

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: АРЦИУХ ДМИТРИЙ ВЛАДИМИРОВИЧ
Должность: Директор Организации
Дата подписания: 05.02.2026 14:41:42
Уникальный программный ключ:
194e9de362a3e118beb1b1af4bc7c157747a8953

**Автономная некоммерческая организация
дополнительного профессионального образования
«Академия Информационных Технологий»**

ОГРН: 1230600003457, ИНН/КПП: 0600010064/060001001

Российская Федерация, 386001, Республика Ингушетия, городской округ Магас,
город Магас, улица Н.С. Хрущева, дом 10

УТВЕРЖДНО

Приказом № 1-ОБР от «7» апреля 2025г.

**ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА
ПОВЫШЕНИЯ КВАЛИФИКАЦИИ**

***«РАЗРАБОТКА НЕЙРОСЕТЕЙ НА PYTHON
С НУЛЯ ДО ПРОЕКТА (ВТОРАЯ СТУПЕНЬ)»***

Магас

2025

1.ОБЩАЯ ХАРАКТЕРИСТИКА ДОПОЛНИТЕЛЬНОЙ ПРОФЕССИОНАЛЬНОЙ ПРОГРАММЫ

1.1. Аннотация программы

Программа повышения квалификации «Разработка нейросетей на Python с нуля до проекта (вторая ступень)» нацелена на освоение методов глубокого обучения в прикладной логике: от подготовки данных и выбора архитектуры до обучения модели, оценки по метрикам и обоснования проектных решений. Актуальность программы связана с потребностью организаций в специалистах, способных быстро спроектировать нейросетевое решение под конкретную задачу, сравнить варианты по качеству и ресурсоёмкости и оформить воспроизводимый результат. Уникальность – в акценте на воспроизводимость и качество: каждое занятие сопровождается демонстрациями и готовыми ноутбуками, задания проверяются в ЛМС, фиксируются метрики и шаги пайплайна.

Обучение проходит в смешанном формате без очных практикумов: синхронные вебинары и демонстрационные разборы дополняются асинхронными лонгридами, видеоматериалами и «лабораторными» домашними заданиями с тестовым контролем в ЛМС и самостоятельной работой. Содержание включает пять блоков: инструменты и работа с данными в Python (pandas, EDA, визуализация); основы машинного обучения и метрики качества (baseline и валидация); построение воспроизводимого ML-пайплайна и оформление кода проекта; глубокое обучение и компьютерное зрение на TensorFlow/Keras и PyTorch (архитектуры CNN/RNN, аугментации, fine-tuning); локальный запуск и MVP (оценка, профилирование, минимальный сценарий использования).

1.2. Цель реализации программы

Совершенствование профессиональных компетенций в области разработки нейросетевых моделей на Python для выполнения трудовых функций:

- Формализация и алгоритмизация поставленных задач для разработки программного кода;
- Написание программного кода с использованием языков программирования, определения и манипулирования данными в базах данных;
- Проверка и отладка программного кода.

1.3. Планируемые результаты обучения (РО)

- Программа направлена на освоение (совершенствование) следующих профессиональных компетенций:

Виды деятельности	Профессиональные компетенции	Практический опыт	Умения	Знания
Разработка и отладка программного кода	Формализация и алгоритмизация поставленных задач для разработки программного кода	Составление формализованных описаний решений поставленных задач в соответствии с требованиями технического задания или внутренних	Использовать методы и приемы формализации поставленных задач Использовать методы и приемы алгоритмизации поставленных задач Использовать	Методы и приемы формализации поставленных задач Языки формализации функциональных спецификаций Методы и приемы

		<p>документов организации</p> <p>Разработка алгоритмов решения поставленных задач в соответствии с требованиями технического задания или внутренних документов организации</p> <p>Проверка корректности алгоритмов решения поставленных задач</p> <p>Оценка и согласование сроков выполнения поставленных задач</p>	<p>программное обеспечение для графического отображения алгоритмов</p> <p>Применять алгоритмы решения типовых задач в соответствующих областях</p> <p>Осуществлять коммуникации с заинтересованным и сторонами</p>	<p>алгоритмизации поставленных задач</p> <p>Нотации и программное обеспечение для графического отображения алгоритмов</p> <p>Алгоритмы решения типичных задач, области и способы их применения</p>
<p>Разработка и отладка программного кода</p>	<p>Написание программного кода с использованием языков программирования, определения и манипулирования данными в базах данных</p>	<p>Создание программного кода в соответствии с техническим заданием (готовыми спецификациями)</p> <p>Оптимизация программного кода с использованием специализированных программных средств</p> <p>Оценка и согласование сроков выполнения поставленных задач</p>	<p>Применять выбранные языки программирования для написания программного кода</p> <p>Использовать выбранную среду программирования и средства системы управления базами данных</p> <p>Использовать возможности имеющейся технической и/или программной архитектуры для написания программного кода</p> <p>Осуществлять коммуникации с заинтересованным и сторонами</p>	<p>Синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования</p> <p>Методологии разработки компьютерного программного обеспечения</p> <p>Методологии и технологии проектирования и использования баз данных</p> <p>Технологии</p>

				программирования Особенности выбранной среды программирования и системы управления базами данных Компоненты программно-технических архитектур, существующие приложения и интерфейсы взаимодействия с ними
Разработка и отладка программного кода	Проверка и отладка программного кода	Анализ и проверка исходного программного кода Отладка программного кода на уровне программных модулей Отладка программного кода на уровне межмодульных взаимодействий и взаимодействий с окружением Оценка и согласование сроков выполнения поставленных задач	Выявлять ошибки в программном коде Применять методы и приемы отладки программного кода Интерпретировать сообщения об ошибках, предупреждения, записи технологических журналов Применять современные компиляторы, отладчики и оптимизаторы программного кода Осуществлять коммуникации с заинтересованным и сторонами	Методы и приемы отладки программного кода Типы и форматы сообщений об ошибках, предупреждений Способы использования технологических журналов, форматы и типы записей журналов Современные компиляторы, отладчики и оптимизаторы программного кода Сообщения о состоянии аппаратных средств

1.4. Программа разработана на основе:

- профессионального стандарта 06.001 «Программист», утвержден приказом Министерства труда и социальной защиты Российской Федерации от 20.07.2022 № 424н.

1.5. Компетенции (трудовые функции) в рамках имеющейся квалификации, качественное изменение которых осуществляется в результате обучения:

- А/01.3 Формализация и алгоритмизация поставленных задач для разработки программного кода;
- А/02.3 Написание программного кода с использованием языков программирования, определения и манипулирования данными в базах данных;
- А/05.3 Проверка и отладка программного кода.

1.6. Область применения программы (целевая аудитория)

Настоящая программа предназначена для начинающих программистов и разработчиков, специалистов, желающих освоить AI и глубокое обучение, аналитиков данных, интересующихся разработкой нейросетевых решений, а также студентов технических специальностей. Сферы применения компетенций: IT и программирование, Data Science и аналитика, финансы и бизнес-аналитика, образование и научные исследования.

Требования к слушателям (категории слушателей): образование — среднее профессиональное или высшее; базовые знания Python (понимание синтаксиса, переменных, циклов) и основ математики (линейная алгебра, статистика); готовность к изучению новых технологий и алгоритмов и работе с метриками качества; уверенные навыки работы с ПК и в ЛМС. Предпочтителен опыт 0–2 года в IT или смежных областях.

1.7. Форма обучения: очно-заочная с применением ЭО и ДОТ.

1.8. Объем программы: 144 академических часа.

1.9. Выдаваемый документ – удостоверение о повышении квалификации установленного образца.

2. СОДЕРЖАНИЕ ПРОГРАММЫ

2.1. Учебный план

№ п/п	Наименование разделов программы	Трудоемк ость, всего (час.)	Контактная работа (час.)						СРС (час.)	Форма аттестации
			Синхронная (час.)			Асинхронная (час.)				
			ЛК	ПР	ЛБ	ЛК	ПР	ЛБ		
1.	Инструменты и работа с данными в Python	29	4	0	4	0	0	5	16	Тест, выполнение ЛР
2.	Основы машинного обучения: задачи, методы, метрики	26	4	0	4	0	0	6	12	Тест, выполнение ЛР
3.	ML-пайплайн: от подготовки данных до рабочей модели	24	4	0	4	0	0	4	12	Тест, выполнение ЛР
4.	Глубокое обучение и компьютерное зрение: распознавание объектов	28	4	0	4	0	0	6	14	Тест, выполнение ЛР
5.	Локальные модели и запуск MVP	29	4	0	6	0	0	5	14	Тест, выполнение ЛР
Итоговая аттестация		8								Выполнение и защита ИАР
	Подготовка ИАР	6								
	Защита ИАР	2								
	Итого (час.)	144	20	0	22	0	0	26	68	

2.2. Учебно-тематический план

№ п/п	Наименование разделов, тем программы	Трудоемк ость, всего (час.)	Контактная работа (час.)						СРС (час.)	Форма контроля
			Синхронная (час.)			Асинхронная (час.)				
			ЛК	ПР	ЛБ	ЛК	ПР	ЛБ		
1	Инструменты и работа с данными в Python	29	4	0	4	0	0	5	16	
1.1.	Инструменты для анализа данных Jupyter Notebook, Google Colab	6	2						4	
1.2.	Создание DataFrame. Работа с существующим набором данных	10			2			2	6	Выполнение ЛР
1.3.	Разведочный анализ. Визуальный анализ	12	2		2			3	5	
1.4.	Текущий контроль	1							1	Тест, выполнение ЛР
2	Основы машинного обучения: задачи, методы, метрики	26	4	0	4	0	0	6	12	

2.1.	Введение в машинное обучение	12	2		2			2	6	Выполнение ЛР
2.2.	Анализ набора данных различными методами	13	2		2			4	5	
2.3.	Текущий контроль	1							1	Тест, выполнение ЛР
2.4.	Промежуточная аттестация									ЛР
3	<i>ML-пайплайн: от подготовки данных до рабочей модели</i>	24	4	0	4	0	0	4	12	
3.1.	ML модель. От данных до внедрения	12	2		2			2	6	Выполнение ЛР
3.2.	Построение модели «Прогнозирование цены автомобиля»	11	2		2			2	5	
3.3.	Текущий контроль	1							1	Тест, выполнение ЛР
4	<i>Глубокое обучение и компьютерное зрение: распознавание объектов</i>	28	4	0	4	0	0	6	14	
4.1.	Анализ данных в Python	13	2		2			2	6	Выполнение ЛР
4.2.	Построение и анализ модели распознавания объектов	16	2		2			4	7	
4.3.	Текущий контроль	1							1	Тест, выполнение ЛР
4.4.	Промежуточная аттестация									ЛР
5.	<i>Локальные модели и запуск MVP</i>	29	4	0	6	0	0	5	14	
5.1.	Обзор локальных бесплатных нейросетей для установки на ПК	13	2		3			2	6	Выполнение ЛР
5.2.	Создание проекта в области глубокого обучения	15	2		3			3	7	
5.3.	Текущий контроль	1							1	Тест, выполнение ЛР
Итоговая аттестация		8								Выполнение и защита ИАР
	Подготовка ИАР	6								
	Защита ИАР	2								
	Итого (час.)	144	20	2	22	0	0	26	68	

2.3. Календарный учебный график

Срок обучения: 9 недель.

Режим занятий: до 24 часов в неделю.

Программа реализуется исключительно с применением ЭО и ДОТ и календарный учебный график формируется по форме календарного рейтинг-плана курса (маршрута обучения), размещаемого в системе дистанционного обучения GetCourse.

2.4. Рабочая программа учебных дисциплин

Наименование компонентов программы	Содержание учебного материала и формы организации деятельности слушателей	Всего (час.)
Раздел 1. Инструменты и работа с данными в Python		29
Тема 1.1. Инструменты для анализа данных Jupyter Notebook, Google Colab	<i>Содержание синхронных лекционных занятий (2 часа):</i> 1. Обзор локальных и облачных средств обработки данных.	2
	<i>СРС (4 часа):</i> 1. Изучение теоретического материала по теме 1.1 2. Изучение интерфейса Google Colab и Jupyter Notebook	4
Тема 1.2. Создание DataFrame. Работа с существующим набором данных	<i>Контактная (синхронная) работа (2 часа):</i> <i>Лабораторная работы</i> – Создание DataFrame, его свойства	2
	<i>Контактная (асинхронная) работа (2 часа):</i> <i>Лабораторная работы</i> – Загрузка данных из файла. Методы для обработки данных	2
	<i>СРС (6 часов):</i> 1. Изучение теоретического материала по теме 1.2 2. Изучение функций библиотек pandas и numpy	6
Тема 1.3. Разведочный анализ. Визуальный анализ	<i>Содержание синхронных лекционных занятий (2 часа):</i> Понятие и назначение разведочного и визуального анализа. Его важность для машинного обучения	2
	<i>Контактная (синхронная) работа (2 часа):</i> <i>Лабораторная работы</i> - Изучаем методы обработки данных на примере датафрейма «Титаник»	2
	<i>Контактная (асинхронная) работа (3 часа):</i> <i>Лабораторная работы</i> - Работа с датафреймом «Титаник» (часть 2)	3
	<i>СРС (5 часов):</i> 1. Изучение теоретического материала по теме 1.3 2. Выполнение дополнительных заданий в датафрейме «Титаник» или в аналогичном	5
Текущий контроль	<i>Тестирование по темам модуля «Инструменты и работа с данными в Python»</i>	1
Раздел 2. Основы машинного обучения: задачи, методы, метрики		26
Тема 2.1 Введение в машинное обучение	<i>Содержание синхронных лекционных занятий (2 часа):</i> 1. Основные понятия машинного обучения (целевая функция, признаки, тестовая и обучающая выборки) 2. Виды обучения (с учителем, без учителя) 3. Понятие нейросети	2
	<i>Контактная (синхронная) работа (2 часа):</i> <i>Лабораторная работы</i> - Выявляем целевую функцию и анализируем признаки на существующем dataframe	2
	<i>Контактная (асинхронная) работа (2 часа):</i> <i>Лабораторная работы</i> - Проводим анализ датафрейма на предмет выявления несущественных признаков и	2

	<i>СРС (6 часов):</i> 1. Изучение теоретического материала по теме 2.1 2. Выполнение дополнительных заданий в датафрейме «Титаник» или в аналогичном	6
Тема 2.2 Анализ набора данных различными методами	Содержание синхронных лекционных занятий (2 часа): 1. Изучение методов регрессионного анализа и случайного леса	2
	<i>Контактная (синхронная) работа (2 часа):</i> <i>Лабораторная работы</i> – Применение регрессионного анализа на конкретном примере	2
	<i>Контактная (асинхронная) работа (4 часа):</i> <i>Лабораторная работы</i> Выполнение дополнительных заданий из лабораторной работы	4
	<i>СРС (5 часов):</i> 1. Изучение теоретического материала по теме 2.2 2. Выполнение дополнительных заданий в датафрейме «Титаник» или в аналогичном	5
Текущий контроль	<i>Тестирование по темам модуля «Основы машинного обучения: задачи, методы, метрики»</i>	1
Промежуточная аттестация	<i>Оценка выполнения домашних работ по модулям 1 и 2</i>	
Раздел 3. ML-пайплайн: от подготовки данных до рабочей модели		24
Тема 3.1 ML модель. От данных до внедрения	Содержание синхронных лекционных занятий (2 часа): 1. Понятие машинного обучения. 2. Процесс построения ML модели от анализа данных до внедрения	2
	<i>Контактная (синхронная) работа (2 часа):</i> <i>Лабораторная работы</i> - Обучение простейшей нейросети на основе готового DataFrame	2
	<i>Контактная (асинхронная) работа (2 часа):</i> <i>Лабораторная работы</i> - Поиск готовых данных пригодных для обучения нейросети	2
	<i>СРС (6 часов):</i> 1. Изучение теоретического материала по теме 3.1 2. Выполнение дополнительных заданий в датафрейме «Пыль» или в аналогичном	6
Тема 3.2 Анализ набора данных различными методами	Содержание синхронных лекционных занятий (2 часа): 1. Обзор инструментов необходимых для обучения нейросетей.	2
	<i>Контактная (синхронная) работа (2 часа):</i> <i>Лабораторная работы</i> - Обучение нейросети «прогнозирование цены автомобиля»	2
	<i>Контактная (асинхронная) работа (2 часа):</i> <i>Лабораторная работы</i> - Обучение нейросети «прогноз цены товара»	2
	<i>СРС (5 часов):</i> 1. Изучение теоретического материала по теме 3.2 2. Выполнение дополнительных заданий в датафрейме «прогноз цены» или в аналогичном	5
Текущий контроль	<i>Тестирование по темам модуля «ML-пайплайн: от подготовки данных до рабочей модели»</i>	1
Раздел 4. Глубокое обучение и компьютерное зрение: распознавание объектов		28
Тема 4.1 Анализ данных в Python	Содержание синхронных лекционных занятий (2 часа): 1. Обзор метрик позволяющих проверить качество обученности нейросети RMSE, MAE и др.	2
	<i>Контактная (синхронная) работа (2 часа):</i> <i>Лабораторная работы</i> - Построение нейросети решающей задачу классификации	2
	<i>Контактная (асинхронная) работа (2 часа):</i>	2

	<i>Лабораторная работы</i> - Оценка качества обученности нейросети	
	<i>СРС (6 часов):</i> 1. Изучение теоретического материала по теме 4.1 2. Выполнение дополнительных заданий в датафрейме «CIFAR» или в аналогичном	6
Тема 4.2 Построение и анализ модели распознавания объектов	<i>Содержание синхронных лекционных занятий (2 часа):</i> 1. Задача распознавания и область ее применения	2
	<i>Контактная (синхронная) работа (2 часа):</i> <i>Лабораторная работы</i> - Построение нейросети решающей задачу распознавания	2
	<i>Контактная (асинхронная) работа (4 часа):</i> <i>Лабораторная работы</i> - Оценка качества обученной нейросети	4
	<i>СРС (7 часов):</i> 1. Изучение теоретического материала по теме 4.2 2. Выполнение дополнительных заданий в датафрейме «распознавание предметов одежды» или в аналогичном	7
Текущий контроль	<i>Тестирование по темам модуля «Глубокое обучение и компьютерное зрение: распознавание объектов»</i>	1
Промежуточная аттестация	<i>Оценка выполнения домашних работ по модулям 3 и 4</i>	
Раздел 5. Локальные модели и запуск MVP		29
Тема 5.1 Обзор локальных бесплатных нейросетей для установки на ПК	<i>Содержание синхронных лекционных занятий (2 часа):</i> 1. Обзор возможностей локальных нейросетей и инструментов для установки на персональный компьютер.	2
	<i>Контактная (синхронная) работа (3 часа):</i> <i>Лабораторная работы</i> - Строим локальную модель	3
	<i>Контактная (асинхронная) работа (2 часа):</i> <i>Лабораторная работы</i> - Анализ качества модели	2
	<i>СРС (6 часов):</i> 1. Изучение теоретического материала по теме 5.1 2. Подбор данных для итогового проекта	6
Тема 5.2 Создание проекта в области глубокого обучения	<i>Содержание синхронных лекционных занятий (2 часа):</i> 1. Структура и требования к проекту в области машинного обучения	2
	<i>Контактная (синхронная) работа (3 часа):</i> <i>Лабораторная работы</i> - Проектируем проект	3
	<i>Контактная (асинхронная) работа (3 часа):</i> <i>Лабораторная работы</i> - Работа над итоговым проектом	3
	<i>СРС (7 часов):</i> 1. Изучение теоретического материала по теме 5.2 2. Доработка проекта	7
Текущий контроль	Тестирование по темам модуля «Локальные модели и запуск MVP»	1
Итоговая аттестация	<i>Подготовка и защита итоговой аттестационной работы</i>	8
Итого		144

2.5. Оценка качества освоения программы

Оценка качества освоения программы включает текущий контроль, промежуточную аттестацию и итоговую аттестацию слушателей.

Текущий контроль проводится в форме контрольных тестов в ЛМС с автоматической проверкой.

Промежуточная аттестация осуществляется в форме лабораторной работы с загрузкой результатов в ЛМС и рецензированием преподавателем.

Итоговая аттестация проводится в форме разработки и оценки модели машинного обучения асинхронно с оцениванием по утвержденным критериям.

Условия допуска к итоговой аттестации:

- выполнены все обязательные лабораторные задания (загружены в ЛМС, получена рецензия «зачтено»);
- пройдены все контрольные тесты;
- выполнена промежуточная аттестация;
- загружены в ЛМС материалы к защите итогового проекта в установленные сроки.

Сроки проверки и обратной связи:

- тесты проверяются автоматически;
- лабораторные задания проверяются преподавателем в ЛМС, рецензия публикуется не позднее трёх рабочих дней с момента загрузки.

2.5.1. Форма(ы) текущей, промежуточной и итоговой аттестации

Текущий контроль проводится в форме контрольных тестов в ЛМС с автоматической проверкой. Результат фиксируется как доля верных ответов (в процентах). Порог успешного выполнения — не менее 50 % правильных ответов

Критерии оценивания текущего контроля:

- каждый вопрос имеет установленный ключ; ответ оценивается как «верно»/«неверно» согласно ключу (для множественного выбора засчитывается только полностью совпадающий набор);
- итог по тесту: «выполнено» при достижении порога, «не выполнено» при недостижении.

Промежуточная аттестация проводится в форме лабораторных работ с загрузкой материалов в ЛМС и рецензированием преподавателем. Для успешного прохождения ПА слушатель выполняет по одной лабораторной работе по каждому из двух предшествующих модулей и размещает полный комплект материалов в ЛМС, согласно утверждённым критериям. Формат демонстрации не предусмотрен.

Критерии оценивания промежуточной аттестации (зачет/незачет):

- соответствие заданию модуля и полнота выполнения всех пунктов;
- корректность решений и валидность результатов (метрики/проверки, воспроизводимость);
- качество оформления: читаемость кода, структура, комментарии, наличие инструкции;
- соблюдение академической добросовестности (оригинальность, корректные заимствования, отсутствие совпадений, не объяснённых ссылками);
- соблюдение сроков сдачи в ЛМС.

Порог зачета: «зачтено» выставляется при выполнении обеих работ и удовлетворении всем критериям; при невыполнении хотя бы одной работы или несоответствии критериям — «не зачтено».

Итоговая аттестация (ИА): индивидуальная разработка и оценка модели машинного обучения (табличные данные: регрессия/классификация в scikit-learn; допускается простая модель распознавания объектов в TensorFlow/Keras) с выполнением и загрузкой пакета материалов в ЛМС. Проверка — рецензирование преподавателем в ЛМС по утверждённым критериям. Защита может проводиться синхронно (вебинар) либо асинхронно (запись видеозащиты + ответы на вопросы в ЛМС в установленное окно).

Требования к содержанию итоговой работы:

Во введении указываются: актуальность темы, цель и задачи работы, объект и предмет исследования, методы и инструменты реализации.

Основная часть включает:

1. Обзорную часть: краткий анализ выбранной области применения нейросетей и обзор аналогичных решений, характеристика архитектур и инструментов (TensorFlow, Keras, PyTorch).
2. Проектирование модели:
 - постановка задачи и выбор целевой метрики (например, Accuracy, F1, ROC-AUC, RMSE/MAE);
 - выбор и обоснование архитектуры (CNN, RNN, MLP и др.);
 - подготовка данных и реализация пайплайна (разделение train/valid/test, аугментации, нормализация);
 - описание функциональных и нефункциональных требований (надёжность, воспроизводимость, масштабируемость).
3. Реализация и тестирование:
 - обучение модели, визуализация обучения и валидации;
 - анализ ошибок, оценка метрик, интерпретация результатов;
 - рекомендации по улучшению архитектуры, гиперпараметров, качества данных.

Заключение: выводы о достигнутых результатах, оценка эффективности модели, направления дальнейшего развития (оптимизация, перенос в продакшн, локальный MVP).

Объём: отчёт 4–6 страниц и/или презентация 5–7 слайдов с выводами и рекомендациями.

Критерии оценивания ИА:

1. Корректность постановки задачи, определение цели, обоснование выбора архитектуры и метрик.
2. Качество подготовки данных и корректность пайплайна обучения/валидации.
3. Полнота реализации и демонстрация работоспособной модели.
4. Обоснованность анализа результатов и интерпретации метрик.
5. Воспроизводимость кода и структура проекта (наличие документации, зависимостей, инструкций).
6. Качество презентации и устного/видеодоклада.

Система оценивания: двухбалльная — «зачтено» / «не зачтено». Слушатель считается аттестованным при оценке «зачтено».

Примерные темы итоговых аттестационных работ:

1. Классификация изображений с использованием свёрточных нейронных сетей.
2. Прогнозирование спроса или временных рядов с помощью рекуррентных архитектур (LSTM, GRU).
3. Обнаружение объектов на изображениях с использованием TensorFlow/Keras.
4. Сравнение архитектур CNN и MLP на задаче классификации табличных данных.
5. Оптимизация нейросетевой модели для работы в ограниченной вычислительной среде (MVP).
6. Применение transfer learning для решения прикладной задачи компьютерного зрения.

Регламент защиты итоговой аттестационной работы.

Онлайн-защита (синхронно, ВКС). Разрешается, если одновременно выполняются условия:

1. Успешно сданы все ПА/текущий контроль по предшествующим модулям и ИАР допущена к защите (допуск в ЛМС).
2. Техническая готовность: стабильный интернет ≥ 10 Мбит/с, микрофон/камера исправны; демонстрация экрана доступна.

3. Идентификация личности: ФИО, документ/пропуск (или корпоративная идентификация) по запросу комиссии.
4. Наличие комиссии в утвержденный слот расписания (временное окно защиты опубликовано в ЛМС).
5. Тайминг и формат: 5–7 минут доклада + 5–10 минут ответы на вопросы; презентация и демонстрация артефактов (ноутбук/репозиторий, метрики).
6. Готовность материалов: загружены в ЛМС не позднее чем за 24 часа до слота: PDF-отчет, презентация, архив/ссылка на код, requirements.txt, инструкция запуска.
7. Академическая добросовестность: оригинальность работы, корректные ссылки на источники и датасеты.

Обязательные элементы защиты: краткая постановка задачи и метрик, архитектура/пайплайн, результаты (метрики/графики), выводы и рекомендации, ответы на вопросы.

Перенос/повтор: допускается при документально подтверждённых причинах (форс-мажор, сбой связи, болезнь) по решению комиссии.

Скринкаст-защита (асинхронно, загрузка в ЛМС). Разрешается, если выполняется одно из условий и все общие требования:

1. Территориальные/временные ограничения слушателя (сменный график/часовой пояс), препятствующие участию в синхронном слоте.
2. Технические ограничения: нестабильный интернет, отсутствие возможности включить камеру/микрофон по объективным причинам.
3. Решение комиссии/куратора для поточных групп (большое число слушателей, дефицит слотов) или как альтернативный формат, предусмотренный УП/УТП.
4. Спецусловия (ОВЗ, медпоказания) — по заявлению и подтверждающим документам.

Требования к скринкасту и пакету сдачи:

- Видео 6–8 минут, формат MP4/MKV, разрешение $\geq 720p$, звук разборчивый.
 - Структура ролика: титул (ФИО, группа/поток, тема), цель/метрики, данные и подготовка, архитектура/параметры, результаты/сравнение с baseline, выводы/рекомендации, демонстрация запуска (коротко).
 - Пакет в ЛМС: PDF-отчёт, презентация, код/ноутбук, requirements.txt, инструкция запуска; при необходимости — зафиксированный random_state и ссылка на данные.
 - Сроки: загрузка не позднее установленного дедлайна (обычно T-0 23:59 местного времени).
 - Проверка и обратная связь: рецензирование по рубрике; при запросе комиссии слушатель готов дать устные уточнения на коротком онлайн-этапе (до 10 минут).
 - Общие правила (для обоих форматов)
 - Критерии зачёта: соответствие заданию, корректность постановки/валидации, реплицируемость кода, интерпретация метрик, качество оформления, соблюдение сроков.
 - Порог: «зачтено» при выполнении всех обязательных критериев; недочёты устранимы — возможна однократная доработка в установленный срок.
 - Плагиат/чужой код без атрибуции — основание для «не зачтено» и пересдачи по решению комиссии.
 - Конфиденциальные данные: использовать анонимизированные/демо-наборы либо приложить разрешение правообладателя.
- Рейтинговая модель курса (100 баллов):
- Тесты – 15 баллов (автопроверка).

- Лабораторные работы – 45 баллов (рецензирование).
- Промежуточная аттестация – 15 баллов (лабораторная работа, рецензирование в ЛМС).
- Итоговая аттестация – 25 баллов (пакет материалов, рецензирование в ЛМС).

Порог «зачтено» по курсу: суммарно ≥ 60 баллов и выполнены все обязательные работы; при невыполнении ИА — «не зачтено» независимо от набранных баллов.

Решение о выдаче удостоверения принимается преподавателем программы/аттестационной комиссией на основании результата ИА («зачтено»).

2.5.2. Оценочные материалы

Оценочные материалы для текущего контроля приведены в Приложении 1.

Оценочные материалы для промежуточной аттестации приведены в Приложении 2.

Оценочные материалы для итоговой аттестации приведены в Приложении 3.

2.5.3. Анализ удовлетворенности требований слушателей

До начала обучения по программе и по итогам обучения проводится анкетирование слушателей для анализа ожидания и удовлетворенности их требований к организации и качеству обучения.

3. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

3.1. Материально-техническое обеспечение

Условия для функционирования электронной информационно-образовательной среды

Электронные информационные и образовательные ресурсы	Вид занятий	Наименование оборудования, программного обеспечения, необходимого слушателю для пользования ЭИОС
Система дистанционного обучения GetCourse, система видеоконференцсвязи	Лекции, Лабораторные работы, Практические занятия, СРС	Компьютер, подключенный к сети Интернет; интернет-браузер; наушники, микрофон, веб-камера, специализированное программное обеспечение, цифровые сервисы

3.2. Учебно-методическое и информационное обеспечение

3.2.1. Основная литература

1. Мюллер А., Гвидо С. Изучаем Python, машинное обучение и искусственный интеллект. — М.: ДМК Пресс, 2021.
2. Вандер Плас Дж. Python для сложных задач. Наука о данных и машинное обучение. — СПб.: Питер, 2020.
3. Лутц М. Изучаем Python. — СПб.: Символ-Плюс, 2020.
4. Гудфеллоу И., Бенджио Й., Курвилл А. Глубокое обучение. — М.: ДМК Пресс, 2018.

3.2.2. Дополнительная литература

1. Рашка С., Мирджалали В. Python и машинное обучение. Машинное и глубокое обучение с использованием Python, scikit-learn и TensorFlow 2. — М.: ДМК Пресс, 2021.
2. Харрис К., Харрис Э. Изучаем Python с помощью командных проектов. — СПб.: Питер, 2022.
3. Бэрри П. Изучаем программирование на Python. — М.: Эксмо, 2020.
4. Никита Кукленков. Программирование на Python для начинающих. — М.: Эксмо, 2022.
5. Артёмьев А.С. и др. Машинное обучение: учебное пособие. — М.: МФТИ, 2021.

3.2.3. Интернет-ресурсы

1. Официальная документация Python (русская версия): <https://docs.python.org/ru/3/>
2. Scikit-learn — документация и примеры: <https://scikit-learn.org/stable/>
3. Kaggle Learn (курсы по Python, pandas, ML): <https://www.kaggle.com/learn>
4. Google Colab: <https://colab.research.google.com/>
5. Real Python: <https://realpython.com/>
6. Хабр / Хабр Карьера — разделы «Python», «Машинное обучение»: <https://habr.com/ru/hub/python/>; https://habr.com/ru/hub/machine_learning/
7. Open Data Science (ods.ai): <https://ods.ai/>
8. eLIBRARY.RU: <https://elibrary.ru/>

3.2.4. Учебно-методическое обеспечение

Учебно-методические материалы программы размещены в электронном курсе в СДО GetCourse. В электронном курсе реализована система обратной связи в формате форума.

3.2.5. Перечень средств обучения

компьютеры с доступом в Интернет;
интерпретатор языка Python (например, PyCharm, GigaIDE или аналог);
IDE для разработки на Python, поддерживающая работу с библиотеками и фреймворками машинного обучения;
система дистанционного обучения GetCourse, включая платформу для тестирования, лабораторных работ и материалов курса;
система видеоконференцсвязи, позволяющая проводить синхронные вебинары, практические занятия и разборы лабораторных работ;
аудиовизуальные средства: видекамера, микрофон, веб-камера для синхронных занятий и коммуникации;
образовательные и информационные ресурсы, включая библиотеки Python и фреймворки для машинного обучения (scikit-learn, TensorFlow, Keras и другие).

3.3. Организация образовательного процесса

Обучение по программе реализовано с применением активных технологий совместного обучения в электронной среде (синхронные и асинхронные занятия). Лекционный материал представляется в виде комплекса видеолекций и мини-видеолекций, записей занятий, текстовых материалов, презентаций, размещаемых в LMS GetCourse. Данные материалы сопровождаются тестом, заданиями и дискуссиями в чате курса. Изучение теоретического материала (СРС) предполагается до и после синхронной части работы.

Синхронные занятия с преподавателем реализуются на базе инструментов видеоконференцсвязи и включают в себя установочные лекции с сопровождающими презентациями, практические занятия, сочетающие в себе групповую и индивидуальную работу. В качестве площадок для совместной синхронной и асинхронной работы будут использованы виртуальные доски и облачные сервисы для совместной работы, в том числе, рабочая тетрадь слушателя.

3.4. Кадровое обеспечение образовательного процесса

Руководитель программы:

Кузьмин Дмитрий Николаевич

Доцент кафедры «Информационные системы», Институт космических и информационных технологий, Сибирский федеральный университет, кандидат педагогических наук.

Эксперт в области преподавания Python для анализа данных и промышленного внедрения (production) для разных возрастных категорий. Преподаёт Python для Data Science и MLOps более

17 лет - в вузах, корпоративных программах и онлайн-курсах. Автор и соавтор образовательных программ, включая программы ДПО на цифровой кафедре СФУ «Python для анализа данных в праве и социальных науках» и «Python в производстве: автоматизация и аналитика». Разрабатывает лабораторные, кейсы и проекты, максимально приближенные к production:

- обработка «грязных» данных;
- оптимизация производительности;
- деплой модели с API и мониторингом;
- работа с большими объемами и распределёнными системами.

Пример оценочных материалы для текущего контроля

1. (SC) Что является основной целью EDA перед моделированием?

A. Подбор гиперпараметров модели

B. Предварительное исследование данных для выявления структуры, проблем и закономерностей

C. Увеличение выборки за счёт аугментаций

D. Снижение времени обучения модели

Ответ: B. Предварительное исследование данных для выявления структуры, проблем и закономерностей.

2. (MC) Какие задачи относятся к целям EDA?

A. Понимание структуры и характеристик данных

B. Выявление проблем качества (пропуски, выбросы)

C. Гарантированное повышение точности любой модели

D. Формирование гипотез о взаимосвязях переменных

E. Выбор информативных признаков

(выберите все подходящие)

Ответ: A, B, D, E.

A. Понимание структуры и характеристик данных; B. Выявление проблем качества; D. Формирование гипотез; E. Выбор информативных признаков.

3. (ORD) Расположите этапы анализа данных в корректной последовательности: 1 — Анализ данных; 2 — Интерпретация результатов; 3 — Сбор данных; 4 — Подготовка данных.

Введите порядок номеров: _____

Ответ: 3 → 4 → 1 → 2.

4. (MATCH) Соотнесите инструмент и назначение:

1. Jupyter Notebook → а) Численные операции с массивами

2. pandas → б) Визуализация графиков

3. NumPy → в) Интерактивная среда разработки/документирования

4. Matplotlib/Seaborn → г) Обработка табличных данных

5. Укажите соответствия в формате «1—?, 2—?, 3—?, 4—?».

Ответ: 1—в, 2—г, 3—а, 4—б.

5. (SC) Что делает `data.head()` в pandas?

A. Строит гистограмму первого столбца

B. Выводит размеры таблицы

C. Показывает первые строки датафрейма

D. Удаляет пропуски из набора данных

Ответ: C. Показывает первые строки датафрейма.

6. (MC) Какие показатели относятся к мерам разброса (вариативности)?

A. Среднее

B. Медиана

- C. Дисперсия
- D. Стандартное отклонение
- E. Мода

(выберите все подходящие)

Ответ: C, D.

C. Дисперсия; D. Стандартное отклонение.

7. (SC) Какой график уместен для зависимости между двумя количественными переменными?

- A. Гистограмма
- B. Диаграмма рассеяния (scatter plot)
- C. Круговая диаграмма
- D. Ящик с усами (box-plot)

Ответ: B. Диаграмма рассеяния (scatter plot).

8. (MC) Какие типичные ошибки новичков описаны в материале?

- A. Игнорирование проверки и подготовки данных
- B. Недооценка статистики и отказ от визуализаций
- C. Слишком ранний выбор простых моделей вместо сложных
- D. Переусложнение моделей без необходимости
- E. Отсутствие документирования шагов анализа

(выберите все подходящие)

Ответ: A, B, D, E.

A. Игнорирование проверки/подготовки данных; B. Недооценка статистики и отказ от визуализаций; D. Переусложнение моделей; E. Отсутствие документирования.

9. (MC) Какие признаки чаще всего признаются ненужными и удаляются?

- A. Постоянные (категориальные) признаки с одним значением
- B. Признаки с высокой долей пропусков
- C. Уникальные идентификаторы строк (случайные ID)
- D. Целевая переменная
- E. Сильно коррелирующие между собой признаки (мультиколлинеарность)

(выберите все подходящие)

Ответ: A, B, C, E.

A. Постоянные категориальные; B. С высокой долей пропусков; C. Уникальные ID; E. Сильно коррелирующие между собой.

10. (TF) Верно или неверно: «EDA помогает обнаружить дисбаланс классов и выбрать корректные стратегии до обучения модели».

- A. Верно
- B. Неверно

Ответ: A. Верно.

Пример оценочных материалов для промежуточной аттестации

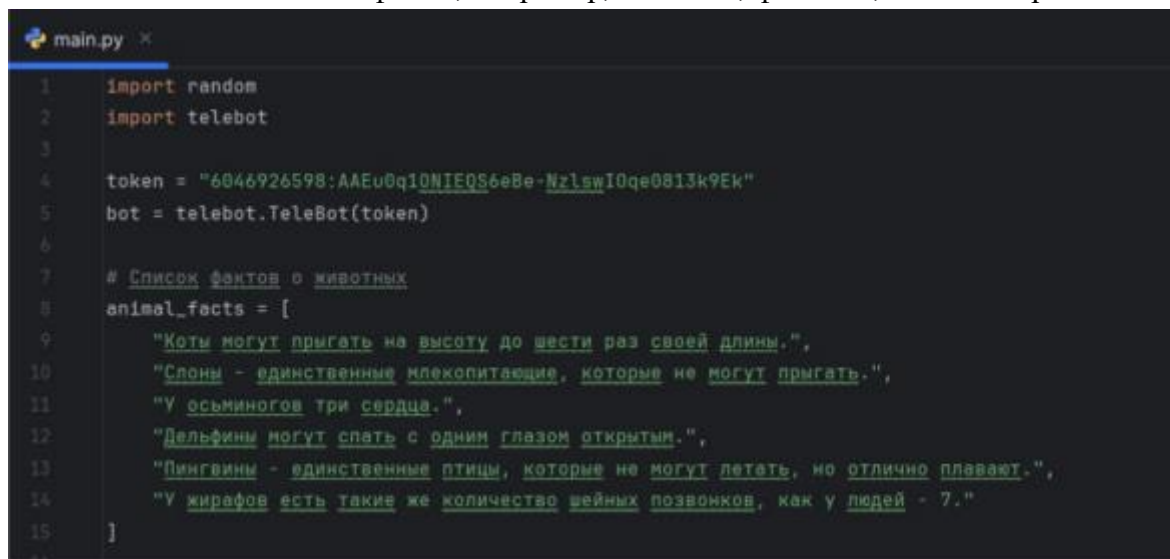
Создание Telegram-бота для генерации случайных фактов

В рамках этого задания мы поработаем со списками и циклом `for`. В качестве задания необходимо будет создать Telegram-бота, который будет отправлять пользователю случайные факты о животных. Пользователь может запрашивать факты, и бот будет отвечать на каждую команду. Бот должен использовать цикл `for` для отправки нескольких фактов за один запрос.

Итак, в рамках этого задания мы будем использовать бота, который ранее создавали на втором задании. Если Вы пропустили этот шаг, то можно вернуться ко второму заданию и вспомнить как создать через BotFather свой собственный бот.

Факты о животных мы будем хранить в списке. Создадим этот список и добавим туда случайные факты о животных.

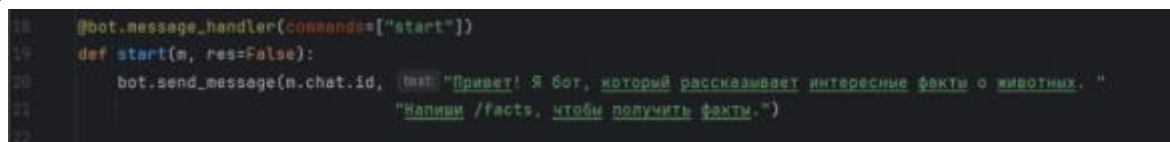
Вы можете написать свои факты, например, о птицах, фильмах, книгах и др.



```

1  import random
2  import telebot
3
4  token = "6046926598:AAEu0q10NIEQ56eBe-NzLswIOqe0813k9Ek"
5  bot = telebot.TeleBot(token)
6
7  # Список фактов о животных
8  animal_facts = [
9      "Коты могут прыгать на высоту до шести раз своей длины.",
10     "Слоны - единственные млекопитающие, которые не могут прыгать.",
11     "У осьминогов три сердца.",
12     "Дельфины могут спать с одним глазом открытым.",
13     "Пингвины - единственные птицы, которые не могут летать, но отлично плавают.",
14     "У жирафов есть такие же количество шейных позвонков, как у людей - 7."
15 ]
  
```

Реализуйте команду `/start`, которая будет отправлять пользователю приветственное сообщение:



```

18 @bot.message_handler(commands=["start"])
19 def start(m, res=False):
20     bot.send_message(m.chat.id, text="Привет! Я бот, который рассказывает интересные факты о животных. "
21                                     "Напиши /facts, чтобы получить факты.")
  
```

Далее, реализуйте команду `/facts`, которая будет отправлять пользователю 3 случайных факта.

Так как по заданию количество фактов, отправляемых пользователю известно заранее, то нам необходимо использовать цикл `for` для перебора и отправки фактов. Генерацию случайных фактов выполните через команду `random.sample(animal_facts, 3)`.

```
24 @bot.message_handler(commands=["facts"])
25 def facts(m, res=False):
26     # Получаем 3 случайных факта
27     selected_facts = random.sample(animal_facts, k=3)
28     # Отправляем факты пользователю
29     for fact in selected_facts:
30         bot.send_message(m.chat.id, fact)
31
32
33 bot.polling(none_stop=True, interval=0)
```

Ваш бот готов!

Не забудьте запустить его на выполнение через команду `polling`!

Вы можете придумать свой собственный бот, который будет хранить список и выдавать пользователю по запросу информацию из этого списка либо выполнить задание, которое мы рассмотрели на уроке.

Пример оценочных материалов для итоговой аттестации

Методические указания к подготовке итоговой аттестационной работе

Основная цель итоговой аттестационной работы – выполнить работу, демонстрирующую уровень подготовленности к самостоятельной профессиональной деятельности.

Процедура защиты итоговой аттестационной работы

ИАР может выполняться как индивидуально, так и в группах 2-3 человека. Защита ИАР включает презентацию работы перед комиссией. Защита ИАР дает возможность продемонстрировать уровень приобретенных слушателем профессиональных компетенций.

Слушатель предоставляет результат выполненной работы в формате PDF, оформленной и отвечающей требованиям к содержанию итоговой аттестационной работы. Список использованных источников литературы приводится в конце ИАР.

Объем презентации следует выбирать исходя из длительности выступления (обычно — не более 5–7 минут). В выступлении должны быть четко обозначены область и актуальность работы, постановка задачи, приведены результаты, полученные слушателем.

Защита ИАР может проходить как в синхронном формате в режиме видеоконференцсвязи, так и в асинхронном формате, когда слушатель или группа слушателей записывают свой доклад по проделанной работе и прикрепляют его в электронный курс вместе с текстом ИАР.

Требования к устному докладу

1. Приветствие, обращение к членам комиссии и представление.
2. Тема итоговой аттестационной работы.
3. Актуальность, цель и задачи работы.
4. Анализ результатов работы.
5. Заключение.

Продолжительность выступления – 7–8 минут.

По результатам защиты ИАР аттестационная комиссия принимает решение о предоставлении слушателям по результатам освоения дополнительной профессиональной программы профессиональной переподготовки права заниматься профессиональной деятельностью в сфере разработки программного обеспечения и выдаче диплома о профессиональной переподготовке.

Примерные темы ИАР

Примерные темы итоговых аттестационных работ:

1. Классификация изображений с использованием свёрточных нейронных сетей.
 2. Прогнозирование спроса или временных рядов с помощью рекуррентных архитектур (LSTM, GRU).
 3. Обнаружение объектов на изображениях с использованием TensorFlow/Keras.
 4. Сравнение архитектур CNN и MLP на задаче классификации табличных данных.
 5. Оптимизация нейросетевой модели для работы в ограниченной вычислительной среде (MVP).
 6. Применение transfer learning для решения прикладной задачи компьютерного зрения.
- Обучающиеся предоставляется право самостоятельно определить тему ИАР.

Структура и содержание итоговой аттестационной работы

Задачи итоговой аттестационной работы должны иметь общетеоретическое или практическое значение.

Структура:

Введение:

- Описание задачи. Актуальность.
- Описание подхода к разработке приложения на языке Python для решения задачи.
- Цель работы.
- Задачи.

1. Обзорная часть

Название обзорной части можно сделать более специфическим, например, «Обзор алгоритмов анализа данных», в зависимости от типа разрабатываемого приложения.

1.1. Обзор задачи

Название также можно сделать более специфическим, например, «Обзор задачи создания web-приложений». В этом разделе необходимо привести постановку задачи. Специфику разработки выбранного типа приложения, краткую историю развития технологий.

1.2. Обзор инструментов для создания приложений

В разделе приводится краткое описание существующих инструментов для создания выбранного типа приложений. Для задач машинного обучения можно привести описание библиотек Pandas и Numpy.

2. Проектирование приложения

2.1. Описание желаемых функций приложения

В общем и целом, для рассматриваемого продукта необходимо сформулировать функциональные и нефункциональные требования.

2.2 Требования к программе или программному изделию

В разделе «Требования к программе или программному изделию» могут содержаться следующие нефункциональные требования:

- требования к надежности;
- требования к составу и параметрам технических средств;
- требования к информационной и программной совместимости;

Нефункциональное требование – это описание присущих свойств или характеристик, которые система должна демонстрировать, или ограничения, которые она должна соблюдать, требования к качеству и так далее.

При формулировании требований к качеству необходимо рассмотреть следующие:

- атрибуты внешнего качества: производительность, надёжность, доступность, масштабируемость, безопасность;
- атрибуты качества в использовании: результативность, эффективность, скорость обучения, точность, утомляемость;
- ограничения: совместимость с оборудованием, системным ПО, ограничения на языки и средства разработки, допущения о квалификации пользователей, состав поставки, лицензионные ограничения.

2.3 Стадии и этапы разработки

В разделе «Стадии и этапы разработки» устанавливают необходимые стадии разработки, этапы и содержание работ (перечень программных документов, которые должны быть разработаны, согласованы и утверждены, если такие предусмотрены).

Например:

Разработка должна быть проведена в три стадии:

- 1) разработка технического задания;*
- 2) рабочее проектирование: дизайн, создание контента, разработка*
- 3) Тестирование, обзор и запуск*
- 4) Поддержка.*

Также здесь приводятся подробное описание стадий разработки.

2.4. Описание архитектуры приложения

Описание основных компонентов приложения и их взаимодействия:

- Основные модули программы
- Используемый инструмент для создания приложения (Google Colab, Jupyter Notebook и

т.д.).

3. Реализация приложения

3.1 Написание программного кода

Приводятся основные фрагменты кода приложения.

3.2 Тестирование приложения

Приводятся примеры использования приложения для определения его работоспособности. Например, скриншоты с результатами распознавания. Нужно загрузить несколько изображений и проанализировать результаты (все ли заявленные функции приложения работают).

Заключение:

Краткое описание полученных результатов:

- Описание задачи
- Описание инструмента создания приложения
- Описание приложения.
- Планируемые дальнейшие шаги.

Порядок выполнения итоговой аттестационной работы

Этапы выполнения ИАР задаются графиком выполнения и контролируются руководителем:

1. Выбор и согласование с руководителем темы ИАР работы.
2. Формулировка цели и задач ИАР работы. Обоснование актуальности выбранного направления работы.
3. Литературно-патентный поиск по тематике работы. Поиск аналогичных решений. Формирование концепции предлагаемых подходов, методов.
4. Разработка методов достижения цели проекта. Выбор и/или проектирование математического, программного, технического и иных видов обеспечений для достижения цели и решения задач ИАР.
5. Моделирование. Разработка программной или программно-аппаратной реализации проекта.
6. Проведение практических испытаний и анализ результатов.
7. Обобщение проведенных исследований. Формирование итоговой аттестационной работы.
8. Публичная защита ИАР.