

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский ядерный университет «МИФИ»
Обнинский институт атомной энергетики –
филиал федерального государственного автономного образовательного учреждения высшего образования
«Национальный исследовательский ядерный университет «МИФИ»
(ИАТЭ НИЯУ МИФИ)

Утверждено на заседании
УМС ИАТЭ НИЯУ МИФИ
Протокол №2-8/2024 От 30.08.2024

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Контейнеризация приложений в Docker

Шифр, название дисциплины

01.04.02 «Прикладная математика и информатика»

Шифр, название специальности/направления подготовки

Математическое моделирование и прикладной анализ данных

Название программы магистратуры

магистр

(Квалификация (степень) выпускника)

Форма обучения: очная

г. Обнинск 2024 г.

Программа составлена в соответствии с требованиями образовательного стандарта высшего образования национального исследовательского ядерного университета «МИФИ» по направлению подготовки 01.04.02 – Прикладная математика и информатика. (квалификация (степень) магистр).

Программу составил:

_____ С.В. Ермаков, доцент, к.ф.-м.н, доцент

Рецензент:

_____ Г.Е. Деев, доцент, к.ф.-м.н, доцент

Программа рассмотрена на заседании ОИКС

(протокол № 5/7 от «30» июля от 2024 г.)

Руководитель направления подготовки 01.04.02
«Прикладная математика и информатика»

_____ Ермаков С.В.

« ____ » _____ 2024 г.

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

В результате освоения ООП магистратуры обучающийся должен овладеть следующими результатами обучения по дисциплине:

Коды компетенций	Результаты освоения ООП <i>Содержание компетенций*</i>	Перечень планируемых результатов обучения по дисциплине**
ПК-6	способен к проектированию и разработке наукоемкого программного обеспечения на основе технического задания	З-ПК-6 Знать основные цели и задачи проектирования и разработки наукоемкого программного обеспечения на основе технического задания. У-ПК-6 Уметь разрабатывать наукоемкое программное обеспечение на основе технического задания. В-ПК-6 Владеть навыками разработки и проектирования наукоемкого программного обеспечения на основе технического задания.
ПК-5	способен четко формулировать цели и задачи научно-прикладных проектов, разрабатывать концептуальные и теоретические модели решаемых задач	З-ПК-5 Знать основные цели и задачи научно-прикладных проектов, разрабатывать концептуальные и теоретические модели решаемых задач. У-ПК-5 Уметь четко формулировать цели и задачи научно-прикладных проектов, разрабатывать концептуальные и теоретические модели решаемых задач В-ПК-5 Владеть навыками разработки теоретических моделей решаемых задач.

2. Место дисциплины в структуре ООП магистратуры

Дисциплина реализуется в рамках профессионального модуля.

Для освоения дисциплины необходимы компетенции, сформированные в рамках изучения следующих дисциплин: Linux, Git

Дисциплина изучается на 1 курсе в 2 семестре.

3. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам занятий) и на самостоятельную работу обучающихся

Общая трудоемкость (объем) дисциплины составляет 2 зачетных единиц (з.е.), 72 академических часа.

3.1. Объем дисциплины по видам учебных занятий (в часах)

	Семестр		
	№ 1	№ 2	Всего
	Количество часов на вид работы:		

Контактная работа обучающихся с преподавателем			
Аудиторные занятия (всего)		32	32
В том числе:			
<i>лекции</i>		16	16
<i>практические занятия</i>		16	16
<i>лабораторные занятия</i>			
Промежуточная аттестация			
В том числе:			
<i>зачет</i>		Зачет с оценкой	Зачет с оценкой
<i>экзамен</i>			
Самостоятельная работа обучающихся (всего)		40	40
В том числе:			
<i>проработка учебного (теоретического) материала</i>		10	10
<i>выполнение индивидуальных заданий</i>		10	10
<i>подготовка ко всем видам контрольных испытаний текущего контроля успеваемости (в течение семестра)</i>		10	10
<i>подготовка ко всем видам контрольных испытаний промежуточной аттестации (по окончании семестра)</i>		10	10
<i>Всего (часы):</i>		72	72
<i>Всего (зачетные единицы):</i>		2	2

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

4.1. Разделы дисциплины и трудоемкость по видам учебных занятий (в академических часах)

№ п/п	Наименование раздела /темы дисциплины	Общая трудоём- кость всего (в часах)	Виды учебных занятий, включая самостоятельную работу обучающихся и трудоёмкость (в часах)				Формы текущего контроля успевае- мости
			Аудиторные учебные занятия			СРО	
			Лек	Сем/Пр	Лаб		
1.			16	16	-	40	
1.1.	Введение в Docker	5	1	1		3	
1.2.	Основы контейнеров	5	1	1	-	3	
1.3.	: Docker образы	5	1	1	-	3	
1.4.	Создание своих Docker	5	1	1	-	3	

	образов						
1.5.	Работа с контейнерами	5	1	1		3	
1.6.	Сети в Docker	7	2	2	-	3	
1.7.	Docker Volumes и управление данными	7	2	2	-	3	
1.8.	Docker Compose	7	2	2	-	3	
1.9.	Оптимизация Docker образов	7	2	2		3	
1.10.	Развертывание и CI/CD с Docker	8	2	2	-	4	
1.11	Итоговый проект	7	1	1		9	

4.2. Содержание дисциплины, структурированное по разделам (темам)

Лекционный курс

№	Наименование раздела /темы дисциплины	Содержание
1.1.	Введение в Docker	Что такое Docker, зачем он нужен.
1.2.	Основы контейнеров	Что такое контейнеры, чем они отличаются от виртуальных машин.
1.3.	Docker образы	Что такое Docker образы, Docker Hub.
1.4.	Создание своих Docker образов	Создание своих Docker образов
1.5.	Работа с контейнерами	Команды управления контейнерами — запуск, остановка, удаление, просмотр состояния.
1.6.	Сети в Docker	Основы сетей в Docker — bridge, host, none, overlay.
1.7.	Docker Volumes и управление данными	Volumes для хранения данных контейнеров
1.8.	Docker Compose	Что такое Docker Compose, зачем он нужен.
1.9	Оптимизация Docker образов	Лучшая практика написания Dockerfile — слои, минимизация размера образов.
1.10	Развертывание и CI/CD с Docker	Развертывание приложений с Docker на серверах, основы CI/CD с Docker.
1.11	Итоговый проект	Итоговый проект

Практические/семинарские занятия

№	Наименование раздела /темы дисциплины	Содержание
1.1.	Введение в Docker	Установка Docker на различных ОС (Linux, macOS, Windows).
1.2.	Основы контейнеров	Создание первого контейнера. Использование docker run для запуска контейнера с hello-world и nginx.
1.3.	Docker образы	Поиск и скачивание образов с Docker Hub. Команды docker pull, docker images.
1.4.	Создание своих Docker образов	Написание своего Dockerfile для простого приложения на Python/Node.js.

1.5.	Работа с контейнерами	Использование команд docker start, docker stop, docker rm, docker ps -a.
1.6.	Сети в Docker	Создание и управление сетями Docker с помощью docker network create и подключения контейнеров.
1.7.	Docker Volumes и управление данными	Создание томов, монтирование их к контейнерам через docker volume create и docker run -v.
1.8.	Docker Compose	Написание docker-compose.yml для создания многоконтейнерного приложения (например, веб-сервер и база данных).
1.9.	Оптимизация Docker образов	Оптимизация уже созданного Dockerfile (например, использование alpine образов).
1.10.	Развертывание и CI/CD с Docker	Настройка простого развертывания на сервере с Docker или интеграция с CI/CD (например, GitLab CI, GitHub Actions).
1.11.	Итоговый проект	Итоговый проект

Лабораторные занятия

Не предусмотрены.

5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

В качестве учебно-методических материалов используется рекомендованная литература.

6. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

6.1. Паспорт фонда оценочных средств по дисциплине

№ п/п	Контролируемые разделы (темы) дисциплины (результаты по разделам)	Код контролируемой компетенции (или её части) / и ее формулировка	Наименование оценочного средства
1.4	Создание своих Docker образов	ПК-5	Контрольная работа № 1
1.6.	Сети в Docker	ПК-6	Контрольная работа № 1
1.8.	Docker Compose	ПК-5	Контрольная работа № 2
1.9.	Оптимизация Docker образов	ПК-6	Контрольная работа № 2

6.2. Типовые контрольные задания или иные материалы

6.2.1. Зачет с оценкой

В билете два теоретических вопроса и один практический

Теоретические вопросы билета:

1. Что такое Docker и какие проблемы он решает?
2. Чем Docker отличается от виртуальных машин?
3. Какие команды используются для проверки установленной версии Docker?
4. Что такое контейнер и как он работает в Docker?

5. Как контейнеры отличаются от виртуальных машин?
6. Как можно просмотреть логи контейнера, используя Docker?
7. Что такое Docker образ и как он связан с контейнерами?
8. Как найти и загрузить образ из Docker Hub с помощью команды `docker pull`?
9. Чем отличается команда `docker pull` от `docker build`?
10. Что такое Dockerfile и какие основные инструкции он поддерживает?
11. Как создать Docker образ с помощью команды `docker build`?
12. Как использовать COPY и ADD в Dockerfile для добавления файлов в контейнер?
13. Какие команды Docker используются для управления жизненным циклом контейнера?
14. Как запустить, остановить и удалить контейнер с помощью Docker?
15. Как просмотреть список всех контейнеров, включая остановленные?
16. Какие типы сетевых драйверов доступны в Docker и в чем их различия?
17. Как создать собственную сеть Docker с помощью команды `docker network create`?
18. Как подключить контейнер к созданной сети?
19. Что такое Docker Volumes и зачем они нужны?
20. Как создать и использовать volume для хранения данных контейнера?
21. Как смонтировать volume в контейнер с помощью команды `docker run -v`?
22. Что такое Docker Compose и как он помогает в разработке многоконтейнерных приложений?
23. Как выглядит базовая структура файла `docker-compose.yml`?
24. Чем отличается команда `docker-compose up` от `docker-compose down`?
25. Какие методы оптимизации Docker образов существуют?
26. Как использование alpine образов помогает уменьшить размер конечного Docker образа?
27. Что такое многослойность образов и как она влияет на размер образа?
28. Как Docker помогает в процессе развертывания приложений?
29. Что такое CI/CD и как Docker используется в этих процессах?
30. Как настроить автоматическую сборку Docker образа с использованием GitLab CI или GitHub Actions?

Критерий оценки – правильность и полнота ответа на вопросы. Оценка выставляется по шкале от 0 до 40 баллов: теоретические вопросы – 30 баллов, 10 баллов – дополнительные вопросы. Зачет считается сданным при оценке не ниже 25 баллов.

6.2.2. Контрольная работа № 1

Оптимизация Docker образа.

Создайте Dockerfile для простого Node.js приложения и оптимизируйте его, используя минимальный образ (например, `node:alpine`). Измерьте размер исходного и оптимизированного образов.

6.2.2. Контрольная работа № 2

Логирование контейнеров.

Настройте систему централизованного логирования с использованием ELK Stack (Elasticsearch, Logstash, Kibana) или другого инструмента для сбора и анализа логов контейнеров. Настройте отправку логов из контейнеров в Elasticsearch и отображение в Kibana.

б) критерии оценивания компетенций (результатов) – правильная работа кода программы, понимание алгоритма метода оптимизации, умение вывести необходимые для алгоритма формулы.

в) описание шкалы оценивания:

Каждая задача оценивается по шкале от 0 до 10 баллов.

Контрольная работа считается выполненной успешно при суммарной оценке не ниже 18 баллов.

6.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Форма аттестации	Наименование оценочного средства	Баллы
Зачет (100 баллов)	Контрольная работа № 1	30
	Контрольная работа № 2	30
	Ответы на билет	40

7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

а) основная учебная литература:

1. Карлос Сантана — "Docker для разработчиков. Полное руководство" — 2020 — 392 с.
2. Джеймс Тёрнбулл — "Docker. Секреты мастерства" — 2020 — 270 с.
3. Маркус Хейцман — "Docker и контейнеризация. Руководство для системных администраторов" — 2020 — 278 с.

б) дополнительная учебная литература:

1. Стивен Картрайт — "Docker и Kubernetes. Руководство для разработчиков и системных администраторов" — 2021 — 450 с.

8. Перечень ресурсов* информационно-телекоммуникационной сети «Интернет» (далее - сеть «Интернет»), необходимых для освоения дисциплины

-

9. Методические указания для обучающихся по освоению дисциплины

Вид учебного занятия	Организация деятельности студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; помечать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначить вопросы, термины, материал, который вызывает трудности, пометить и попытаться найти ответ в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на консультации, на практическом занятии.

Практические занятия	Проработка рабочей программы, уделяя особое внимание целям и задачам, структуре и содержанию дисциплины. Работа с конспектом лекций, просмотр рекомендуемой литературы. Изучение выбранной предметной области на примерах решения задач семинарских занятий, индивидуальных домашних заданий.
Курсовая работа	Не предусмотрена
Контрольная работа	Ознакомиться с основной и дополнительной литературой, включая справочные издания, зарубежные источники, основополагающие термины. Попрактиковаться в решении аналогичных домашних задач по всем темам контрольных работ.
Лабораторная работа	Не предусмотрена.
Подготовка к зачету	При подготовке к зачету необходимо ориентироваться на конспекты лекций и рекомендуемую литературу.

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем (при необходимости)

Издательская система LaTeX для подготовки докладов, презентаций и учебного материала.

11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Видеопроектор, компьютер, издательская система LaTeX для подготовки докладов, презентаций и учебного материала.

12. Иные сведения и (или) материалы

12.1. Перечень образовательных технологий, используемых при осуществлении образовательного процесса по дисциплине

Часов в интерактивной форме – 8.

В ходе практических занятий происходит публичное обсуждение каждой решаемой задачи. При этом студенты высказывают свои мнения по выбору наиболее простого способа поиска оптимального решения.

После решения домашних работ на консультациях проводится разбор допущенных студентами ошибок.

12.2. Формы организации самостоятельной работы обучающихся (темы, выносимые для самостоятельного изучения; вопросы для самоконтроля; типовые задания для самопроверки)

Некоторые темы изучаются студентами самостоятельно. Для изучения используется приведённая в списке основная и дополнительная литература. Контроль освоения материала осуществляется при проверке контрольных работ, домашнего задания и на зачете.

№	Тема и часть, изучаемая (осваиваемая) самостоятельно
1.1	Kubernetes
1.2	CI/CD (Continuous Integration / Continuous Deployment)
1.3	DevOps практики

№	Тема и часть, изучаемая (осваиваемая) самостоятельно
1.4	Микросервисы
1.5	Сетевые технологии в Docker
1.6	Безопасность Docker
1.7	Облачные технологии
1.8	Docker Swarm
1.9	Мониторинг и логирование
1.10	Системы управления контейнерами

Вопросы и задания для самоконтроля по всем темам:

1. Что такое контейнер и чем он отличается от виртуальной машины?
2. Какие ключевые компоненты включает в себя архитектура Docker?
3. В чем разница между Docker образом и контейнером?
4. Как создать новый Docker образ с использованием Dockerfile?
5. Как можно просмотреть список всех локальных Docker образов на вашем устройстве?
6. Как запустить, остановить и удалить контейнер с помощью команд Docker?
7. Как получить логи контейнера с помощью Docker?
8. Как создать собственную сеть Docker с помощью команды `docker network create`?
9. Чем отличаются Docker Volumes от bind mounts?
10. Что такое Docker Compose и в чем его основное преимущество?
11. Как создать файл `docker-compose.yml` для запуска многоконтейнерного приложения?
12. Какие методы можно использовать для уменьшения размера Docker образа?
13. Что такое CI/CD и как Docker интегрируется в эти процессы?
14. Как управлять безопасностью Docker контейнеров и образов?
15. В чем отличие между Docker Swarm и Kubernetes как системами оркестрации контейнеров?

12.3. Краткий терминологический словарь

Агрегация	процесс объединения данных, например, вычисление сумм, средних значений, минимумов и максимумов по группам данных.
Уникальные идентификаторы	уникальные значения, которые помогают идентифицировать строки или записи в таблицах, часто используются для объединения таблиц
Колоночное хранение	способ хранения данных, при котором значения для каждого столбца таблицы хранятся отдельно, что ускоряет выполнение аналитических запросов в больших объемах данных