

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: АРЦЮХ ДМИТРИЙ ВЛАДИМИРОВИЧ
Должность: Директор Организации
Дата подписания: 03.08.2026 15:24:24
Уникальный программный ключ:
194e9de362a3e118beb1b1af4bc7c1577477a952

**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 № 16-ОБР от «31» июля 2026г.

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
«Системная аналитика»**

Магас 2026

1.Пояснительная записка

Уровень освоения программы: стартовый

Направленность (профиль) программы – техническая

Настоящая программа реализуется в очно-заочной форме с применением ЭО и ДОТ. Аудиторные занятия (лекции и практические занятия) проводятся очно в синхронном формате. Самостоятельная работа осуществляется с применением ЭО и ДОТ в электронной информационно-образовательной среде Заказчика. Программа реализуется в формате пятидневного практического интенсива: команды слушателей работают над единой сквозной бизнес-задачей от реального заказчика — от интервью и сбора требований до финальной защиты артефактов. Каждый день включает лекцию, постановку задачи, самостоятельную работу команды и встречу с ментором; дополнительно предусмотрены очные встречи с бизнес-заказчиком (промежуточное ревью и финальная защита).

Технические требования для слушателя: ноутбук с предустановленным программным обеспечением —

- Git (клиент командной строки или графическая оболочка);
- учётная запись на российском Git-сервисе (GitVerse, GitFlic);
- таск-трекер (YouTrack, Planfix или локально развёрнутая Jira);
- редактор для диаграмм Draw.io;
- среда разработки (IntelliJ IDEA, VS Code или Eclipse);
- браузер (Chrome, Firefox или Edge);
- офисный пакет;
- доступ к ИИ-инструментам (DeepSeek, GigaChat, YandexGPT, Qwen).

Необходим стабильный Wi-Fi или проводной интернет для одновременной работы всех участников. Условия обучения: артефакты и рабочие материалы фиксируются командами в репозитории и на досках для совместной работы (Miro, Trello) в течение всего периода обучения; связь с ментором и организаторами поддерживается в рабочих чатах.

1.1. Актуальность Программы

Актуальность обусловлена тем, что перевод требований бизнеса на язык, понятный разработке, требует не формального описания «как есть», а системного мышления: умения выявлять неявные потребности заказчика, отделять слова от реальных проблем, формализовать сценарии использования и проектировать структуру данных и API так, чтобы решение можно было безопасно развивать. Программа построена на работе с реальной бизнес-

задачей от практикующего заказчика и с применением ИИ-инструментов для генерации и проверки артефактов, что отражает актуальную практику работы современного системного аналитика.

1.2. Категории обучающихся

Взрослые (от 18 лет) — обучающиеся и выпускники «Школы 21» направлений «Бизнес/системная аналитика», уже обладающие базовыми навыками уровня Junior+ и стремящиеся повысить квалификацию до уровня Middle через практику на реальном производственном кейсе.

1.3. Цель и задачи программы

Цель Программы – повышение инженерного уровня специалистов с Junior+ до Middle через внедрение практико-ориентированного обучения на реальных производственных кейсах, формирование системного мышления и создание верифицируемого портфолио, подтверждающего квалификацию перед работодателями.

Реализация поставленной цели предусматривает решение ряда задач.

Задачи Программы

- Развить у участников практические навыки системного проектирования, анализа архитектуры через отработку реальных производственных сбоев.
- Сформировать культуру документирования решений и самоанализа через обязательное ведение записей об архитектурных решениях и инженерных дневников.
- Обучить осознанно использовать инструменты искусственного интеллекта для генерации артефактов, но с обязательной верификацией и доработкой, чтобы избежать слепого копирования.
- Сформировать навыки полного цикла аналитической работы: от сбора требований и моделирования до проектирования API-контрактов и передачи артефактов в разработку.

1.4. Сроки реализации

Срок реализации: 5 дней

Трудоемкость: 32 часа

1.5. Планируемые результаты обучения

Навыки:

- Осознанно использовать ИИ-инструменты для генерации и проверки артефактов, критически оценивать выдачу и корректировать её с учётом

бизнес-правил.

- Проверять артефакты по чек-листу готовности, выявлять пробелы и формулировать план доработок.
- Собирать полный пакет артефактов и одностраничное резюме для передачи в команду разработки с указанием ключевых решений и рисков.

Знания:

- Понимание методов выявления неявных требований и когнитивных искажений при интервьюировании заказчика.
- Знание форматов описания требований и структуры OpenAPI-спецификации с правилами версионирования.
- Владение критериями качества и чек-листами готовности для каждого типа артефакта.

Умения:

- Проводить интервью с заказчиком, фиксировать требования в таблице «Сказал → Нужно → Риск» и выявлять скрытые потребности.
- Строить ER-диаграммы с выделением агрегатов и инвариантов, а также State Diagrams для описания жизненного цикла сущностей.
- Проектировать OpenAPI-спецификации с учётом обратной совместимости, версионирования и полного набора статус-кодов и примеров.

1.6. Календарный учебный график

Количество учебных недель – 1.

Количество учебных дней – 5.

Режим занятий: до 6 часов в день.

Общая трудоёмкость программы — 32 часа: 22 часа аудиторных занятий (2,5 часа лекций и 19,5 часа практических занятий), 8 часов самостоятельной работы и 2 часа промежуточной аттестации. Аудиторные занятия проводятся очно, самостоятельная работа — с применением ЭО и ДОТ в электронной информационно-образовательной среде Заказчика.

2.Содержание программы

2.1. Учебный план

№п/п	Наименование разделов/дисциплин/модулей	Общая трудоемкость	Всего аудиторных часов	в т.ч		Самостоятельная работа	Форма промежуточной аттестации/контроля (при наличии)
				Лекции/теория	Практические занятия		
1	Сбор требований и интервью с заказчиком	6	4,5	0,5	4,0	1,5	—
2	User stories и use cases	6	4,5	0,5	4,0	1,5	—
3	Проектирование доменной модели: ER-диаграмма и State Diagram	6	4,5	0,5	4,0	1,5	Текущий контроль: Промежуточное ревью с заказчиком
4	Проектирование API-контрактов и версионирование	6	4,5	0,5	4,0	1,5	—
5	Контроль качества, доработка и защита	6	4,0	0,5	3,5	2,0	—
	Промежуточная аттестация	2	—	—	—	—	Защита проекта
	Итого часов	32	22	2,5	19,5	8	

2.2. Учебно-тематический план

	Виды учебных занятий	Содержание
Модуль 1: Сбор требований и интервью с заказчиком		
Тема 1.1 Сбор требований: методы выявления неявных требований и когнитивные искажения в интервью	Теоретическое занятие - 0,5 часа	Методы сбора требований; разбор когнитивных искажений (проклятие знания, эффект подтверждения, эффект якоря); закон Хайрама и понятие API/системы как контракта.
	Практическое занятие - 1,5 часа	Анализ кода предыдущей реализации (Telegram-бот); подготовка вопросов к интервью, распределение ролей в команде.
Тема 1.2 Интервью с заказчиком и фиксация требований	Практическое занятие - 2,5 часа	Проведение интервью с заказчиком; фиксация требований в таблице «Сказал → Нужно → Риск»; формирование глоссария проекта.
Модуль 2: User stories и use cases		
Тема 2.1 User stories и use cases: формализация требований через основной и альтернативные сценарии	Теоретическое занятие - 0,5 часа	Формат «Как [роль], я хочу [действие], чтобы [цель]»; критерии качественной user story (INVEST); разбор примера «плохо → хорошо».
	Практическое занятие - 1,5 часа	Формулировка user stories команды по итогам интервью с заказчиком.

Тема 2.2 Use case: основной поток, альтернативы, исключения	Практическое занятие - 2,5 часа	Детализация use cases для ключевых сценариев (создание заявки, назначение, выполнение, отказ, переоткрытие, отчётность); проверка покрытия сценариев по ролям.
Модуль 3: Проектирование доменной модели		
Тема 3.1 Моделирование бизнес-процессов и проектирование доменной модели: агрегаты, инварианты и транзакционные границы	Теоретическое занятие - 0,5 часа	Понятие агрегата и корня агрегата; инварианты и транзакционные границы; принцип «Just Barely Good Enough» в объёме документации.
	Практическое занятие - 1,5 часа	Выделение агрегатов и инвариантов для своего проекта; построение ER-диаграммы.
Тема 3.2 State Diagram и промежуточное ревью с заказчиком	Практическое занятие - 2,5 часа	Построение State Diagram для жизненного цикла заявки; встреча с заказчиком — промежуточное ревью собранных требований и первой формализации.
Модуль 4: Проектирование API-контрактов и версионирование		
Тема 4.1 Проектирование API-контрактов: подходы и управление версиями	Теоретическое занятие - 0,5 часа	API-контракт как договор между командами; подходы к версионированию (URL, Header, Query param); статус-коды и единый формат ошибки.
	Практическое занятие - 1,5 часа	Первичная проработка эндпоинтов и схем данных на основе use cases и ER-диаграммы.

Тема 4.2 OpenAPI-спецификация v1 и v2	Практическое занятие - 2,5 часа	Написание OpenAPI v1 и v2; проверка обратной совместимости по чек-листу; описание стратегии миграции и сроков поддержки v1.
Модуль 5: Контроль качества, доработка и защита		
Тема 5.1 Критерии качества требований и передача артефактов между командами	Теоретическое занятие - 0,5 часа	Definition of Done / Quality Gate; чек-лист готовности артефактов по каждому типу документа; формат Given-When-Then для acceptance criteria.
	Практическое занятие - 1,5 часа	Самопроверка всех артефактов по чек-листу готовности.
Тема 5.2 Подготовка к защите	Практическое занятие - 2,0 часа	Встреча с ментором, репетиция презентации; доработка презентации и финальная проверка артефактов.
Итоговая аттестация	Защита проектов (2 часа)	

Текущий контроль – промежуточное ревью с заказчиком (день 3), ежедневная обратная связь ментора

Промежуточная аттестация - защита пакета артефактов перед бизнес-заказчиком (день 5)

2.3. Оценочные материалы

Текущий контроль (Quality Gate) проводится ежедневно ментором в форме проверки артефактов дня по чек-листу готовности; каждый тип артефакта проверяется по своим критериям:

- ТЗ — измеримые нефункциональные требования, описанные ограничения, критерии приёмки для каждого сценария.
- User stories и use cases — acceptance criteria у каждой истории (в формате Given-When-Then), охват всех ролей, альтернативные потоки и исключения.
- ER-диаграмма и State Diagram — выделенные агрегаты и инварианты, описанные транзакционные границы, полнота переходов состояний.
- OpenAPI-спецификация — примеры запросов и ответов, статус-коды (в том числе 409, 400, 500), единый формат ошибки, заложенное версионирование.

Правило проверки: артефакт проверяют минимум два человека — автор и другой участник (или ментор) — и ИИ-инструмент; все замечания фиксируются и устраняются до передачи дальше по цепочке.

Промежуточная аттестация — защита пакета артефактов перед бизнес-заказчиком (день 5, 5–7 минут на команду). Структура выступления: контекст → ключевые решения → артефакты → риски → итог. Обратную связь и оценку по итогам защиты дают три стороны: бизнес-заказчик (соответствие решения потребностям), ментор (качество командной работы и самостоятельность), консультант (полнота, непротиворечивость и реализуемость артефактов). Решение по итогам защиты: зачтено/незачтено

Критерии оценивания:

Для оценки «зачтено» необходимо в совокупности: представить полный пакет всех заявленных артефактов; строго соблюсти структуру выступления (контекст → решения → артефакты → риски → итог) и регламент (5–7 минут); обеспечить участие всех членов команды в защите; явно привязать решение к исходной бизнес-задаче заказчика; обеспечить непротиворечивость данных между артефактами (сроки, бюджет, архитектура); по каждому критическому риску предложить меру предотвращения или план Б; дать содержательные ответы на все уточняющие вопросы комиссии.

Оценка «незачтено» выставляется, если нарушено хотя бы одно из указанных требований, а также в случаях: отсутствие демонстрационного материала (прототип, расчеты, схема), грубое нарушение регламента или этики, плагиат, либо если по итогам защиты комиссия приняла решение «отправлено на доработку» (что означает концептуальную несостоятельность). При варианте «принято с доработками» в ведомость ставится «зачтено»

условно — при условии устранения замечаний в установленный срок; неисполнение этого условия влечёт пересмотр оценки на «незачтено».

3. Организационно-педагогические условия реализации программы

3.1. Материально-технические условия реализации программы

Для осуществления образовательного процесса по программе в очно-заочной форме (аудиторные занятия — очно по месту нахождения Заказчика, самостоятельная работа — с применением ЭО и ДОТ) необходима следующая материально-техническая база:

- аудитория, оборудованная рабочими местами для одновременной работы всех обучающихся, с возможностью групповой работы по командам;
- ноутбук или ПК на каждое рабочее место (минимально: 8 ГБ ОЗУ, 2 ядра, 50 ГБ свободного места, ОС Windows / macOS / Linux);
- стабильный доступ в интернет (Wi-Fi или Ethernet) — для работы с облачными сервисами и участия в созвонах без сбоя;
- проектор или экран для демонстрации лекций и презентаций;
- оборудованная электронная информационно-образовательная среда (для самостоятельной работы с применением ЭО и ДОТ), обеспечивающая доступ к учебным материалам и отслеживание прогресса.

3.2. Учебно-методическое и информационное обеспечение программы

Комплект методических материалов, разработанный для программы: шаблоны глоссария, таблицы требований, user story, use case, чек-листа обратной совместимости и финальной проверки артефактов, шаблон презентации для защиты, форма обратной связи от заказчика, краткое описание сквозной задачи (бриф), а также комплект из пяти лекционных презентаций по темам программы.

Для осуществления образовательного процесса по программе необходимо использование следующего программного обеспечения и информационных справочных систем:

- браузер (Chrome — основной вариант; альтернативы Firefox, Edge) — для работы с веб-интерфейсами (Miro, Swagger Editor, облачные документы);
- офисный пакет (MS Office — основной вариант; альтернативы LibreOffice, Google Docs) — для оформления ТЗ, глоссария и презентаций;
- таск-трекер (YouTrack, Planfix или локально развёрнутая Jira) — для постановки задач, распределения ответственности и трекинга прогресса;
- редактор для диаграмм Draw.io — для построения ER-диаграмм, State Diagram, use case диаграмм и контекстных схем;

- текстовый редактор (VS Code с плагинами для YAML и Markdown) — для написания OpenAPI-спецификаций и оформления документов;
- генератор OpenAPI (Swagger Editor онлайн) — для написания, визуализации и валидации OpenAPI-спецификаций;
- средство для презентаций (PowerPoint, Google Slides или Keynote) — для подготовки и демонстрации финальной защиты;
- Git (клиент командной строки или графическая оболочка) и учётная запись на российском Git-сервисе (GitVerse, GitFlic) — для версионирования кода и артефактов;
- среда разработки (IntelliJ IDEA, VS Code или Eclipse) — для анализа предоставленного кода легаси-реализации;
- облачное хранилище (Яндекс Диск — основной вариант; альтернативы Google Drive, Облако Mail.ru) — для хранения шаблонов, артефактов и итоговых документов;
- сервис видеосвязи (Яндекс Телемост — основной вариант; альтернативы Zoom, Google Meet, MS Teams) — для встреч с заказчиком и менторских сессий;
- ИИ-инструменты для генерации и проверки артефактов: DeepSeek (основной, бесплатный, стабильно работает в России), GigaChat (для работы с конфиденциальными данными заказчика), YandexGPT, Qwen (резервный вариант).

3.3. Литература

3.3.1. Интернет-ресурсы

- RFC 7231 (HTTP/1.1 Semantics and Content) — официальное описание семантики статус-кодов HTTP. <https://www.rfc-editor.org/rfc/rfc7231>
- Agile Alliance — Definition of Done: glossary-статья о критериях готовности в Agile/Scrum. <https://www.agilealliance.org/glossary/definition-of-done/>
- Martin Fowler — материалы по Domain-Driven Design (агрегаты, инварианты, ограниченные контексты). <https://martinfowler.com/tags/domain%20driven%20design.html>
- OpenAPI Initiative — официальная спецификация OpenAPI. <https://www.openapis.org/>

3.4. Кадровое обеспечение программы

№	Модуль/Дисциплина в соответствии с УП	ФИО преподавателя	Уч.степень/Звание	Должность
1	Модуль 1. Сбор требований и интервью с заказчиком	Исмаил Ужахов	--	Заведующий лабораторией робототехники АНО ДПО «АЦР»
2	Модуль 2. User stories и use cases	Исмаил Ужахов	--	Заведующий лабораторией робототехники АНО ДПО «АЦР»
3	Модуль 3. Проектирование доменной модели: ER-диаграмма и State Diagram	Магомед Беков	--	Support АНО ДПО «АЦР»
4	Модуль 4. Проектирование API-контрактов и версионирование	Амин Тутаев	--	Support АНО ДПО «АЦР»
5	Модуль 5. Контроль качества, доработка и защита	Магомед Беков	--	Support АНО ДПО «АЦР»