Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 02.11.2023 11:08:12
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение
высшего образования
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
Факультет информационных технологий

УТВЕРЖДАЮ
Декан факультета
«Информационные технологии»
_____/Д.Г.Демидов/
«16» мая 2022

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Основы непрерывной интеграции автоматизированных информационных систем»

Направление подготовки/специальность:

09.03.02 Информационные системы и технологии

Профиль/специализация:

Автоматизированные системы обработки информации и управления

Квалификация
Бакалавр

Форма обучения
Очная

Москва 2022 г.

Разработчик(и):
к.т.н, доцент кафедры
«Информатика и информационные технологии»  / П.С. Новиков /

Согласовано:
Заведующий кафедрой «Информатики и
информационных технологий», к.т.н.  / Е.В.Булатников/

Содержание

1 Цели и задачи дисциплины 4
2 Место дисциплины в структуре образовательной программы 4
3 Структура и содержание дисциплины 5
3.1 Виды учебной работы и трудоемкость (по формам обучения) 5
3.2 Тематический план изучения дисциплины (по формам обучения) 5
3.3 Содержание дисциплины 6
3.4 Тематика семинарских/практических и лабораторных занятий 8
3.5 Тематика курсовых проектов (курсовых работ) 12
4 Учебно-методическое и информационное обеспечение 12
4.1 Нормативные документы и ГОСТы 12
4.2 Основная литература 12
4.3 Дополнительная литература 13
4.4 Электронные образовательные ресурсы 13
4.5 Лицензионное и свободно распространяемое программное обеспечение 13
4.6 Современные профессиональные базы данных и информационные справочные системы 13
5 Материально-техническое обеспечение 14
6 Методические рекомендации 14
6.1 Методические рекомендации для преподавателя по организации обучения 14
6.2 Методические указания для обучающихся по освоению дисциплины 15
7 Фонд оценочных средств 15
7.1 Методы контроля и оценивания результатов обучения 15
7.2 Шкала и критерии оценивания результатов обучения 15
7.3 Оценочные средства 17

1 Цели и задачи дисциплины

Цель курса: формирование у студентов системы знаний и навыков в области непрерывной интеграции автоматизированных информационных систем, изучение основных инструментов, используемых в интеграции современных автоматизированных систем, а также приобретение практических навыков работы с ними для повышения профессиональной компетентности и расширения возможностей трудоустройства в сфере программирования.

Задачи курса:

- Изучить основные понятия и принципы непрерывной интеграции автоматизированных систем.
- Освоить процесс разработки программного обеспечения с использованием непрерывной интеграции.
- Изучить особенности внедрения непрерывной интеграции в проекты, включая управление изменениями кода и обеспечение безопасности.
- Овладеть навыками работы с инструментами и технологиями, используемыми в непрерывной интеграции.

Обучение по дисциплине «Основы непрерывной интеграции автоматизированных информационных систем» направлено на формирование у обучающихся следующих компетенций:

Код и наименование компетенций	**Индикаторы достижения компетенции**
ПК-4. Способен проводить интеграцию программных модулей и компонент и верификацию выпусков программного продукта.	ИПК-4.1. Знает методы и способы интеграции программных модулей информационных и автоматизированных систем по обработке информации. ИПК-4.2. Умеет проводить верификацию выпусков информационных и автоматизированных систем по обработке информации и их систем управления. ИПК-4.3. Имеет навыки применения программного обеспечения для верификации версий используемых ИТ в информационных и автоматизированных систем по обработке информации и их систем управления в виде готовых продуктов.

2 Место дисциплины в структуре образовательной программы

Дисциплина относится к части, формируемой участниками образовательных отношений учебных дисциплин обязательной части Блока 1.

Дисциплина взаимосвязана логически и содержательно-методически со следующими дисциплинами и практиками ООП:

- Управление проектами;
- Разработка технической документации;
- Системы управления разработкой программного обеспечения.

3 Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 3 зачетных единицы (108 академических часа).

3.1 Виды учебной работы и трудоемкость (по формам обучения)

3.1.1. Очная форма обучения

№ п/п	Вид учебной работы	Количество часов	Семестр
			8
1	**Аудиторные занятия**	**40**	40
	В том числе:		
1.1	Лекции		
1.2	Семинарские/практические занятия		
1.3	Лабораторные работы	40	40
2	**Самостоятельная работа**	**68**	68
3	**Промежуточная аттестация**		
	Экзамен/зачет/диф.зачет		зачет
	Итого:	**108**	108

3.2 Тематический план изучения дисциплины (по формам обучения)

3.2.1 Очная форма обучения

№ п/п	Разделы/темы дисциплины	Трудоемкость, час					
		Всего	Аудиторная работа				Самостоятельная работа
			Лекции	Семинарские/ практические занятия	Лабораторные занятия	Практическая подготовка	
1	Тема 1. Основы непрерывного развертывания. Проблема развертывания программного обеспечения. Стратегия управления конфигурациями. Непрерывная интеграция. Реализация стратегии непрерывного развертывания.	27			10		17
2	Тема 2. Конвейер развертывания. Структура конвейера развертывания. Сценарии сборки и развертывания. Стадии фиксации. Автоматические приемочное тестирование. Развертывание и выпуск приложения.	27			10		17
3	Тема 3. Процесс поставки. Управление инфраструктурой.	27			10		17

	Управление средами. Управление данными. Управление компонентами. Управление зависимостями. Управление версиями.						
4	Тема 4. Управление процессом непрерывного развертывания.	27			10		17
	Промежуточная аттестация: зачет						
Итого		**108**			**40**		**68**

3.3 Содержание дисциплины

Т-1	Основы непрерывного развертывания.	

Краткое содержание (перечень рассматриваемых вопросов) темы:

- Проблема развертывания программного обеспечения.
- Стратегия управления конфигурациями.
- Непрерывная интеграция.
- Реализация стратегии непрерывного развертывания.

Контрольные вопросы:

1. Что представляет собой проблема развертывания программного обеспечения, и почему она важна в разработке информационных систем?
2. Какие основные этапы включает в себя стратегия управления конфигурациями в контексте непрерывной интеграции?
3. Что такое непрерывная интеграция, и какие преимущества она приносит в разработке программного обеспечения?
4. Какие ключевые компоненты конвейера развертывания используются для реализации непрерывной интеграции?
5. Какие сценарии сборки и развертывания могут быть интегрированы в конвейер развертывания?
6. Какие стадии фиксации играют роль в обеспечении надежности процесса непрерывной интеграции?
7. Что такое автоматизированное приемочное тестирование, и как оно связано с непрерывной интеграцией?
8. Какие методы развертывания и выпуска приложений могут быть использованы в контексте непрерывной интеграции?
9. Как управление инфраструктурой, средами, данными, компонентами и зависимостями связано с успешной реализацией непрерывной интеграции?
10. Какие шаги и практические рекомендации можно предложить для эффективной реализации и управления стратегией непрерывного развертывания программного обеспечения?

Т-2	Конвейер развертывания.	

Краткое содержание (перечень рассматриваемых вопросов) темы:

- Структура конвейера развертывания.
- Сценарии сборки и развертывания.
- Стадии фиксации.
- Автоматические приемочное тестирование.
- Развертывание и выпуск приложения.

Контрольные вопросы:

1. Какова структура конвейера развертывания, и какие этапы включает в себя этая структура?

2. Какие типы сценариев сборки и развертывания программного обеспечения могут быть реализованы в рамках конвейера развертывания?
3. Какие основные цели стадии фиксации в процессе непрерывной интеграции, и почему они важны?
4. Какие виды автоматического приемочного тестирования используются в непрерывной интеграции, и какие преимущества они предоставляют?
5. Какие шаги и процедуры могут быть включены в процесс развертывания и выпуска приложений, чтобы обеспечить эффективность и надежность этого процесса?
6. Какие инструменты и технологии могут быть использованы для автоматизации сценариев сборки и развертывания в конвейере развертывания?
7. Какие методы и стратегии можно применить для обеспечения безопасности при автоматическом приемочном тестировании?
8. Какие роли играют контейнеры и виртуализация в процессе развертывания и выпуска приложений?
9. Какие этапы и процессы включает в себя развертывание приложения на продуктивное окружение, и как это связано с концепцией непрерывной интеграции?
10. Как можно оценить успех развертывания и выпуска приложения в контексте непрерывной интеграции, и какие метрики можно использовать для оценки качества и производительности процесса?

Т-3	Процесс поставки.	

Краткое содержание (перечень рассматриваемых вопросов) темы:

- Управление инфраструктурой.
- Управление средами.
- Управление данными.
- Управление компонентами.
- Управление зависимостями.
- Управление версиями.

Контрольные вопросы:

1. Что включает в себя управление инфраструктурой в контексте непрерывной интеграции, и почему это важно для процесса разработки?
2. Какие методы и инструменты используются для управления средами, в которых происходит непрерывная интеграция?
3. Какое значение имеет управление данными в процессе непрерывной интеграции, и какие аспекты данных необходимо учитывать?
4. Какие компоненты программного обеспечения подвергаются управлению в рамках непрерывной интеграции, и какие преимущества это предоставляет для процесса разработки?
5. Какие виды зависимостей могут возникать между компонентами при разработке информационных систем, и как управление зависимостями способствует успешной непрерывной интеграции?
6. Какие методы используются для управления версиями программного обеспечения в контексте непрерывной интеграции, и как это влияет на разработку и развертывание?
7. Какие риски могут возникнуть при непрерывной интеграции, если управление инфраструктурой не настроено должным образом, и как их можно избежать?
8. Какие лучшие практики можно применить для управления средами разработки и тестирования с целью обеспечения стабильности и консистентности процесса непрерывной интеграции?
9. Какие методы управления данными могут быть полезны для эффективного тестирования и анализа результатов в процессе непрерывной интеграции?

10. Какие инструменты для управления компонентами, зависимостями и версиями программного обеспечения вы бы предложили для успешной непрерывной интеграции?

Т-4	Управление процессом непрерывного развертывания.	

Краткое содержание (перечень рассматриваемых вопросов) темы:

- Непрерывное развертывание.
- Инфраструктура как код.
- Мониторинг и обратная связь.

Контрольные вопросы:

1. Что представляет собой концепция "Непрерывного развертывания" в контексте разработки программного обеспечения?
2. Какие основные преимущества обеспечивает практика непрерывного развертывания для разработки и развертывания приложений?
3. Какие ключевые принципы "Инфраструктуры как код" и как они связаны с автоматизацией управления инфраструктурой?
4. Какие инструменты и технологии используются для реализации "Инфраструктуры как код"?
5. Каким образом "Инфраструктура как код" содействует непрерывной интеграции и развертыванию?
6. Что включает в себя процесс "Мониторинга и обратной связи" в контексте непрерывного развертывания, и какие метрики могут быть отслеживаемы?
7. Каким образом мониторинг производственной среды связан с обнаружением и реагированием на проблемы в реальном времени?
8. Каким образом обратная связь от пользователей и тестировщиков может влиять на процесс непрерывного развертывания?
9. Какие методы и инструменты используются для автоматизации мониторинга и сбора обратной связи?
10. Как можно обеспечить безопасность и надежность при практике непрерывного развертывания и мониторинге важных параметров системы?

3.4 Тематика семинарских/практических и лабораторных занятий

Л-1	Обзор возможностей GitLab	

Выполняемые практические задания:

- Создание проекта и репозитория в GitLab
- Управление задачами (Issues) и задачами для разработчиков (Epics)
- Совместная работа и управление доступом

Контрольные вопросы:

1. Что такое GitLab, и какие основные возможности он предоставляет для управления разработкой и совместной работой в команде?
2. Как создать новый проект в GitLab, и какие настройки проекта доступны для конфигурации?
3. Какие основные компоненты системы управления задачами (Issues) предоставляет GitLab, и как они могут быть использованы в процессе разработки?
4. Как настраивается система непрерывной интеграции и развертывания (CI/CD) в GitLab, и какие преимущества она предоставляет для разработчиков?
5. Как можно управлять доступом к проектам и репозиториям в GitLab, и какие уровни доступа доступны для настройки?

Л-2	Установка GitLab.	

Выполняемые практические задания:

- Установка GitLab.
- Создание административной учетной записи.
- Создание проекта и репозитория.
- Работа с репозиторием.

Контрольные вопросы:

1. Что такое GitLab, и какие основные функции он предоставляет?
2. Какие шаги необходимо выполнить для установки GitLab на сервере? Какие системные требования нужно учесть?
3. Как создать административную учетную запись в GitLab после установки? Какие права предоставляются администратору?
4. Какой командой можно клонировать удаленный репозиторий GitLab на локальную машину?
5. Какие основные шаги нужно выполнить для инициализации нового репозитория в GitLab и отправки первого коммита?

Л-3	Изучение архитектуры GitLab.	

Выполняемые практические задания:

- Анализ структуры GitLab-репозитория.
- Исследование архитектуры микросервисов.
- Анализ баз данных.
- Изучение архитектуры высокой доступности (HA) GitLab.

Контрольные вопросы:

1. Какие основные компоненты архитектуры GitLab вы смогли выявить при анализе структуры репозитория? Какие функции они выполняют?
2. Какие микросервисы входят в архитектуру GitLab, и какие функциональные области они обслуживают? Дайте примеры сервисов и их назначения.
3. Какие базы данных используются GitLab, и какие данные они хранят? Какие таблицы присутствуют в базах данных, и как они взаимосвязаны?
4. Какие механизмы обеспечивают высокую доступность и отказоустойчивость в архитектуре GitLab? Укажите основные методы и инструменты.
5. Что необходимо сделать, чтобы настроить GitLab в конфигурации с высокой доступностью (HA)? Какие шаги и компоненты потребуются для обеспечения отказоустойчивости системы?

Л-4	Управление пользователями в GitLab и доступ по SSH.	

Выполняемые практические задания:

- Создание нового пользователя.
- Назначение прав доступа.
- Настройка доступа по SSH.
- Удаление пользователя и отзыв прав доступа.

Контрольные вопросы:

1. Какой процесс необходим для создания нового пользователя в GitLab, и какие роли доступа могут быть назначены этому пользователю?
2. Как можно настроить права доступа к определенному проекту в GitLab для конкретных пользователей или групп? Какие уровни доступа доступны, и как они различаются?
3. Как создать и настроить SSH-ключ для аутентификации в GitLab? Какие шаги необходимо выполнить, чтобы клонировать репозиторий через SSH?
4. Как можно отозвать права доступа у пользователя или группы в GitLab? Какие последствия это может иметь для их доступа к проектам?

5. Почему важно обеспечивать безопасность ключей SSH и управление доступом в системе управления версиями, такой как GitLab? Какие меры безопасности можно рекомендовать для защиты от несанкционированного доступа и утечек данных?

Л-5	Знакомство с CI/CD и CI/CD Pipeline.	

Выполняемые практические задания:

- Настройка среды CI/CD.
- Создание и настройка CI-пайплайна.
- Создание и настройка CI-пайплайна.
- Мониторинг и откат изменений.

Контрольные вопросы:

1. Что означают аббревиатуры CI и CD в контексте разработки программного обеспечения, и какие основные задачи они решают?
2. Какие инструменты или системы управления версиями вы знаете, которые могут использоваться в процессе CI/CD?
3. Какие этапы обычно включает в себя CI/CD-пайплайн, и какие задачи выполняются на каждом из этих этапов?
4. В чем заключается принцип "непрерывной интеграции" (CI) в CI/CD, и почему он важен для успешной разработки программного обеспечения?
5. Какие методы и практики безопасности можно внедрить в CI/CD-пайплайн, чтобы обеспечить безопасность приложения в процессе разработки и развертывания?

Л-6	Выполнение тестов в GitLab	

Выполняемые практические задания:

- Настройка проекта в GitLab.
- Написание простых тестов.
- Настройка и выполнение CI/CD пайплайна.
- Интеграция с отчетами о тестировании.

Контрольные вопросы:

1. Что такое система непрерывной интеграции (CI), и как она связана с выполнением тестов в GitLab?
2. Какие компоненты .gitlab-ci.yml файла используются для определения этапов CI/CD пайплайна в GitLab?
3. Какие виды тестов вы можете выполнить в GitLab, и какие средства для их запуска предоставляет GitLab?
4. Как можно интегрировать отчеты о тестировании (например, JUnit, Jest) в GitLab и почему это важно для процесса разработки?
5. Что такое GitLab Runner, и как он связан с выполнением тестов в GitLab? Какие виды Runner доступны, и какой выбрать для выполнения тестов в различных сценариях?

Л-7	GitLab runners	

Выполняемые практические задания:

- Настройка и регистрация GitLab Runner.
- Создание и настройка CI/CD пайплайна.
- Настройка переменных окружения и секретов.
- Интеграция с контейнеризацией.

Контрольные вопросы:

1. Что такое GitLab Runner, и какую роль он играет в процессе CI/CD?

2. Какие команды используются для установки и регистрации GitLab Runner на локальной машине?
3. Как создать файл .gitlab-ci.yml, и какие элементы он должен содержать?
4. Какие преимущества предоставляют переменные окружения и секреты в настройке CI/CD пайплайнов?
5. Как можно настроить GitLab Runner для выполнения заданий в контейнерах Docker, и почему это полезно в процессе CI/CD?

Л-8	GitLab SaaS	

Выполняемые практические задания:

- Создание репозитория и базовая работа с Git.
- Работа с ветками и слиянием (branching and merging).
- Интеграция с CI/CD (непрерывная интеграция и доставка).
- Управление доступом и совместной работой.

Контрольные вопросы:

1. Что такое система контроля версий (СКВ) и какие преимущества она предоставляет при разработке программного обеспечения?
2. Какие основные операции можно выполнять с Git? Перечислите и объясните их.
3. Какие основные различия между локальным и удаленным репозиторием Git? Как они взаимодействуют друг с другом?
4. Что такое ветки (branches) в Git и для чего они используются? Как создать новую ветку и выполнить слияние (merge) веток?
5. Какие функциональности предоставляет GitLab для управления проектами и командной работой? Как можно настроить интеграцию непрерывной интеграции и доставки (CI/CD) с помощью GitLab?

Л-9	Построение конвейера CI/CD	

Выполняемые практические задания:

- Настройка среды CI/CD.
- Создание пайплайна CI/CD.
- Интеграция с контейнеризацией.
- Развертывание на тестовом и продукционном сервере.

Контрольные вопросы:

1. Что такое CI и CD, и в чем заключается их основное различие?
2. Какие инструменты и технологии используются для автоматизации процесса CI/CD?
3. Какие преимущества приносит внедрение конвейера CI/CD в разработку программного обеспечения?
4. Какие этапы включают в себя типичный конвейер CI/CD, и какие задачи выполняются на каждом этапе?
5. Какие лучшие практики и методологии связаны с CI/CD, и какие проблемы разработчики могут столкнуться при его внедрении?

Л-10	Создание образа Docker	

Выполняемые практические задания:

- Создание базового образа.
- Создание кастомного образа на основе приложения.
- Многоконтейнерное приложение.
- Оптимизация размера образа.

Контрольные вопросы:

1. Что такое Docker и в чем состоит его основное назначение?

2. В чем разница между контейнером и виртуальной машиной, и какие преимущества предоставляют контейнеры в сравнении с виртуализацией?
3. Какие ключевые компоненты Docker и их функции (например, Docker Engine, Dockerfile, Docker Hub)?
4. Как создать Docker-образ с помощью Dockerfile? Какие инструкции часто используются в Dockerfile для создания образа?
5. Какие команды используются для управления контейнерами в Docker (например, запуск, остановка, удаление, перезапуск)? Как можно передать аргументы и параметры контейнеру при его запуске?

Л-11	Загрузка приложения на сервер	

Выполняемые практические задания:

- Загрузка приложения на сервер.
- Загрузка и развертывание приложения.
- Резервное копирование и восстановление.
- Обновление приложения.

Контрольные вопросы:

1. Какой веб-сервер вы выбрали для развертывания приложения на сервере, и почему? Какие альтернативы рассматривались?
2. Какие шаги и инструменты вы использовали для загрузки приложения на сервер? Опишите процесс от выбора хостинга/сервера до фактической установки приложения.
3. Каким образом вы обеспечиваете безопасность приложения на сервере? Какие меры безопасности были применены, чтобы защитить сервер и данные?
4. Как производится обновление вашего приложения на сервере? Как вы управляете версиями и минимизируете простой при обновлении?
5. Каковы методы и частота резервного копирования данных вашего приложения? Как быстро и эффективно вы можете восстановить приложение в случае сбоя?

3.5 Тематика курсовых проектов (курсовых работ)

Курсовые проекты/задания по данной дисциплине не предусмотрены.

4 Учебно-методическое и информационное обеспечение

4.1 Нормативные документы и ГОСТы

1. https://fgos.ru/fgos/fgos-01-03-02-prikladnaya-matematika-i-informatika-9/2.

"Положения об организации образовательного процесса в Московском Политехническом университете"

4.2 Основная литература

1. Фишерман, Л. В. Git. Практическое руководство. Управление и контроль версий в разработке программного обеспечения : руководство / Л. В. Фишерман. — Санкт-Петербург : Наука и Техника, 2021. — 304 с. — ISBN 978-5-94387-547-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/191470
2. Сейерс, Э. Х. Docker на практике / Э. Х. Сейерс, А. Милл ; перевод с английского Д. А. Беликов. — Москва : ДМК Пресс, 2020. — 516 с. — ISBN 978-5-97060-772-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/131719

3. Моуэт, Э. Использование Docker / Э. Моуэт ; научный редактор А. А. Маркелов ; перевод с английского А. В. Снастина. — Москва : ДМК Пресс, 2017. — 354 с. — ISBN 978-5-97060-426-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/93576
4. Васильева, М. А. Система контроля версий. Основы командной разработки / М. А. Васильева, К. М. Филипченко. — Санкт-Петербург : Лань, 2022. — 144 с. — ISBN 978-5-507-44630-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/261089

4.3 Дополнительная литература

1. Кочер, П. С. Микросервисы и контейнеры Docker : руководство / П. С. Кочер ; перевод с английского А. Н. Киселева. — Москва : ДМК Пресс, 2019. — 240 с. — ISBN 978-5-97060-739-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123710
2. Годзурас, Э. Docker Compose для разработчика : руководство / Э. Годзурас ; перевод с английского А. Н. Киселева. — Москва : ДМК Пресс, 2022. — 220 с. — ISBN 978-5-93700-203-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/348110
3. Васильева, М. А. Система контроля версий. Основы командной разработки / М. А. Васильева, К. М. Филипченко. — Санкт-Петербург : Лань, 2022. — 144 с. — ISBN 978-5-507-44630-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/261089

4.4 Электронные образовательные ресурсы

ЭОР находится в разработке.

4.5 Лицензионное и свободно распространяемое программное обеспечение

При изучении дисциплины потребуется ряд лицензионного и свободно распространяемого программного обеспечения:

1. **GitLab Community Edition (CE).** GitLab CE является открытым и свободно распространяемым программным обеспечением под лицензией MIT.
2. **Git.** Git - это распределенная система контроля версий, которая также свободно распространяется и поддерживается лицензией GPL.
3. **Docker.** Docker - это свободное программное обеспечение, которое позволит виртуализировать и управлять контейнерами для развертывания приложений на сервере
4. **PostgreSQL**. GitLab использует PostgreSQL как базу данных по умолчанию. PostgreSQL - это система управления реляционными базами данных.
5. **Nginx**. Для обеспечения проксирования и управления входящими запросами к GitLab, Nginx используется по умолчанию в GitLab CE. Nginx также распространяется под лицензией MIT и бесплатно доступен.

4.6 Современные профессиональные базы данных и информационные справочные системы

1. https://elenph.org/
2. https://www.philosophy.ru/

3. https://iphlib.ru/library

5 Материально-техническое обеспечение

Для проведения занятий необходимы проекторы и экраны. Компьютеры или ноутбуки для преподавателя и студентов также требуются для доступа к учебным ресурсам, выполнения практических заданий и демонстрации программного обеспечения. Устойчивое интернет-соединение необходимо для доступа к онлайн-ресурсам и совместной работы студентов.

Помимо этого, требуется специальное программное обеспечение, связанное с учебным материалом и практическими заданиями, для обеспечения эффективного обучения.

6 Методические рекомендации

6.1 Методические рекомендации для преподавателя по организации обучения

Методика преподавания дисциплины и реализация компетентностного подхода в изложении и восприятии материала предусматривает использование следующих активных и интерактивных форм проведения групповых, индивидуальных, аудиторных занятий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков у обучающихся:

- посещение лекций;
- выполнение практических заданий на семинарах;
- индивидуальные и групповые консультации студентов преподавателем, в том числе в виде защиты выполненных заданий в рамках самостоятельной работы;
- посещение профильных конференций и работа на мастер-классах экспертов и специалистов индустрии.

Самостоятельная внеаудиторная работа студентов составляет 50% от общего объема дисциплины и состоит из:

- выполнению практических заданий;
- чтения литературы и освоения дополнительного материала в рамках тематики дисциплины;
- подготовки к промежуточной аттестации.

При подготовке к занятиям следует предварительно проработать материал занятия, предусмотрев его подачу точно в отведенное для этого время занятия. Следует подготовить необходимые материалы – теоретические сведения, задачи и др. При проведении занятия следует контролировать подачу материала и решение заданий с учетом учебного времени, отведенного для занятия.

При проверке работ и отчетов следует учитывать не только правильность выполнения заданий, но и оптимальность выбранных методов решения, правильность выполнения всех его шагов.

При организации и проведения промежуточной аттестации в практико-ориентированной форме следует использовать утвержденные кафедрой Методические рекомендации.

6.2 Методические указания для обучающихся по освоению дисциплины

Изучение дисциплины осуществляется в строгом соответствии с целевой установкой в тесной взаимосвязи учебным планом. Основой теоретической подготовки студентов являются аудиторные занятия.

В процессе самостоятельной работы студенты закрепляют и углубляют знания, полученные во время аудиторных занятий, дорабатывают конспекты и записи, готовятся к проведению и обрабатывают результаты лабораторных работ, готовятся к промежуточной аттестации, а также самостоятельно изучают отдельные темы учебной программы.

На занятиях студентов, в том числе предполагающих практическую деятельность, осуществляется закрепление полученных, в том числе и в процессе самостоятельной работы, знаний. Особое внимание обращается на развитие умений и навыков установления связи положений теории с профессиональной деятельностью будущего специалиста.

Самостоятельная работа осуществляется индивидуально. Контроль самостоятельной работы организуется в двух формах:

- самоконтроль и самооценка студента;
- контроль со стороны преподавателей (текущий и промежуточный).

Критериями оценки результатов самостоятельной работы студента являются:

- уровень освоения студентом учебного материала;
- умения студента использовать теоретические знания при выполнении практических задач;
- сформированность компетенций;
- оформление материала в соответствии с требованиями.

7 Фонд оценочных средств

7.1 Методы контроля и оценивания результатов обучения

В процессе обучения используются следующие оценочные формы самостоятельной работы студентов, оценочные средства текущего контроля успеваемости и промежуточных аттестаций.

7.2 Шкала и критерии оценивания результатов обучения

Показателем оценивания компетенций на различных этапах их формирования является достижение обучающимися планируемых результатов обучения по дисциплине (модулю).

Показатель:	**Критерии оценивания**			
	Допороговое значение	**Пороговое значение**		
	2	**3**	**4**	**5**
ПК-4. Способен проводить интеграцию программных модулей и компонент и верификацию выпусков программного продукта.				
ИПК-4.1. Знает методы и способы интеграции программных модулей информационных и автоматизированных систем по обработке информации.	Обучающийся демонстрирует полное отсутствие или недостаточное соответствиематериалу дисциплины знаний, указанных в индикаторах компетенций	Обучающийся демонстрирует неполное соответствие следующих знаний, указанных в индикаторах компетенций	Обучающийся демонстрирует частичное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины «Знать»	Обучающийся демонстрирует полное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины

	дисциплины «Знать» (см. п. 3).	дисциплины «Знать» (см. п. 3). Допускаются значительные ошибки, проявляется недостаточность знаний, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании знаниями при их переносе на новые ситуации.	(см. п. 3). Но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	«Знать» (см. п. 3). Свободно оперирует приобретенными знаниями.
ИПК-4.2. Умеет проводить верификацию выпусков информационных и автоматизированных систем по обработке информации и их систем управления.	Обучающийся не умеет или в недостаточной степени умеет выполнять действия, указанных в индикаторах компетенций дисциплины «Уметь» (см. п. 3).	Обучающийся демонстрирует неполное соответствие умений, указанных в индикаторах компетенций дисциплины «Уметь» (см. п. 3). Допускаются значительные ошибки, проявляется недостаточность умений, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании умениями при их переносе на новые ситуации.	Обучающийся демонстрирует частичное соответствие умений, указанных в индикаторах компетенций дисциплины «Уметь» (см. п. 3). Умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе умений на новые, нестандартные ситуации.	Обучающийся демонстрирует полное соответствие умений, указанных в индикаторах компетенций дисциплины «Уметь» (см. п. 3). Свободно оперирует приобретенными умениями, применяет их в ситуациях повышенной сложности.
ИПК-4.3. Имеет навыки применения программного обеспечения для верификации версий используемых ИТ в информационных и автоматизированных систем по обработке информации и их систем управления в виде готовых продуктов.	Обучающийся не владеет или в недостаточной степени владеет приемами, методами и иными умениями, указанными в индикаторах компетенций дисциплины «Владеть» (см. п. 3).	Обучающийся в неполном объеме владеет приемами, методами и иными умениями, указанными в индикаторах компетенций дисциплины «Владеть» (см. п. 3). Допускаются	Обучающийся частично владеет приемами, методами и иными умениями, указанными в индикаторах компетенций дисциплины «Владеть» (см. п. 3). Навыки освоены, но допускаются незначительные ошибки, неточности,	Обучающийся в полном объеме владеет приемами, методами и иными умениями, указанными в индикаторах компетенций дисциплины «Владеть» (см. п. 3). Свободно применяет полученные навыки в ситуациях

		значительные ошибки, проявляется недостаточность владения навыками по ряду показателей. Обучающийся испытывает значительные затруднения при применении навыков в новых ситуациях.	затруднения при аналитических операциях, переносе умений на новые, нестандартные ситуации.	повышенной сложности.

7.3 Оценочные средства

Промежуточная аттестация обучающихся в форме зачёта проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом по данной дисциплине (модулю), при этом учитываются результаты текущего контроля успеваемости в течение семестра. Оценка степени достижения обучающимися планируемых результатов обучения по дисциплине (модулю) проводится преподавателем, ведущим занятия по дисциплине (модулю) методом экспертной оценки. По итогам промежуточной аттестации по дисциплине (модулю) выставляется оценка «зачтено» или «не зачтено».

К промежуточной аттестации допускаются только студенты, выполнившие все виды учебной работы, предусмотренные рабочей программой дисциплины. При этом используется балльно-рейтинговая система, включающая следующие критерии оценки.

Критерий	Значение критерия
Выполнение и защита лабораторных работ в срок	+5 баллов за каждую защищенную на отлично лабораторную работу;
Невыполнение и/или не защита (защита с оценкой «неудовлетворительно») лабораторных работ.	+1 балл за каждую защищенную на хорошо лабораторную работу.
Выполнение зачетного задания	Максимальное значение критерия – не более 20 баллов.

Максимальная сумма набираемых по дисциплине баллов – 100. С началом каждого нового семестра изучения дисциплины набранные баллы обнуляются и рейтинг студента ведется заново. Перевод набранных баллов в оценку промежуточной аттестации производится согласно следующей таблице.

Оценка по балльно-рейтинговой системе	Оценка по итоговой аттестации
0 … 49	Не зачтено
55 … 100	Зачтено

7.3.1 Типовой билет для промежуточной аттестации

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(МОСКОВСКИЙ ПОЛИТЕХ)

ЗАЧЕТНЫЙ БИЛЕТ №1
по дисциплине
«ОСНОВЫ НЕПРЕРЫВНОЙ ИНТЕГРАЦИИ АВТОМАТИЗИРОВАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ»
направление подготовки 09.03.02 Информационные системы и технологии

ВОПРОСЫ:

1. Объясните, что такое непрерывная интеграция (CI) и почему она важна в разработке программного обеспечения. Укажите основные цели и преимущества CI.
2. Опишите процесс непрерывной интеграции с использованием GitLab CI/CD. Какие основные шаги и компоненты включаются в этот процесс?
3. Что такое Docker? Какие преимущества Docker предоставляет для упаковки и развертывания приложений? Укажите ключевые термины, связанные с Docker.
4. Как вы создадите Docker-контейнер для веб-приложения? Опишите шаги от создания Dockerfile до запуска контейнера.

Утверждено: _____ / _____ / «__» _____ 20__г.