

MINISTRY OF SCIENCE AND HIGHER EDUCATION OF THE RUSSIAN FEDERATION

**Federal State Autonomous Educational Institution of Higher Education  
«National Research Lobachevsky State University of Nizhny Novgorod»**

Институт информационных технологий, математики и механики

---

УТВЕРЖДЕНО

решением Ученого совета ННГУ

протокол № 10 от 02.12.2024 г.

**Working programme of the discipline**

Development of deep learning systems

---

Higher education level

Master degree

---

Area of study / speciality

02.04.02 - Fundamental Informatics and Information Technology

---

Focus /specialization of the study programme

Artificial Intelligence and Data Analysis

---

Mode of study

full-time

---

Nizhny Novgorod

Year of commencement of studies 2025

## 1. Место дисциплины в структуре ОПОП

Дисциплина Б1.В.ДВ.01.02 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<br>(код, содержание компетенции)   | Планируемые результаты обучения по дисциплине (модулю), в соответствии с индикатором достижения компетенции   |  | Наименование оценочного средства     |                               |
|--|---|--|--------------------------------------|-------------------------------|
|  | Индикатор достижения компетенции<br>(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 | <p>ПК-8.1: Знает методику разработки новых алгоритмических, методических и технологических решений</p> <p>ПК-8.2: Умеет применять полученные знания для разработки новых алгоритмических, методических и технологических решений</p> <p>ПК-8.3: Имеет практический опыт составления технического задания на разработку информационной системы</p> | <p>ПК-8.1:<br/>Знает понятия качества кода, системной архитектуры, переносимости программного обеспечения и docker.</p> <p>Знает современные практики программирования, направленные на поддержание высокого качества систем в условиях непрерывной доработки.</p> <p>/Knows the concepts of code quality, system architecture, software portability, and docker.</p> <p>Knows modern programming practices aimed at maintaining high-quality systems in conditions of continuous processing.</p> <p>ПК-8.2:<br/>Умеет использовать практики организации проектной работы, рефакторинга, разработки через тестирование на примере разработки программного обеспечения систем глубокого обучения.</p> <p>/Is able to use the practices of organizing project work, refactoring, test-driven development on the example of software development for deep learning systems.</p> | Собеседование<br>Практическая задача | Зачёт:<br>Контрольные вопросы |

|  |  |  |  |  |
|--|--|--|--|--|
|  |  | ПК-8.3:<br>Обладает навыками обеспечения переносимости программного обеспечения при разработке систем глубокого обучения / Has the skills to ensure software portability during the deep learning systems development. |  |  |
|--|--|--|--|--|

### 3. Структура и содержание дисциплины

#### 3.1 Трудоемкость дисциплины

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

#### 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<br>(часы) | в том числе   |  |             | Самостоятельная<br>работа<br>обучающегося,<br>часы |
|--|-----------------|---|--|-------------|--|
|  |                 | Контактная работа (работа во взаимодействии с преподавателем),<br>часы из них |  |             |  |
|  |                 | Занятия<br>лекционного<br>типа  | Занятия<br>семинарского<br>типа<br>(практические<br>занятия/ лабора<br>торные<br>работы), часы | Всего       |  |
|  | 0<br>Ф<br>0     | 0<br>Ф<br>0   | 0<br>Ф<br>0  | 0<br>Ф<br>0 |  |
| Введение. GitHub - технология. Открытые источники комментированного кода в задачах компьютерного зрения и глубокого обучения / Introduction. GitHub - technology. Open sources of commented code in computer vision and deep learning. | 6               | 2   | 2  | 4           | 2  |

|  |     |    |    |    |    |
|--|-----|----|----|----|----|
| Современные стили программирования. Роль стиля в качестве кода. Несовместимость версий. Книга Roberts C. Martin «Clean Code», 2008 / Modern programming styles. The role of style in code quality. Incompatibility of versions. The book by 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 / Practices of writing and maintaining clean code. Key concepts of code quality, their impact. Examples of the impact of violating the single-responsibility principle (SRP) on code cleanliness. Disadvantages of using comments. | 11  | 2  | 3  | 5  | 6  |
| Разработка через тестирование (Test-Driven Development). Что такое TDD, его назначение. Цикл разработки по TDD. Атрибуты хорошего unit-теста / Test-Driven Development. What is TDD, its purpose. TDD development cycle. Attributes of a good unit test.   | 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" / Docker conventions: dockerfile; docker build docker run; volumes; docker registry: hub.docker.com, registry.gitlab.com, etc. Docker practices: Distribution package; CI/CD environment; "Clean Room".   | 31  | 4  | 3  | 7  | 24 |
| Аттестация   | 0   |    |    |    |    |
| КСР  | 1   |    |    | 1  |    |
| Итого  | 108 | 16 | 16 | 33 | 75 |

### Contents of sections and topics of the discipline

- 1) Введение. GitHub - технология. Открытые источники комментированного кода в задачах компьютерного зрения и глубокого обучения / Introduction. GitHub - technology. Open sources of commented code in computer vision and deep learning.
- 2) Современные стили программирования. Роль стиля в качестве кода. Несовместимость версий. Книга Roberts C. Martin «Clean Code», 2008 / Modern programming styles. The role of style in code quality. Incompatibility of versions. The book by 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 / Practices of writing and maintaining clean code. Key concepts of code quality, their impact. Examples of the impact of violating the single-responsibility principle (SRP) on code cleanliness. Disadvantages of using comments.
- 4) Разработка через тестирование (Test-Driven Development). Что такое TDD, его назначение. Цикл разработки по TDD. Атрибуты хорошего unit-теста / Test-Driven Development. What is TDD, its purpose. TDD development cycle. Attributes of a good unit test.
- 5) Виртуальные машины и контейнеры. Концепция докер-контейнеров: виртуализация операционных систем, группы контейнеров, выделенные корни, виртуальные сетевые адаптеры / Virtual machines and containers. The concept of docker containers: virtualization of operating systems, container groups, dedicated roots, virtual network adapters.
- 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" / Docker conventions: dockerfile; docker build docker run; volumes; docker registry: hub.docker.com, registry.gitlab.com, etc. Docker practices: 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](http://www.objectmentor.com/resources/articles/Principles_and_Patterns.pdf) .

#### 5. Assessment tools for ongoing monitoring of learning progress and interim certification in the discipline (module)

##### 5.1 Model assignments required for assessment of learning outcomes during the ongoing monitoring of learning progress with the criteria for their assessment:

##### 5.1.1 Model assignments (assessment tool - Interview) to assess the development of the competency ПК-8:

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

1. GitHub – технология / GitHub - technology.
2. Открытые источники комментированного кода в задачах компьютерного зрения и глубокого обучения / Open sources of commented code in computer vision and deep learning.
3. Современные стили программирования / Modern programming styles.
4. Роль стиля в качестве кода. Книга Roberts C. Martin «Clean Code», 2008 / The role of style in code quality. The book by Roberts C. Martin "Clean Code", 2008.
5. Несовместимость версий / Incompatibility of versions.
6. Практики написания и поддержания чистого кода. Ключевые понятия качества кода, их влияние / Practices of writing and maintaining clean code. Key concepts of code quality, their impact.

##### Assessment criteria (assessment tool — Interview)

| Grade | Assessment criteria  |
|-------|--|
| pass  | Студент дал развернутый ответ на все вопросы без существенных ошибок / The student gave a detailed answer to all the questions without significant errors. |
| fail  | При ответе студент допускает грубые ошибки в основном материале / When answering, the student makes gross mistakes in the main material                    |

##### 5.1.2 Model assignments (assessment tool - Practical task) to assess the development of the competency ПК-8:

Практическое задание / Practical task.

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./

The practical task for each student is an individual task to ensure portability of a medium complexity DL-system using the Docker and others studied technologies including autotesting. As the initial deep learning system, one can be chosen one's own DL-application or an application selected from the site of open source applications (<https://paperswithcode.com/>). The choice is agreed upon with the teacher. The project is presented by the author during the practical lesson and tested by the student assigned by the teacher.

### Assessment criteria (assessment tool — Practical task)

| Grade | Assessment criteria  |
|-------|--|
| pass  | Выполнены все или большая часть этапов решения задачи или задача решена с незначительными недочетами. Результаты работы представлены преподавателю всрок. / All or most of the stages of solving the problem have been completed, or the problem has been solved with minor flaws. The results of the work have been presented to the teacher on time.   |
| fail  | 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). / Not all practical tasks were completed or were not completed in full (an incomplete description of the stages of completing the tasks was provided, an incorrect answer was received, the results of the work were not presented to the teacher). |

## 5.2. Description of scales for assessing learning outcomes in the discipline during interim certification

### 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. Невозможность оценить полноту знаний вследствие отказа | Уровень знаний ниже минимальных требований. Имели место грубые ошибки | Минимально допустимый уровень знаний. Допущено много негрубых ошибок | Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько | Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько | Уровень знаний в объеме, соответствующем программе подготовки. Ошибок нет. | Уровень знаний в объеме, превышающем программу подготовки. |

|               |  |  |  |   |   |  |   |
|---------------|--|--|--|---|---|--|---|
|               | обучающегося от ответа   |  |  | негрубых ошибок   | несущественных ошибок   |  |   |
| <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 | Продemonстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном объеме | Продemonстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания в полном объеме, но некоторые с недочетами | Продemonстрированы все основные умения. Решены все основные задачи с отдельными и несущественными недочетами, выполнены все задания в полном объеме | Продemonстрированы 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 некоторыми недочетами                                      | Продemonстрированы базовые навыки при решении стандартных задач с некоторыми недочетами   | Продemonстрированы базовые навыки при решении стандартных задач без ошибок и недочетов  | Продemonстрированы навыки при решении нестандартных задач без ошибок и недочетов   | Продemonстрирован творческий подход к решению нестандартных задач |

### Scale of assessment for interim certification

| Grade |                | Assessment criteria   |
|-------|----------------|---|
| pass  | outstanding    | All the competencies (parts of competencies) to be developed within the discipline have been developed at a level no lower than "outstanding", the knowledge and skills for the relevant competencies have been demonstrated at a level higher than the one set out in the programme. |
|       | excellent      | All the competencies (parts of competencies) to be developed within the discipline have been developed at a level no lower than "excellent",  |
|       | very good      | All the competencies (parts of competencies) to be developed within the discipline have been developed at a level no lower than "very good",  |
|       | good           | All the competencies (parts of competencies) to be developed within the discipline have been developed at a level no lower than "good",   |
|       | satisfactory   | All the competencies (parts of competencies) to be developed within the discipline have been developed at a level no lower than "satisfactory", with at least one competency developed at the "satisfactory" level.   |
| fail  | unsatisfactory | At least one competency has been developed at the "unsatisfactory" level.   |
|       | poor           | At least one competency has been developed at the "poor" level.   |

### 5.3 Model control assignments or other materials required to assess learning outcomes during the interim certification with the criteria for their assessment:

#### 5.3.1 Model assignments (assessment tool - Control questions) to assess the development of the competency ПК-8

1. GitHub – технология / GitHub - technology.
2. Открытые источники комментированного кода в задачах компьютерного зрения и глубокого обучения / Open sources of commented code in computer vision and deep learning.
3. Современные стили программирования / Modern programming styles.
4. Роль стиля в качестве кода. Книга Roberts C. Martin «Clean Code», 2008 / The role of style in code quality. The book by Roberts C. Martin "Clean Code", 2008.
5. Несовместимость версий / Incompatibility of versions.
6. Практики написания и поддержания чистого кода. Ключевые понятия качества кода, их влияние / Practices of writing and maintaining clean code. Key concepts of code quality, their impact.
7. Примеры влияния нарушения single-responsibility principle (SRP) на чистоту кода. Недостатки использования комментариев / Examples of the impact of violating the single-responsibility principle (SRP) on code cleanliness. Disadvantages of using comments.
8. Разработка через тестирование (Test-Driven Development). Что такое TDD, его назначение. Цикл разработки по TDD. Атрибуты хорошего unit-теста / Test-Driven Development. What is TDD, its purpose. TDD development cycle. Attributes of a good unit test.
9. Виртуальные машины и контейнеры. Концепция докер-контейнеров: виртуализация операционных систем, группы контейнеров, выделенные корни, виртуальные сетевые адаптеры / Virtual machines and containers. The concept of docker containers: virtualization of operating systems, container groups, dedicated roots, virtual network adapters.
10. Докер-соглашения: dockerfile; docker build docker run; volumes; docker registry: hub.docker.com, registry.gitlab.com, etc. / Docker conventions: 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” / Docker practices: Distribution package; CI/CD environment; “Clean Room”.

#### Assessment criteria (assessment tool — Control questions)

| Grade | Assessment criteria   |
|-------|---|
| pass  | Студент ответил на большую часть вопросов возможно с незначительными недочетами / The student answered most of the questions, possibly with minor flaws   |
| fail  | При ответе студент допускает грубые ошибки в основном материале и решении стандартных задач / When answering, the student makes gross mistakes in the basic material and solving standard tasks |

### 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](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](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 - Fundamental Informatics and Information Technology.

Авторы: 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.