

# ІТМО

**А.В. Духанов, А.А. Лаушкина**

## **ОСНОВЫ ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ С ПРИМЕНЕНИЕМ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ**



**Санкт-Петербург  
2025**

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

УНИВЕРСИТЕТ ИТМО

**А.В. Духанов, А.А. Лаушкина**  
**ОСНОВЫ ПРОЕКТИРОВАНИЯ**  
**ИНФОРМАЦИОННЫХ СИСТЕМ С**  
**ПРИМЕНЕНИЕМ БОЛЬШИХ ЯЗЫКОВЫХ**  
**МОДЕЛЕЙ**

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ

РЕКОМЕНДОВАНО К ИСПОЛЬЗОВАНИЮ В УНИВЕРСИТЕТЕ ИТМО  
по направлению подготовки 09.03.02, 27.03.04, 27.03.05, 38.03.05, 45.03.04  
в качестве Учебно-методического пособия для реализации основных  
профессиональных образовательных программ высшего образования  
бакалавриата

**ИТМО**

Санкт-Петербург  
2025

Духанов А.В., Лаушкина А.А., Основы проектирования информационных систем с применением больших языковых моделей– СПб: Университет ИТМО, 2025. – 50 с.

Рецензент(ы):

Чистякова Тамара Балабековна, доктор технических наук, профессор, заведующий кафедрой систем автоматизированного проектирования и управления федерального государственного бюджетного образовательного учреждения высшего образования «Санкт-Петербургский государственный технологический институт (технический университет)», федеральное государственное бюджетное образовательное учреждение высшего образования «Санкт-Петербургский государственный технологический институт»;

Учебно-методическое пособие предназначено для формирования представлений и навыков применения классических и передовых инструментов для проектирования информационных систем. В составе практических работ рассматриваются вопросы системного анализа, архитектур информационных системы, подготовки технических заданий, а также разнообразных видов графических инструментов. Ряд работ предполагают использование больших языковых моделей для генерации скриптов, позволяющих описывать структуру и функции информационных систем с применением нотаций популярных графических языков. Материал подготовлен так, что он может эффективно использоваться при обучении студентов технических и экономических направлений, а также школьников старших классов, имеющих общее представление о структуре и составе информационных систем.

The logo of ITMO University, consisting of the letters 'ITMO' in a bold, black, sans-serif font. The 'I' and 'T' are connected, and the 'O' is a solid circle.

ИТМО (Санкт-Петербург) — национальный исследовательский университет, научно-образовательная корпорация. Альма-матер победителей международных соревнований по программированию. Приоритетные направления: IT и искусственный интеллект, фотоника, робототехника, квантовые коммуникации, трансляционная медицина, Life Sciences, Art&Science, Science Communication.

Лидер федеральной программы «Приоритет-2030», в рамках которой реализуется программа «Университет открытого кода». С 2022 ИТМО работает в рамках новой модели развития — научно-образовательной корпорации. В ее основе академическая свобода, поддержка начинаний студентов и сотрудников, распределенная система управления, приверженность открытому коду, бизнес-подходы к организации работы. Образование в университете основано на выборе индивидуальной траектории для каждого студента.

ИТМО пять лет подряд — в сотне лучших в области Automation & Control (кибернетика) Шанхайского рейтинга. По версии SuperJob занимает первое место в Петербурге и второе в России по уровню зарплат выпускников в сфере IT. Университет в топе международных рейтингов среди российских вузов. Входит в топ-5 российских университетов по качеству приема на бюджетные места. Рекордсмен по поступлению олимпиадников в Петербурге. С 2019 года ИТМО самостоятельно присуждает ученые степени кандидата и доктора наук.

© Университет ИТМО, 2025

© Духанов А.В., Лаушкина А.А., 2025

## Содержание

Введение.....	4
Ресурсный анализ систем .....	5
Идентификация архитектуры информационной системы в соответствии с названием и кратким описанием .....	12
Разработка технического задания на создание информационной системы.....	20
Разработка графической модели процессов с применением стандарта IDEF0 .....	26
Разработка структурной UML-диаграммы.....	34
Разработка поведенческой UML-диаграммы.....	40
Заключение .....	49
Список литературы .....	50

## Введение

Представляем вашему вниманию системные и графические инструменты для проектирования информационных систем (ИС). Они позволяют проанализировать производственные или бытовые процессы и подготовить проект информационной системы или более сложного решения для их частичной или полной автоматизации. Процессы и соответствующие информационные системы предлагается рассматривать с разных сторон: их структуры, функции и ресурсы. Для достижения соответствующих целей в указаниях предлагается провести системно-ресурсный анализ объектов и процессов автоматизации, строить модели информационных систем (включая поведенческие и структурные с помощью UML-языков), а также решать задачи в рамках подготовки технических заданий для их разработки.

В учебно-методическом пособии предложены шесть практических работ:

1. Ресурсный анализ систем.
2. Идентификация архитектуры информационной системы.
3. Разработка технического задания информационной системы.
4. Разработка графической модели процессов с применением стандарта IDEF0.
5. Разработка структурной UML-диаграммы.
6. Разработка поведенческой UML-диаграммы.

В дополнение к традиционным инструментам, в части работ предлагается использовать большие языковые модели (БЯМ), в том числе для генерации скриптов UML-диаграмм с применением сервиса PlantUML. В таких работах приведены инструкции для формирования необходимых запросов (промттов для получения желаемого результата). Также указываются особенности и ограничения в применении БЯМ, включая эффекты «галлюцинаций» (генерация недостоверной информации) или задание контекста.

Данное учебно-методическое пособие формирует основу для развития навыков в решении задач проектирования, реализации и внедрения информационных систем, в том числе с применением технологий искусственного интеллекта. Оно будет полезно при изучении дисциплины «Проектирование информационных систем» образовательных программ бакалавриата «Мобильные и сетевые технологии», «Интеллектуальные системы в гуманитарной сфере». Также данное пособие может быть полезно в освоении дисциплин «Основы процессов внедрения информационных систем», «Интеллектуальные технические системы» в рамках программ бакалавриата в области инноватики и технологического менеджмента, в том числе «Инноватика», «Бизнес-информатика», «Управление технологическими инновациями».

## Ресурсный анализ систем

**Цель работы.** Освоить навыки аналитического разбора систем, ее структуры и содержания с применением принципа базовых видов ресурсов.

**Практическое задание.** Пожалуйста, выберите процесс из числа приведенных вариантов (см. ниже) и проведите его ресурсный анализ с применением принципа базовых видов ресурсов. Результат отразите в форме таблицы: по вертикали – роли или компоненты системы (не путать с ресурсами), по горизонтали – шаги процесса (см. ниже). В полученных внутренних ячейках таблицы, пожалуйста, приведите преобразования ресурсов. К примеру, когда студент (роль) проходит от дома к остановке (этап), то он расходует энергию (E) и время (T), использует информацию о маршруте (I), возможность его прохождения (Adm), свои компетентностные возможности как человек (C), чтобы переместить себя, одежду и рюкзак с содержимым (M) из дома к остановке (Env).

Процесс должен состоять не менее чем из трех шагов, а соответствующая система содержать не менее двух компонент или роли.

**Исходные данные:** наименование процесса и ключевые слова.

**Требуемый результат:** таблица, описывающая ресурсы базовых видов, которые имеют место для каждого шага процесса, каждой роли, участвующей в процессе.

Таблица 1.1 – Пример оформления таблицы, описывающей ресурсы базовых видов.

Роли \ Шаги	Вызов такси (заказ)	Подача такси к пассажиру	Поездка до места назначения
Пассажир	М: смартфон, пассажир (его тело). E: смартфон расходует электрическую энергию, пассажир расходует биологическую и тепловую энергию. Env: пассажир, формируя заказ в приложении смартфона, находится в некотором	М: смартфон, пассажир (его тело). E: во время ожидания автомобиля смартфон расходует электрическую энергию, пассажир расходует биологическую и тепловую энергию. Env: пассажир ожидает автомобиль в некотором пространстве,	М: смартфон, пассажир (его тело). E: во время поездки смартфон расходует электрическую энергию, пассажир расходует биологическую и тепловую энергию. Env: во время поездки пассажир находится в пространстве автомобиля (в пассажирском

Роли \ Шаги	Вызов такси (заказ)	Подача такси к пассажиру	Поездка до места назначения
	<p>пространстве, например, в кофе.  Т: пассажир тратит время на формирование заказа.  I: приложение на смартфоне, адрес подачи и адрес назначения.  Adm: доступность сервиса для пассажира.  С: пассажир обладает компетенцией в применении приложения на смартфоне.</p>	<p>например, около парковки кафе.  Т: на ожидание автомобиля расходуется время.  I: приложение на смартфоне, сведения о месте нахождения подаваемого автомобиля.  Adm: доступность сервиса для пассажира, включая возможность отслеживания подаваемого автомобиля;  С: пассажир обладает компетенцией в применении приложения на смартфоне.</p>	<p>кресле).  Т: на совершение поездки расходуется время.  I: приложение на смартфоне, сведения о текущей поездке.  Adm: доступность сервиса для пассажира, включая возможность получать сведения о текущей поездке;  С: пассажир обладает компетенцией в применении приложения на смартфоне и поведения в автомобиле во время поездки.</p>
Водитель	<p>М: смартфон, водитель (его тело).  Е: смартфон расходует электрическую энергию, водитель расходует биологическую и тепловую энергию.  Env: водитель находится в пространстве автомобиля.  Т: водитель тратит время на открытие и изучение заказа,</p>	<p>М: смартфон, водитель (его тело).  Е: во время подачи автомобиля смартфон расходует электрическую энергию, водитель расходует биологическую и тепловую энергию.  Env: водитель находится в пространстве своего автомобиля (в водительском кресле).</p>	<p>М: смартфон, водитель (его тело).  Е: во время поездки смартфон расходует электрическую энергию, водитель расходует биологическую и тепловую энергию.  Env: во время поездки водитель находится в пространстве автомобиля (в водительском кресле).</p>

<b>Роли \ Шаги</b>	<b>Вызов такси (заказ)</b>	<b>Подача такси к пассажиру</b>	<b>Поездка до места назначения</b>
	<p>отображаемого на смартфоне.</p> <p>I: приложение на смартфоне, адрес подачи и адрес назначения.</p> <p>Adm: доступность сервиса для водителя.</p> <p>C: водитель обладает компетенцией в применении приложения на смартфоне.</p>	<p>T: на подачу автомобиля расходуется время.</p> <p>I: приложение на смартфоне, сведения о месте нахождения автомобиля, сведения о статусе заказа и пассажира.</p> <p>Adm: доступность сервиса для водителя, включая возможность отслеживания статуса пассажира;</p> <p>C: водитель обладает компетенциями в управлении автомобилем и применении приложения на смартфоне.</p>	<p>T: на совершение поездки расходуется время.</p> <p>I: приложение на смартфоне, сведения о текущей поездке.</p> <p>Adm: право управлением транспортным средством, доступность сервиса для водителя, включая возможность получать сведения о текущей поездке (в том числе навигационные данные);</p> <p>C: водитель обладает компетенциями в управлении автомобилем и применении приложения на смартфоне.</p>

### **Теоретические вводные и описание способов решения**

Принцип базовых видов ресурсов позволяет рассматривать содержимое системы как совокупность базовых видов ресурсов. В части случаев компоненты системы или их составляющие могут относиться к базовым видам ресурсов, но в большинстве случаев представляют собой комбинацию ресурсов. Приведем базовые виды ресурсов [1].

1. М – материальный ресурс или вещество. Из вещества и материи состоят объекты, которые можно потрогать, собирать/разбирать в реальном пространстве. Вещество может выступать как мера однородности материи. Примеры: деревянный брусок, металлический уголок, вода в сосуде.



2. E – энергетический ресурс. Энергия отражает изменчивость материи, ее переходы из одного вида в другой. Примеры: тепло, электричество, потенциальная энергия, кинетическая энергия.

3. C – человеческий, компетентностный (competency), интеллектуальный ресурс. Человек как носитель компетенций является социальным и профессиональным ресурсом и в решении задач и проблем может выступать как индивидуальный актер или часть команды. В ряде случаев в качестве компетентностного ресурса можно рассматривать интеллектуальные системы (в том числе системы искусственного интеллекта), включая беспилотные средства доставки, системы поддержки принятия решений.

4. Env – пространственный ресурс. Пространство является местом, где объекты не только размещаются, но и могут быть измерены по своим сторонам (для трехмерного пространства – длина, ширина, высота). Примеры: пространство аудитории (можно рассматривать как площадь, так и трехмерное пространство, ограниченное стенами аудитории); пространство салона самолета.

5. T – временной ресурс. Время выступает как мера, в которой проходят изменения, процессы. От скорости, быстроты процесса зависит время, которое необходимо затратить на его реализацию. Очень ценный ресурс. В философии время – необратимое течение (протекает в одном направлении).

6. Adm – ресурс влияния или административный ресурс, определяющий возможности решения задачи, доступ к сервису, разрешение на какое-либо действие. В частности, когда студент посещает лекцию, то в качестве административного ресурса может быть право посетить соответствующую аудиторию в указанное по расписанию время, задать вопросы преподавателю в рамках соответствующей сессии.

7. I – информационный ресурс. Информация выступает как отражение порядка, структурированности и самоорганизации материи, объектов, социума (совокупности людей и иных ресурсов, связанных с ними). Примеры: расписание занятий, рекомендации научного руководителя, отчет о выполнении практической работы, табличное представление данных.

Рассмотрим ресурсный анализ на примере приобретения компьютерной мыши в некотором магазине.

M – мышь, оборудование магазина, денежные средства (если они электронные, то рассматривается виртуальный вид ресурса);

E – биологическая энергия участников сделки, энергозатраты оборудования магазина (например, устройство считывания штрих-кода и проведения платежной операции).

C – компетенции участников сделки (продавец – знание товара и умение продавать, покупатель – умение выбрать товар и купить его); возможно, интеллектуальные электронные средства обслуживания сделки в пределах магазина.

Env – пространство магазина, в котором осуществляется сделка; если требуется, то, возможно, рассмотреть пространство, которое займет упаковка с мышью, когда требуется учитывать пространство рюкзака покупателя.

T – время, которое затрачивается участниками сделки.

Adm – право покупателя приобрести мышшь, как и любой другой товар, имеющийся в магазине; право продавца-кассира проконсультировать покупателя, оформить покупку, принять платеж и отпустить товар.

I – информация, которая сопровождается в ходе сделки: информация о мышши (ее возможности, параметры и др.), стоимость мышши, условия покупки.

Следующий пример показывает заполнение ячейки искомой таблицы. Здесь рассматриваемая роль – водитель; шаг – проезд на автобусе между остановками (когда водитель на автобусе возит пассажиров).

Ресурсы:

M – расход топлива на тепловую, а затем механическую энергию;

E – расход механической энергии на преодоление расстояния; расход биологической энергии водителя, в том числе для управление автобусом;

C – водитель использует свои компетенции для управления транспортом;

Env – автобус перемещается между остановками; у автобуса есть внутреннее пространство, в котором размещается водитель;

T – тратится на проезд между остановками;

I – водитель использует информацию о маршруте и технологии управления автобусом;

Adm – проездной студента; разрешающие документы для водителя, дающие право на управление транспортным средством: водительское удостоверение, путевые листы; возможности использования городской транспортной инфраструктуры.

### **Рекомендации по формированию вариантов заданий**

Варианты содержат название процесса и ключевые слова, которые его отражают. Ключевые слова могут отражать ресурсы, роли, действия и(или) иные параметры процесса (при необходимости можно адаптировать примеры в зависимости от специфики дисциплины):

- перелет пассажирского самолета (самолет, топливо, пассажиры, пилоты, наземные службы аэропортов);
- обучение студента на курсе (студенты, преподаватель, учебные материалы, информационные системы, аудитория);
- выстрел пули из пистолета (пистолет, патрон, стрелок, порох, гильза, пуля);
- работа торговой палатки по продаже шаурмы (клиент, хозяин, повар-продавец, ингредиенты, шаурмница, денежные средства);
- поездка на такси (автомобиль, водитель, пассажир, топливо, денежные средства);

- разработка компьютерной программы на заказ (представитель заказчика, программист, менеджер, программные библиотеки, компьютерная программа, денежные средства);
- работа торговой точки, реализующей товар одного вида (оптовый склад, товар, продавец, хозяин, водитель, автомобиль, денежные средства);
- доставка еды (еда, курьер, клиент, поставщик еды, электрический самокат);
- всход растения (почва с питательными веществами, семя, росток, вода, солнечная энергия);
- приготовление супа (человек, ингредиенты, энергия плиты, суп);
- представление информации из ОЗУ компьютера на монитор в числовом виде (ОЗУ, видеокарта, монитор, электрическая энергия);
- заполнение формы заявления с помощью шариковой ручки (заявитель, чернила, лист с чистой формой, лист с заполненной формой, настольная лампа);
- посещение кафе (заказ еды, оплаты еды, меню);
- посещение врача (запись к врачу, визит к врачу, диагноз, назначение лечения);
- покупка рецептурного препарата в аптеке (фармацевт, препарат, покупатель, рецепт, оплата).

#### **Рекомендуемое содержание отчетных материалов**

Результат формируется в виде PDF-файла со следующим содержанием:

1. ФИО обучающегося;
2. название настоящей практической работы;
3. цель работы;
4. постановка задачи в соответствии с выбранным вариантом задания;
5. краткий текст объемом в 200–250 слов, раскрывающий процесс с указанием базовых видов ресурсов;
6. таблица ресурсного анализа: по горизонтали – шаги (этапы), по вертикали – роли (компоненты);
7. рефлексивное заключение: что удалось полезного извлечь из процесса и результата выполнения работы.

#### **Контрольные вопросы**

1. Какой вид ресурса позволяет отразить интеллектуальные способности человека?
2. Что такое энергетический ресурс?
3. Требуется показать, что у одного из сотрудников есть доступ к сервису генерации текстов с помощью большой языковой модели. Какой вид ресурса следует задействовать?
4. Предположим, что требуется вскипятить воду в электрическом чайнике. Перечислите виды ресурсов, которые здесь используются?

5. Перечислите виды ресурсов, которые необходимо иметь в виду при анализе процесса проведения лекции в виртуальной комнате системы видеоконференцсвязи (роли: преподаватель, студент).

## **Идентификация архитектуры информационной системы в соответствии с названием и кратким описанием**

**Цель работы.** Освоить навыки идентификации ИС, классов, к которым относится ИС, а также возможные архитектуры, соответствующие ее функционалу. Предполагается, что данная практическая работа будет способствовать развитию навыков анализа и классификации ИС, а также обоснованно применять существующие подходы к проектированию ИС.

**Практическое задание.** В рамках практического задания предоставляется название и краткое описание проектируемой ИС (допускается свой вариант). На основе данной информации необходимо:

1. Определить классы, к которым может принадлежать ИС, и обосновать ее принадлежность (оформить в виде таблицы: №, название класса, краткое обоснование принадлежности).

2. Определить возможные архитектуры ИС и обосновать их соответствие функционалу системы (оформить в виде таблице: №, архитектура, краткое обоснование соответствия).

ИС может принадлежать и(или) соответствовать разным классам и архитектурам соответственно. Варианты названий ИС и описаний приведены ниже. Допускается предлагать свое название и описание.

**Исходные данные:** наименование ИС и ее краткое описание.

**Требуемый результат:** список идентифицированных классов и архитектур, которым соответствует ИС (с обоснованием выбора/идентификации).

### **Теоретические вводные**

Информационные системы — это системы, предназначенные для сбора, обработки, хранения и передачи информации. Классификация ИС позволяет систематизировать их по различным признакам, что облегчает анализ и выбор необходимой системы.

ИС можно классифицировать по различным признакам. С помощью классификации возможно разделять системы, например, по технологическим особенностям (например, клиент-серверные или мобильные) или по сфере применения (например, коммерческие или научно-технические):

1. По уровню управления:
  - низкий (системы, применяемые для автоматизации работы отдельных сотрудников или рабочих групп);
  - средний (системы, предназначенные для решения задач подразделений организации);
  - верхний (системы, используемые для принятия стратегических решений на уровне высшего руководства).

2. По функциональным областям: медицинские, финансовые, образовательные, промышленные и др.

3. По степени интеграции:

- неинтегрированные (разрозненные системы, работающие независимо друг от друга);
- частично интегрированные (системы, имеющие ограниченную связь друг с другом, например, обмен данными через общие файлы);
- полностью интегрированные (системы, объединенные в единую сеть с общим хранилищем данных и единым пользовательским интерфейсом).

4. По технологическим особенностям: клиент-серверные, веб-ориентированные, мобильные, облачные.

5. По способу взаимодействия с пользователем: текстовый, графический и голосовой интерфейс.

6. По типу обрабатываемых данных: текстовые, числовые, графические, звуковые и видео данные.

7. По сфере применения: коммерческие, информационно-справочные, управленческие и научно-технические.

8. По уровню автоматизации: экспертные системы, автоматизированные системы управления, интеллектуальные системы и др.

9. По масштабу: локальные, корпоративные, глобальные и распределенные системы.

Вышеприведенный список не является исчерпывающим, поскольку информационные и смежные технологии развиваются быстрыми темпами, поэтому рекомендуется отслеживать новейшие подходы.

Чтобы показать свойства ИС по ее структуре, назначению, топологии компонент и пр., введено понятие «архитектура» ИС. Здесь под архитектурой ИС понимается организация компонентов системы, определяющая их взаимодействие, распределение функций и принципы обработки данных, что, в свою очередь, влияет на производительность, масштабируемость, надежность и другие ключевые характеристики системы, обеспечивая её соответствие функциональным и нефункциональным требованиям.

Существует несколько архитектур ИС [2]:

1. Монолитная архитектура: все компоненты системы (UI, бизнес-логика, БД) объединены в единое приложение.

Плюсы: простота разработки и развертывания.

Минусы: сложность масштабирования, низкая отказоустойчивость.

2. Клиент-серверная архитектура: состоит из клиента (запрашивает данные) и сервера (обрабатывает запросы) (веб-приложения, базы данных).

Плюсы: четкое разделение ответственности.

Минусы: зависимость от сервера, риск перегрузки.

3. Многоуровневая (n-уровневая) архитектура: система разделена на слои (3-уровневая архитектура (UI → сервер приложений → БД)).

Плюсы: гибкость, легкое обслуживание.

Минусы: сложность настройки взаимодействия.

4. Сервис-ориентированная архитектура (SOA): система состоит из независимых сервисов, общающихся через стандартные протоколы (SOAP, REST).

Плюсы: повторное использование сервисов.

Минусы: высокие накладные расходы.

5. Микросервисная архитектура: усовершенствованная SOA, где сервисы являются мелкими, автономными и развертываются независимо.

Плюсы: гибкость, масштабируемость.

Минусы: сложность управления.

6. Событийно-ориентированная архитектура: компоненты реагируют на события (например, Kafka).

Плюсы: асинхронность, масштабируемость.

Минусы: сложность отладки.

7. Архитектура на основе шины данных (ESB): централизованная шина для интеграции сервисов (например, MuleSoft).

Плюсы: унификация взаимодействия.

Минусы: риск узкого места.

8. Пиринговая (P2P): узлы равноправны (например, BitTorrent, блокчейн).

Плюсы: отказоустойчивость.

Минусы: сложность обеспечения безопасности.

9. Облачная архитектура: использование облачных сервисов (IaaS, PaaS, SaaS). (AWS, Azure).

Плюсы: масштабируемость.

Минусы: зависимость от провайдера.

10. Serverless Architecture: запуск кода без управления серверами (например, AWS Lambda).

Плюсы: экономия ресурсов.

Минусы: ограничения выполнения.

11. FaaS (Function as a Service): подвид serverless, где выполняются отдельные функции (Google Cloud Functions).

Плюсы: высокая гибкость и масштабируемость.

Минусы: сложность интеграции с традиционными системами.

12. Компонентно-ориентированная архитектура: система строится из переиспользуемых компонентов (например, COM, EJB).

Плюсы: облегчение обслуживания и обновления

Минусы: необходимость стандартизации интерфейсов, возможны проблемы с совместимостью.

13. Message-Oriented Architecture: обмен сообщениями через брокеры (например, RabbitMQ).

Плюсы: асинхронность коммуникации, гибкость и независимость компонентов.

Минусы: усложнение отладки.

14. Мультитенантная архитектура: один экземпляр приложения обслуживает нескольких пользователей или организаций (SaaS).

Плюсы: эффективность использования ресурсов, упрощение обновления и обслуживания.

Минусы: сложность обеспечения безопасности.

15. Архитектура с контейнерами и оркестрацией: использование Docker и Kubernetes для управления микросервисами.

Плюсы: портативность, гибкость развертывания и масштабирования.

Минусы: сложность настройки и управления.

16. Архитектура на основе реконфигурируемых вычислений: аппаратная адаптация под задачи (например, FPGA).

Плюсы: высокая производительность.

Минусы: ограниченная гибкость.

Вышеприведенный список не является исчерпывающим ввиду быстрых темпов развития информационных технологий и изменяющихся потребностей в их применении.

**Инструкция по использованию больших языковых моделей для выполнения практического задания.** В настоящей работе могут применяться большие языковые модели для получения ответа на вопросы соответствия заданной ИС классам и(или) принадлежности к той или иной архитектуре (см. выше). В этом случае рекомендуется провести ряд действий с сервисом БЯМ, которые позволят получить ответ в текстовой форме. Первое действие состоит в подготовке «промпта» — текстового запроса к БЯМ. Он может состоять как из основного запроса, например, «Укажи, к каким классам может относиться информационная система», так из дополнительных атрибутов, включая структуру и объем ответа. В промпт можно включать специальные атрибуты, которые определяют способ формирования ответа, например, последовательное рассуждение БЯМ. Качество ответа БЯМ зависит от четкости и детализации запроса. Полученный ответ следует критически проанализировать и оценить его качество. При необходимости ответ корректируется или формируется уточненный запрос к БЯМ. Ниже приведены основные шаги работы с сервисом БЯМ для получения результата настоящей практической работы.

1. Получение от сервиса БЯМ уточненного описания ИС. В промпт необходимо включить первичное описание выбранной ИС в соответствии с



вариантом задания, а также указать, что ожидается получить уточненное описание системы с перечнем основных компонентов и функций.

2. Получение от сервиса БЯМ списка классов, которым может соответствовать ИС. При этом рекомендуется внести в текущий промпт ответ, полученный на предыдущем шаге. Например, предлагается реализовать запрос к БЯМ «Информационная система "EcoGuardian" предназначена для мониторинга экологической обстановки и прогнозирования рисков и содержит следующие компоненты..., функции... К каким классам информационных систем она может принадлежать? Рассуждай последовательно. Составь таблицу с обоснованием.». На основе полученного ответа сформировать и проверить таблицу, при необходимости отредактировать её.

3. Получение от сервиса БЯМ списка архитектур, которым может соответствовать ИС. Выполняется аналогично шагу №2.

4. Анализ возможных ошибок. Необходимо реализовать запрос к сервису БЯМ, к примеру: «Какие ошибки могут возникнуть при выборе архитектуры для системы "EcoGuardian"? Перечисли 3–5 ошибок и объясни, почему они критичны. Рассуждай последовательно.»

5. Итоговая работа. На основе всех полученных данных необходимо провести анализ полученных результатов и внести корректировки (если требуется).

#### **Общие рекомендации по использованию БЯМ:**

- Проверять и корректировать полученные ответы.
- Формулировать запросы четко и конкретно как по структуре, так и по содержанию.
- Использовать БЯМ как вспомогательный инструмент, а не замену собственному анализу.

**Типичные ошибки, которые допускаются студентами при выполнении практического задания.** Во-первых, необходимо тщательно анализировать функционал системы, чтобы корректно определить классы ИС. Ошибки на этом этапе могут привести к некорректному выбору классов, что отрицательно скажется на всей архитектуре ИС. Во-вторых, важно обосновывать выбор архитектуры, анализировать, насколько она соответствует требованиям ИС. В-третьих, следует учитывать возможность множественной принадлежности объектов, поскольку игнорирование этого аспекта может привести к ошибкам в логике системы и усложнить работу с данными. В-четвертых, необходимо обеспечивать логическую связь между классами и архитектурными решениями.

#### **Рекомендации по формированию вариантов заданий**

Варианты содержат названия ИС, а также их краткое описание, которые отражают основное предназначение каждой ИС. Для выполнения данной

практической работы необходимо выбрать один из предложенных вариантов или же предложить свой.

1. EcoGuardian

Описание: интеллектуальная система мониторинга экологической обстановки в реальном времени, способная прогнозировать и предупреждать о природных катастрофах и экологических рисках.

2. MediConnect

Описание: универсальная платформа для обмена медицинскими данными между пациентами, врачами и медицинскими учреждениями, обеспечивающая безопасность и конфиденциальность информации.

3. AgroVision

Описание: система управления умным сельским хозяйством, использующая датчики и дроны для оптимизации урожайности и снижения затрат на ресурсы.

4. EduSphere

Описание: глобальная образовательная платформа с адаптивным обучением, предлагающая индивидуальные учебные планы и виртуальные классы с дополненной реальностью.

5. FinOptima

Описание: автоматизированный финансовый консультант, анализирующий расходы и доходы пользователя, предоставляющий персонализированные инвестиционные рекомендации.

6. CityPulse

Описание: платформа для управления умным городом, объединяющая данные от транспортных систем, коммунальных служб и сервисов безопасности для улучшения качества жизни горожан.

7. HealthNavigator

Описание: приложение для мониторинга здоровья, использующее носимые устройства для отслеживания жизненных показателей и предлагающее рекомендации по улучшению самочувствия.

8. TransLogix

Описание: интеллектуальная логистическая система, оптимизирующая маршруты доставки и управление складскими запасами с помощью искусственного интеллекта.

9. LegalAssist AI

Описание: автоматизированный юридический помощник, способный анализировать правовые документы и предоставлять консультации на основе актуального законодательства.

10. EnergyWise

Описание: система управления энергопотреблением в домах и предприятиях, позволяющая снижать расходы и переходить на возобновляемые источники энергии.

#### 11. SafeHome

Описание: интегрированная система домашней безопасности, использующая ИИ для распознавания угроз и автоматического реагирования на чрезвычайные ситуации.

#### 12. TravelMate

Описание: персональный туристический помощник, предлагающий маршруты, бронирование и рекомендации, основанные на предпочтениях и прошлом опыте путешествий пользователя.

#### 13. RetailGenie

Описание: аналитическая платформа для розничной торговли, отслеживающая поведение покупателей и помогающая оптимизировать ассортимент и ценообразование.

#### 14. AutoPilot Assistant

Описание: система поддержки водителя с функциями автономного вождения, обеспечивающая безопасность и комфорт при управлении транспортным средством.

#### 15. MindSync

Описание: нейронный интерфейс, позволяющий управлять устройствами и компьютерами с помощью мыслей, открывая новые возможности для взаимодействия человека и технологий.

### **Рекомендуемое содержание отчетных материалов**

Результат формируется в виде PDF-файла со следующим содержанием:

1. ФИО обучающегося;
2. название настоящей практической работы;
3. цель работы;
4. постановка задачи в соответствии с выбранным вариантом задания;
5. таблица №... с названием класса, краткое обоснование принадлежности;
6. таблица № ... с названием архитектур, краткое обоснование соответствия;
7. рефлексивное заключение: что удалось полезного извлечь из процесса и результата выполнения работы.

### **Контрольные вопросы**

1. Какие основные признаки используются для классификации ИС?
2. Как определяется соответствие ИС определенному классу?
3. Почему одна и та же ИС может соответствовать разным архитектурам?
4. В чем ключевые отличия монолитной и микросервисной архитектур?
5. В каких сценариях предпочтительна пиринговая (P2P) архитектура?

6. В чем отличие архитектуры без сервера (Serverless) от традиционного подхода с выделенными серверами?
7. Какую роль играет мультитенантность в разработке облачных решений?
8. Какие преимущества дает архитектура на основе реконфигурируемых вычислений?

## **Разработка технического задания на создание информационной системы**

**Цель работы.** Освоить навыки разработки технических заданий на создание ИС, научиться определять её масштаб, формулировать функциональные и нефункциональные требования, а также учитывать технологические, качественные и интерфейсные особенности.

**Практическое задание.** Требуется разработать общее или частное техническое задание (ТЗ) в зависимости от масштаба ИС. Для большой ИС частное техническое задание разрабатывается для одного из отдельных компонент.

Для выбранного варианта задания необходимо определить масштаб системы и выбрать вид ТЗ. Например, если система небольшая, то строится частное ТЗ. Возможно предложить свой вариант. Для своего варианта также необходимо определить масштаб соответствующей системы и выбрать тип технического задания.

Техническое задание должно иметь следующую структуру:

1. Общие сведения о разработке – краткое описание системы: что за система или компонент, для чего она предназначена, какие основные технологии применяются при ее разработке (один абзац, 5–6 предложений).

2. Основание для разработки – 1–2 предложения (например, договор..., ведомственная программа развития – допускается придумать).

3. Назначение разработки – 1–3 предложения, кратко раскрывающие цель разработки (зачем система или компонент нужен) и где будет использоваться.

4. Требования к системе:

4.1 Функциональные требования – перечень 3–4 функций системы/компонента и бизнес-процессов, которые они поддерживают, с кратким описанием (можно в форме таблицы), одна функция должна быть интеграционной – взаимодействия с другими смежными системами/компонентами/сервисами.

4.2 Требования к структуре – список 3–4 компонент или модулей (для компонента), их назначение (можно показать в виде таблицы и(или) структурной схемы).

4.3 Требования к интерфейсу – перечень 2–4 основных форм и интерфейсных элементов с указанием их функциональности по отношению к поддерживаемым бизнес-процессам (например, форма представления результатов прогнозирования, содержащая графики реальных и спрогнозированных значений, предназначенные для просмотра и сравнения соответствующих значений).

4.4 Технологические требования – список основных технологий, методов и(или) алгоритмов, которые необходимо задействовать при разработке; обоснование требований (2–4 единицы).

4.5 Технические требования – аппаратное и программное обеспечение, совместимость с платформами, стандартами.

4.6 Требования к качеству результатов – 2–4 предложения с метриками и соответствующими значениями (допускается оформление в виде таблицы).

5. Требования к программной документации – несколько предложений с упоминанием не менее трех типов документов, например, описание программы, руководство системного программиста, описание применения, программа и методика испытаний со ссылками на стандарты, которые регламентируют соответствующие документы (допускается оформление в виде таблицы).

6. Порядок выполнения работ: указывается 4–7 основных шагов выполнения работы с их описанием в 1–3 предложения, а также получаемым после выполнения каждого шага результатом, или дается отсыл на соответствующий стандарт. В шагах должны присутствовать испытания и приемка основного продукта.

**Исходные данные:** наименование ИС и ее краткое описание.

**Требуемый результат:** текст ТЗ.

### **Теоретические вводные и описание способов решения**

Техническое задание – документ, определяющий требования к разрабатываемой системе, её структуре, функциональности, интерфейсам и техническим характеристикам [3].

К основным элементам ТЗ можно отнести следующий перечень, но не ограничиваться им:

- Общие сведения о разработке – краткое описание системы, её цели и применяемых технологий.
- Основание для разработки – причины и нормативные документы, обуславливающие необходимость создания системы.
- Назначение разработки – определение целей и областей применения системы.
- Требования к системе:
  - Функциональные требования (перечень функций и поддерживаемых бизнес-процессов);
  - Требования к структуре (перечень модулей и их назначение);
  - Требования к интерфейсу (основные формы и элементы управления);
  - Технологические требования (применяемые технологии и методы);
  - Требования к качеству (метрики и ожидаемые показатели);
  - Технические требования (аппаратное и программное обеспечение).

- Требования к программной документации – перечень необходимых документов и требований к оформлению.
- Порядок выполнения работ – основные этапы разработки и испытаний, а также порядок сдачи-приемки.
- Приложения – схемы и диаграммы, таблицы и дополнительные материалы.

Процесс разработки ТЗ включает несколько этапов, каждый из которых играет ключевую роль в создании качественного документа, соответствующего требованиям заказчика и стандартам. К основным шагам по разработке ТЗ можно отнести:

1. Сбор требований – на этом этапе проводится исследование предметной области, в ходе которого собираются исходные данные о системе. К основным задачам можно отнести: выявление целей и задач будущей системы, определение основных пользователей и их потребности, изучение аналогичных решений и выявление их преимуществ/недостатков, сбор информации о технических, организационных и бизнес-ограничениях.

2. Анализ и уточнение требований – проводится анализ требований с целью их детализации, устранения противоречий и согласования с техническими возможностями. К основным задачам можно отнести: определение функциональных и нефункциональных требований, разработка сценариев использования системы (Use Cases), анализ возможных рисков и ограничений, определение интеграционных возможностей с другими системами, а также уточнение технических и ресурсных требований. Результатом должны стать четко сформулированные и классифицированные требования к системе.

3. Составление предварительной версии ТЗ – создается черновой вариант ТЗ, включающий основные разделы: описание системы и ее назначения; функциональные и нефункциональные требования; требования к интерфейсу, технологиям, качеству; требования к программной документации; этапы выполнения работ и критерии приемки. Для достижения этих задач можно использовать различные методы и инструменты, но не ограничиваясь ими, например, использовать ГОСТ 34.602–89, специальные шаблоны и стандарты, а также применять БЯМ для генерации.

4. Согласование ТЗ с заказчиком – после подготовки предварительного варианта документа его необходимо обсудить с заказчиком и ключевыми заинтересованными сторонами. К основным задачам можно отнести: проверка соответствия ТЗ бизнес-целям и ожиданиям; обсуждение возможных изменений и дополнений; уточнение спорных моментов (сроки, технологии, интеграция); сбор обратной связи.

5. Формализация и утверждение ТЗ – на финальном этапе вносятся последние правки, после чего документ утверждается и становится обязательным к исполнению. К основным задачам можно отнести: финальное редактирование и

устранение замечаний; оформление документа в соответствии со стандартами; подписание ТЗ всеми заинтересованными сторонами; размещение ТЗ в системе документации. Результатом должно стать утвержденное ТЗ, являющееся основным документом для разработки ИС.

**Инструкция по использованию БЯМ для выполнения практического задания.** В настоящей работе может предусматриваться применение больших языковых моделей для получения ответа на вопросы, связанные со структурой и содержанием ТЗ для выбранной ИС. В этом случае рекомендуется провести ряд действий с сервисом БЯМ (можно использовать ChatGPT, Gemini, Claude, Mistral и другие модели. Также можно использовать специализированные плагины или инструменты для генерации текстов, такие как Copilot или Jasper), которые позволят получить ответ в текстовой форме. Первое действие состоит в подготовке промпта. Он может состоять как из основного запроса, например, «Сгенерируй структуру технического задания на разработку <название системы> в соответствии с ГОСТ 34.602-89». В промпт возможно включать специальные атрибуты, которые определяют способ формирования ответа, например, последовательное рассуждение БЯМ. Качество ответа БЯМ зависит от четкости и детализации запроса. Полученный ответ следует критически проанализировать и оценить его качество. При необходимости ответ корректируется или формируется уточненный запрос к БЯМ. Ниже приведены основные шаги работы с сервисом БЯМ для получения результата настоящей практической работы.

1. Получение от сервиса БЯМ уточненного описания структуры ТЗ. В промпт необходимо включить первичное описание выбранной ИС в соответствии с вариантом задания, а также указать, что ожидается получить уточненное описание структуры ТЗ. В качестве примера запроса можно привести следующий текст: «Сгенерируй структуру технического задания на разработку <название системы и её описание> в соответствии с <требования к оформлению>.»

2. Получение от сервиса БЯМ формулировок требований к системе. При этом рекомендуется внести в текущий промпт ответ, полученный на предыдущем шаге. Например, «Перечисли основные требования для системы <название системы и её описание> в соответствии с техническим заданием <основные разделы ТЗ>, опиши их кратко в виде списка или таблицы с пояснениями.»

3. Получение от сервиса БЯМ формулировок согласно остальным компонентам ТЗ. Выполняется аналогично шагу №2.

4. Проверка логики и полноты ТЗ. После составления черновика можно попросить БЯМ проанализировать документ, например: «Проверь логичность и полноту моего ТЗ. Какие аспекты требуют уточнения или доработки?».

5. Оптимизация формулировок. Если какой-то раздел написан нечетко или слишком объемно, можно использовать запрос: «Сократи и сделай более точной следующую формулировку: <текст>».

**Общие рекомендации по использованию БЯМ:**



1. Не копируйте текст БЯМ без изменений – редактируйте его под свою задачу.
2. Проверяйте соответствие требованиям ГОСТ, поскольку БЯМ может не учитывать все необходимые стандарты.
3. Используйте БЯМ как помощника, а не как единственный и достоверный источник – критически относитесь к выходным данным БЯМ, добавляя собственные идеи и дополнения.

**Типичные ошибки, которые допускаются студентами при выполнении практического задания.** Во-первых, не следует допускать неясных, неоднозначных или расплывчатых формулировок требований, поскольку это затрудняет их интерпретацию и может привести к неверной реализации. Важно формулировать требования четко и однозначно, избегая двусмысленности. Во-вторых, необходимо учитывать взаимодействие системы с другими системами и не оставлять эти аспекты без описания. В-третьих, следует приводить аргументы в пользу использования выбранных методов и технологий. В-четвертых, требуется задавать четкие и достижимые метрики, поскольку их отсутствие или указание недостижимых показателей затрудняет контроль качества и оценку результата. Например, если система должна обрабатывать миллион запросов в секунду, но не обладает соответствующей инфраструктурой, такое требование будет невыполнимым. В-пятых, необходимо согласовывать порядок выполнения работ и учитывать их полноту, поскольку несогласованный или неполный план может привести к задержкам, некорректному распределению ресурсов и недостижимости завершения проекта.

#### **Рекомендации по формированию вариантов заданий**

Вариант системы берется из практического задания «Идентификация архитектуры информационной системы в соответствии с названием и кратким описанием». Допускается придумать/предложить свой вариант.

#### **Рекомендуемое содержание отчетных материалов**

Результат формируется в виде PDF-файла со следующим содержанием:

1. ФИО обучающегося;
2. название настоящей практической работы;
3. цель работы;
4. постановка задачи в соответствии с выбранным вариантом задания, включая масштаб системы и тип технического задания;
5. текст технического задания, включающего в себя общие сведения о разработке, основании для разработки, назначении разработки, требованиях к системе (функциональные требования, требования к структуре, требования к интерфейсу, технологические требования, требования к качеству результатов, технические требования), требованиях к программной документации, порядке выполнения работ;

6. рефлексивное заключение: что удалось извлечь полезного из процесса и результата выполнения работы.

**Контрольные вопросы**

1. Какова цель разработки ТЗ?
2. Какие основные разделы должно включать ТЗ?
3. В чем разница между общим и частным ТЗ?
4. Как формулируются функциональные и нефункциональные требования?
5. В чем заключается разница между функциональными и техническими требованиями?
6. Какие нормативные документы могут регламентировать требования к ТЗ?

## **Разработка графической модели процессов с применением стандарта IDEF0**

**Цель работы.** Освоить навыки моделирования бизнес-процессов с использованием методологии IDEF0 (разработать модели, демонстрирующие процессы функционирования проектируемой ИС).

**Практическое задание.** В рамках данной работы необходимо создать модель бизнес-процессов, в которых будет применяться проектируемая ИС. Например, при разработке системы обучения (Learning Management System) следует отразить процессы:

- Разработка курса.
- Обработка запросов потенциальных обучаемых.
- Проведение курсов, включая аттестацию (зачет, экзамен).
- Выпуск обучающихся.

Модель должна включать минимум три диаграммы:

1. Основная диаграмма (top-диаграмма, только один блок).
2. Декомпозиция основной диаграммы с числом блоков от трех до пяти.
3. Дальнейшая декомпозиция выбранных блоков основной диаграммы

(не менее двух блоков).

Каждая диаграмма:

1. Оформляется на листе формата А4 (горизонтальная ориентация).
2. Сопровождается названием, указанием точки зрения, цели построения и номером диаграммы (А-0, А0, А1).
3. Обеспечивается пояснением объемом 3–5 предложений.

**Исходные данные:** ИС и ее описание.

**Требуемый результат:** диаграммы IDEF0 (top-диаграмма, основная диаграмма, декомпозиции выбранных блоков основной диаграммы).

### **Теоретические вводные и описание способов решения**

Методология SADT (Structured Analysis and Design Technique) – методология описания искусственных систем средней сложности.

- IDEF0 (как часть SADT) используется для формализации системы и её среды на ранних этапах жизненного цикла.
- Диаграммы SADT-модели оформляются стандартным образом на бланках. Каждая SADT-диаграмма содержит блоки и дуги.

Блоки изображают функции моделируемой системы, а дуги связывают блоки вместе и отображают взаимодействия и взаимосвязи между ними [4].

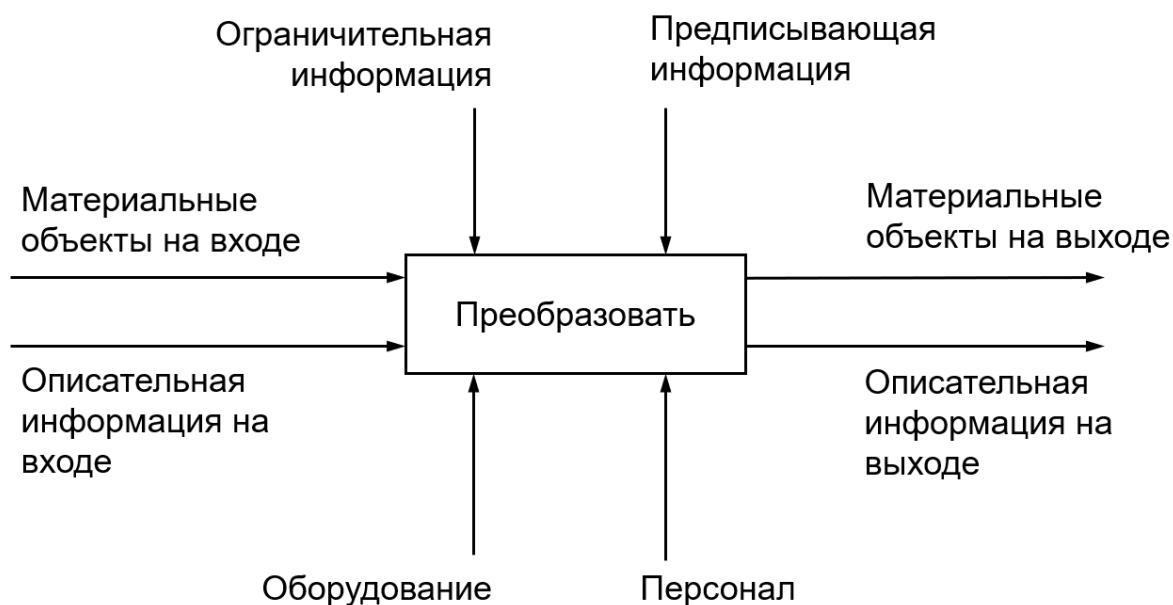


Рисунок 4.1 – IDEF0: блок с дугами (top диаграммы).

### Основные принципы IDEF0.

1. Функциональная декомпозиция:
  - Разбиение сложной системы на иерархические уровни: от общего к частному. Каждый уровень отображает детализацию процессов, входящих в состав более высокого уровня.
2. Графическая и текстовая документация:
  - Визуальная модель (графическая часть) сопровождается пояснительным текстом для упрощения интерпретации.
3. Контекстное представление процессов:
  - Каждый процесс рассматривается в контексте его окружения: входы, выходы, управляющие воздействия, механизмы исполнения.

### Основные элементы IDEF0.

1. Функциональный блок:
  - Основная единица модели. Обозначается прямоугольником, в котором указывается название процесса (например, «Обработка заявки», «Регистрация студента», «Разработка учебного курса» и пр.).

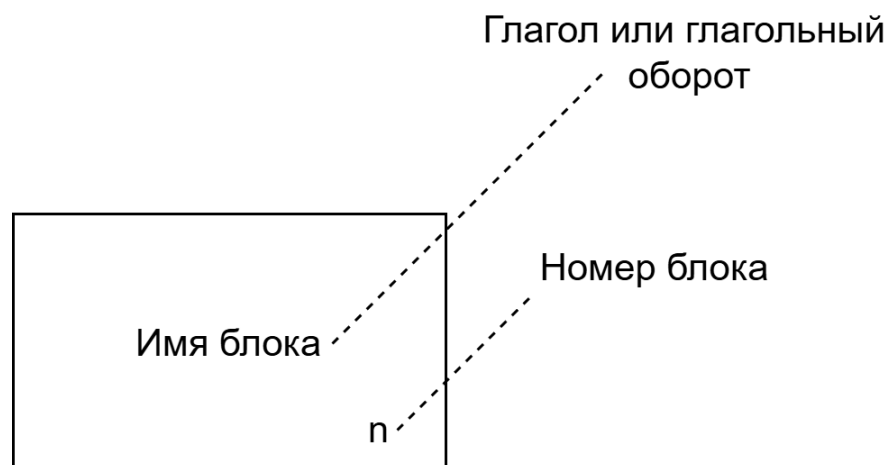


Рисунок 4.2 – Функциональный блок.

2. Дуги взаимодействия:

- Используются для связи между функциональными блоками и показывают характер взаимодействий. Как правило, существует четыре типа дуг:

- Входные данные – поступают слева в блок, преобразуются в результате выполнения процесса.

- Выходные данные – выходят справа из блока, представляют результат работы функции.

- Управляющие воздействия – поступают сверху, определяют правила или ограничения работы процесса (например, регламенты, инструкции, законодательные нормы).

- Механизмы выполнения – поступают снизу, представляют ресурсы или исполнителей, необходимые для выполнения функции (например, сотрудники, программное обеспечение, оборудование).

3. Контекстная диаграмма (A-0):

- Это первичная диаграмма, задающая общий контекст системы.

- Она показывает систему как единый блок, обозначая его входы, выходы, управляющие воздействия и механизмы.

В пояснительном тексте к контекстной диаграмме должна быть указана цель построения диаграммы и зафиксирована точка зрения, где [4]:

- Цель — это краткая формулировка причины создания модели, отвечающая на вопросы: почему этот процесс должен быть смоделирован? что должна показать модель? что может получить читатель исходя из представления модели?

- Точка зрения — это указание на должностное лицо или подразделение организации, с позиции которого разрабатывается модель. Точка зрения определяет основное направление развития модели и уровень необходимой детализации. Важно учитывать, что корректное и четкое фиксирование точки зрения позволяет визуальнo разгрузить модель. Это может

быть достигнуто путем отказа от детализации отдельных элементов, которые являются избыточными, исходя из выбранной точки зрения на систему.

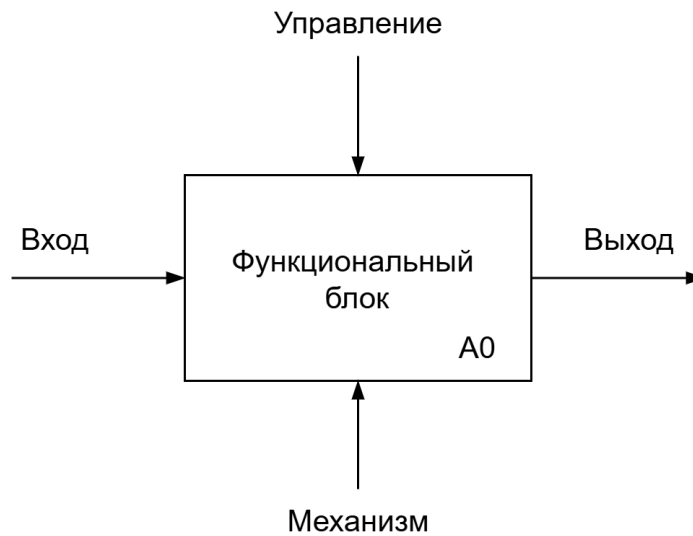


Рисунок 4.3 – Виды дуг.

#### 4. Декомпозиция:

- Каждый функциональный блок может быть детализирован на отдельной диаграмме.
- Детализация идёт по уровням ( $A0 \rightarrow A1, A2$  и т. д.), вплоть до уровня, на котором можно четко определить каждый элемент процесса.

#### **Графические особенности IDEF0.**

1. Ограниченное количество функциональных блоков на одной диаграмме. Рекомендуется использовать не более 3–6 блоков на одном уровне, чтобы сохранить читабельность.
2. Туннелирование дуг. Если поток данных или управления не отображается на следующем уровне, то он обозначается туннелем (маленькая черточка на конце дуги).

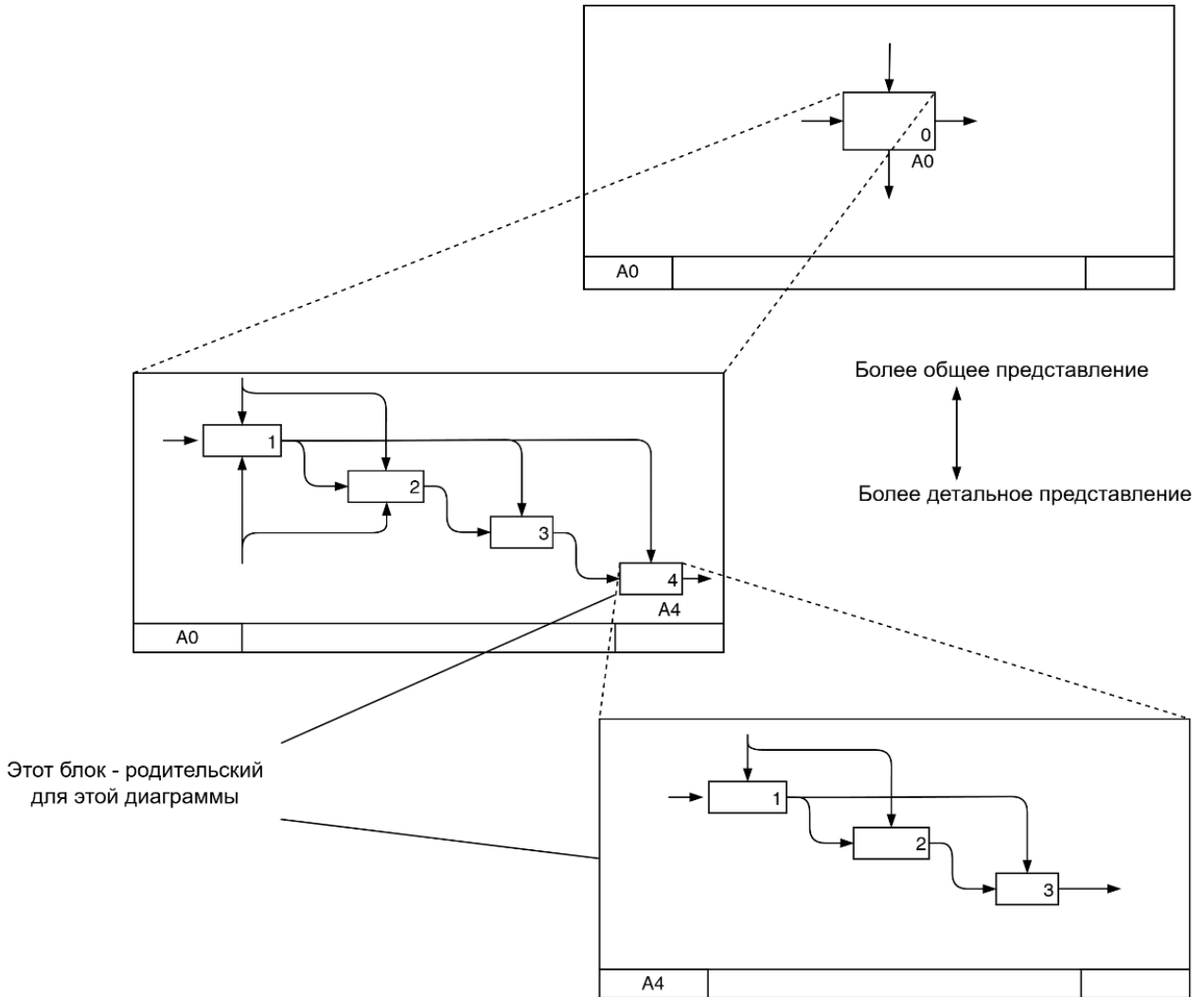


Рисунок 4.4 – Функциональная декомпозиция.

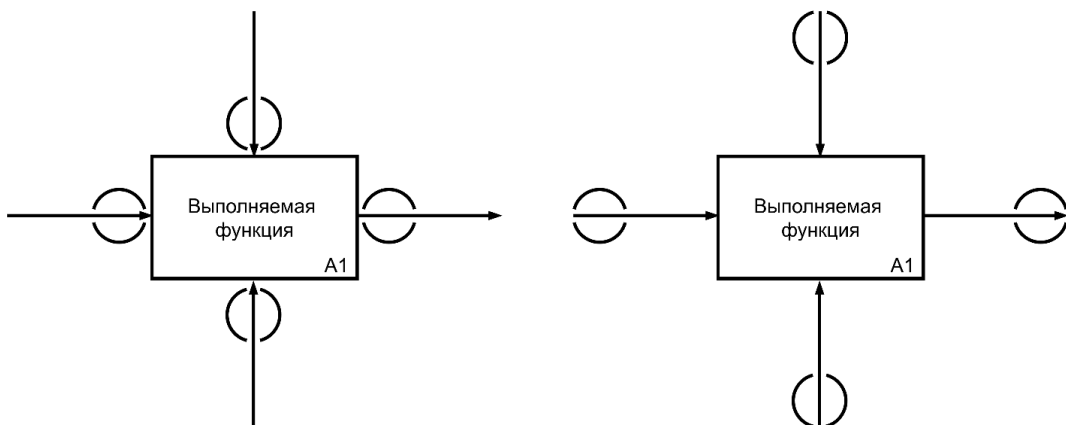


Рисунок 4.5 – Туннели (вместо «Выполняемая функция» указывается соответствующий глагольный оборот).

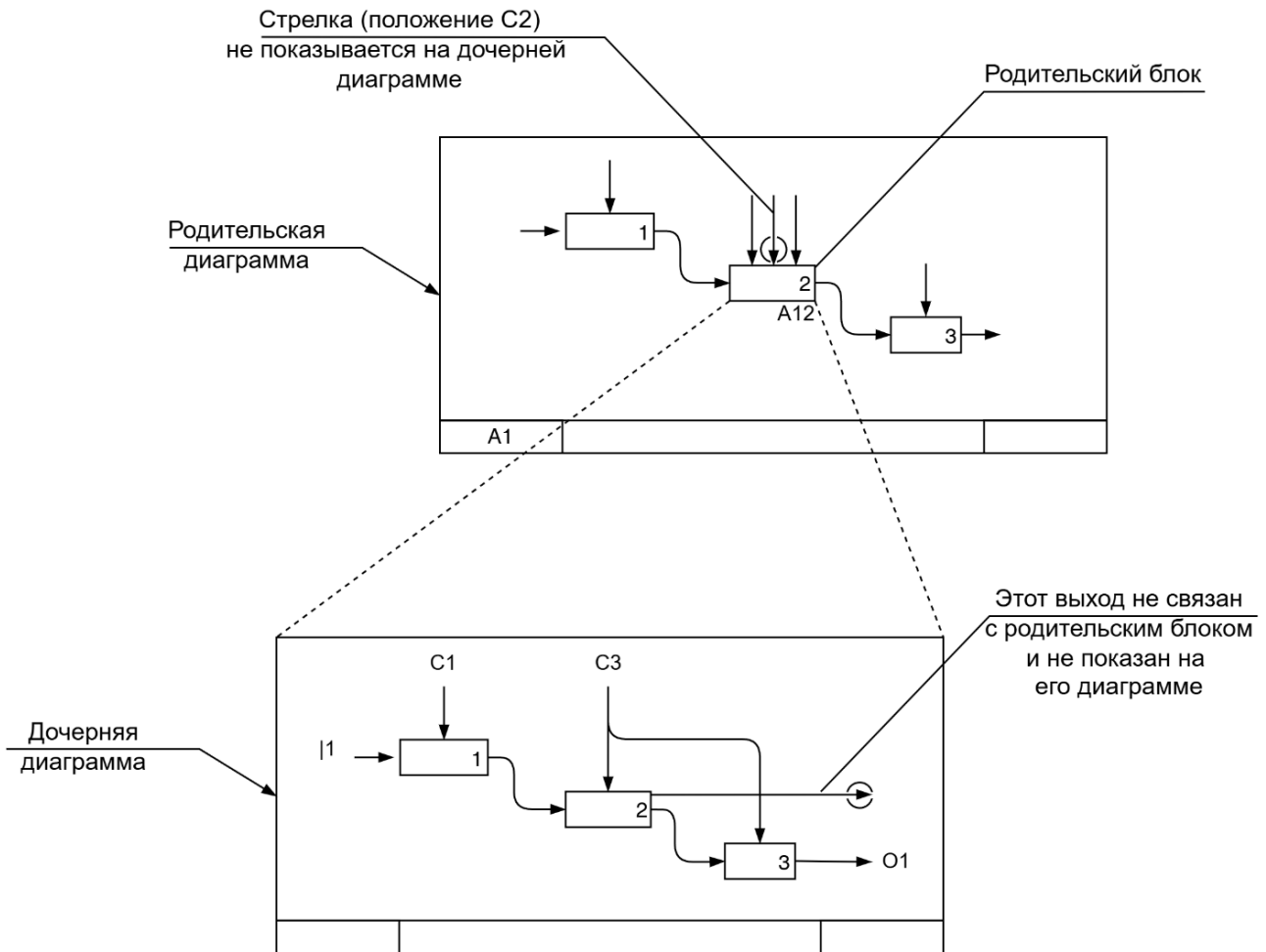


Рисунок 4.6 – Туннели при отображении на родительской и дочерней диаграмме.

3. Нумерация диаграмм.

- Первая диаграмма A-0 – верхний уровень.
- Затем идет диаграмма A0, которая раскрывает основные функции.
- Далее — A1, A2 и т. д., обозначающие декомпозицию отдельных блоков.

**Типичные ошибки, которые допускаются студентами при выполнении практического задания.** Во-первых, диаграммы IDEF0 должны содержать четкие заголовки, отражающие точку зрения и цель модели. Отсутствие этой информации затрудняет интерпретацию диаграммы и снижает её информативность. Во-вторых, при построении моделей SADT необходимо корректно использовать дуги и блоки. Некорректное соединение элементов и(или) их неправильное расположение могут исказить смысл диаграммы и привести к искаженному представлению процессов. В-третьих, при создании диаграмм BPMN важно учитывать рекомендуемое количество элементов. Чрезмерное усложнение схемы затрудняет её восприятие и анализ, что в свою очередь может



снизить эффективность коммуникации между участниками процесса. В-четвёртых, диаграммы должны сопровождаться исчерпывающими пояснениями.

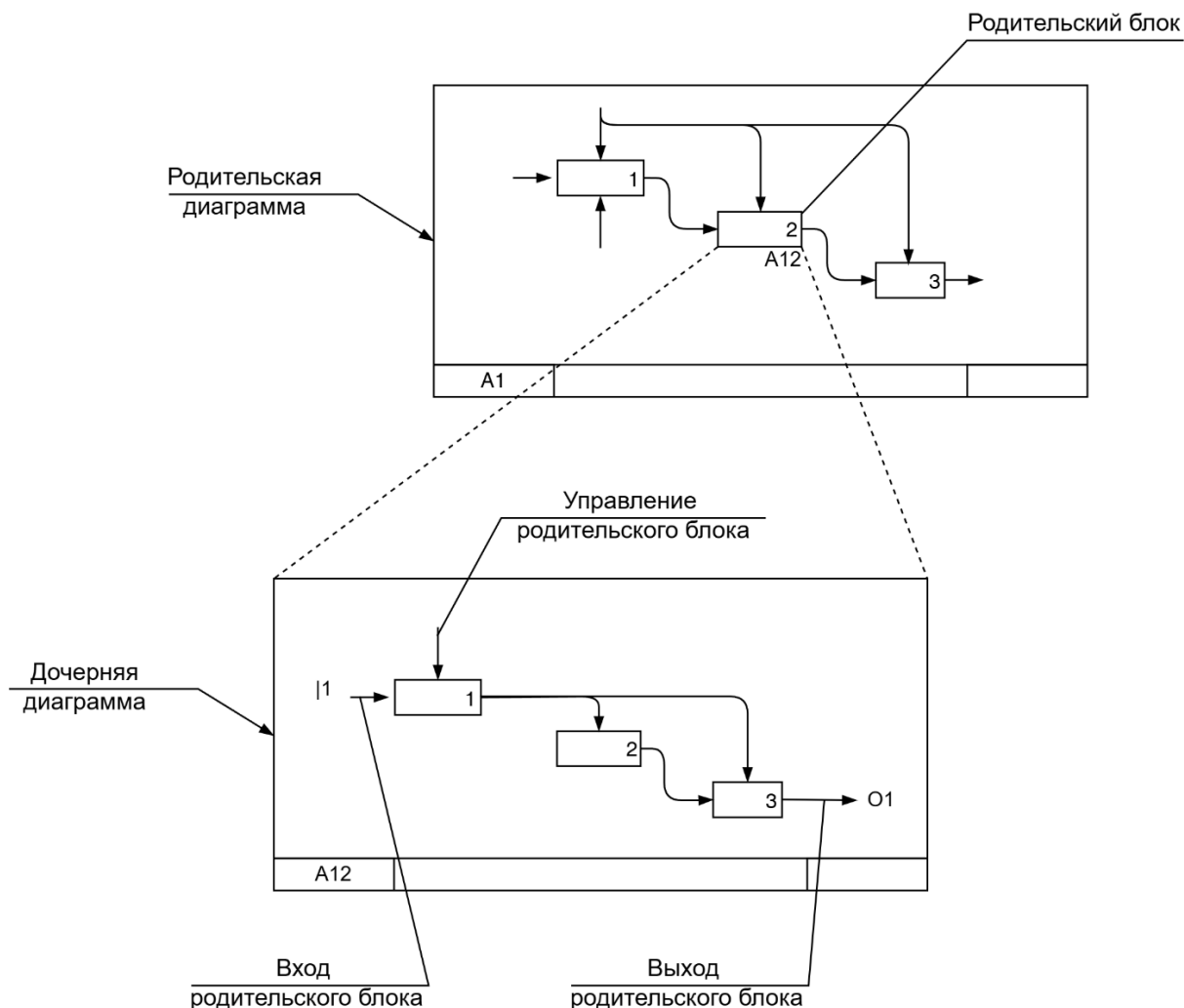


Рисунок 4.7 – Связь родительского блока и дочерней диаграммы.

Недостаток комментариев и(или) их полное отсутствие усложняют понимание модели и могут привести к разночтениям. В-пятых, следует избегать использования неформализованных обозначений, которые не соответствуют принятому стандарту, что может привести к разночтению, затруднить её использование в дальнейшем и затягиванию процесса работы в целом.

### Рекомендации по формированию вариантов заданий

Вариант системы берется из практического задания «Идентификация архитектуры информационной системы в соответствии с названием и кратким описанием». Допускается придумать/предложить свой вариант.

### Рекомендуемое содержание отчетных материалов

Результат формируется в виде PDF-файла со следующим содержанием:

1. ФИО обучающегося;
2. название настоящей практической работы;

3. цель работы;
4. постановка задачи;
5. диаграммы полученной модели с пояснениями к каждой из них (5–10 предложений);
6. рефлексивное заключение;
7. бонусная часть (если делаете): промпт и ответ в форме скрипта; скриншот визуализации скрипта; пояснения (5–10 предложений).

#### **Контрольные вопросы**

1. В чем разница между IDEF0 и BPMN?
2. Какие элементы используются в IDEF0?
3. Как осуществляется декомпозиция в IDEF0?
4. Какие ключевые компоненты присутствуют в BPMN-диаграммах?
5. В чем смысл точек зрения в IDEF0?
6. Какой тип потоков используется в BPMN?
7. Какие ошибки наиболее распространены при моделировании в IDEF0?
8. Как можно проверить корректность BPMN-модели?

## Разработка структурной UML-диаграммы

**Цель работы.** Освоить навыки проектирования структурных UML-диаграмм, позволяющих моделировать архитектуру ИС. Развить навыки аналитического мышления, необходимого для структурированного представления программных решений.

**Практическое задание.** Необходимо создать UML-диаграмму, отражающую структуру ИС. Можно использовать один из видов структурных диаграмм:

- Диаграмму классов (Class Diagram).
- Диаграмму компонентов (Component Diagram).

Требования к работе:

- Число классов или компонентов — не менее 5.
- Для диаграммы компонентов требуется наличие вложенных компонентов.
- При необходимости должны быть отображены базы данных и другие элементы архитектуры.
- К диаграмме необходимо приложить текстовое пояснение (5–10 предложений).

Дополнительные задания:

- Сгенерировать код диаграммы с помощью БЯМ и визуализировать его.
- Создать подкаст по теме работы.

**Исходные данные:** наименование ИС и ее краткое описание.

**Требуемый результат:** структурная диаграмма UML, описывающая содержание системы.

### Теоретические вводные и описание способов решения

**UML (Unified Modeling Language)** – распространенный язык графического моделирования, который позволяет строить модели как процессов, так и объектов (систем) в зависимости от постановки задачи и представляемой стороны (объекта и(или) процесса) [5].


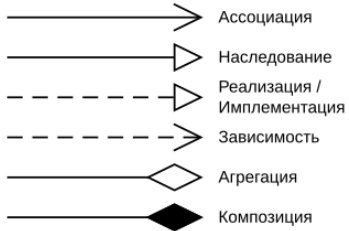
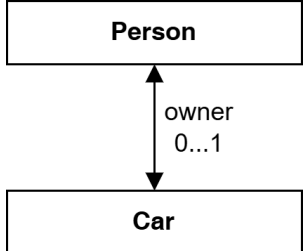
UML-диаграммы делятся на две категории:

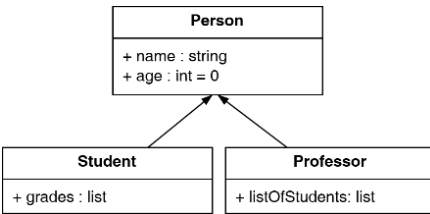
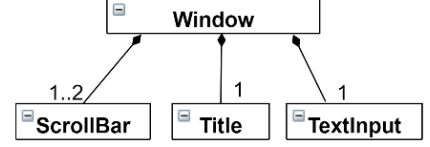
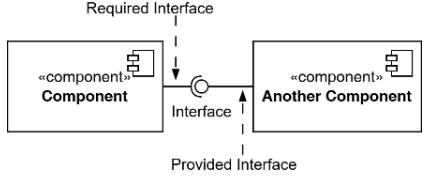
1. структурные диаграммы;
2. поведенческие диаграммы.

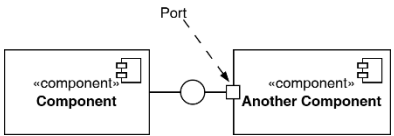
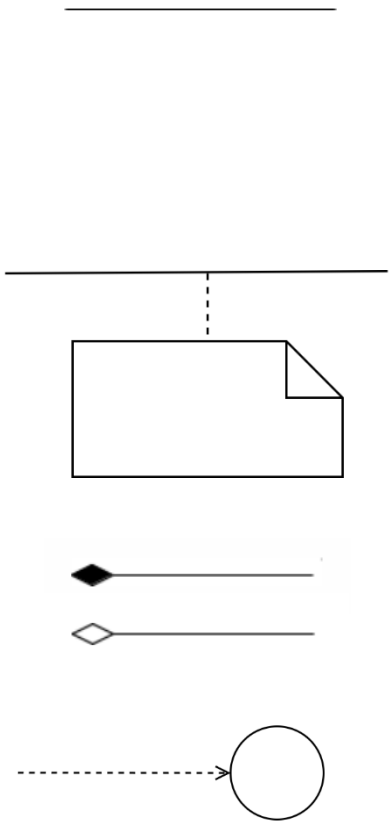
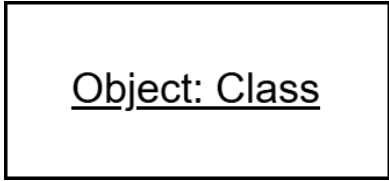
В данной практической работе речь будет идти о структурных диаграммах.

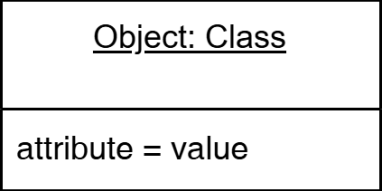
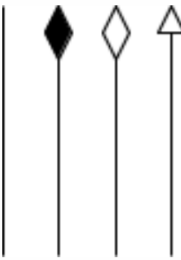
Таблица 5.1 – Структурные диаграммы.

Вид структурной диаграммы	Графические элементы
---------------------------	----------------------

Вид структурной диаграммы	Графические элементы	
<p><b>Диаграмма классов</b> - демонстрирует общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей (отношений) между ними</p>	<p>Классы – класс с определенным полями и методами, которые могут меняться (переопределяться), добавляться в иерархии наследования</p>	
	<p>Атрибуты – описывают свойство в виде строки текста внутри прямоугольника класса</p>	
	<p>Методы – представляют поведение или функциональность класса</p>	
	<p>Видимость</p>	<p>[ + ] Публичный ключ (Public)  [ - ] Приватный (Private)  [ # ] Защищенный (Protected)  [ / ] Производный (Derived) (может быть совмещен с другими)  [ ~ ] Пакет (Package)</p>
	<p>Взаимосвязи – это особый тип логических отношений между сущностями, показанных на диаграммах классов и объектов</p>	
	<p>Ассоциации – это непрерывная линия между двумя классами, направленная от исходного класса к целевому классу</p>	

Вид структурной диаграммы	Графические элементы	
	<p>Наследование – показывает, что один из двух связанных классов (подтип) является частной формой другого (надтипа), который называется обобщением первого</p>	
	<p>Агрегация – это разновидность ассоциации при отношении между целым и его частями (объекты могут существовать независимо)</p>	
	<p>Композиция – более строгий вариант агрегации</p>	
	<p>Мощность отношений (Кратность) – число связей между каждым экземпляром класса (объектом) в начале линии с экземпляром класса в её конце</p>	
<p><b>Диаграмма компонентов -</b> статическая структурная диаграмма, которая показывает разбиение программной системы на структурные компоненты и связи (зависимости) между ними</p>	<p>Компонент – сущность, предназначенная для выполнения какой-либо функции, решения задачи или класса задач</p>	
	<p>Интерфейс – указывает на то, какие возможности или функции предоставляет конкретный компонент для взаимодействия с другими компонентами или системами</p>	

Вид структурной диаграммы	Графические элементы	
	<p>Порт – указывает на интерфейс компонента, через который он предоставляет или получает данные, вызывает методы или осуществляет обмен сообщениями</p>	
	<p>Связи:  Ассоциации – семантическая связь, которая может возникнуть между типизированными экземплярами.  Ограничение – условие, которое выражается в текстовой форме (на естественном языке или в машиночитаемом языке) для целей объявления некоторых семантических элементов.  Композиция и агрегация (см. диаграмму классов).  Зависимость – показывает, что использование одних компонент требует задействования других</p>	
<p><b>Диаграмма объектов</b> – фиксируются конкретные экземпляры с их состояниями</p>	<p>Объект обозначается в виде прямоугольника, в котором имя объекта и его класс подчеркнуты, а также разделены двоеточием</p>	

Вид структурной диаграммы	Графические элементы	
	<p>Атрибуты объекта – каждому объекту присваиваются атрибуты. В отличие от классов, атрибутам объектов должны быть присвоены значения, потому что на диаграмме классов рассматривается конкретный временной отрезок</p>	
	<p>Связи – используются для представления взаимосвязи между двумя объектами (по аналогии с диаграммой классов)</p>	

**Типичные ошибки, которые допускаются студентами при выполнении практического задания.** Во-первых, следует очень внимательно относиться к выбору связей между элементами системы, чтобы избежать их некорректного использования. Например, композиция должна применяться только тогда, когда один объект полностью владеет другим и отвечает за его жизненный цикл, тогда как агрегация уместна в случаях, когда объекты могут существовать независимо. Во-вторых, важно уделять внимание именованию классов и компонентов, так как некорректные или неинформативные названия могут затруднить понимание структуры системы в целом. Например, использование абстрактных имен затрудняет дальнейшую поддержку и развитие проекта/системы. В-третьих, необходимо тщательно проверять наличие всех значимых элементов в проектируемой системе, включая методы, атрибуты и зависимости. Их отсутствие может привести к тому, что система не сможет корректно выполнять свои функции и/или окажется плохо масштабируемой. Наконец, следует правильно определять кратность отношений между элементами, чтобы избежать логических ошибок в модели. Например, если связь между объектами подразумевает возможность множественных связей, но ошибочно задана как «1:1» вместо «1:\*», это может привести к ограничениям, не соответствующим реальным требованиям системы.

### **Рекомендации по формированию вариантов заданий**

Вариант системы берется из практического задания «Идентификация архитектуры информационной системы в соответствии с названием и кратким описанием». Допускается придумать/предложить свой вариант.

### **Рекомендуемое содержание отчетных материалов**

Результат формируется в виде PDF-файла со следующим содержанием:

1. Ф.И.О. студента, название практической работы и постановка задачи;
2. промпты и ответы в форме скриптов; скриншот визуализации одного из скриптов с пояснениями (5–10 предложений) – если применяется БЯМ;
3. финальная диаграмма с пояснениями (5–10 предложений);
4. рефлексивное заключение.

### **Контрольные вопросы**

1. Опишите этапы проектирования системы с использованием UML-диаграмм. На каком этапе используются структурные диаграммы, а на каком – поведенческие?
2. Каковы принципы декомпозиции компонентов в диаграмме компонентов?
3. Чем отличается тип связи «композиция» от «агрегация»?
4. Предположим, что у вас есть система управления интернет-магазином. Какие виды структурных UML-диаграмм будут наиболее полезны на разных этапах разработки?
5. Можно ли с помощью UML-диаграмм описать работу системы искусственного интеллекта? Если да, то какие виды диаграмм для этого подходят лучше всего?



## Разработка поведенческой UML-диаграммы

**Цель работы.** Освоить навыки моделирования работы (динамики) ИС с использованием поведенческой диаграммы UML. Развить навыки построения диаграмм активностей и(или) диаграмм последовательностей. Закрепить умение представлять бизнес-логику процессов в графической форме.

**Практическое задание.** В рамках практического задания должна быть смоделирована работа ИС в виде поведенческой диаграммы UML. Для этого может быть выбран один из типов диаграмм:

- диаграммы активностей;
- диаграммы последовательности.

При моделировании диаграммы необходимо соблюдение следующих условий:

- число состояний в диаграмме активностей – не менее шести;
- число акторов в диаграмме последовательностей – не менее трех;
- число сообщений – не менее шести.

Диаграмма должна сопровождаться пояснительным текстом (5–10 предложений), описывающим её назначение и основные процессы, отображённые на схеме. В данной работе разрешено использование БЯМ для автоматизации построения диаграмм и генерации пояснительного текста.

**Исходные данные:** ИС и её краткое описание.

**Требуемый результат:** поведенческая диаграмма UML, которая описывает один из процессов применения системы.

### Теоретические вводные и описание способов решения

UML (Unified Modeling Language) – язык графического моделирования, который позволяет строить модели как процессов, так и объектов (систем) в зависимости от постановки задачи и представляемой стороны (объекта и(или) процесса).

UML-диаграммы делятся на две категории:

1. структурные диаграммы;
2. поведенческие диаграммы.

В данной практической работе речь будет идти о поведенческих диаграммах.

Поведенческие диаграммы UML используются для моделирования динамических аспектов системы. Они описывают, как объекты взаимодействуют друг с другом во времени, какие процессы происходят и в каком порядке выполняются действия.

К поведенческим диаграммам UML относятся:

- Диаграмма активностей (Activity Diagram) – моделирует последовательность выполняемых действий.

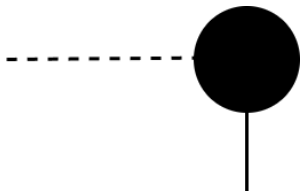
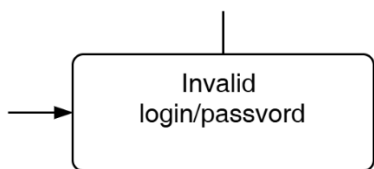
- Диаграмма случаев использования (Use Case Diagram) – описывает взаимодействие пользователей и системы.
- Диаграмма взаимодействий (Interaction Overview Diagram) – сочетает элементы диаграмм активностей и последовательностей, и подразделяется на:
  - Диаграмма последовательностей (Sequence Diagram) – показывает порядок взаимодействий между объектами.
  - Диаграмма коммуникаций (Communication Diagram) – аналог диаграммы последовательностей, на которой изображаются взаимодействия между частями композитной структуры или ролями кооперации. Отличается от диаграммы последовательностей тем, что явно указываются отношения между объектами, а время не используется.
  - Диаграмма временного анализа (Timing Diagram) – отображает изменение состояния объектов по времени (отображает внутреннюю динамику взаимодействия некоторого набора компонент системы).
- Диаграмма конечных автоматов (State Machine Diagram) – показывает изменение состояний объекта в зависимости от событий.

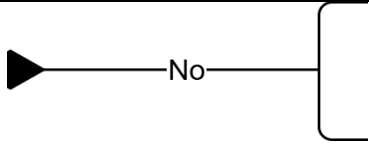
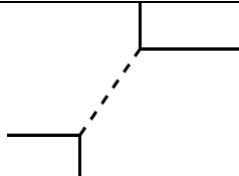
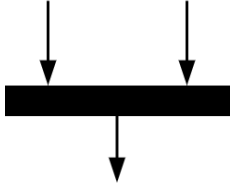
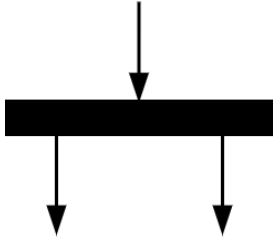
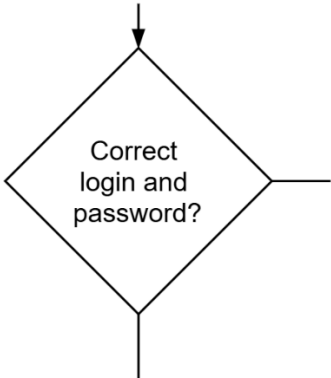
В рамках данной практической работы акцент сделан на диаграммы активностей и диаграммы последовательностей.



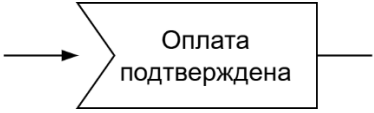
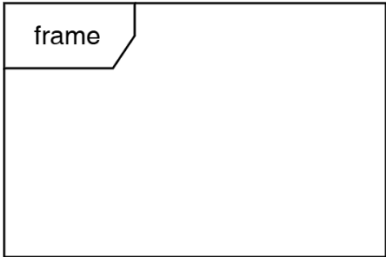
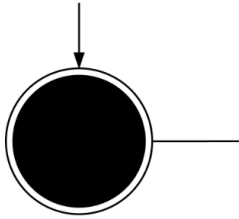
Диаграмма активностей используется для представления потока управления и данных внутри системы. Она похожа на блок-схемы, но обладает более широкими возможностями моделирования.

В таблице ниже приведен перечень основных элементов диаграммы активностей, их описание и графическое представление.

Таблица 6.1 – Поведенческие диаграммы. Диаграмма активностей

Вид диаграммы	Графические элементы	
Диаграмма активностей — отображает динамику системы. Представляет собой блок-схему действий состояний, схожую с блок-схемой алгоритмов, в том числе операций, ветвлений, циклов и терминальных состояний	Начальное состояние – обозначает начало процесса. Используется самостоятельно или с элементом «Комментарий», объясняющим условия старта процесса	
	Активное состояние – главный строительный блок диаграммы. Описывает состояние/действие,	

Вид диаграммы	Графические элементы	
	составляющее моделируемый процесс	
	Переход – обозначает переход из одного состояния в другое состояние	
	Сообщение — обозначает сообщение/поток данных, получаемых в состоянии	
	Синхронизатор, узел соединения – используется при переходе нескольких состояний/действий в одно. Может быть как горизонтальным, так и вертикальным	
	Узел разделения (разветвитель) – используется при переходе из одного состояния/действия в несколько. Может быть как горизонтальным, так и вертикальным.	
	Узел принятия решений – обозначает решение (аналог шлюза в BPMN). Из решения доступны как минимум 2 перехода с указанными условиями (в основном, «да» / «нет»)	

Вид диаграммы	Графические элементы	
	Комментарий – комментарий к узлу	
	Узел передачи сигнала – действие, которое на основе своих входов создает экземпляр сигнала и передает его объекту цели (другому узлу)	
	Узел приема сигнала – действие, которое ожидает наступление некоторого сигнала	
	Фрейм (рамка) – рамка, которая ограничивает состояния и связи между частями диаграммы для применения над ними тех или иных операций, например зацикливания	
	Комментарий – состояние завершения активностей (терминальное состояние)	

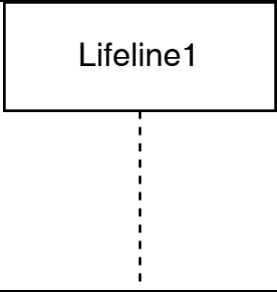
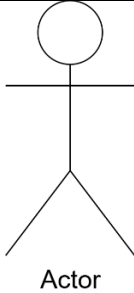
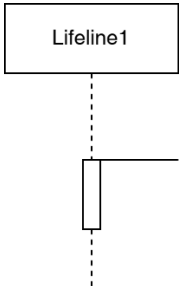
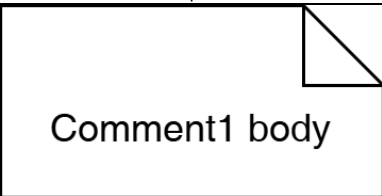
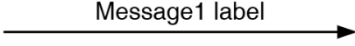
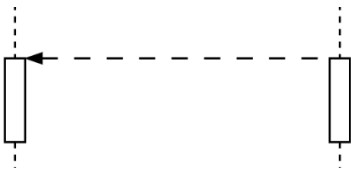
В качестве примеров применения диаграммы активностей можно привести следующий список, но не ограничиваться им:

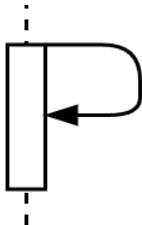
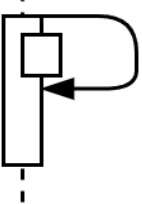
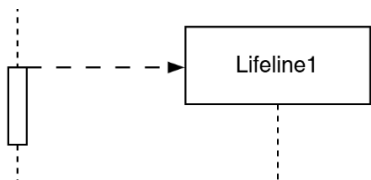
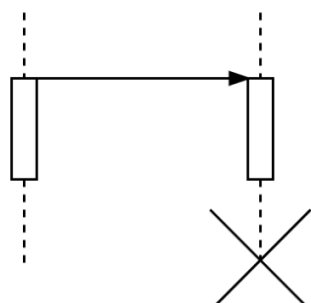
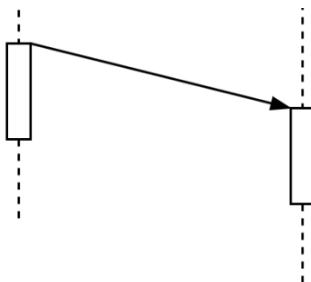
- моделирование бизнес-процессов;
- описание алгоритмов работы системы;
- анализ потоков данных;
- определение последовательности выполнения операций.

Диаграмма последовательностей показывает, как объекты взаимодействуют между собой во времени, передавая сообщения. Она удобна для моделирования сценариев использования системы.

В таблице ниже приведен перечень основных элементов диаграммы последовательностей, их описание и графическое представление.

Таблица 6.2 – Поведенческие диаграммы. Диаграмма последовательностей

Вид диаграммы	Графические элементы	
<p>Диаграмма последовательностей — описывает взаимодействие между объектами, участвующими в некотором сценарии. На диаграмме взаимодействие упорядочено в хронологическом порядке и отображено с применением «линий жизни» объектов</p>	<p>Линия жизни — указывает время деятельности конкретного участника</p>	
	<p>Актор — тип роли, которую играет сущность, взаимодействующая с субъектом (например, путем обмена сигналами и данными)</p>	
	<p>Активация — старт выполнения какой-либо операции</p>	
	<p>Комментарий — примечание к какому-либо элементу или стрелке</p>	
	<p>Вызываемое сообщение — сообщение-обращение к узлу для выполнения какой-либо операции</p>	
	<p>Ответное (возвратное) сообщение — ответное сообщение (сообщение о реакции) от вызываемого узла к вызывающему</p>	

Вид диаграммы	Графические элементы	
	<p>Собственное сообщение — собственное сообщение задает представления для коммуникаций в деятельности системы: узел/актор сообщает о промежуточном или финальном результате</p>	
	<p>Рекурсивное сообщение — сообщение об инициации и завершении вложенных (рекурсивных) действий в акторе/узле</p>	
	<p>Сообщение о создании — сообщение о создании (инициации) линии жизни другого актора/узла</p>	
	<p>Сообщение о прекращении — сообщение, инициирующее прекращение линии жизни актора/узла</p>	
	<p>Собственное сообщение — отражает латентность (сдвиг по времени) между посланием сообщения и стартом соответствующей операции</p>	

В качестве примеров применения диаграммы последовательностей можно привести следующий список, но не ограничиваться им:

- Описание последовательности выполнения операций.

- Анализ взаимодействий объектов в системе.
- Документирование сценариев использования системы.
- Разработка архитектуры программного обеспечения.

**Инструкция по использованию БЯМ для выполнения практического задания.** В настоящей работе может предусматриваться применение БЯМ для автоматизации процессов построения UML-диаграмм. БЯМ могут быть использованы для генерации описания модели, проверки корректности структуры диаграммы, выявления ошибок и уточнения логики взаимодействий. Для эффективного применения БЯМ рекомендуется выполнить ряд последовательных действий, которые позволят получить корректные результаты в текстовой или кодовой форме. Первое действие состоит в формировании промпта к БЯМ. Промпт может содержать как основной запрос, например: «Сгенерируй диаграмму последовательностей для процесса авторизации в системе», так и дополнительные параметры, такие как желаемый формат ответа (текст, код на PlantUML и т. д.), уровень детализации, ограничения по количеству элементов [6]. Качество ответа БЯМ зависит от четкости запроса. Полученный результат следует критически оценить, проверить на соответствие требованиям задания и при необходимости скорректировать путем уточнения запроса.

Ниже приведены основные шаги работы с сервисом БЯМ для выполнения данной практической работы, но не ограничиваясь ими.

1. Получение от сервиса БЯМ уточненного описания модели системы. На первом этапе необходимо сформировать общее представление о модели, которая будет отображена на UML-диаграмме. Для этого в промпт включается текстовое описание системы, выбранной для моделирования, а также уточняется формат ответа. Например: «Опиши процесс <...> в терминах UML. Определи, какие объекты участвуют во взаимодействии и какие ключевые процессы должны быть отражены в диаграмме.». На основе ответа необходимо проверить, соответствует ли предложенное описание системе, которая подлежит моделированию. При необходимости можно скорректировать запрос, добавив уточняющие детали.

2. Выбор типа диаграммы и генерация структуры. После уточнения модели следует выбрать тип UML-диаграммы: диаграмма активностей или диаграмма последовательностей. В зависимости от выбора формируется запрос к БЯМ. Пример запроса для диаграммы активностей: «Создай текстовое описание диаграммы активностей для процесса <название процесса>. Опиши основные действия и их последовательность.». Пример запроса для диаграммы последовательностей: «Составь текстовое описание диаграммы последовательностей для <...>. Укажи не менее трех акторов и шести сообщений.». Результат анализа БЯМ необходимо проверить: все ли основные процессы учтены, логична ли последовательность действий, нет ли пропущенных элементов. Для получения графического представления диаграммы можно

запросить у БЯМ код на языке PlantUML, который затем можно импортировать в соответствующие UML-редакторы.

3. Анализ ошибок и корректировка модели. После построения диаграммы важно проверить ее корректность. Для этого можно воспользоваться БЯМ, запросив анализ ошибок. Пример запроса: «Какие ошибки могут быть в следующей диаграмме последовательностей <текст описания диаграммы>? Проведи анализ и предложи исправления.». На основе полученных рекомендаций возможна доработка диаграммы.

**Общие рекомендации по использованию БЯМ.** Использование БЯМ позволяет упростить процесс создания поведенческих UML-диаграмм, но важно правильно формулировать запросы и критически оценивать полученные результаты. Ввиду этого предлагается обобщённый порядок работы с БЯМ, который может быть уточнен или изменен в зависимости от требований к ИС: описать систему и запросить уточненное описание; выбрать тип диаграммы и получить ее текстовое описание; сгенерировать код UML (при необходимости); провести анализ ошибок и исправить диаграмму; провести итоговую проверку и внести финальные правки. Говоря об общих рекомендациях, рекомендуется не копировать текст БЯМ без изменений, проверять соответствие поставленным требованиям, использовать БЯМ как помощника, а не как единственный источник информации.

**Типичные ошибки, которые допускаются студентами при выполнении практического задания.** Во-первых, не следует путать акторов и объекты системы. В диаграмме последовательностей акторы – это внешние сущности (пользователи или другие системы), взаимодействующие с моделью, а объекты системы – это внутренние компоненты. Во-вторых, важно корректно определять последовательность взаимодействий. Ошибкой является пропуск ключевых шагов процесса или нарушение логики их выполнения. В-третьих, необходимо учитывать потоки управления и данных. В диаграмме активностей студенты могут забыть о решающих условиях (развилках), что приводит к линейному сценарию без альтернативных путей. В-четвертых, следует избегать перегруженности диаграммы. Частая ошибка – добавление избыточных элементов, которые не несут ценности для модели. В-пятых, важно правильно оформлять сообщения в диаграмме последовательностей

#### **Рекомендации по формированию вариантов заданий**

Вариант системы берется из практического задания «Идентификация архитектуры информационной системы в соответствии с названием и кратким описанием». Допускается придумать/предложить свой вариант.

#### **Рекомендуемое содержание отчетных материалов**

Результат формируется в виде PDF-файла со следующим содержанием:

1. ФИО обучающегося;
2. название настоящей практической работы;



3. цель работы;
4. постановка задачи в соответствии с выбранным вариантом задания;
5. промпты и ответы в форме скриптов; скриншот визуализации одного из скриптов с пояснениями (5–10 предложений)
6. финальная диаграмма с пояснениями (5–10 предложений);
7. рефлексивное заключение.

#### **Контрольные вопросы**

1. Какие типы поведенческих диаграмм UML существуют?
2. В чем различие между диаграммой активностей и диаграммой последовательностей?
3. Какие ключевые элементы присутствуют в диаграмме активностей?
4. Как моделируются взаимодействия между объектами в диаграмме последовательностей?
5. Как можно использовать БЯМ при разработке UML-диаграмм?

## **Заключение**

В настоящем учебно-методическом пособии рассмотрены средства и способы решения задач в проектировании ИС, включая разработку ТЗ, ресурсный анализ и формирование (в том числе генерация с применением БЯМ) UML-диаграмм.

В настоящем учебно-методическом пособии предлагается применение инструментов системного анализа для идентификации и понимания как ИС, так и соответствующих объектов/процессов автоматизации. В них включено рассмотрение вопросов идентификации или выбора архитектуры проектируемой системы, а также подготовки ТЗ на ее разработку. Применение БЯМ с учетом их особенностей позволяет существенно ускорить моделирование информационных систем, в том числе с применением UML-языка.

После успешного выполнения практических работ у студентов будут поставлены или развиты навыки в быстром решении задач разработки, реализации и внедрении ИС за счет применения как классических, так и передовых средств моделирования и анализа соответствующих процессов.

## Список литературы

1. Шушура О.М. Методические указания к самостоятельной работе по дисциплине «Системный анализ и проектирование компьютерных систем» для студентов специальности «Системный анализ и управление», Донецкий национальный технический университет, Донецк, 2009 – 62 с.
2. Архитектурные решения информационных систем : Учебник для вузов / А. И. Водяхо, Л. С. Выговский, В. А. Дубенецкий, В. В. Цехановский. — 3-е изд., стер. — Санкт-Петербург : Лань, 2022. — 356 с. — ISBN 978-5-507-44710-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/254624> (дата обращения: 07.04.2025). — Режим доступа: для авториз. пользователей.
3. ГОСТ 34.602—2020. Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы. Изд. Офиц. – М., 2021. 12 с.
4. Марка, Д.А.. Методология структурного анализа и проектирования [Пер. с англ.] / Дэвид А. Марка, Клемент Л. МакГоуэн; Предисл. Д. Т. Росса. — [Москва] : Фирма "Мета Технология", 1993. — 240 с. : ил. : 29 см.; ISBN 5-7395-0007-9 (В пер.).
5. Новиков, Ф. А. Учебно-методическое пособие по дисциплине «Анализ и проектирование на UML» : учебно-методическое пособие / Ф. А. Новиков. — Санкт-Петербург : НИУ ИТМО, 2007. — 286 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/43540> (дата обращения: 07.04.2025). — Режим доступа: для авториз. пользователей.
6. Ананьева Е. Как быстро построить UML-диаграмму через ChatGPT. [электронный ресурс]: режим доступа: <https://getanalyst.ru/database/create-uml-in-chatGPT>, загл. с экрана (посл. доступ 12.03.2025).

Лаушкина Анастасия Александровна  
Духанов Алексей Валентинович

## **Основы проектирования информационных систем с применением больших языковых моделей**

**Учебно-методическое пособие**

В авторской редакции

Редакционно-издательский отдел Университета ИТМО

Зав. РИО

Н.Ф. Гусарова

Подписано к печати

Заказ №

Тираж

Отпечатано на ризографе



**Редакционно-издательский отдел**  
**Университета ИТМО**  
197101, Санкт-Петербург, Кронверкский пр., 49, литер А