

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 президиума Ученого совета ННГУ

протокол № 1 от 16.01.2024 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

2024 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	<p>ПК-8.1: Знать основы ИТ в области КС и иметь навыки анализа современного состояния науки и информационных технологий в области КС</p> <p>ПК-8.2: Иметь навыки проектирования и разработки и развития ИТ-решений на основе анализа современного состояния науки и информационных технологий в области КС</p> <p>ПК-8.3: Иметь навыки управления разработкой и развитием ИТ-решений на основе анализа современного состояния науки и информационных технологий в области КС</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</p>	Задачи	Зачёт: Контрольные вопросы

		<p>высокого качества системы в условиях непрерывной переработки.</p> <p>Умеет использовать практики рефакторинга, разработки через тестирование и непрерывной интеграции.</p> <p>Владеет навыками рефакторинга, разработки через тестирование и использования паттернов MVP и DDD.</p>		
--	--	--	--	--

### 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> <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	
	о ф о	о ф о	о ф о	о ф о	о ф о
GitHub - технология.	23	4	4	8	15
Современные стили программирования.	28	4	4	8	20

Разработка через тестирование	28	4	4	8	20
Стандарты C++ программирования	28	4	4	8	20
Аттестация	0				
КСР	1			1	
Итого	108	16	16	33	75

### Содержание разделов и тем дисциплины

Введение. GitHub - технология. Открытые источники комментированного кода в задачах компьютерного зрения и глубокого обучения. Современные стили программирования. Роль стиля в качестве кода. Несовместимость версий. Книга Roberts C. Martin «Clean Code», 2008. Практики написания и поддержания чистого кода. Ключевые понятия качества кода, их влияние. Примеры влияния нарушения singleresponsibility principle (SRP) на чистоту кода. Недостатки использования комментариев. Разработка через тестирование (Test-Driven Development). Что такое TDD, его назначение. Цикл разработки по TDD. Атрибуты хорошего unit-теста. Виртуальные машины и контейнеры. Концепция докер-контейнеров:

виртуализация операционных систем, группы контейнеров, выделенные корни, виртуальные сетевые адаптеры. Докер-соглашения: dockerfile; docker build docker run; volumes; docker registry: hub.docker.com, registry.gitlab.com, etc. Докер-практики: Distribution package; CI/CDenvironment; “Clean Room”. Принципы ООП. SOLID: SRP, OCP, DIP (его реализация, связь с OCP), LSP, ISP, DIP в DDD. Закон Деметера. Стандарты C++ программирования. Паттерн Domain-Driven Design. Диаграмма DDD. Назначение сборки Application. Способ стыковки Domain и Infrastructure. DIP в DDD. Дизайн слоя доступа к данным. Типичные проблемы архитектуры при отсутствии слоя Data Access. Способы организации Data Access. Преимущества Data Access Layer. Дизайн презентационного слоя (паттерн Model-View-Presenter). Типичные проблемы с UI. Диаграмма Passive View. Назначение и преимущества применения Passive View. 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.

Самостоятельная работа обучающихся осуществляется в виде работы с рекомендованной обязательной и дополнительной литературой:

1. Гибкая методология разработки программного обеспечения. Курс на Интуит: <http://www.intuit.ru/studies/courses/583/439/info>.
2. Технологии командной разработки программного обеспечения информационных систем. Курс на Интуит: <http://www.intuit.ru/studies/courses/4806/1054/info>.
3. Анализ и оценка методов разработки программного обеспечения (Agile). Курс на Интуит: <http://www.intuit.ru/studies/courses/3505/747/info>.

Дополнительная литература:

1. Скопин И. Основы менеджмента программных проектов. <http://www.intuit.ru/studies/courses/38/38/info>.
2. Meyer B. Объектно-ориентированное программирование и программная инженерия. Курс

на Интуит: <http://www.intuit.ru/studies/courses/2309/609/info>.

3. Кознов Д. Введение в программную инженерию. Курс на Интуит:

<http://www.intuit.ru/studies/courses/497/353/info>

## **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 - Задачи) для оценки сформированности компетенции ПК-8:**

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**

6. Supply/Demand.
7. Ипотечный калькулятор.
8. Определение эластичности спроса.
9. Расчет месячной зарплаты (оклад, отработанные часы, сверхурочные, адм. отпуска).

#### **АЛГОРИТМЫ**

10. Алгоритм Хаффмана.
11. Парсер строк + калькулятор выражений.
12. Сортировка (не "пузырьковая").
13. Поиск дихотомией.
14. Решение квадратного уравнения.

## ДОПОЛНИТЕЛЬНЫЕ ТЕМЫ

15. Подсчет статистических величин (мат. ожидание, дисперсия, моменты разных порядков).
16. Поиск в тексте регулярных выражений (упрощенные).
17. Вычисление расстояний между float векторами (метрики Linf, L1, L2, L3, L4).

### Критерии оценивания (оценочное средство - Задачи)

Оценка	Критерии оценивания
зачтено	Задача решена полностью или решена основная часть задачи, или задача решена с недочетами
не зачтено	Задача не решена или сделан первый этап решения задачи

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

### Шкала оценивания сформированности компетенций

Уровень сформированности компетенций (индикатор 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	Продемонстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном	Продемонстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания в полном объеме, но	Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания в полном объеме, но некоторые с	Продемонстрированы все основные умения. Решены все основные задачи с отдельными несущественными недочетами	Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме без недочетов

			объеме	некоторые с недочетами	недочетами	и, выполнены все задания в полном объеме	
<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. 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.

### Критерии оценивания (оценочное средство - Контрольные вопросы)

Оценка	Критерии оценивания
зачтено	Владение основным и дополнительным материалом достаточное или с незначительными ошибками и погрешностями
не зачтено	владение материалом, необходимым по данному предмету, недостаточно. Работу за время семестра можно оценить как неудовлетворительную

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

Основная литература:

1. Барков И. А. Объектно-ориентированное программирование : учебник для вузов / Барков И. А. - 2-е изд., стер. - Санкт-Петербург : Лань, 2023. - 700 с. - Книга из коллекции Лань - Информатика. - ISBN 978-5-507-47113-3., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=864800&idb=0>.



Дополнительная литература:

1. Зыков С. В. Объектно-ориентированное программирование : учебник и практикум / С. В. Зыков. - 2-е изд. - Москва : Юрайт, 2023. - 151 с. - (Высшее образование). - ISBN 978-5-534-16941-6. - Текст : электронный // ЭБС "Юрайт"., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=870794&idb=0>.
2. Объектно-ориентированное программирование / Иванова Г.С., Ничушкина Т.Н. - Москва : МГТУ им. Н.Э. Баумана, 2014., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=661368&idb=0>.

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

Домашняя страница учебного курса <https://github.com/UNN-VMK-Software/agilecourse-theory>.

## **7. Материально-техническое обеспечение дисциплины (модуля)**

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

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 13.12.2023, протокол № 3.