

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 президиума Ученого совета ННГУ

протокол № 1 от 16.01.2024 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 2024

## 1. Место дисциплины в структуре ОПОП

Дисциплина Б1.В.ДВ.01.03 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	<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: Знает понятия, используемые при разговоре о качестве кода и архитектуры системы (объектной иерархии). Умеет использовать практики ГМР по организации проектной работы, проектирования и разработки ПО на практике. Владеть стратегиями управления, стратегиями кодирования и внедрения ХР</p> <p>ПК-8.2: Знает проблемы графического интерфейса пользователя и слоя доступа к данным с точки зрения сложности его поддержки и развития. Умеет применять методологию «Экстремальное программирование» Владеет практиками экстремального программирования ХР, парного программирования.</p> <p>ПК-8.3: Знает современные практики программирования, направленные на поддержание высокого качества системы в условиях непрерывной переработки. Умеет использовать</p>	<p>Практическая задача Собеседование</p>	<p>Зачёт: Контрольные вопросы</p>

		практики рефакторинга, разработки через тестирование и непрерывной интеграции. Владеет навыками рефакторинга, разработки через тестирование и использования паттернов MVP и DDD.		
--	--	---	--	--

### 3. Структура и содержание дисциплины

#### 3.1 Трудоемкость дисциплины

	очная
<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> <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 (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/лабораторные 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.	5	1	2	3	2
Современные стили программирования. Роль стиля в качестве кода. Несовместимость версий. Книга Roberts C. Martin «Clean Code», 2008.	12	1	2	3	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.	5	1	2	3	2
Разработка через тестирование (Test-Driven Development). Что такое TDD, его назначение. Цикл разработки по TDD. Атрибуты хорошего unit-теста.	10	1	2	3	7
Виртуальные машины и контейнеры. Концепция докер-контейнеров: виртуализация операционных систем, группы контейнеров, выделенные корни, виртуальные сетевые адаптеры.	11	2	2	4	7
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".	10	2	1	3	7
Принципы ООП. SOLID: SRP, OCP, DIP (его реализация, связь с OCP), LSP, ISP, DIP в DDD. Закон Деметера. Стандарты C++ программирования.	10	2	1	3	7
Паттерн Domain-Driven Design. Диаграмма DDD. Назначение сборки Application. Способ стыковки Domain и Infrastructure. DIP в DDD.	18	2	1	3	15
Дизайн слоя доступа к данным. Типичные проблемы архитектуры при отсутствии слоя Data Access. Способы организации Data Access. Преимущества Data Access Layer.	18	2	1	3	15
Дизайн презентационного слоя (паттерн Model-View-Presenter). Типичные проблемы с UI. Диаграмма Passive View. Назначение и преимущества применения Passive View.	4	1	1	2	2
Методология «Экстремальное программирование» (XP). Ценности и практики XP. Стратегия управления и кодирования. Парное программирование. Методология Scrum и схема спринта. Методология Kanban. Рефакторинг.	4	1	1	2	2
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.
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".
7. Принципы ООП. SOLID: SRP, OCP, DIP (его реализация, связь с OCP), LSP, ISP, DIP в DDD. Закон Деметера. Стандарты C++ программирования.
8. Паттерн Domain-Driven Design. Диаграмма DDD. Назначение сборки Application. Способ стыковки Domain и Infrastructure. DIP в DDD.
9. Дизайн слоя доступа к данным. Типичные проблемы архитектуры при отсутствии слоя Data Access.

Способы организации Data Access. Преимущества Data Access Layer.

10. Дизайн презентационного слоя (паттерн Model-View-Presenter). Типичные проблемы с UI.

Диаграмма Passive View. Назначение и преимущества применения Passive View.

11. Методология «Экстремальное программирование» (XP). Ценности и практики XP. Стратегия управления и кодирования. Парное программирование. Методология Scrum и схема спринта. Методология Kanban. Рефакторинг.

#### **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. Домашняя страница учебного курса <https://github.com/UNN-VMK-Software/agile-course-theory>.

2. Мартин Р. Принципы и паттерны ООП

[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 - Practical task) to assess the development of the competency ПК-8:**

Защита проекта по разработке системы глубокого обучения средней сложности с использованием изученных технологий.

Альтернативно: реализовать приложения, выполняющие следующие простые прикладные задачи:

#### **СТРУКТУРЫ ДАННЫХ**

1. Реализация структуры данных «Стек».
2. Реализация структуры данных «Дек».
3. Реализация структуры данных «Очередь».
4. Реализация структуры данных «Дерево» (построение, поиск по значению).
5. Реализация структуры данных «Левосторонняя куча».

#### **ЭКОНОМИКА**

1. Supply/Demand.
2. Ипотечный калькулятор.
3. Определение эластичности спроса.
4. Расчет месячной зарплаты (оклад, отработанные часы, сверхурочные, адм. отпуска).

#### **АЛГОРИТМЫ**

1. Алгоритм Хаффмана.

2. Парсер строк + калькулятор выражений.
3. Сортировка (не "пузырьковая").
4. Поиск дихотомией.
5. Решение квадратного уравнения.

## ДОПОЛНИТЕЛЬНЫЕ ТЕМЫ

1. Подсчет статистических величин (мат. ожидание, дисперсия, моменты разных порядков).
2. Поиск в тексте регулярных выражений (упрощенные).
3. Вычисление расстояний между float векторами (метрики L<sub>inf</sub>, L<sub>1</sub>, L<sub>2</sub>, L<sub>3</sub>, L<sub>4</sub>).

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

Grade	Assessment criteria
pass	Выполнены все или большая часть этапов решения задачи или задача решена с незначительными недочетами. Код и результаты работы представлены преподавателю в срок.
fail	Выполнены не все лабораторные работы или выполнены не в полном объеме (представлено не полное описание этапов выполнения заданий, код работает некорректно, результаты работы не представлены преподавателю).

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

1. GitHub - технология. Открытые источники комментированного кода в задачах компьютерного зрения и глубокого обучения.
2. Современные стили программирования. Практики написания и поддержания чистого кода.
3. Разработка через тестирование (Test-Driven Development).
4. Виртуальные машины и контейнеры. Концепция докер-контейнеров.
5. 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".

### Assessment criteria (assessment tool — Interview)

Grade	Assessment criteria
pass	Студент дал развернутый ответ на все вопросы без существенных ошибок.
fail	При ответе студент допускает грубые ошибки в основном материале.

## 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 обучающегося от ответа	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 негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько несущественных ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Ошибок нет.	Уровень знаний в объеме, превышающем 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	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. 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.

	<b>excellent</b>	All the competencies (parts of competencies) to be developed within the discipline have been developed at a level no lower than "excellent",
	<b>very good</b>	All the competencies (parts of competencies) to be developed within the discipline have been developed at a level no lower than "very good",
	<b>good</b>	All the competencies (parts of competencies) to be developed within the discipline have been developed at a level no lower than "good",
	<b>satisfactory</b>	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.
<b>fail</b>	<b>unsatisfactory</b>	At least one competency has been developed at the "unsatisfactory" level.
	<b>poor</b>	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. Введение. GitHub - технология. Открытые источники комментирован-ного 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".
7. Принципы ООП. SOLID: SRP, OCP, DIP (его реализация, связь с OCP), LSP, ISP, DIP в DDD. Закон Деметера. Стандарты C++ программирования.
8. Паттерн Domain-Driven Design. Диаграмма DDD. Назначение сборки Application. Способ стыковки Domain и Infrastructure. DIP в DDD.
9. Дизайн слоя доступа к данным. Типичные проблемы архитектуры при отсутствии слоя Data Access. Способы организации Data Access. Преимущества Data Access Layer.
10. Дизайн презентационного слоя (паттерн Model-View-Presenter). Типичные проблемы с UI. Диаграмма Passive View. Назначение и преимущества применения Passive View.
11. Методология «Экстремальное программирование» (XP). Ценности и практики XP. Стратегия управления и кодирования. Парное программирование. Методология Scrum и схема спринта. Методология Kanban. Рефакторинг.

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



Grade	Assessment criteria
pass	Студент ответил на большую часть вопросов возможно с незначительными недочетами.
fail	При ответе студент допускает грубые ошибки в основном материале и решении стандартных задач.

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

Основная литература:

1. Гибкая методология разработки программного обеспечения. - Москва : ИНТУИТ, 2016., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=662976&idb=0>.
2. Технологии командной разработки программного обеспечения информационных систем / Долженко А.И. - Москва : ИНТУИТ, 2016., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=663275&idb=0>.

Дополнительная литература:

1. Основы менеджмента программных проектов / Скопин И.Н. - Москва : ИНТУИТ, 2016., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=663174&idb=0>.
2. Объектно-ориентированное программирование и программная инженерия / Мейер Б. - Москва : ИНТУИТ, 2016., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=663014&idb=0>.
3. Введение в программную инженерию / Кознов Д.В. - Москва : ИНТУИТ, 2016., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=662933&idb=0>.

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

1. Домашняя страница учебного курса <https://github.com/UNN-VMK-Software/agile-course-theory>.
2. Мартин Р. Принципы и паттерны ООП  
[http://www.objectmentor.com/resources/articles/Principles\\_and\\_Patterns.pdf](http://www.objectmentor.com/resources/articles/Principles_and_Patterns.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.

Author(s): Турлапов Вадим Евгеньевич, доктор технических наук, доцент.

Заведующий кафедрой: Мееров Иосиф Борисович, кандидат технических наук.

Программа одобрена на заседании методической комиссии от 13.12.2023, протокол № 3.