

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 № 15 от 24.12.2025 г.

Рабочая программа дисциплины

Разработка и внедрение моделей машинного обучения

Уровень высшего образования
Бакалавриат

Направление подготовки / специальность
02.03.02 - Фундаментальная информатика и информационные технологии

Направленность образовательной программы
Системное программирование

Форма обучения
очная

г. Нижний Новгород

2026 год начала подготовки

1. Место дисциплины в структуре ОПОП

Дисциплина Б1.В.ДВ.05.04 Разработка и внедрение моделей машинного 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-3: Способен создавать и исследовать новые математические модели в естественных науках, промышленности и бизнесе, с учетом возможностей современных информационных технологий и программирования и компьютерной техники	<p>ПК-3.1: Знает методы анализа и исследования математических моделей в области фундаментальной информатики и информационных технологий</p> <p>ПК-3.2: Умеет определять ключевые свойства и ограничения системы</p>	<p>ПК-3.1: Знает понятия качества кода, системной архитектуры, переносимости программного обеспечения и docker.</p> <p>ПК-3.2: Умеет использовать практики организации проектной работы, рефакторинга, разработки через тестирование на примере разработки программного обеспечения систем машинного обучения.</p>	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	32
- занятия семинарского типа (практические занятия / лабораторные работы)	0
- 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/лабораторные работы), часы	Всего	
0 Ф 0	0 Ф 0	0 Ф 0	0 Ф 0	0 Ф 0	
GitHub - технология.	23	8	0	8	15
Современные стили программирования.	28	8	0	8	20
Разработка через тестирование	28	8	0	8	20
Стандарты C++ программирования	28	8	0	8	20
Аттестация	0				
КСР	1			1	
Итого	108	32	0	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 В. 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 - Практическая задача) для оценки сформированности компетенции ПК-3:

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. 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 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 отдельными 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	При решении стандартных задач не продемонстрированы базовые	Имеется минимальный набор навыков для	Продемонстрированы базовые навыки при решении	Продемонстрированы базовые навыки при решении	Продемонстрированы навыки при решении	Продемонстрирован творческий подход к решению

	навыков вследствие отказа обучающегося от ответа	навыки. Имели место грубые ошибки	решения стандартных задач с некоторыми недочетами	стандартных задач с некоторыми недочетами	стандартных задач без ошибок и недочетов	нестандартных задач без ошибок и недочетов	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-3

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

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 02.03.02 - Фундаментальная информатика и информационные технологии.

Автор(ы): Золотых Николай Юрьевич, доктор физико-математических наук, доцент.

Заведующий кафедрой: Золотых Николай Юрьевич, доктор физико-математических наук.

Программа одобрена на заседании методической комиссии от 17.12.2025, протокол № протокол №6.