

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Нижегородский государственный университет
им. Н.И. Лобачевского»**

Институт информационных технологий, математики и механики

УТВЕРЖДЕНО

решением Ученого совета ННГУ

протокол № 10 от 02.12.2024 г.

Рабочая программа дисциплины

Разработка систем глубокого обучения

Уровень высшего образования

Магистратура

Направление подготовки / специальность

02.04.02 - Фундаментальная информатика и информационные технологии

Направленность образовательной программы

Искусственный интеллект

Форма обучения

очная

г. Нижний Новгород

2025 год начала подготовки

1. Место дисциплины в структуре ОПОП

Дисциплина Б1.В.ДВ.05.03 Разработка систем глубокого обучения относится к части, формируемой участниками образовательных отношений образовательной программы.

2. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине (модулю), в соответствии с индикатором достижения компетенции		Наименование оценочного средства	
	Индикатор достижения компетенции (код, содержание индикатора)	Результаты обучения по дисциплине	Для текущего контроля успеваемости	Для промежуточной аттестации
ПК-8: Способен к разработке новых алгоритмических, методических и технологических решений в конкретной сфере профессиональной деятельности	ПК-8.1: Знает методику разработки новых алгоритмических, методических и технологических решений ПК-8.2: Умеет применять полученные знания для разработки новых алгоритмических, методических и технологических решений ПК-8.3: Имеет практический опыт составления технического задания на разработку информационной системы	ПК-8.1: Знает понятия качества кода, системной архитектуры, переносимости программного обеспечения и docker. Знает современные практики программирования, направленные на поддержание высокого качества систем в условиях непрерывной доработки. ПК-8.2: Умеет использовать практики организации проектной работы, рефакторинга, разработки через тестирование на примере разработки программного обеспечения систем глубокого обучения. ПК-8.3: Обладает навыками обеспечения переносимости программного обеспечения при разработке систем глубокого обучения.	Собеседование Практическая задача	Зачёт: Контрольные вопросы

3. Структура и содержание дисциплины

3.1 Трудоемкость дисциплины

	очная
--	-------

Общая трудоемкость, з.е.	3
Часов по учебному плану	108
в том числе	
аудиторные занятия (контактная работа):	
- занятия лекционного типа	16
- занятия семинарского типа (практические занятия / лабораторные работы)	16
- КСР	1
самостоятельная работа	75
Промежуточная аттестация	0
	Зачёт

3.2. Содержание дисциплины

(структурированное по темам (разделам) с указанием отведенного на них количества академических часов и виды учебных занятий)

Наименование разделов и тем дисциплины	Всего (часы)	в том числе			
		Контактная работа (работа во взаимодействии с преподавателем), часы из них			Самостоятельная работа обучающегося, часы
		Занятия лекционного типа	Занятия семинарского типа (практические занятия/ лабора- торные работы), часы	Всего	
	0 Ф 0	0 Ф 0	0 Ф 0	0 Ф 0	0 Ф 0
Введение. GitHub - технология. Открытые источники комментированного кода в задачах компьютерного зрения и глубокого обучения	6	2	2	4	2
Современные стили программирования. Роль стиля в качестве кода. Несовместимость версий. Книга Roberts C. Martin «Clean Code», 2008	13	2	2	4	9
Практики написания и поддержания чистого кода. Ключевые понятия качества кода, их влияние. Примеры влияния нарушения single-responsibility principle (SRP) на чистоту кода. Недостатки использования комментариев	11	2	3	5	6
Разработка через тестирование (Test-Driven Development). Что такое TDD, его назначение. Цикл разработки по TDD. Атрибуты хорошего unit-теста	19	2	3	5	14
Виртуальные машины и контейнеры. Концепция докер-контейнеров: виртуализация операционных систем, группы контейнеров, выделенные корни, виртуальные сетевые адаптеры.	27	4	3	7	20
Докер-соглашения: dockerfile; docker build docker run; volumes; docker registry: hub.docker.com, registry.gitlab.com, etc. Докер-практики: Distribution package; CI/CD environment; “Clean Room”	31	4	3	7	24
Аттестация	0				
КСР	1			1	
Итого	108	16	16	33	75

Содержание разделов и тем дисциплины

- 1) Введение. GitHub - технология. Открытые источники комментированного кода в задачах компьютерного зрения и глубокого обучения.
- 2) Современные стили программирования. Роль стиля в качестве кода. Несовместимость версий. Книга Roberts C. Martin «Clean Code», 2008.
- 3) Практики написания и поддержания чистого кода. Ключевые понятия качества кода, их влияние. Примеры влияния нарушения single-responsibility principle (SRP) на чистоту кода. Недостатки использования комментариев.
- 4) Разработка через тестирование (Test-Driven Development). Что такое TDD, его назначение. Цикл разработки по TDD. Атрибуты хорошего unit-теста.
- 5) Виртуальные машины и контейнеры. Концепция докер-контейнеров: виртуализация операционных систем, группы контейнеров, выделенные корни, виртуальные сетевые адаптеры.
- 6) Докер-соглашения: dockerfile; docker build docker run; volumes; docker registry: hub.docker.com, registry.gitlab.com, etc. Докер-практики: Distribution package; CI/CD environment; “Clean Room”.

4. Учебно-методическое обеспечение самостоятельной работы обучающихся

Самостоятельная работа обучающихся включает в себя подготовку к контрольным вопросам и заданиям для текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведенным в п. 5.

1. Home page of the course <https://github.com/UNN-VMK-Software/agile-course-theory>.
2. Martin R. OOP principles and patterns.
http://www.objectmentor.com/resources/articles/Principles_and_Patterns.pdf.

5. Фонд оценочных средств для текущего контроля успеваемости и промежуточной аттестации по дисциплине (модулю)

5.1 Типовые задания, необходимые для оценки результатов обучения при проведении текущего контроля успеваемости с указанием критериев их оценивания:

5.1.1 Типовые задания (оценочное средство - Собеседование) для оценки сформированности компетенции ПК-8:

Вопрос для оценки компетенции ПК-8/ Questions for assessing the competence PK-8

1. GitHub – технология.
2. Открытые источники комментированного кода в задачах компьютерного зрения и глубокого обучения.
3. Современные стили программирования.
4. Роль стиля в качестве кода. Книга Roberts C. Martin «Clean Code», 2008.
5. Несовместимость версий.
6. Практики написания и поддержания чистого кода. Ключевые понятия качества кода, их влияние.

Критерии оценивания (оценочное средство - Собеседование)

Оценка	Критерии оценивания
зачтено	Студент дал развернутый ответ на все вопросы без существенных ошибок.
не зачтено	При ответе студент допускает грубые ошибки в основном материале.

5.1.2 Типовые задания (оценочное средство - Практическая задача) для оценки сформированности компетенции ПК-8:

Практическое задание.

Практическое задание для каждого студента — индивидуальное задание по обеспечению переносимости DL-системы средней сложности с использованием Docker и других изученных технологий, включая автотестирование. В качестве исходной системы глубокого обучения может быть выбрано собственное DL-приложение или приложение, выбранное с сайта приложений с открытым исходным кодом (<https://paperswithcode.com/>). Выбор согласовывается с преподавателем. Проект представляется автором всем участникам практического занятия в режиме on-line. В течение представления, инструкция по запуску проекта, и проверке корректности результата, тестируется студентом, назначенным преподавателем.

Критерии оценивания (оценочное средство - Практическая задача)

Оценка	Критерии оценивания
зачтено	Выполнены все или большая часть этапов решения задачи или задача решена с незначительными недочетами. Результаты работы представлены преподавателю в срок.
не зачтено	Выполнены не все практические задания или выполнены не в полном объеме (представлено не полное описание этапов выполнения заданий, получен неверный ответ, результаты работы не представлены преподавателю).

5.2. Описание шкал оценивания результатов обучения по дисциплине при промежуточной аттестации

Шкала оценивания сформированности компетенций

Уровень сформированности компетенций (индикатора достижения компетенций)	плохо	неудовлетворительно	удовлетворительно	хорошо	очень хорошо	отлично	превосходно
	не зачтено		зачтено				
<u>Знания</u>	Отсутствие знаний теоретического материала.	Уровень знаний ниже минимальных требований.	Минимально допустимый уровень	Уровень знаний в объеме, соответствующий	Уровень знаний в объеме, соответствующий	Уровень знаний в объеме, соответствующий	Уровень знаний в объеме, превышающий

	Невозможность оценить полноту знаний вследствие отказа обучающегося от ответа	Имели место грубые ошибки	знаний. Допущено много негрубых ошибок	ющем программе подготовки . Допущено несколько негрубых ошибок	ющем программе подготовки . Допущено несколько несущественных ошибок	ующем программе подготовк и. Ошибок нет.	м программу подготовки.
<u>Умения</u>	Отсутствие минимальных умений. Невозможность оценить наличие умений вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы основные умения. Имели место грубые ошибки	Продемонстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном объеме	Продемонстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания в полном объеме, но некоторые с недочетами	Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания в полном объеме, но некоторые с недочетами	Продемонстрированы все основные умения. Решены все основные задачи с отдельным и несущественными недочетами, выполнены все задания в полном объеме	Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме без недочетов
<u>Навыки</u>	Отсутствие базовых навыков. Невозможность оценить наличие навыков вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы базовые навыки. Имели место грубые ошибки	Имеется минимальный набор навыков для решения стандартных задач с некоторым и недочетами	Продемонстрированы базовые навыки при решении стандартных задач с некоторым и недочетами	Продемонстрированы базовые навыки при решении стандартных задач без ошибок и недочетов	Продемонстрированы навыки при решении нестандартных задач без ошибок и недочетов	Продемонстрирован творческий подход к решению нестандартных задач

Шкала оценивания при промежуточной аттестации

Оценка		Уровень подготовки
зачтено	превосходно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «превосходно», продемонстрированы знания, умения, владения по соответствующим компетенциям на уровне выше предусмотренного программой
	отлично	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично».
	очень хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «очень хорошо»
	хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «хорошо».
	удовлетворительно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «удовлетворительно», при этом хотя бы одна компетенция сформирована на уровне «удовлетворительно»

не зачтено	неудовлетворительно	Хотя бы одна компетенция сформирована на уровне «неудовлетворительно».
	плохо	Хотя бы одна компетенция сформирована на уровне «плохо»

5.3 Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения на промежуточной аттестации с указанием критериев их оценивания:

5.3.1 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ПК-8

1. GitHub – технология.
2. Открытые источники комментированного кода в задачах компьютерного зрения и глубокого обучения.
3. Современные стили программирования.
4. Роль стиля в качестве кода. Книга Roberts C. Martin «Clean Code», 2008.
5. Несовместимость версий.
6. Практики написания и поддержания чистого кода. Ключевые понятия качества кода, их влияние.
7. Примеры влияния нарушения single-responsibility principle (SRP) на чистоту кода. Недостатки использования комментариев.
8. Разработка через тестирование (Test-Driven Development). Что такое TDD, его назначение. Цикл разработки по TDD. Атрибуты хорошего unit-теста / Test-Driven Development.
9. Виртуальные машины и контейнеры. Концепция докер-контейнеров: виртуализация операционных систем, группы контейнеров, выделенные корни, виртуальные сетевые адаптеры.
10. Докер-соглашения: dockerfile; docker build docker run; volumes; docker registry: hub.docker.com, registry.gitlab.com, etc.
11. Докер-практики: Distribution package; CI/CD environment; “Clean Room”.

Критерии оценивания (оценочное средство - Контрольные вопросы)

Оценка	Критерии оценивания
зачтено	Студент ответил на большую часть вопросов возможно с незначительными недочетами.
не зачтено	При ответе студент допускает грубые ошибки в основном материале и решении стандартных задач.

6. Учебно-методическое и информационное обеспечение дисциплины (модуля)

Основная литература:

1. Deep Learning Applications and Intelligent Decision Making in Engineering. - IGI Global, 2021. - 1 online resource. - ISBN 9781799821106. - ISBN 9781799821083. - Текст : электронный., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=858228&idb=0>.

Дополнительная литература:

1. Alex Magana. Version Control with Git and GitHub : Discover the Most Popular Source Control Solutions Used by Developers Worldwide. - Packt Publishing, 2018. - 1 online resource. - ISBN 9781789800739. - ISBN 9781789808971. - Текст : электронный., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=854526&idb=0>.

Программное обеспечение и Интернет-ресурсы (в соответствии с содержанием дисциплины):

- 1) Introduction to Github. © 2025 GitHub <https://github.com/skills/introduction-to-github>
- 2) Github Collection: Learn to Code. Master programming by recreating your favorite technologies from scratch. <https://github.com/collections/learn-to-code>
- 3) Github Practical tutorials. Project based learning. <https://github.com/practical-tutorials/project-based-learning>
- 4) Google Style Guides. <https://google.github.io/styleguide/>
- 5) The Clean Coder: A Code of Conduct for Professional Programmers (Robert C. Martin Series). 1st Edition by Robert Martin. 2011. <https://archive.org/details/cleancodercodeof0000mart>
- 6) James W. Grenning. Test Driven Development for Embedded C (Pragmatic Programmers). 1st Ed. <https://books-library.net/files/books-library.net-07281709Ee1R6.pdf>
- 7) Davide Fucci, et al. A Dissection of the Test-Driven Development Process: Does It Really Matter to Test-First or to Test-Last? 2018. 19p. https://oa.upm.es/50842/1/INVE_MEM_2018_276502.pdf
- 8) Containers vs Virtual Machines: A Detailed Comparison for Developers? <https://cloud.google.com/discover/containers-vs-vms>
- 9) Aashish Nair. Containers vs Virtual Machines: A Detailed Comparison for Developers. <https://www.datacamp.com/blog/containers-vs-virtual-machines>
- 10) Docker. Building best practices. <https://docs.docker.com/build/building/best-practices/>
- 11) Docker. Content & first 16 pages / Tutorialspoint. 145 p. https://www.tutorialspoint.com/docker/docker_tutorial.pdf
- 12) Prakhar Srivastav. Docker for beginners. <https://docker-curriculum.com/>
- 13) James Turnbull. The Docker Book. 2016. <https://github.com/AngelSanchezT/books-1/blob/master/docker/the-docker-book.pdf>

7. Материально-техническое обеспечение дисциплины (модуля)

Учебные аудитории для проведения учебных занятий, предусмотренных образовательной программой, оснащены мультимедийным оборудованием (проектор, экран), техническими средствами обучения, компьютерами.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечены доступом в электронную информационно-образовательную среду.

Программа составлена в соответствии с требованиями ОС ННГУ по направлению подготовки/специальности 02.04.02 - Фундаментальная информатика и информационные технологии.

Автор(ы): Турлапов Вадим Евгеньевич, доктор технических наук, доцент.

Заведующий кафедрой: Мееров Иосиф Борисович, кандидат технических наук.

Программа одобрена на заседании методической комиссии от 02.12.2024, протокол № 5.