

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук  
Магистерская программа  
«Исследования и предпринимательство  
в искусственном интеллекте»

СОГЛАСОВАНО

Научный руководитель,  
доцент ФКН департамента  
программной инженерии, к.т.н.

\_\_\_\_\_ С. Л. Макаров  
«\_\_\_\_\_» \_\_\_\_\_ 2025 г.

УТВЕРЖДАЮ

Академический руководитель  
образовательной программы  
«Исследования и предпринимательство  
в искусственном интеллекте»

\_\_\_\_\_ Д. С. Лялин  
«\_\_\_\_\_» \_\_\_\_\_ 2025 г.

**Курсовая работа  
(проектная)**

на тему: **Разработка мультиагентной текстовой стратегической игры на  
основе оркестрируемых языковых моделей**

по направлению подготовки 01.04.02 «Прикладная математика и информатика»

ВЫПОЛНИЛ

студент группы ИПИИ  
образовательной программы  
01.04.02 «Прикладная  
математика и информатика»

\_\_\_\_\_ Н. С. Пеганов  
«\_\_\_\_\_» \_\_\_\_\_ 2025 г.

# Реферат

Работа посвящена разработке мультиагентной текстовой стратегической игры на основе оркестрируемых языковых моделей и исследованию применимости современных LLM в качестве автоматизированного мастера игры в жанре военно-политических игр (ВПИ).

В работе рассмотрен полный цикл создания подобной системы, включая проектирование архитектуры мультиагентной среды, оркестрацию языковых моделей для выполнения различных игровых функций, а также анализ результатов пилотного запуска с реальными пользователями. Особое внимание уделяется методам преодоления типичных проблем языковых моделей в контексте игрового процесса: галлюцинации, сохранение долгосрочного контекста и вычислительные ограничения.

Результаты исследования демонстрируют потенциал современных LLM для автоматизации роли вердера в текстовых стратегических играх, выявляют ключевые технические и игровые ограничения существующих подходов, а также предлагают набор технических решений для повышения качества игрового опыта, включая применение систем на основе RAG и локальных моделей для снижения стоимости эксплуатации игровой системы.

Данная работа состоит из 21 страницы, 5 глав, 5 листингов, 1 таблицы, 2 приложений. Использовано 23 источника.

**Ключевые слова:** мультиагентная система; языковые модели; оркестрация LLM; текстовые стратегические игры; искусственный интеллект в играх; RAG; автоматизация игрового мастера.

# Abstract

This paper is dedicated to developing a multi-agent text-based strategy game using orchestrated large language models, and investigating the applicability of modern LLMs as automated game masters in the military-political games (WPG) genre.

The work covers the complete cycle of creating such a system, including designing the architecture of a multi-agent environment, orchestrating language models to perform various game functions, and analyzing the results of a pilot launch with real users. Special attention is paid to methods for overcoming typical language model problems in the gaming context: hallucinations, maintaining long-term context, and computational limitations.

The research results demonstrate the potential of modern LLMs for automating the role of game arbiter in text-based strategy games, identify key technical and gameplay limitations of existing approaches, and propose a set of technical solutions to enhance gaming experience, including the application of RAG-based systems and local models to reduce the operational costs of the game system.

The paper contains 21 pages, 5 chapters, 5 listings, 1 table, 2 appendices. 23 sources are used.

**Keywords:** multi-agent system; large language models; LLM orchestration; text-based strategy games; artificial intelligence in games; RAG; automated game mastering.

# Содержание

|  |    |
|--|----|
| Реферат . . . . .                            | 2  |
| Abstract . . . . .                           | 3  |
| Используемые определения и термины . . . . . | 5  |
| Введение . . . . .                           | 7  |
| 0.1 Актуальность темы . . . . .              | 8  |
| 0.2 Цель и задачи исследования . . . . .     | 8  |
| 0.3 Объект и предмет исследования . . . . .  | 10 |
| 0.4 Методология исследования . . . . .       | 11 |
| 0.5 Практическая значимость . . . . .        | 12 |
| Глава 1 Обзор литературы . . . . .           | 13 |
| 1.1 Какая-то подглава . . . . .              | 13 |
| 1.1.1 Какая-то подподглава . . . . .         | 13 |
| 1.1.1.1 Какой-то параграф . . . . .          | 13 |
| 1.1.1.2 Какой-то параграф . . . . .          | 13 |
| 1.1.1.3 Какой-то параграф . . . . .          | 13 |
| 1.1.1.4 Какой-то параграф . . . . .          | 13 |
| 1.1.2 Какая-то подподглава . . . . .         | 13 |
| 1.1.2.1 Какой-то параграф . . . . .          | 13 |
| 1.1.2.2 Какой-то параграф . . . . .          | 13 |
| 1.1.2.3 Какой-то параграф . . . . .          | 13 |
| 1.1.2.4 Какой-то параграф . . . . .          | 14 |
| Выводы по главе . . . . .                    | 14 |
| Глава 2 Методология исследования . . . . .   | 15 |
| Глава 3 Первичный прототип системы . . . . . | 16 |
| Глава 4 Финальная версия системы . . . . .   | 17 |
| Глава 5 Обсуждение результатов . . . . .     | 18 |
| Заключение . . . . .                         | 19 |
| Список использованных источников . . . . .   | 21 |
| Приложение А . . . . .                       | 22 |
| Приложение Б . . . . .                       | 25 |

## Используемые определения и термины

**RAG (Retrieval-Augmented Generation)** – метод улучшения генерации текста языковыми моделями путем предварительного извлечения релевантной информации из внешних источников.

**Большая языковая модель (Large Language Model, LLM)** – языковая модель значительного размера, способная генерировать когерентный текст и выполнять различные языковые задачи.

**Верд (вердикт)** – текстовое описание результатов выполнения приказов игроков, составляемое вердером.

**Виртуальное государство** – поджанр ВПИ, концентрирующийся на отыгрыше политики в рамках одного государства с детализацией механик принятия государственных решений.

**Военно-политические игры (ВПИ, Military-Political Games, WPG)** – жанр текстовых стратегических игр, в которых игроки управляют государствами, политическими фракциями или организациями, взаимодействуя посредством письменных приказов и получая ответные вердикты от мастера игры.

**Галлюцинации LLM** – явление, при котором языковая модель генерирует фактически неверную информацию, представляя её как достоверную.

**Калькулятор** – принцип организации игровой механики, при котором взаимодействие с миром осуществляется через численные переменные и формулы, обеспечивающие определенный уровень автоматизации.

**Классическая ВПИ** – проект, где в рамках игровой механики предусмотрен отыгрыш правителя государства с возможностью действий во всех сферах государственной политики.

**Командно-штабная игра (КШИ)** – разновидность ВПИ с акцентом на военной составляющей, где игроки делятся на команды, отыгрывающие офицерский состав заранее определенных армий.

**Мультиагентная система** – система, состоящая из нескольких взаимодействующих интеллектуальных агентов, каждый из которых выполняет определенную функцию.

**Оркестрация языковых моделей** – процесс координации нескольких языковых моделей или компонентов для последовательного выполнения сложных задач.

**Приказ** – сформулированная в текстовой форме воля игрока, использующая находящиеся под контролем игрока силы для преобразования внешнего мира в рамках установленных игрой правил.

**Рестарт** – перезапуск проекта и смена игровой сессии.

**Ролеплей (РП)** – принцип организации игровой механики, при котором взаимодействие с игровой реальностью осуществляется через прямое текстовое взаимодействие игрока и судьи.

**Сессия** – игровой процесс, проходящий в рамках одной «игровой реальности», непрерывный процесс отыгрыша в определённой вселенной без удаления игроков и обнуления прогресса.

**Сеттинг** – совокупность особенностей среды, в рамках которой протекает игра, включая историю мира, технологический уровень, географию и культурные особенности.

**Судья (вердер)** – человек, наделенный полномочиями определять реакцию внешнего мира на действия игрока и описывать в текстовом формате итоги приказов, аналог гейм-мастера.

**Языковая модель (Language Model, LM)** – алгоритмическая система, обученная предсказывать и генерировать текст на естественном языке.

# Введение

Современные достижения в области искусственного интеллекта и, в частности, больших языковых моделей (LLM) открывают новые перспективы для традиционных интерактивных развлечений [1; 2]. Одной из областей, где применение искусственного интеллекта имеет значительный потенциал, являются текстовые стратегические игры, такие как военно-политические игры (ВПИ) — жанр, сочетающий элементы стратегии, ролевой игры и коллективного сторителлинга [3].

ВПИ представляют собой текстовые игры, в которых игроки управляют сложными структурами (государствами, военными силами, политическими организациями) путем написания приказов, а судья (или вердер) оценивает эти приказы и формирует вердикты — текстовые описания результатов действий [4]. Этот процесс требует от судьи глубокого понимания игрового мира, механик, а также способности генерировать связные и логичные повествования, что делает эту роль одной из самых трудоемких в организации игры. Кроме того, традиционно судья ограничен в скорости обработки приказов, что создает естественный «потолок» для темпа игры и количества участников.

Большие языковые модели, такие как GPT-4, демонстрируют впечатляющие способности к пониманию контекста, следованию инструкциям и генерации связных текстов [5]. Эти характеристики потенциально позволяют им выполнять роль судьи в ВПИ, автоматизируя процесс создания вердиктов и значительно ускоряя игровой процесс. Однако использование LLM в таком качестве сопряжено с рядом технических и методологических вызовов, включая проблему галлюцинаций, ограничения контекстного окна и сложности в поддержании долговременной согласованности [6; 7].

В данной работе представлена разработка мультиагентной текстовой стратегической игры на основе оркестрируемых языковых моделей — системы, использующей несколько специализированных LLM-агентов для выполнения различных функций судьи в ВПИ. Исследование включает как теоретическое обоснование подхода, так и практическую реализацию в виде работающего прототипа, протестированного реальными игроками. Особое внимание уделяется механизмам обеспечения целостности игрового мира, преодоления ограничений LLM и создания интуитивно понятного пользовательского опыта.

Работа основывается на междисциплинарном подходе, объединяющем методы искусственного интеллекта, игрового дизайна и нарративных исследований [8; 9]. Представленная система не только демонстрирует практическое применение современных LLM в новой предметной области, но и открывает перспективы для создания более масштабных и динамичных текстовых игр, доступных широкой аудитории.

## 0.1. Актуальность темы

Текстовые стратегические игры жанра военно-политических игр (ВПИ) занимают особую нишу в игровой индустрии, предоставляя уникальный опыт коллективного стратегического взаимодействия. Ключевым ограничением данного жанра является высокая зависимость от человека-вердера (судьи), который обрабатывает игровые ситуации и формирует нарративную основу игры [10]. Это делает организацию подобных игр трудоемкой, снижает их доступность и ограничивает масштабы игрового сообщества [10].

Современное развитие больших языковых моделей (LLM) создает предпосылки для решения данной проблемы [11]. Недавние исследования показывают, что автоматизация создания интерактивного контента с помощью ИИ может существенно изменить подход к разработке игр с текстовой основой [12]. Особую актуальность приобретает архитектурный подход на основе мультиагентных систем, где различные аспекты игрового взаимодействия обрабатываются специализированными ИИ-агентами [13]. Подобная оркестрация позволяет преодолеть ограничения единичных моделей через разделение обязанностей и специализацию [13]. В случае ВПИ это особенно важно, поскольку игра требует компетенций в различных областях: политике, экономике, военном деле, дипломатии [10].

Разработка мультиагентной системы для ВПИ также имеет значение в контексте растущего интереса к цифровой гуманитаристике и инструментам совместного повествования [14]. Исследования показывают, что комбинация человеческого творчества и ИИ-ассистирования открывает новые горизонты для коллективного творчества и обмена идеями [15].

С практической точки зрения, создание автоматизированной системы для проведения ВПИ может возродить интерес к жанру, сделать его доступным для более широкой аудитории и заложить основу для новых форм социального взаимодействия в цифровых средах [12]. В научном плане проект представляет ценность как исследование применимости мультиагентных систем для поддержания сложных нарративных структур и последовательных игровых вселенных [16].

Таким образом, разработка мультиагентной системы на основе оркестрируемых языковых моделей для автоматизации ВПИ представляет собой актуальную задачу, решение которой способно обогатить как теорию искусственного интеллекта, так и практику игрового дизайна.

## 0.2. Цель и задачи исследования

Основная цель исследования заключается в разработке и оценке мультиагентной системы на основе оркестрируемых языковых моделей для автоматизации роли вердера в текстовых стратегических играх жанра ВПИ, а также в определении эффективности и практической применимости такого подхода для создания интересного и последова-



тельного игрового опыта.

Для достижения поставленной цели сформулированы следующие задачи:

- 1) Провести анализ существующих подходов к организации военно-политических игр и выявить ключевые аспекты, требующие автоматизации.
- 2) Провести customer development с опытными игроками ВПИ для выявления их ожиданий, болевых точек и требований к автоматизированной системе проведения игр.
- 3) Разработать архитектуру мультиагентной системы, включающую специализированных ИИ-агентов для различных аспектов игрового процесса (обработка приказов, проверка на соответствие эпохе, оценка экономических показателей, формирование вердиктов).
- 4) Реализовать первичный прототип системы (RELOAD WPG) на основе единой языковой модели и провести его тестирование с реальными игроками для выявления ограничений и потенциальных улучшений.
- 5) На основе полученной обратной связи спроектировать и реализовать усовершенствованную версию системы с применением:
  - Локальных языковых моделей для снижения стоимости эксплуатации
  - RAG-системы для минимизации галлюцинаций и точного доступа к информации
  - Оптимизированных механик взаимодействия для улучшения игрового опыта
  - Специализированной системы симуляции боевых действий
- 6) Провести сравнительный анализ первичного прототипа и усовершенствованной системы по критериям:
  - Качество генерируемых вердиктов
  - Устойчивость к галлюцинациям
  - Способность поддерживать долгосрочную согласованность игрового мира
  - Удовлетворенность пользователей
  - Вычислительная эффективность
- 7) Определить принципиальные ограничения и возможности применения оркестрируемых языковых моделей в контексте автоматизации текстовых стратегических игр.
- 8) Сформулировать рекомендации для дальнейшего развития ИИ-ассистированных текстовых игр на основе полученных результатов.

Решение данных задач позволит не только создать функциональную систему для проведения военно-политических игр с минимальным участием человека-вердера, но и внести вклад в понимание того, как мультиагентные системы на основе языковых моделей могут применяться для создания сложных интерактивных нарративных сред.

### 0.3. Объект и предмет исследования

**Объектом исследования** являются текстовые стратегические игры жанра военно-политических игр (ВПИ), в которых игроки управляют государствами или иными сложными структурами посредством текстовых приказов, обрабатываемых и интерпретируемых вердером (судьей).

**Предметом исследования** выступает процесс автоматизации роли вердера с помощью мультиагентной системы на основе оркестрируемых языковых моделей, включая:

- Методы организации взаимодействия между игроками и ИИ-вердером в текстовом формате
- Архитектурные решения для создания мультиагентной системы, способной эффективно обрабатывать и интерпретировать игровые приказы
- Способы оркестрации различных языковых моделей для выполнения специализированных функций в контексте игрового процесса
- Методы преодоления ограничений языковых моделей (галлюцинации, поддержание долгосрочного контекста, согласованность генерируемого контента) для создания качественного игрового опыта
- Принципы проектирования пользовательского интерфейса и взаимодействия с ИИ-вердером, обеспечивающие максимальную доступность и понимание игрового процесса
- Критерии оценки эффективности и качества автоматизированной системы проведения ВПИ с точки зрения игрового опыта и технической реализации

Исследование фокусируется на изучении баланса между творческими аспектами генерации контента языковыми моделями и необходимостью сохранения игровой логики, исторической или жанровой достоверности и общей связности игрового мира. Особое внимание уделяется выявлению оптимальных подходов к декомпозиции задач вердера между различными агентами мультиагентной системы для достижения максимальной эффективности и качества игрового процесса.

## 0.4. Методология исследования

В основу методологии исследования положен комплексный подход, сочетающий методы программной инженерии, искусственного интеллекта и пользовательского дизайна. Исследование разделено на несколько взаимосвязанных этапов, каждый из которых имеет свою методологическую основу.

На подготовительном этапе применяются методы анализа предметной области и customer development для выявления ключевых требований и ограничений в контексте автоматизации ВПИ. Используется метод экспертных интервью с опытными игроками и организаторами ВПИ, а также анализ существующих проектов в данной области. Для структурирования полученной информации применяется методология Jobs-to-be-Done [17], позволяющая выявить основные потребности пользователей и сформулировать критерии успешности системы.

При разработке архитектуры мультиагентной системы применяется методология итеративного проектирования и прототипирования, опирающаяся на принципы микросервисной архитектуры [18]. Для проектирования взаимодействия между агентами используется подход, основанный на исследованиях в области оркестрации языковых моделей [19], с учетом специфики задачи автоматизации роли вердера.

Для оценки эффективности языковых моделей в контексте генерации игрового контента применяются методы, заимствованные из области оценки генеративных систем [20], включая как количественные метрики (согласованность, соответствие тематике, информационная точность), так и качественные критерии (нарративная убедительность, игровая ценность).

В процессе разработки прототипа и его усовершенствованной версии используется методология Agile с короткими итерациями и постоянным взаимодействием с конечными пользователями [21]. Это позволяет оперативно вносить изменения в архитектуру и функциональность системы на основе получаемой обратной связи.

Для преодоления ограничений языковых моделей, в частности проблемы галлюцинаций, применяется методология RAG (Retrieval-Augmented Generation) [22], адаптированная к специфике игрового контекста с динамически меняющейся информационной базой.

Тестирование системы проводится с использованием методов качественной оценки пользовательского опыта [23], включая структурированные опросы, интервью и наблюдение за игровыми сессиями. Для количественной оценки применяются метрики вычислительной эффективности, точности и согласованности генерируемого контента.

Таким образом, методология исследования представляет собой комплексный подход, сочетающий теоретические и эмпирические методы, что обеспечивает всестороннее изучение проблемы автоматизации ВПИ с помощью мультиагентных систем на основе языковых моделей.

## 0.5. Практическая значимость

Практическая значимость данного исследования определяется несколькими аспектами.

Во-первых, разработанная мультиагентная система на основе оркестрируемых языковых моделей может быть непосредственно использована для проведения военно-политических игр, что позволит существенно снизить трудозатраты на их организацию и сделает формат более доступным для широкого круга игроков. Это особенно актуально для сообщества ВПИ, которое сталкивается с проблемой ограниченности человеческих ресурсов для роли судей [3].

Во-вторых, предложенные архитектурные решения по организации мультиагентной системы могут быть адаптированы для автоматизации других типов текстовых игр и интерактивных нарративных сред, где требуется сочетание следования правилам с творческой генерацией контента. Подход к оркестрации языковых моделей, разработанный в рамках исследования, может найти применение в образовательных симуляциях, тренировочных системах и других областях, где необходима интеллектуальная обработка текстовых запросов.

В-третьих, методы преодоления ограничений языковых моделей, разработанные в процессе исследования (в частности, применение RAG для снижения галлюцинаций в динамическом контексте), имеют самостоятельную практическую ценность и могут использоваться в различных приложениях, требующих поддержания согласованности генерируемого контента на протяжении длительных интеракций.

Наконец, разработанный в рамках исследования инструментарий для проведения ВПИ с ИИ-вердером может способствовать популяризации этого жанра игр и стимулировать появление новых подходов к интеграции искусственного интеллекта в игровую индустрию. Это открывает перспективы для коммерциализации подобных решений и создания новых ниш на рынке интерактивных развлечений.

# Глава 1. Обзор литературы

Текст главы 1

## 1.1. Какая-то подглава

Текст подглавы

### 1.1.1. Какая-то подподглава

Текст подподглавы

#### 1.1.1.1. Какой-то параграф

Текст параграфа

#### 1.1.1.2. Какой-то параграф

Текст параграфа

#### 1.1.1.3. Какой-то параграф

Текст параграфа

#### 1.1.1.4. Какой-то параграф

Текст параграфа

### 1.1.2. Какая-то подподглава

Текст подподглавы

#### 1.1.2.1. Какой-то параграф

Текст параграфа

#### 1.1.2.2. Какой-то параграф

Текст параграфа

#### 1.1.2.3. Какой-то параграф

Текст параграфа

#### **1.1.2.4. Какой-то параграф**

Текст параграфа

#### **Выводы по главе**

Текст Текст Текст Текст Текст Текст Текст

## Глава 2. Методология исследования

Текст главы 2

## Глава 3. Первичный прототип системы



## Глава 4. Финальная версия системы

## Глава 5. Обсуждение результатов

## Заключение

Текст заключения

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Language Models are Few-Shot Learners / Т. В. Brown [и др.] // Advances in Neural Information Processing Systems. — 2020. — Т. 33. — С. 1877—1901.
2. Training language models to follow instructions with human feedback / L. Ouyang [и др.] // Advances in Neural Information Processing Systems. — 2022. — Т. 35. — С. 27730—27744.
3. Каталог ВПИ [Электронный ресурс] : Сообщество военно-политических игр. — URL: <https://vk.com/catalogwpg> (дата обр. 15.10.2023).
4. Что такое военно-политические игры (ВПИ)? [Электронный ресурс] : Введение в жанр текстовых стратегических игр. — URL: <https://dtf.ru/id417564/853668-cto-takoe-voenno-politicheskie-igry-vpi> (дата обр. 04.05.2025).
5. *OpenAI*. GPT-4 Technical Report : тех. отч. / OpenAI. — 2023.
6. Sparks of Artificial General Intelligence: Early experiments with GPT-4 / S. Bubeck [и др.] // arXiv preprint arXiv:2303.12712. — 2023.
7. *Liu S., Ogren P., Peng N.* Evaluating the Factual Consistency of Large Language Models Through News Summarization // Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. — 2023. — С. 10574—10587.
8. Wordcraft: Story Writing With Large Language Models / A. Yuan [и др.] // 27th International Conference on Intelligent User Interfaces. — ACM. 2022. — С. 841—852.
9. Generative Agents: Interactive Simulacra of Human Behavior / J. S. Park [и др.] // Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology. — 2023.
10. Что такое военно-политические игры (ВПИ)? [Электронный ресурс] : Каталог ВПИ. — URL: <https://dtf.ru/u/417564-katalog-vpi/853668-cto-takoe-voenno-politicheskie-igry-vpi> (дата обр. 04.05.2025).
11. Как работают большие языковые модели? [Электронный ресурс] : Cloud.ru. — URL: [https://cloud.ru/docs/aicloud/mlspace/concepts/tutorials/llm/tutorials\\_llm\\_how\\_it\\_works.html](https://cloud.ru/docs/aicloud/mlspace/concepts/tutorials/llm/tutorials_llm_how_it_works.html) (дата обр. 04.05.2025).
12. 10 лучших генераторов игр с искусственным интеллектом [Электронный ресурс] : Unite.AI. — URL: <https://www.unite.ai/ru/best-ai-game-generators/> (дата обр. 04.05.2025).
13. Многоагентная система [Электронный ресурс] : Википедия. — URL: [https://ru.wikipedia.org/wiki/%D0%9C%D0%BD%D0%BE%D0%B3%D0%BE%D0%B0%D0%B3%D0%B5%D0%BD%D1%82%D0%BD%D0%B0%D1%8F\\_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0](https://ru.wikipedia.org/wiki/%D0%9C%D0%BD%D0%BE%D0%B3%D0%BE%D0%B0%D0%B3%D0%B5%D0%BD%D1%82%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) (дата обр. 04.05.2025).

14. Цифровые гуманитарные науки [Электронный ресурс] : Википедия. — URL: [https://ru.wikipedia.org/wiki/%D0%A6%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D1%8B%D0%B5\\_%D0%B3%D1%83%D0%BC%D0%B0%D0%BD%D0%B8%D1%82%D0%B0%D1%80%D0%BD%D1%8B%D0%B5\\_%D0%BD%D0%B0%D1%83%D0%BA%D0%B8](https://ru.wikipedia.org/wiki/%D0%A6%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D1%8B%D0%B5_%D0%B3%D1%83%D0%BC%D0%B0%D0%BD%D0%B8%D1%82%D0%B0%D1%80%D0%BD%D1%8B%D0%B5_%D0%BD%D0%B0%D1%83%D0%BA%D0%B8) (дата обр. 04.05.2025).
15. Искусственный интеллект в искусстве [Электронный ресурс] : Википедия. — URL: [https://ru.wikipedia.org/wiki/%D0%98%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9\\_%D0%B8%D0%BD%D1%82%D0%B5%D0%BB%D0%BB%D0%B5%D0%BA%D1%82\\_%D0%B2\\_%D0%B8%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%B5](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D0%BB%D0%BB%D0%B5%D0%BA%D1%82_%D0%B2_%D0%B8%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%B5) (дата обр. 04.05.2025).
16. Мультиагентные системы в построении виртуальных пространств [Электронный ресурс] : Хабр. — URL: <https://habr.com/ru/companies/microsoft/articles/419129/> (дата обр. 04.05.2025).
17. Competing Against Luck: The Story of Innovation and Customer Choice / С. М. Christensen [и др.]. — New York : HarperBusiness, 2016.
18. *Newman S.* Building Microservices: Designing Fine-Grained Systems. — 2-е изд. — Sebastopol, CA : O'Reilly Media, 2021.
19. HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face / S. Shen [и др.] // arXiv preprint arXiv:2303.17580. — 2023.
20. Evaluation Methods for Large Language Models: A Systematic Survey / Y. Zhou [и др.] // arXiv preprint arXiv:2307.03109. — 2023.
21. *Martin R. C.* Clean Agile: Back to Basics. — Upper Saddle River, NJ : Pearson Education, 2019.
22. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks / P. Lewis [и др.] // Advances in Neural Information Processing Systems. Т. 33. — 2020. — С. 9459—9474.
23. *Albert W., Tullis T.* Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics. — 2-е изд. — Burlington, MA : Morgan Kaufmann, 2013.

## Пример приложения

Пример приложения. Какой-то текст. Какой-то текст. Какой-то текст. Какой-то текст. Какой-то текст. Какой-то текст. Какой-то текст. Какой-то текст. Какой-то текст. Какой-то текст.

Ссылка на приложение Б.

Тут ссылка на листинг 1.

А тут ссылка на листинг 3.

```

1 | @Deprecated("Reason")
2 | fun findScriptDefinition(project: Project, script: SourceCode): ScriptDefinition?
   | {
3 |     val scriptDefinitionProvider = ScriptDefinitionProvider.getInstance(project) ?:
   |         return null
4 |     ?: throw IllegalStateException("Unable to get script definition: ...")
5 |
6 |     return scriptDefinitionProvider.findDefinition(script) ?:
   |         scriptDefinitionProvider.getDefaultDefinition() // Comment
7 | }

```

Листинг 1 — Пример какого-то кода на Kotlin

```

1 | class Main {
2 |     public static ScriptDefinition findScriptDefinition(Project project, SourceCode
   |         script) {
3 |         ScriptDefinitionProvider scriptDefinitionProvider = ScriptDefinitionProvider.
   |             getInstance(project);
4 |         if (scriptDefinitionProvider == null) {
5 |             if (null == null) {
6 |                 throw IllegalStateException("Unable to get script definition: ...");
7 |             } else {
8 |                 return null;
9 |             }
10 |        }
11 |
12 |        ScriptDefinition definition = scriptDefinitionProvider.findDefinition(script);
13 |        if (definition == null) {
14 |            return scriptDefinitionProvider.getDefaultDefinition(); // Comment
15 |        } else {
16 |            return definition;
17 |        }
18 |    }
19 | }

```

Листинг 2 — Пример какого-то кода на Java

```

...
13 aload_2
14 dup
15 ifnonnull 28
18 new #17 // NullPointerException
21 dup
22 ldc #19 // String null cannot be cast to non-null String
24 invokespecial #23 // NullPointerException."<init>"(String)
27 athrow

...
46 aload_2
47 dup
48 ifnonnull 61
51 new #17 // NullPointerException
54 dup
55 ldc #19 // String null cannot be cast to non-null String
57 invokespecial #23 // NullPointerException."<init>"(String)
60 athrow

...

```

Листинг 3 — Пример JVM-байткода

```

...
13: aload_2
14: dup
15: ifnonnull 28
18: new #17 // NullPointerException
21: dup
22: ldc #19 // String null cannot be cast to non-null String
24: invokespecial #23 // NullPointerException."<init>"(String)
27: athrow

...
46: aload_2
47: dup
48: ifnonnull 61
51: new #17 // NullPointerException
54: dup
55: ldc #19 // String null cannot be cast to non-null String
57: invokespecial #23 // NullPointerException."<init>"(String)
60: athrow

...

```

Листинг 4 — Пример JVM-байткода 2

А тут ссылка на таблицу 1.

Таблица 1 — Пример таблицы

| Col1 | Col2 | Col2  | Col3 |
|------|------|-------|------|
| 1    | 6    | 87837 | 787  |
| 2    | 7    | 78    | 5415 |
| 3    | 545  | 778   | 7507 |
| 4    | 545  | 18744 | 7560 |
| 5    | 88   | 788   | 6344 |

```
1 | procedure RUN(packages, hashes)
2 |   queue[svace.parallel_max]
3 |   for item ∈ zip(packages, hashes)
4 |     ps = create(item)
5 |     if !queue.full()
6 |       queue.put(ps)
7 |     else
8 |       first = queue.get()
9 |       first.wait()
10 |    end if
11 |  end for
12 | end procedure
```

Листинг 5 — Пример псевдокода на алгоритмическом языке



## Ещё один пример приложения

Пример приложения