифференцированный зачет

**Разработчик:** А.В. Созыкин, канд. техн. наук, доцент

## 1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
УК-2 Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	УК-2.1 Формулирует совокупность взаимосвязанных задач в рамках поставленной цели работы, обеспечивающих ее достижение. Определяет ожидаемые результаты решения поставленных задач
ОПК-5 Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем	ОПК-5.2 Способен оценивать концепции и атрибуты качества программного обеспечения (надежности, безопасности, удобства использования), в том числе роль людей, процессов, методов, инструментов и технологий обеспечения качества
ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.2 Умеет применять языки программирования для решения прикладных задач
	ОПК-6.3 Знает методы тестирования программного кода на ошибки и способен проводить тестирование на различных уровнях (модульное, интеграционное, системное)
	ОПК-6.4 Имеет навыки программирования и тестирования программных продуктов
ПК-3 Способен проектировать, разрабатывать, интегрировать, проверять на работоспособность программное обеспечение	ПК-3.1 Различает синтаксис языков программирования, особенности программирования на этих языках, стандартные библиотеки языков программирования
	ПК-3.2 Умеет выбирать языки программирования для написания программного кода с учетом технического задания
	ПК-3.3 Умеет излагать основные принципы построения и виды архитектуры программного обеспечения, методы и средства проектирования программного обеспечения, методологии разработки программного обеспечения и технологии программирования
ПК-6 Способен разрабатывать и внедрять стандарты и процессы разработки, производить их мониторинг и обновления	ПК-6.1 Знает, как создавать стандарты и методологии разработки программного обеспечения в организации
	ПК-6.3 Владеет навыками мониторинга и обновления стандартов с учетом изменяющихся требований и технологий

## 2. Показатели оценивания компетенций

В результате изучения дисциплины «Современные методы DevOps» обучающийся должен:

### знать:

- представление о строении, функционировании зрительного анализатора;
- представление о психофизиологических и информационных моделях бинокулярного зрения;
- принципы функционирования видеоинтерфейса применительно к системам VR/AR.

### уметь:

- принципы функционирования и методологию разработки распределенных систем применительно к задачам создания систем VR/AR;
- строение и принципы функционирования существующих и перспективных графических API.

### владеть:

- методологией разработки ПМО всех звеньев систем VR/AR (включая графическое ядро, подсистемы управления виртуальной средой, видеоинтерфейс и др.);
- объектно-ориентированной методологией проектирования и разработки программного кода для всего спектра задач создания систем VR/AR.

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

В чем разница между изменяемыми и неизменяемыми объектами в Python? Каковы преимущества использования изменяемых или неизменяемых типов? Что такое «вызов по назначению»?

2. Что такое закрытие в Python? Когда мы сможем это использовать?
3. Что такое декоратор в Python? Как реализовать собственный декоратор? (Декоратор с параметрами)
4. Что такое диспетчер контекста в Python? использование
5. Генераторы и итераторы в питоне? Реализация, использование
6. Наследование классов, мпо, миксины
7. GIL
8. Многопроцессорность и многопоточность, в чем разница
9. Как мы можем сделать частные атрибуты в классе?
10. Интерфейс класса для создания хешируемых объектов.

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

1. Докажите, что если  $m$ ,  $n$  - два взаимно простых целых числа разной четности, то числа  $m^2 - n^2$  и  $2mn$  также взаимно просты.
2. Напишите и докажите общую формулу для количества различных представлений данного целого числа  $n$  в виде суммы двух квадратов. (Представители, которые не получены друг от друга путем изменения знаков и порядка слов, считаются разными.)
3. На основе полученной формулы выведите нижнюю границу максимального числа равных расстояний между заданными  $n$  точками на плоскости, используя правильную прямоугольную решетку.
4. Постройте правильный пятиугольник с помощью циркуля и линейки.
5. Постройте правильный 15-угольник, используя циркуль и линейку.
6. Вам дается один сегмент. Требуется построить с помощью циркуля и линейки отрезок длины  $x$ , удовлетворяющий уравнению
7. Основываясь на предыдущем задании, докажите, что правильный семиугольник нельзя построить с помощью циркуля и линейки.
8. Докажите, что трисекция угла невозможна.
9. Опишите все возможные комбинации количества черных и белых шаров в урне для голосования, чтобы при случайном вылове двух шаров в выборке без возврата, вероятность вылова двух белых шаров составляла точно 0,5.
10. Рассмотрим соотношение сторон  $a$ ,  $b$ ,  $c$  треугольника, в котором треугольник с вершинами в основании биссектрис равнобедренный. Предполагая, что стороны, сходящиеся на стороне  $c$  большого треугольника, равны, сведем это соотношение к следующему
11. Далее мы рассматриваем куб, определяемый первым из трех уравнений (отказ от требования, чтобы  $a$ ,  $b$ ,  $c$  были сторонами треугольника). Покажите, что полученный куб неразложим, то есть определяющий его многочлен не учитывается.
12. В дополнение к этому, покажите, что наш куб неособен, то есть нет ни одной точки на его проективизации, в которой каждое направление касалось бы (или того же самого, в котором все три первые частных производных многочлена, определяющего его, вырождают ).

Примеры экзаменационных билетов

Билет №1

1. Напишите и докажите общую формулу для количества различных представлений данного целого числа  $n$  в виде суммы двух квадратов.
2. Докажите, что трисекция угла невозможна.

Билет №2

1. Рассмотрим соотношение сторон  $a$ ,  $b$ ,  $c$  треугольника, в котором треугольник с вершинами в основании биссектрис равнобедренный.



2. Опишите все виды комбинаций чисел черных и белых шаров в урне для голосования, чтобы, если два шара случайно выловлены в выборке и не вернулись, вероятность вылова двух белых шаров была точно 0,5.

#### Критерии оценивания

Оценка «отлично (10)» выставляется студенту, проявившему всестороннее, систематическое и глубокое знание материала образовательной программы, самостоятельно выполнившего все задания, предусмотренные программой, глубоко изучившему основную и дополнительную литературу, рекомендованную программой. , активно работает в классе и понимает основные научные концепции по изучаемой дисциплине, проявил творческий подход и научный подход в понимании и представлении материала образовательной программы, ответ на который характеризуется использованием богатых и адекватных терминов, а также последовательным и логичным изложением материала;

Оценка «отлично (9)» выставляется студенту, который продемонстрировал всестороннее систематическое знание материала образовательной программы, самостоятельно выполнил все задачи, предусмотренные программой, глубоко усвоил основную литературу и знаком с рекомендуемой дополнительной литературой. по программе, активно проработал на занятиях, показал системность знаний по дисциплине, достаточную для дальнейшего изучения, а также умение самостоятельно расширять ее, ответ которой отличается точностью используемых терминов, а изложение материала в нем последовательное и логичное;

Оценка «отлично (8)» выставляется студенту, который проявил полное знание материала образовательной программы, не допускает существенных неточностей в своем ответе, самостоятельно выполнил все задания, предусмотренные программой, изучил основную литературу, рекомендованную учебной программой. программа, активно проработанная на занятиях, показала системность его знаний по дисциплине, достаточных для дальнейшего изучения, а также способность самостоятельно их усиливать;

Оценка «хорошо (7)» выставляется студенту, который проявил достаточно полное знание материала образовательной программы, не допускает существенных неточностей в ответе, самостоятельно выполнил все задания, предусмотренные программой, изучил основную рекомендованную литературу по программе, активно работал на занятиях, проявил системность своих знаний по дисциплине, достаточных для дальнейшего изучения, а также способность самостоятельно их усиливать;

Оценка «хорошо (6)» выставляется студенту, который проявил достаточно полное знание материала образовательной программы, не допускает существенных неточностей в своем ответе, самостоятельно выполнил основные задачи, предусмотренные программой, изучил основную литературу. рекомендован программой, показал систематичность своих знаний по дисциплине, достаточную для дальнейшего изучения;

Оценка «хорошо (5)» дается студенту, продемонстрировавшему знание материала основной образовательной программы в объеме, необходимом для дальнейшего обучения и будущей работы по профессии, который, не проявляя достаточной активности на уроках, тем не менее самостоятельно выполнял, овладел основными задачами, предусмотренными программой, освоил основную литературу, рекомендованную программой, допустил ошибки в их выполнении и ответе во время тестирования, но имеет необходимые знания для исправления этих ошибок самостоятельно;

Оценка «удовлетворительно (4)» дается студенту, обнаружившему знание материала основной образовательной программы в объеме, необходимом для дальнейшего обучения и будущей работы по профессии, который, не проявляя достаточной активности на уроках, тем не менее самостоятельно выполнял, выполнил основные задачи, предусмотренные программой, изучил основную литературу, но допустил ошибки в их выполнении и в своем ответе во время теста, но имеет необходимые знания для исправления этих ошибок под руководством преподавателя;

Оценка «удовлетворительно (3)» выставляется обучающемуся, проявившему знание материала основной образовательной программы в объеме, необходимом для дальнейшего обучения и будущей работы по профессии, не проявившего активности на занятиях, самостоятельно выполнившего основные задания, предусмотренные законодательством. программа, но допускающая ошибки в их выполнении и в ответе при тестировании, но обладающая необходимыми знаниями для устранения под руководством преподавателя наиболее существенных ошибок;

Оценка «неудовлетворительно (2)» выставляется студенту, который показал пробелы в знаниях или недостаток знаний по значительной части материала основной образовательной программы, не выполнил самостоятельно основные задания, требуемые программой, допустил принципиальные ошибки в выполнении предусмотренных программой задач, не имеющего возможности продолжить учебу или начать профессиональную деятельность без дополнительной подготовки по данной дисциплине;

Оценка «неудовлетворительно (1)» ставится студенту при отсутствии ответа (отказ от ответа) или когда представленный ответ не соответствует сути вопросов, содержащихся в задании.

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

Во время дифференцированного зачета студенту разрешается использовать программу дисциплины.