

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

УТВЕРЖДАЮ

Заведующий кафедрой
электроники
Усков Г.К.



31.01.2024 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
Б1.В.ДВ.03.01 Средства непрерывной интеграции и непрерывной
доставки приложений

1. Код и наименование направления подготовки/специальности:

09.03.01 Информатика и вычислительная техника

2. Профиль подготовки/специализация:

Программно-аппаратные средства информационных систем

3. Квалификация (степень) выпускника:

бакалавр

4. Форма обучения:

очная

5. Кафедра, отвечающая за реализацию дисциплины:

электроники

6. Составители программы:

Ряполов М.П., к.ф.-м.н.

7. Рекомендована:

НМС физического факультета 23.06.2021, № протокола: 6

8. Учебный год: *2027/2028*

Семестр(ы): *8*

9. Цели и задачи учебной дисциплины

Основная цель освоения учебной дисциплины — знакомство студентов с современными практиками разработки программного обеспечения и запуска его в эксплуатацию

Задачи учебной дисциплины:

- Изучение концепции непрерывной интеграции и непрерывной доставки приложений, а так же методологии devops.
- Практика установки и настройки платформы для реализации технологий CI/CD
- Практика написания кода с его автоматическим тестированием при отправке в репозиторий системы управления версиями
- Настройка механизмов непрерывного развертывания

10. Место учебной дисциплины в структуре ООП:

Дисциплина входит в вариативную часть и является дисциплиной по выбору. Для её изучения студенты должны пройти курсы Б1.О.11 Информатика, Б1.О.13 Вычислительная техника, Б1.О.14 Операционные системы, Б1.О.15 Программирование, Б1.О.16 Сети и телекоммуникации, Б1.О.17 Защита информации, Б1.О.18 Базы данных и Б1.В.17 Системное администрирование.

11. Планируемые результаты обучения по дисциплине/модулю (знания, умения, навыки), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями) и индикаторами их достижения:

Код	Название компетенции	Код(ы)	Индикатор(ы)	Планируемые результаты обучения
ПК-2	Способен осуществлять концептуальное, функциональное и логическое проектирование систем среднего и крупного масштаба и сложности	ПК-2.1	Знает возможности современных и перспективных средств разработки программных продуктов, технических средств, методы и средства проверки работоспособности программного обеспечения, методы и средства рефакторинга и оптимизации программного кода	Использует инструменты для автоматизированного развертывания приложений корректно
		ПК-2.2	Проводит анализ требований, вырабатывает варианты реализации программного продукта с учетом требований	На основе требований к программному обеспечению проектирует линии непрерывной интеграции и доставки приложений до потребителя
		ПК-2.3	Проводит оценку и обоснование рекомендуемых решений	Пишет обоснование по использованию инструментов для развертывания и тестирования приложений
		ПК-2.4	Анализирует возможности удовлетворения требований к программному обеспечению	Анализирует пути доставки приложений и вырабатывает рекомендации по оптимизации и улучшению
		ПК-2.5	Оценивает время и трудоемкость реализации программного обеспечения с учетом требований	Оценивает трудоемкость работ по настройке непрерывной интеграции и развертыванию приложений
ПК-4	Способен разрабатывать компоненты системных	ПК-4.4	Применяет методы и средства сборки модулей и компонентов программного обеспечения, разработки процедур для	Использует инструменты сборки и доставки программного обеспечения в рамках GitHub, GitLab, Bitbucket

	программных продуктов		развертывания программного обеспечения, миграции и преобразования данных, создания программных интерфейсов	
		ПК-4.5	Разрабатывает и документирует программные интерфейсы	Проектирует, реализует и тестирует поддержку проектов через API GitHub, GitLab, Bitbucket
		ПК-4.6	Разрабатывает процедуры сборки модулей и компонентов программного обеспечения	Проектирует, реализует и тестирует системы сборки программных модулей с помощью инструментов GitHub, GitLab, Bitbucket

12. Объем дисциплины в зачетных единицах/час.— 4/144.

Форма промежуточной аттестации **экзамен**

13. Трудоемкость по видам учебной работы

Вид учебной работы		Трудоемкость		
		Всего	По семестрам	
			8	
Аудиторные занятия		32	32	
в том числе:	лекции	22	22	
	практические			
	лабораторные	10	10	
Самостоятельная работа		76	76	
в том числе: курсовая работа (проект)				
Форма промежуточной аттестации (экзамен – ___ час.)		36	36	
Итого:		144	144	

13.1. Содержание дисциплины

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК *
1. Лекции			
1.1	Методология DevOps, инструменты CI/CD	История концепции непрерывной доставки приложений. Основные определения и концепции DevOps и CI/CD. Требования для применения этих концепций на практике.	-
1.2	Процесс непрерывного развёртывания.	Ветки разработки, тестирование одобрение на публикацию и разворачивание веб приложения.	-
1.3	Документирование проектов	Использование встроенных комментариев для автоматической генерации комментариев. Инструменты автоматической генерации документации. Использование внешних сервис для генерации документации (wiki движки, локальные файлы)	-
1.4	Платформа GitHub	Риски облачного решения, основной функционал, лицензионные ограничения. Запуск автоматического тестирования и разворачивания приложений.	-
1.5	Платформа GitLab	Локальная установка, настройка. Лицензионные	-

		ограничения. Запуск автоматического тестирования и разворачивания приложений.	
2. Лабораторные занятия			
2.1	Установка проформы Gitlab	Локальная установка и настройка Gilab	-
2.2	Создание проекта веб приложения		-
2.3	Настройка автоматического тестирования	Настройка автоматического тестирования в GitLab	-
2.4	Настройка публикации приложения в Gilab	Настройка публикации приложения в Gilab	-
2.5	Формулирование требований	Формулирование требований к разрабатываемому веб приложению, написание планов его поэтапной реализации	-
2.6	Документирование приложения	Документирование создаваемого приложения с использованием встроенных комментариев и системы генерации документации Doxygen, написание документации в вики движках	-
2.7	Процесс непрерывного развёртывание приложения	Разработка проекта с поэтапным тестирование и развёртыванием приложения	-

13.2. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование темы (раздела) дисциплины	Виды занятий (количество часов)				Всего
		Лекции	Практические	Лабораторные	Самостоятельная работа	
1.1	Методология DevOps, инструменты CI/CD	4			6	10
1.2	Процесс непрерывного развёртывания.	4		2	10	16
1.3	Документирование проектов	4		4	25	33
1.4	Платформа GitHub	4			10	14
1.5	Платформа GitLab	6		4	25	35
	Итого:	22		10	76	108

14. Методические указания для обучающихся по освоению дисциплины

Для успешного освоения дисциплины необходимо изучение лекционных материалов и выполнение лабораторных заданий. Все лабораторные задания представляют собой один проект и оцениваются не только по конкретному этапу, но и по наличию в проекте истории действий на предыдущих этапах. Часть практики по наработке навыков уходит на самостоятельную работу, в рамках которой студенты должны разработать самостоятельный проект. Полученный проект будет базой для их экзаменационной работы. Лекционные и лабораторные занятия могут проводиться в очном режиме или дистанционно с использованием видеоконференций. Лабораторные задания демонстрируются лично и сдаются в виде файлов или репозиториях с программным кодом.

15. Перечень основной и дополнительной литературы, ресурсов интернет, необходимых для освоения дисциплины

а) основная литература:

№ п/п	Источник
1.	Херинг, М. DevOps для современного предприятия : учебное пособие / М. Херинг ; перевод с английского М. А. Райтмана. — Москва : ДМК Пресс, 2020. — 232 с. — ISBN 978-5-97060-836-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/140580 . — Режим доступа: для авториз. пользователей.
2.	Грувер, Г. Запуск и масштабирование DevOps на предприятии / Г. Грувер. — Москва : ДМК Пресс, 2018. — 80 с. — ISBN 978-5-97060-704-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/116130 . — Режим доступа: для авториз. пользователей.
3.	Скрынник, О. В. DevOps для ИТ-менеджеров: концентрированное структурированное изложение передовых идей / О. В. Скрынник. — 2-е изд. — Москва : ДМК Пресс, 2019. —

126 с. — ISBN 978-5-97060-692-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/112933 . — Режим доступа: для авториз. пользователей.

б) дополнительная литература:

№ п/п	Источник
4.	Sander van Vugt Red Hat RHCSA/RHCE 7 Cert Guide: Red Hat Enterprise Linux 7 (EX200 and EX300) / Sander van Vugt — Pearson IT Certification, 2015. — 1008 p.
5.	Бреснахэн, Кристин. Linux на практике / Кристин Бреснахэн, Ричард Блум ; [пер. с англ. С. Черников] .— Санкт-Петербург [и др.] : Питер, 2017 .— 381 с.

в) информационные электронно-образовательные ресурсы (официальные ресурсы интернет)*:

№ п/п	Ресурс
6.	Github URL - https://github.com/
7.	Gitlab. The complete DevOps platform URL - https://about.gitlab.com/

16. Перечень учебно-методического обеспечения для самостоятельной работы

№ п/п	Источник
8.	Херинг, М. DevOps для современного предприятия : учебное пособие / М. Херинг ; перевод с английского М. А. Райтмана. — Москва : ДМК Пресс, 2020. — 232 с. — ISBN 978-5-97060-836-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/140580 . — Режим доступа: для авториз. пользователей.

17. Образовательные технологии, используемые при реализации учебной дисциплины, включая дистанционные образовательные технологии (ДОТ), электронное обучение (ЭО), смешанное обучение):

При реализации учебной дисциплины лекции и практические занятия могут проводиться очно в аудитории или с использованием видеоконференцсвязи в дистанционном формате. Все задания в обязательном порядке сдаются на портале электронного обучения. Текущая и промежуточные аттестации в дополнение к отправке ответов на задания могут сопровождаться собеседованием. Собеседование, так же как и занятия, может проводиться как очно, так и в дистанционном режиме с помощью видеоконференций.

18. Материально-техническое обеспечение дисциплины:

Для проведения занятий требуется компьютерный класс с проектором или большеформатным экраном. На компьютерах должно быть установлено программное обеспечение для создания виртуальных машин (VirtualBox, KVM либо HyperV). Возможна работа с виртуальными машинами не локально, а через портал самообслуживания в ЦОД вуза. Необходим доступ в интернет для использования внешних общедоступных сервис.

19. Оценочные средства для проведения текущей и промежуточной аттестаций

Порядок оценки освоения обучающимися учебного материала определяется содержанием следующих разделов дисциплины:

№ п/п	Наименование раздела дисциплины (модуля)	Компетенция(и)	Индикатор(ы) достижения компетенции	Оценочные средства
1.	Методология DevOps, инструменты CI/CD	ПК-2	ПК-2.1, ПК-2.2	Тесты, практикоориентированные задания
2.	Процесс непрерывного развертывания.	ПК-2	ПК-2.1, ПК-2.2	Тесты, практикоориентированные задания

№ п/п	Наименование раздела дисциплины (модуля)	Компетенция(и)	Индикатор(ы) достижения компетенции	Оценочные средства
3.	Документирование проектов	ПК-2	ПК-2.3, ПК-2.4, ПК-2-5	Тесты, практикоориентированные задания
4.	Платформа GitHub	ПК-4	ПК-4.4, ПК-4.5, ПК-4.6	Тесты, практикоориентированные задания, индивидуальные проекты
5.	Платформа GitLab	ПК-4	ПК-4.4, ПК-4.5, ПК-4.6	Тесты, практикоориентированные задания, индивидуальные проекты
Промежуточная аттестация форма контроля - экзамен				Набор практических заданий

20 Типовые оценочные средства и методические материалы, определяющие процедуры оценивания

20.1 Текущий контроль успеваемости

Контроль успеваемости по дисциплине осуществляется с помощью практикоориентированных заданий, тестовых заданий и индивидуальных проектов.

Примеры практикоориентированных заданий:

1. Создание репозитория для проекта на GitHub
2. Создание репозитория для проекта на GitLab
3. Реализация разработки проекта с помощью 3-х веток для текущей разработки, тестирования и выкладывания финальных версии.
4. Встраивание в код автоматических тестов.
5. Настройка правил установки приложения.

Примеры тестовых заданий:

Выберите один или несколько ответов:

Основными задачами DevOps являются:

- Автоматизация тестирования программного проекта
- Автоматизация сохранения кода в центральном репозитории
- Автоматизация установки приложения на рабочих станциях разработчиков
- Автоматизация установки проекта на сервере для промышленного использования

Введите короткий ответ:

Хранилище кода проекта, необходимое для организации его автоматического тестирования

СУВ, система управления версиями, version control system, vcs

Соедините термин с его определением:

Commit	Операций отправки кода в локальный репозиторий
Push	Отправка изменения в локальном репозитории на сервер.
Pull	Получение изменений с сервера в локальный репозиторий и применение их в текущий код проекта
Fetch	Получение из серверного репозитория изменений без применения их к коду проекта

Примеры индивидуальных проектов:

1. Разработка веб сайта с простым блогом и книгой отзывов.
2. разработка веб приложения для организации онлайн чата.

20.2 Промежуточная аттестация

Промежуточная аттестация по дисциплине осуществляется с помощью практико-ориентированного задания.

Примерные задания:

1. Настройка репозитория проекта на Gitlab и организация автоматического тестирования кода при отправке его на сервер. Оповещение руководителя проекта о результатах автоматической проверки и отправка его на ручное тестирование.
2. Настройка сборки и установки проекта на GitHub в репозитории с 3-мя ветками разработки.

При проведении промежуточной аттестации обучающийся получает одно случайное задание и выполняет его в течении ограниченного времени. Оценивается функционал полученного в результате решения. Допускается использование подготовленной в течении семестра собственной документации и материалов из открытых источников. По окончании времени или завершения выполнения задания обучающийся демонстрирует работающий функционал.

Задание считается полностью выполненным, если весь описанный в нём функционал работает и обучающийся может это продемонстрировать. В этом случае ему выставляется оценка «отлично». Если функционал реализован не более, чем на 75%, либо обучающийся не может его продемонстрировать самостоятельно, он получает оценку «хорошо». При функционале реализованном на 50% оценка «удовлетворительно». При реализации функционала менее, чем на 30% обучающийся получает оценку «неудовлетворительно».