

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: АРЦИОХ ДМИТРИЙ ВЛАДИМИРОВИЧ
Должность: Директор Организации
Дата подписания: 05.12.2025 18:46:09
Уникальный программный ключ:
с3e9ce6eee7d75b5b63095041353570976b44378

**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

УТВЕРЖДЕНО

Приказом № 1-067 от «07» 09 2025г.

**ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА
ПОВЫШЕНИЯ КВАЛИФИКАЦИИ**

«РАЗРАБОТКА СИСТЕМ AI: PYTHON (ПЕРВАЯ СТУПЕНЬ)»

Магас

2025

1. ОБЩАЯ ХАРАКТЕРИСТИКА ДОПОЛНИТЕЛЬНОЙ ПРОФЕССИОНАЛЬНОЙ ПРОГРАММЫ

1.1. Аннотация программы

Программа повышения квалификации «Разработка систем AI: Python (первая ступень)» нацелена на формирование практических навыков программирования на Python и построения базовых моделей машинного обучения. Актуальность программы связана с ростом спроса на специалистов, умеющих быстро собирать прототипы ИИ-моделей и проверять их качество на реальных данных. Уникальность – в «hands-on» формате: каждое занятие заканчивается работающим кодом, а все решения оцениваются по метрикам.

Обучение проходит в смешанном формате: синхронные вебинары и лабораторные работы дополняются асинхронными заданиями с проверкой в ЛМС и самостоятельной практикой. Содержание охватывает пять блоков: основы Python и алгоритмизация; подготовка и анализ табличных данных в pandas; базовые алгоритмы ML в scikit-learn (регрессия, классификация) с оценкой качества по точности, полноте, F-мере и ROC-кривой; построение прототипов моделей; введение в глубокое обучение на TensorFlow/Keras для распознавания объектов. Во всех модулях — кейсы, контроль воспроизводимости, работа с baseline и улучшениями.

1.2. Цель реализации программы

Совершенствование профессиональных компетенций в области разработки и отладки программного кода с использованием языка программирования 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 документов организации	Использовать методы и приемы формализации поставленных задач Использовать методы и приемы алгоритмизации поставленных задач Использовать программное обеспечение для	Методы и приемы формализации поставленных задач Языки формализации функциональных спецификаций Методы и приемы алгоритмизации поставленных

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

				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. Перечень профессиональных компетенций в рамках имеющейся квалификации, качественное изменение которых осуществляется в результате обучения:

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

1.6. Требования к слушателям (категории слушателей)

Настоящая программа предназначена для начинающих программистов и разработчиков, специалистов, желающих освоить технологии искусственного интеллекта, аналитиков данных, интересующихся разработкой.

Требования к слушателям: наличие среднего профессионального или высшего образования; базовые знания Python (синтаксис, переменные, условия/циклы) и основ математики (линейная алгебра, статистика); готовность к освоению новых технологий и алгоритмов; уверенные навыки работы с ПК и в ЛМС. Предпочтителен опыт работы 0–2 года в ИТ или смежных областях. Возрастной ориентир 18–45 лет.

1.7. Форма обучения: очно-заочная с применением ЭО и ДОТ. Программа реализуется исключительно с использованием дистанционных образовательных технологий

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	ГР	ЛБ	ЛК	ГР	ЛБ			
1.	Основы работы в Python	34	4	0	0	6	0	0	4	20	Тест, выполнение ЛР
2.	Обработка данных в Python	36	6	0	0	6	0	0	6	18	Тест, выполнение ЛР
3.	Алгоритмы машинного обучения	20	4	0	0	2	0	0	6	8	Тест, выполнение ЛР
4.	Построение моделей	30	4	0	0	4	0	0	8	14	Тест, выполнение ЛР
5.	Оценка качества модели	16	2	0	0	4	0	0	2	8	Тест, выполнение ЛР
Итоговая аттестация		8									Выполнение и защита ИАР
	Подготовка ИАР	6									
	Защита ИАР	2									
Итого (час.)		144	20	0	0	22	0	0	26	72	

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	ГР	ЛБ	ЛК	ГР	ЛБ			
1	Основы работы в Python	34	4	0	0	6	0	0	4	20	
1.1.	Введение в программирование. Интерпретатор. IDE. Переменные и типы данных	10	2		2					6	
1.2.	Основы структур данных, работа с файлами	12	2		2				2	6	Выполнение ЛР
1.3.	Создание телеграмм бота	11			2				2	7	
1.4.	Текущий контроль	1								1	Тест, выполнение ЛР
2	Обработка данных в Python	36	6	0	0	6	0	0	6	18	
2.1.	Введение в анализ данных	8	2						2	4	Выполнение ЛР

2.2.	Инструменты для анализа данных Jupyter Notebook, Google Colab	8	2		2				4	
2.3.	Создание DataFrame. Работа с существующим набором данных	9			2			2	5	Выполнение ЛР
2.4.	Разведочный анализ	10	2		2			2	4	
2.5.	Текущий контроль	1							1	Тест, выполнение ЛР
2.6.	Промежуточная аттестация									
3	<i>Алгоритмы машинного обучения</i>	20	4	0	2	0	0	6	8	
3.1.	Введение в машинное обучение	8	2					2	4	Выполнение ЛР
3.2.	Анализ набора данных различными методами	11	2		2			4	3	
3.3.	Текущий контроль	1							1	Тест, выполнение ЛР
4	<i>Построение моделей</i>	30	4	0	4	0	0	8	14	
4.1.	ML модель. От данных до внедрения	9	2					2	5	Выполнение ЛР
4.2.	Посторенние модели "Прогнозирование цены автомобиля"	10	2		2			2	4	Выполнение ЛР
4.3.	Построение модели "Вероятность дефолта"	10			2			4	4	
4.4.	Текущий контроль	1							1	Тест, выполнение ЛР
4.5.	Промежуточная аттестация									Тестирование
5.	<i>Оценка качества модели</i>	16	2	0	4	0	0	2	8	
5.1.	Анализ данных в Python	8	2		2				4	
5.2.	Построение и анализ модели распознавания объектов	7			2			2	3	Тест, выполнение ЛР
5.3.	Текущий контроль	1							1	
Итоговая аттестация		8								Выполнение и защита ИАР
	Подготовка ИАР	6								
	Защита ИАР	2								
Итого (час.)		144	20	2	22	0	0	26	72	

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. Основы работы в Python		34
Тема 1.1. Введение в программирование. Интерпретатор. IDE. Переменные и типы данных	Содержание синхронных лекционных занятий (2 часа): 1. Что такое программирование и язык программирования 2. Достоинства и недостатки языка Python 3. Обзор IDE для языка Python (PyCharm, GigaIDE и др) 4. Простейшие программы на Python	2
	Контактная (синхронная) работа (2 часа): Лабораторная работы – Автоматическая фиксация скорости – Игра – «угадай случайное число». – Среднеарифметическое	2
	СРС (6 часов): 1. Изучение теоретического материала по теме 1.1 2. Решение задач (Числа Фибоначчи, Самая длинная строка)	6
Тема 1.2. Основы структур данных, работа с файлами	Содержание синхронных лекционных занятий (2 часа): 1. Управляющие конструкции языка Python 2. Структуры данных (списки, словари, кортежи) 3. Работа с файлами 4. Основы ООП	2
	Контактная (синхронная) работа (2 часа): Лабораторная работы – Анализ текста на ошибки – Универсальный код речей	2
	Контактная (асинхронная) работа (2 часа): Лабораторная работы – Эмулятор домашнего питомца	2
	СРС (6 часов): 1. Изучение теоретического материала по теме 1.2 2. Решение задач (Загранпаспорт, Самое длинное слово)	6
Тема 1.3. Создание Telegram-бота	Контактная (синхронная) работа (2 часа): Лабораторная работы – Создание телеграм бота «Повторюшка»	2
	Контактная (асинхронная) работа (2 часа): Лабораторная работы – Создание телеграм бота «Прогноз погоды»	2
	СРС (7 часов): 1. Изучение теоретического материала по теме 1.3 2. Создание телеграм бота «Генератор фактов»	7
Текущий контроль	Тестирование по теме «Основы работы в Python»	1
Раздел 2. Обработка данных в Python		36
Тема 2.1 Введение в анализ данных	Содержание синхронных лекционных занятий (2 часа): 1. Что такое анализ данных. Где он применяется 2. Требования к данным. Методы их достижения	2

	3. Алгоритмы обработки данных	
	Контактная (асинхронная) работа (2 часа): Лабораторная работы – Изучаем методы обработки данных на примере датафрейма «Титаник»	2
	СРС (4 часа): 1. Изучение теоретического материала по теме 2.1 2. Выполнение дополнительных заданий в датафрейме «Титаник» или в аналогичном	4
Тема 2.2 Инструменты для анализа данных Jupyter Notebook, Google Colab	Содержание синхронных лекционных занятий (2 часа): 1. Альтернативные среды разработки. Сравним локальную работу в Jupyter и облачную в Colab 2. Разница в форматах файлов ru и ipynb 3. Обзор рабочей среды ноутбуков	2
	Контактная (синхронная) работа (2 часа): Лабораторная работы – Работа с датафреймом «Титаник» (часть 2)	2
	СРС (4 часа): 1. Изучение теоретического материала по теме 2.2 2. Выполнение дополнительных заданий в датафрейме «Титаник» или в аналогичном	4
Тема 2.3 Создание DataFrame. Работа с существующим набором данных	Контактная (синхронная) работа (2 часа): Лабораторная работы – Создание датафрейма «Игра престолов»	2
	Контактная (асинхронная) работа (2 часа): Лабораторная работы – Создаем DataFrame из своего набора данных	2
	СРС (5 часов): 1. Изучение теоретического материала по теме 2.3 2. Создание и первичная обработка собственного DataFrame	5
Тема 2.4 Разведочный анализ	Содержание синхронных лекционных занятий (2 часа): 1. Построение различных видов графиков 2. Применение графиков для анализа данных	2
	Контактная (синхронная) работа (2 часа): Лабораторная работы – Графический анализ данных на датафрейме «Титаник»	2
	Контактная (асинхронная) работа (2 часа): Лабораторная работы – Анализ датафрейма «Герои»	2
	СРС (4 часа): 1. Изучение теоретического материала по теме 2.4 2. Выполнение дополнительных заданий в датафрейме «Герои»	4
Текущий контроль	Тестирование по теме «Разведочный и графический анализ данных»	1
Промежуточная аттестация	Оценка выполнения домашних работ по модулям 1 и 2	
Раздел 3. Алгоритмы машинного обучения		20
Тема 3.1 Введение в машинное обучение	Содержание синхронных лекционных занятий (2 часа): 1. Основные понятия машинного обучения (целевая функция, признаки, тестовая и обучающая выборки) 2. Виды обучения (с учителем, без учителя) 3. Понятие нейросети	2
	Контактная (асинхронная) работа (2 часа): Лабораторная работы – Знакомство с возможностями языка Python для машинного обучения	2

	СРС (4 часа): 1. Изучение теоретического материала по теме 3.1 2. Поиск задач, решаемых при помощи машинного обучения	4
Тема 3.2 Анализ набора данных различными методами	Содержание синхронных лекционных занятий (2 часа): Обучение нейросети методами линейной регрессии и случайного леса. Области применимости данных моделей	2
	Контактная (синхронная) работа (2 часа): Лабораторная работы – Считаем линейную регрессию	2
	Контактная (асинхронная) работа (4 часа): Лабораторная работы Обучаем простую нейросеть	4
	СРС (3 часа): 1. Изучение теоретического материала по теме 3.2 2. Обучение простой нейросети для решения задачи найденной в п. 3.1	3
Текущий контроль	Тестирование по теме «Основы машинного обучения»	1
Раздел 4. Построение моделей		30
Тема 4.1 Построение модели «Прогнозирование цены автомобиля»	Содержание синхронных лекционных занятий (2 часа): 1. ML модель. От данных до внедрения	2
	Контактная (асинхронная) работа (2 часа): Лабораторная работы – Обучение нейросети «Прогнозирование цены автомобиля «Форд»	2
	СРС (5 часа): 1. Изучение теоретического материала по теме 4.1 2. Обучение нейросети «Прогнозирование цены автомобиля «Тойота»	5
Тема 4.2 Построение модели «Вероятность дефолта»	Содержание синхронных лекционных занятий (2 часа): 1. Обучение нейросети методом «случайный лес»	2
	Контактная (синхронная) работа (2 часа): Лабораторная работы – Работа с датасетом «Вероятность дефолта»	2
	Контактная (асинхронная) работа (4 часа): Лабораторная работы Обучение нейросети «Вероятность дефолта»	2
	СРС (4 часа): 1. Изучение теоретических материалов 2. Обучение нейросети «Вероятность дефолта» (продолжение)	4
Тема 4.3 Сравнение методов обучения нейросетей	Контактная (синхронная) работа (2 часа): Лабораторная работы – Изучаем метрики позволяющие оценить качество обученности нейросети	2
	Контактная (асинхронная) работа (4 часа): Лабораторная работы – Сравниваем качество обученности нейросетей из предыдущих заданий	4
	СРС (4 часа): 1. Изучение теоретического материала по теме 4.2	4
Текущий контроль	Тестирование по темам «Методы обучения нейросетей и метрики качества обученности нейросети»	1
Промежуточная аттестация	Оценка выполнения домашних работ по модулям 3 и 4	
Раздел 5. Оценка качества модели		16
Тема 5.1 Анализ данных в	Содержание синхронных лекционных занятий (2 часа):	2

<i>Python</i>	1. Систематизируем инструменты: numpy, pandas, scipy и matplotlib/plotly 2. Рассмотрим требования к итоговому проекту	
	<i>Контактная (синхронная) работа (2 часа):</i> <i>Лабораторная работы</i> – Распознавание объектов на изображениях из набора данных CIFAR-10	2
	<i>СРС (4 часов):</i> 1. Изучение теоретического материала по теме 5.1 2. Закрепление навыков обучения нейросетей	4
	<i>Тема 5.2 Построение и анализ модели распознавания объектов</i> <i>Контактная (синхронная) работа (2 часа):</i> <i>Лабораторная работы</i> – Закрепляем полученные навыки на обработке датасета «Распознавание предметов одежды»	2
	<i>Контактная (асинхронная) работа (2 часа):</i> <i>Лабораторная работы</i> – Закрепляем полученные навыки на обработке датасета «Распознавание предметов одежды»	2
	<i>СРС (3 часа):</i> 1. Изучение теоретического материала по теме 5.2 2. Закрепление полученных навыков	3
Текущий контроль	<i>Тестирование по теме «Задача распознавания объектов»</i>	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): разработка и защита итоговой аттестационной работы по программе «Разработка систем AI: Python (первая ступень)». Работа выполняется индивидуально либо в группе 2–3 человек с указанием личного вклада. Защита может проводиться синхронно (вебинар) либо асинхронно (запись видеозащиты + ответы на вопросы в ЛМС в установленное окно).

Требования к содержанию итоговой работы:

- постановка задачи, актуальность, цель и задачи; выбор целевой метрики; описание данных и источников, план предобработки/признаков;
- baseline и ожидаемое улучшение; схема валидации (train/valid или K-fold);
- реализация модели(ей): логистическая регрессия, случайный лес (scikit-learn) и/или простая модель распознавания объектов (TensorFlow/Keras); результаты по метрикам (RMSE/MAE или ROC-AUC/F1/Precision/Recall) и их интерпретация;
- требования к воспроизводимости: фиксированный random_state (при необходимости), версии библиотек, структурированный код;
- выводы и рекомендации по улучшению модели/данных.

Структура устного доклада: приветствие и представление; тема; актуальность, цель и задачи; полученные результаты и их анализ; заключение.

Примерные направления тем: анализ данных; прогнозирование показателей (напр., цена автомобиля/прибыль); классификация (напр., вероятность дефолта); распознавание объектов.

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

Критерии оценивания ИА:

1. Корректная постановка задачи, подготовка данных и выбор целевых метрик.
2. Наличие baseline, честная валидация и обоснованное сравнение результатов.
3. Качество и интерпретация метрик; выводы по ошибкам модели.
4. Воспроизводимость и качество кода (структура, читаемость, requirements).
5. Соответствие формату и срокам загрузки в ЛМС.

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

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 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: видеокамера, 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 объёмами и распределёнными системами.

Пример оценочных материалы для текущего контроля

1. Что такое Jupyter Notebook?

- A. Библиотека Python для визуализации данных
- B. Веб-интерфейс для написания и выполнения кода
- C. Сервис облачных вычислений
- D. Фреймворк для машинного обучения

Верно: B. Веб-интерфейс для написания и выполнения кода

2. Какой файл имеет расширение .ipynb?

- A. JavaScript-файл
- B. HTML-документ
- C. Документы Word
- D. Блокнот Jupyter

Верно: D. Блокнот Jupyter

3. Google Colab позволяет пользователям хранить файлы проекта непосредственно в...

- A. Dropbox
- B. Google Drive
- C. OneDrive
- D. Локальном хранилище компьютера

Верно: B. Google Drive

4. Основное преимущество использования Google Colab перед локальным Jupyter Notebook заключается в...

- A. Бесплатных GPU и TPU ресурсах
- B. Лучшей поддержке Python библиотек
- C. Большем количестве встроенных инструментов
- D. Отсутствии надобности настройки окружения

Верно: A. Бесплатных GPU и TPU ресурсах

5. Для запуска ядра Jupyter используется команда...

- A. `python notebook.py`
- B. `jupyter lab`
- C. `pip install jupyter`
- D. `jupyter notebook`

Верно: D. jupyter notebook

6. В каком облаке работает Google Colab?

- A. Amazon Web Services
- B. Microsoft Azure
- C. Google Cloud Platform
- D. Yandex.Cloud

Верно: C. Google Cloud Platform

7. Для установки пакета в ячейке Jupyter достаточно написать команду...

- A. `%install package_name`
- B. `import package_name`
- C. `conda install package_name`
- D. `!pip install package_name`

Верно: D. `!pip install package_name`

8. Какой тип графического представления доступен в обоих сервисах (Jupyter и Colab)?

- A. Matplotlib
- B. Seaborn
- C. Plotly
- D. Все вышеперечисленные

Верно: D. Все вышеперечисленные

9. Версия интерпретатора Python, используемого в Google Colab, обычно...

- A. Всегда последняя версия Python
- B. Старая версия Python
- C. Определяется самим пользователем
- D. Зафиксирована разработчиками Google

Верно: D. Зафиксирована разработчиками Google

10. Где хранится временная сессия при работе в Google Colab?

- A. В браузере пользователя
- B. Во внешней памяти
- C. В виртуальной машине сервера
- D. Локально на жестком диске ПК

Верно: C. В виртуальной машине сервера

Пример оценочных материалов для промежуточной аттестации

1. Лабораторная работа «Основы структур данных, работа с файлами».

Вам необходимо написать программу для отдела, который занимается выдачей загранпаспортов. Программа должна осуществлять транслитерацию фамилии, имени, отчества по следующим требованиям:

A (a) -> A (a) Ж (ж) -> Zh (zh) Н (н) -> N (n) Ф (ф) -> F (f) Ъ (ъ) -> Ie (ie) Б (б) -> B (b) З (з) -> Z (z) О (о) -> O (o) X (х) -> Kh (kh) Э (э) -> E (e) В (в) -> V (v) И (и) -> I (i) П (п) -> P (p) Ц (ц) -> Ts (ts) Ю (ю) -> Iu (iu) Г (г) -> G (g) Ы (ы) -> Y (y) Р (р) -> R (r) Ч (ч) -> Ch (ch) Я (я) -> Ia (ia) Д (д) -> D (d) К (к) -> K (k) С (с) -> S (s) Ш (ш) -> Sh (sh) Ь (ь) -> не пишется Е (е) -> E (e) Л (л) -> L (l) Т (т) -> T (t) Щ (щ) -> Shch (shch) Ё (ё) -> E (e) М (м) -> M (m) У (у) -> U (u) Ы (ы) -> Y (y)

Например, для строки: Попов Василий Вячеславович, формат вывода будет: Popov Vasili Viacheslavovich

Напишите функцию, которая возвращает самую длинную неповторяющуюся подстроку для входной строки. Если несколько подстрок совпадают по длине, функция возвращает ту, которая встречается первой.

Например:

sssssss -> s

abcdefab -> abcdef

abcabcbb -> abc

2. Лабораторная работа «Создание датафрейма».

Основное задание

1. Создайте датафрейм, состоящий из нескольких строк (порядка 10), содержащий данные о различных металлах и их свойствах. Для каждой записи необходимо указать:

- Название (название металла)
- Плотность (плотность металла в г/см³)
- Температура плавления (температура плавления в градусах Цельсия)
- Применение (основные области применения металла, например, строительство, электроника и т.д.)
- Доступность (логическое значение: True, если металл доступен для использования, False, если нет)

2. Выведите на экран:

- Содержимое датафрейма.
- Размерность датафрейма.
- Названия признаков (столбцов).
- Первые 4 строки таблицы.
- Все металлы с плотностью в диапазоне [7, 11] г/см³.
- Все доступные металлы.

3. Отправьте файл с кодом в формате .ipynb в качестве ответа на задание. Название файла должно содержать вашу фамилию, например, Ivanov-task1.ipynb.

Дополнительное задание

Дополнительное задание не является обязательным для выполнения (его можно не выполнять).

1. Преобразуйте датафрейм:

- Добавьте в конец таблицы еще 3 объекта такой же структуры, задав значения признаков.
- Добавьте в таблицу еще один столбец - Устойчивость к коррозии. Если металл является коррозионно-стойким, то значение в столбце должно быть равно True, если нет — False.

2. Выведите на экран содержимое обновленного датафрейма.

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

Темы выполнения ИАР, могут быть следующими:

1. Анализ данных
2. Прогнозирование прибыли от инвестиционной деятельности
3. Моделирование рыночных цен на акции компании
4. Применение методов машинного обучения при анализе данных

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.

Список использованных источников

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.

Критерии оценивания итоговой аттестационной работы

Оценка «отлично» ставится, если слушатель при выполнении ИАР глубоко усвоил аналитический материал по теме итоговой работы, исчерпывающе, последовательно, четко и логически стройно излагает актуальность, методы решения задач, уверенно использует современные информационные технологии для достижения цели итоговой работы, в ходе выполнения ИАР применил язык программирования Python системно на экспертном уровне. Контролировал весь цикл программной разработки в проекте.

Оценка «хорошо» выставляется слушателю, если он усвоил аналитический материал по теме итоговой работы, отразил в итоговой работе актуальность, методы решения производственно-технологических задач, при выполнении ИАР самостоятельно разработал отдельные модули программы.

Оценка «удовлетворительно» ставится, если слушатель усвоил аналитический материал по теме итоговой работы, отразил в итоговой работе актуальность, методы решения производственно-технологических задач, при выполнении ИАР под контролем преподавателя разработал программное обеспечение.

Защита итоговой аттестационной работы является обязательной.