



МОСКОВСКИЙ УНИВЕРСИТЕТ ИМ. С.Ю.ВИТТЕ

Кафедра «Информационных систем»

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ПО ОРГАНИЗАЦИИ ПОДГОТОВКИ, ВЫПОЛНЕНИЮ И ЗАЩИТЕ
КУРСОВОЙ РАБОТЫ ПО ДИСЦИПЛИНЕ
«ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ»**

Москва

Авторы-составители:

Блощук А.А., к.т.н., доцент кафедры информационных систем

Королькова И.А., руководитель образовательной программы

Рецензент:

Зайцев С.А., к.т.н., доцент, декан факультета информационных технологий

Методические указания рассмотрены на заседании кафедры информационных систем.

Методические указания по организации подготовки, выполнению и защите курсовой работы по дисциплине «Проектирование информационных систем»: учебно-методические материалы // А.А. Блощук, И.А. Королькова М.: МУ им. С.Ю. Витте, 2021. – 83 с.

В предлагаемых методических указаниях для обучающихся по направлению подготовки 38.03.05 Бизнес-информатика, направленность (профиль): «Электронный бизнес» изложены основные подходы и принципы подготовки курсовой работы по дисциплине «Проектирование информационных систем», рассматривается порядок аттестации, формулируются критерии оценки курсовой работы. Приведены основные требования к содержанию курсовой работы.

Указания предназначены для обучающихся кафедры информационных систем.

Содержание

1. ОБЩИЕ ПОЛОЖЕНИЯ.....	4
2. ВЫБОР ТЕМЫ КУРСОВОЙ РАБОТЫ И ПОДБОР ИСПОЛЪЗУЕМЫХ ИСТОЧНИКОВ.....	5
3. СТРУКТУРА КУРСОВОЙ РАБОТЫ.....	7
3.1 Общие требования к структуре.....	7
3.2 Основные теоретические сведения о проектировании автоматизированной информационной системы.....	9
3.3 План и методика выполнения курсовой работы.....	19
3.3.1 Стадия формирования концепции ИС.....	19
3.3.1.1 Общая характеристика объекта проектирования.....	20
3.3.1.2 Выделение функциональной подсистемы ИС и описание бизнес- процесса.....	21
3.3.1.3 Разработка технического задания на создание системы управления бизнес-процессом.....	23
3.3.2 Стадия проектирования ИС.....	24
3.3.2.1 Постановка задачи.....	24
3.3.2.2 Разработка локального классификатора информации.....	26
3.3.2.3 Информационное моделирование бизнес-процесса.....	31
3.3.2.4 Разработка динамической модели работы подсистемы.....	34
3.3.3 Стадия реализации проектных решений.....	35
3.3.3.1 Выбор ПО для реализации функциональной подсистемы.....	36
3.3.3.2 Реализация интерфейса подсистемы.....	42
3.3.3.3 Разработка инструкции для пользователя функциональной подсистемы.....	43
4. ОФОРМЛЕНИЕ КУРСОВОЙ РАБОТЫ.....	44
5. ПОРЯДОК АТТЕСТАЦИИ И ЗАЩИТЫ КУРСОВОЙ РАБОТЫ.....	46
Приложение 1.....	48
Приложение 2.....	49
Приложение 3.....	51
Приложение 4.....	52
Приложение 5.....	83

1. ОБЩИЕ ПОЛОЖЕНИЯ

Курсовая работа является важным элементом учебного процесса и представляет собой итоговый предэкзаменационный этап в изучении дисциплины; представляет собой логически завершенное и оформленное в виде текста изложение обучающимся содержания исследованных проблем, задач и методов их решения в уголовно-правовой сфере.

Значение курсовой работы заключается в том, что обучающийся самостоятельно выполняет ее в письменной форме, развивая тем самым навыки творческой деятельности, овладевает методами современных научных исследований, углублено изучает вопросы, темы, разделы учебной дисциплины и формирует компетенции.

Написание курсовых работ предусмотрено учебным планом и выступает обязательным элементом программы обучения по значимым дисциплинам направления подготовки.

Курсовая работа дает обучающемуся возможность углубить, систематизировать и закрепить теоретические и практические знания, приобрести навыки творческого поиска и обработки информации, систематизированного и логического изложения материала, работы с нормативными актами, статистическими материалами, литературой, справочным и библиографическим материалом, Интернет-ресурсами по теме.

Цель курсовой работы состоит в:

- ~ систематизации и углублении теоретических и практических знаний;
- ~ приобретении навыков самостоятельной работы с различными источниками, документами и т.п.;
- ~ овладении методами и способами аналитической и научно-исследовательской деятельности;
- ~ умении выявлять проблемы и находить способы их решения;

В процессе выполнения курсовой работы формируются компетенции в соответствии с направлением и профилем подготовки.

Вопросы, рассматриваемые в курсовой работе, основываются на:

- ~ существующей законодательной базе Российской Федерации;
- ~ обобщении отечественного и зарубежного опыта;
- ~ формировании личной точки зрения и критического взгляда автора на рассматриваемые проблемы.

В ходе написания курсовой работы обучающийся должен показать умение использовать общетеоретические и специальные знания по выбранной проблематике.

В результате выполнения курсовой работы по данной дисциплине студент

должен научиться:

- определять границы автоматизированной информационной системы управления бизнес-процессами предприятия и составляющих ее функциональных подсистем;
- определять внутренние и внешние информационные потоки;

- выделять функциональные и обеспечивающие подсистемы информационной системы управления бизнес-процессами предприятия;
- выполнять постановку задачи на проектирование ИС на примере ее функциональной подсистемы;
- применять математический аппарат для обоснования выбора оптимального проектного решения;
- применять методы моделирования для описания и анализа предметной области и проектирования информационной системы.

2. ВЫБОР ТЕМЫ КУРСОВОЙ РАБОТЫ И ПОДБОР ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

Тематика курсовых работ, требования к ним и указания по их выполнению доводятся до сведения обучающихся в начале соответствующего семестра.

Тема курсовой работы у всех обучающихся единая – **«Проектирование информационной системы для решения задач профессиональной деятельности»**. Данная тема указывается на титульном листе.

Решение прикладной задачи выполняется по одному из предложенных вариантов предметных областей. **Выбор варианта предметной области обучающийся осуществляет самостоятельно** из перечня в Приложении 2. Выбранный вариант предметной области указывается во введении, в целях и задачах курсовой работы.

Для подготовки к выполнению курсовой работы рекомендуется самостоятельное изучение учебной и научной литературы, использование справочной литературы и интернет - ресурсов. По мере изучения темы следует разобрать приведенные задачи с обязательным выполнением задач для самостоятельного решения.

Подготовка к написанию курсовой работы во многом зависит от правильной подготовки к выполнению работы, которую можно условно разделить на следующие этапы.

1 этап. Предусматривает выбор варианта предметной области, ее осмысление, определение целевых установок, на основе чего важно наметить главные вопросы, подлежащие рассмотрению, и их краткое содержание.

2 этап. Включает подбор литературы по теме курсовой работы, работу с каталогами библиотек, библиографическими указателями, Интернет-ресурсами, справочно-правовыми системами и т.п..

При выборе литературы возникает множество трудностей, которые можно избежать, если придерживаться нескольких *правил*:

обращать внимание на мета-данные книги. В оглавлении важно отметить те разделы и параграфы, которые представляют интерес для

раскрытия темы. В предисловии можно найти ответы на такие вопросы, как цель написания книги, основные направления исследования, общий характер работы;

обращать внимание на год издания, т.к. требуется соблюсти условие об актуальности используемых данных. **Актуальной считается литература (за исключением словарей и фундаментальных первоисточников) сроком после выхода не более 5 лет;**

целесообразно при ознакомлении с литературными источниками делать выписки, конспекты, рефераты и т.п. Изучение литературы должно быть глубоким, сплошным;

при конспектировании литературы и ее последующего использования в своей курсовой работе обучающемуся необходимо сформулировать **личное мнение** по рассматриваемой проблеме. В этом случае используется правило, когда автор выступает во множественном числе и вместо «я» употребляет «на наш взгляд», «по нашему мнению»;

при написании курсовой работы следует активно использовать, научные статьи, опубликованные в ведущих IT-изданиях и на интернет-ресурсах для IT-специалистов, официальные образовательные Интернет-ресурсы, а также официальные сайты разработчиков программного обеспечения, IT-компаний и других организаций, являющихся источниками необходимой информации для раскрытия темы курсовой работы. Например, обучающиеся могут пользоваться интернет-ресурсом для IT-специалистов - <https://habr.com/ru/all/> и электронной научной библиотекой – <https://www.elibrary.ru/defaultx.asp>.

3 этап. Выполнение курсовой работы по плану, изложенному в разделе 3 данных методических указаний.

Курсовая работа посвящена разработке функциональной подсистемы исследуемого объекта, в качестве которого может рассматриваться производственное предприятие, государственное учреждение, частная коммерческая фирма, а также их структурные подразделения. При этом предусматривается закрепление умений и навыков типового автоматизированного проектирования с применением современных методологий и инструментальных средств.

В процессе курсовой работы студент должен:

- Провести предпроектное обследование исследуемого объекта.
- Формализовать результаты предпроектного обследования в виде комплекса моделей.
- Выполнить постановку задачи на проектирование функциональной подсистемы информационной системы управления бизнеспроцессами.
- Разработать элементы информационного и программного обеспечения функциональной подсистемы.

- Спроектировать интерфейс пользователя функциональной подсистемы.
- Разработать проектную документацию на функциональную подсистему.
- Оформить пояснительную записку, графическую и программную часть проекта.

Кроме того, в состав проекта могут быть включены дополнительные разделы, связанные с научно-исследовательской работой.

3. СТРУКТУРА КУРСОВОЙ РАБОТЫ

3.1 Общие требования к структуре

При выполнении курсовой работы необходимо придерживаться ряда требований к ее структуре. Прежде всего, все курсовые работы должны состоять из введения, аналитической главы, практической главы, заключения, списка литературы. К основному тексту даются приложения, которые размещаются в конце работы.

Типовые требования к курсовой работе определяют, обязательные к применению, стандарты:

ГОСТ 7.32.-2017. Отчет о научно-исследовательской работе. Структура и правила оформления;

ГОСТ 7.0.100-2018. Библиографическая запись. Библиографическое описание. Общие требования и правила составления;

ГОСТ Р 7.0.5-2008 Библиографическая ссылка. Общие требования и правила составления;

Структура курсовой работы также определяется ее исследовательскими задачами, т.е. каждый параграф в ней должен работать на решение одной задачи в рамках достижения сформулированной цели.

Структура курсовой работы должна включать в себя следующие составные части:

- титульный лист (приложение 3);
- оглавление (приложение 4);
- введение;
- текст работы, структурированный по главам (параграфам, разделам);
- заключение;
- список литературы;
- приложения (при необходимости).

Оглавление должно быть создано с помощью инструмента «Автосодержание» текстового редактора и **собираться автоматически из заголовков глав и параграфов.**

Во **введении** автор обосновывает тему курсовой работы, ее актуальность, степень разработанности, кратко характеризуя современное состояние научной проблемы (вопроса), которой посвящена работа,

определяет цель, задачи, объект и предмет исследования.

Актуальность темы. Обоснование того, почему нужно заниматься исследованием выбранной темы. **Обоснование актуальности темы** показывает главное – суть проблемной ситуации (противоречивой ситуации требующей своего разрешения). Актуальность темы означает ее связь с конкретными потребностями пользователей, отражает важность, своевременность выбранной темы, ее значимость.

Цели и задачи. Количество и название задач как правило совпадает с количеством параграфов работы, впереди ставится соответствующий глагол (рассмотреть, охарактеризовать и т.п.). От доказательства актуальности выбранной темы обучающийся должен логично перейти к **определению цели работы**. Как правило, целью исследования является разработка предложений или методических рекомендаций на основе анализа существующей ситуации и полученных практических результатов анализа объекта исследования.

Задачи указывают на основные направления работы и начинаются с совершенных глаголов: рассмотреть..., охарактеризовать..., проанализировать..., оценить..., выявить..., разработать..., спроектировать..., автоматизировать..., разработать рекомендации..., предложить методику..., повысить экономическую эффективность... и т.п.

Основные разделы курсовой работы – главы (не менее двух) и параграфы, которые содержат систематизированное изложение и анализ основных проблем исследуемой темы. В главах и параграфах излагаются теоретические аспекты предметной области на основе анализа опубликованной литературы, формулируются ключевые аспекты автоматизации, выделяются 5-6 основных предметно значимых сущностей, планируются общие принципы ведения различных видов учета в разрезе этих сущностей (теоретическая часть), создается информационная база, решаются задания представленные в главе 2 данных методических указаний, описываются полученные результаты и приводятся скриншоты конфигурации (практическая часть). Содержание теоретической и практической частей курсовой работы определяется в зависимости от выбранной предметной области.

Заключение представляет собой краткое последовательное, логически стройное изложение полученных и описанных в основной части результатов, выводов исследования, построенных на анализе соотношения полученных результатов с общей целью и конкретными задачами исследования. Число выводов не должно быть большим, обычно оно определяется количеством поставленных задач.

Список литературы представляет собой оформленный в соответствии с установленными правилами перечень использованных в процессе исследования избранной темы: законов и подзаконных нормативных правовых актов, учебной и научной литературы, материалов периодической печати, материалов юридической практики, электронных ресурсов. Список литературы должен иметь следующую структуру:

нормативные правовые акты;
учебники, учебные пособия, монографии, комментарии к
законодательству, словари, энциклопедии;
научные статьи, материалы из периодических изданий;
диссертации, авторефераты диссертаций;
материалы юридической практики, справочно-
статистические материалы;
электронные ресурсы.

При использовании в тексте курсовой работы положений, цитат, заимствованных из литературы, обучающийся обязан делать ссылки на них в соответствии с установленными правилами. Заимствования текста без ссылки на источник не допускаются.

Содержание курсовой работы должно соответствовать следующим основным требованиям:

самостоятельность исследования;
наличие анализа специальной литературы и интернет-источников по предметной области;
наличие работоспособного файла с исходным программным кодом (например, конфигурации 1С – для проектов на базе 1С:Предприятие; файлов .py – для проектов на Python и т.д.), сопутствующих файлов (например, файлов базы данных);
логичность изложения содержания курсовой работы, подробное описание процесса разработки конфигурации, аргументированность выводов и предложений;
научно-практическая значимость курсовой работы.

В процессе выполнения курсовой работы запрещено пользоваться устаревшим ПО, таким как BPWin и ERWin!

3.2 Основные теоретические сведения о проектировании автоматизированной информационной системы

Структура автоматизированной информационной системы.

Каждый исследуемый объект можно рассматривать как систему, стремящуюся в своем функционировании к достижению определенной цели. В качестве примера можно назвать систему образования, энергетическую, транспортную, производственную систему и т.д. Для того чтобы рассматриваемая в качестве исследуемого объекта система выполняла заданную функцию и при этом достигала требуемого результата, необходимо управлять ею. Для управления сложными системами существуют системы управления. Для повышения эффективности систем управления исследуемыми объектами используются новейшие технические, технологические и программные средства.

Управление связано с обменом информацией между компонентами системы, а также системы с внешней средой. В процессе управления получают сведения о состоянии системы в каждый момент времени, о достижении (или не достижении) заданной цели с тем, чтобы воздействовать на систему и обеспечить выполнение управленческих решений. Таким образом, любой системе управления исследуемым объектом соответствует своя автоматизированная информационная система.

АИС – совокупность информации, экономико-математических методов и моделей, технических, технологических и программных средств и специалистов, предназначенная для обработки информации и принятия управленческих решений.

Основными составными частями АИС являются: аппарат управления; автоматизированная информационная технология; комплекс функциональных подсистем; комплекс обеспечивающих подсистем. Структура АИС с указанием основных информационных потоков между ее компонентами представлена на рисунке 1.

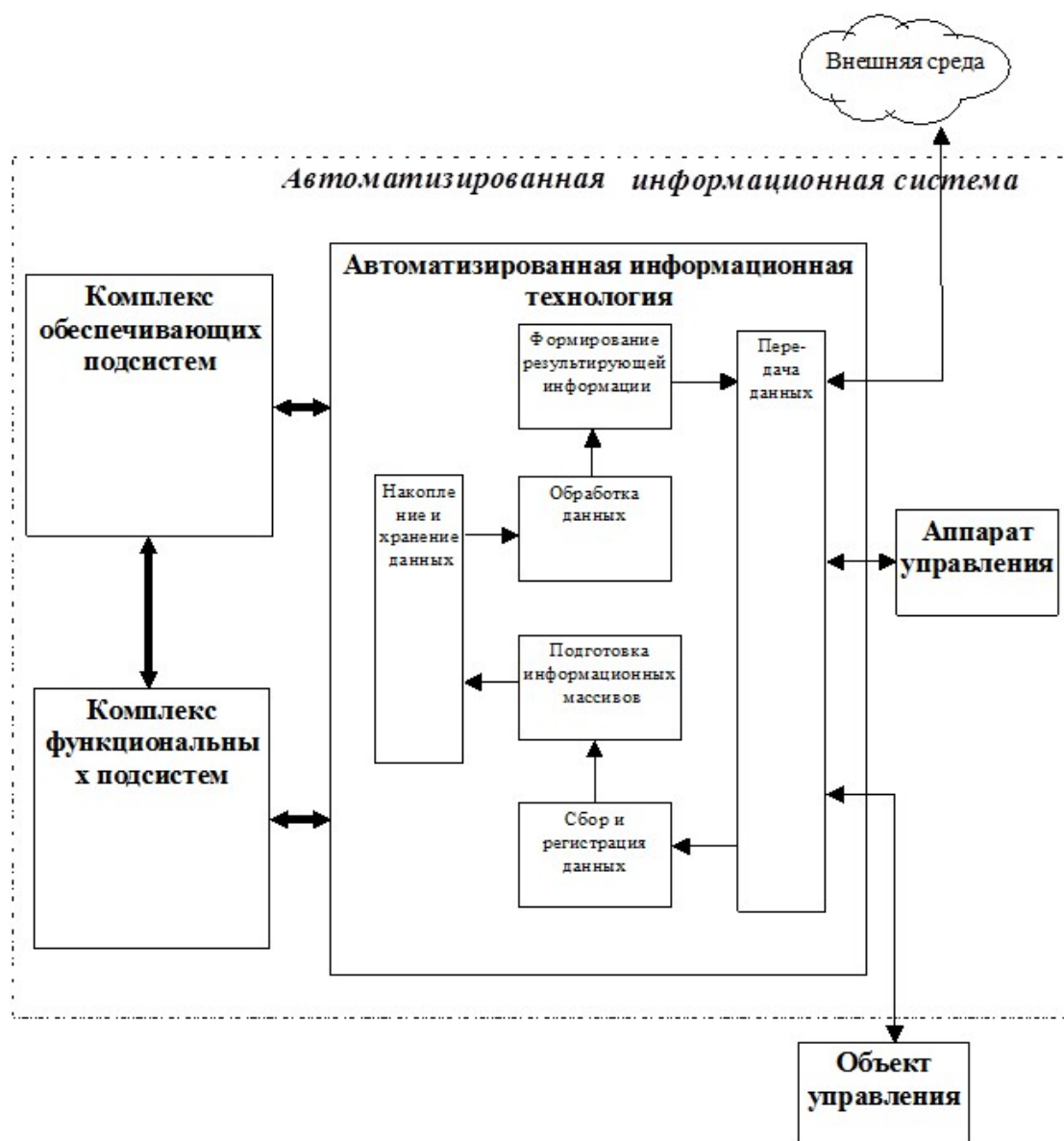


Рисунок 1. – Структура АИС

Обеспечивающие подсистемы предназначены для поддержания функционирования АИС в заданном режиме. В процессе проектирования АИС разрабатывают следующие виды обеспечения: информационное, лингвистическое, математическое, программное, техническое, организационное, правовое, эргономическое.

Функциональная подсистема ИС представляет собой комплекс исследуемых задач, тесно связанных между собой с помощью информационных, материальных и финансовых потоков. Состав функциональных подсистем во многом определяется особенностями ИС, ее отраслевой принадлежностью, размером, характером деятельности предприятия.

Состав, порядок и принципы взаимодействия функциональных подсистем устанавливаются с учетом цели, стоящей перед исследуемым

объектом. Каждая функциональная подсистема характеризуется следующими элементами:

- своим специфическим объектом управления;
- внешними входами и выходами;
- внутренней сравнительно замкнутой информационной системой;
- особым кругом задач, возникающих и решаемых в процессе управления;
- перечнем лиц из аппарата управления, исполняющих в соответствии со своими должностными обязанностями различные роли в процессе решения экономических, производственных, управленческих задач.

Примерами функциональных подсистем могут служить: подсистема маркетинга, подсистема бухгалтерского учета, подсистема оперативного учета хода производства, подсистема организационно-распорядительного документооборота, подсистема материально-технического снабжения.

Например, функциональную подсистему бухгалтерского учета можно описать следующим образом:

объект управления – процессы бухгалтерского и налогового учета;

внешние входы – исходная информация от производственных подразделений, аппарата управления предприятием, органов государственного управления; **внешние выходы** – бухгалтерская и налоговая отчетность, отчеты для руководства предприятия; **внутренняя сравнительно замкнутая ИС** – 1С: Бухгалтерия; **некоторые задачи функциональной подсистемы** – начисление зарплаты, ведение кассовых операций, обработка документации по поступлению товарно-материальных ценностей, подготовка отчетов и т.д.; **аппарат управления** – бухгалтер, главный бухгалтер, материально-ответственное лицо, кассир и т.д.

Аналогично приведенному примеру в курсовой работе выполняется описание функциональной подсистемы в заданной предметной области. При этом в качестве **объекта управления** может рассматриваться бизнес-процесс, административный процесс, производственный процесс, а также, в случае выполнения курсовой работы для конкретного предприятия, деятельность его структурного подразделения. **Внешние входы** процесса соответствуют внешним (по отношению к процессу или подразделению) источникам информации, необходимой для его выполнения. **Внешние выходы** соответствуют внешним (по отношению к процессу или подразделению) потребителям информации, для которых предназначены результаты выполнения процесса или результаты функционирования подразделения. **Внутренняя сравнительно замкнутая информационная система,**

сопровождающая выполнение функций процесса, может быть описана только в случае выполнения курсовой работы на базе конкретного предприятия, где уже внедрена информационная система. **Задачи функциональной подсистемы** формулируются в соответствии с документами, регламентирующими выполнение процесса или работу подразделения. К таким документам могут относиться регламент процесса, инструкция, правила, положение о структурном подразделении. **Аппарат управления** указывается в соответствии с организационно-штатной структурой предприятия или в соответствии с регламентом, описывающим данный бизнес-процесс.

При описании элементов функциональных подсистем следует придерживаться терминологии, закрепленной в регламентных и нормативных документах.

Современные методологии проектирования ИС отдают предпочтение процессному подходу, который предполагает смещение акцентов от управления отдельными структурными подразделениями исследуемого объекта на управление сквозными бизнес-процессами, связывающими деятельность всех подразделений.

Каждый бизнес-процесс проходит через ряд подразделений, т. е. в его выполнении участвуют различные специалисты. Чаще всего приходится сталкиваться с ситуацией, когда собственно процессами никто не управляет, а управляют лишь подразделениями. Более того, организационная структура предприятий строится без учета возможностей оптимизации деловых процессов, обеспечивающих необходимые функции. Процессный подход позволяет устранить фрагментарность в работе, организационные и информационные разрывы, дублирование, нерациональное использование финансовых, материальных и кадровых ресурсов.

Основной принцип процессного подхода определяет структурирование системы управления в соответствии с деятельностью и бизнес-процессами предприятия, а не в соответствии с его организационно-штатной структурой. Процессный подход требует комплексного изучения различных сторон жизни организации правовых основ и правил деятельности, организационной структуры, функций и показателей результатов их исполнения, интерфейсов, ресурсного обеспечения, организационной культуры. В ходе анализа бизнеспроцессов детально исследуются сферы ответственности подразделений, их руководителей и сотрудников. Это позволяет установить адреса владельцев бизнес-процессов, в результате чего процессы перестают быть бесхозными, создаются условия для разработки и внедрения систем стимулирования и ответственности за конечные результаты, определяются моменты и процедуры передачи ответственности.

В результате анализа создается модель деятельности «как есть». Обработка этой модели с помощью различных аналитических методов позволяет проверить, насколько бизнес-процессы рациональны, а также определить, является ли та или иная операция ориентированной на общественно значимый конечный результат или излишней бюрократической процедурой.

В результате применения процессного подхода создаются процессные потоковые модели, описывающие последовательное преобразование материальных и информационных потоков в ходе реализации какой-либо производственной функции или функции управления. На верхних уровнях модели описывается логика взаимодействия участников процесса, на нижних - технология работы отдельных специалистов на своих рабочих местах.

Процессная модель исследуемого объекта строится с учетом следующих положений:

- Верхний уровень модели должен отражать только контекст деятельности (т.е., контекстная диаграмма должна отражать взаимодействие моделируемого предприятия с внешним миром).
- На втором уровне должны быть отражены тематически сгруппированные бизнес-процессы предприятия и их взаимосвязи в виде основных направлений деятельности.
- Каждое из направлений деятельности должно быть детализировано на бизнес-процессы.
- Детализация бизнес-процессов осуществляется посредством бизнес-функций (или задач, решаемых при выполнении бизнес-процессов).
- Бизнес-функции описываются последовательностью элементарных технологических операций.
- Описание элементарной операции осуществляется с помощью мини спецификации.

Курсовая работа предполагает построение процессной модели предметной области, начиная с третьего уровня. Таким образом, контекстная диаграмма функциональной модели «как есть» будет представлять собой описание системы управления бизнес-процессом (рис. 2).



Рисунок 2. – Пример описания функциональной подсистемы в нотации IDEF0

Декомпозиция контекстной диаграммы описывает перечень задач, решаемых сотрудниками предприятия при выполнении указанного бизнеспроцесса (рис. 3).

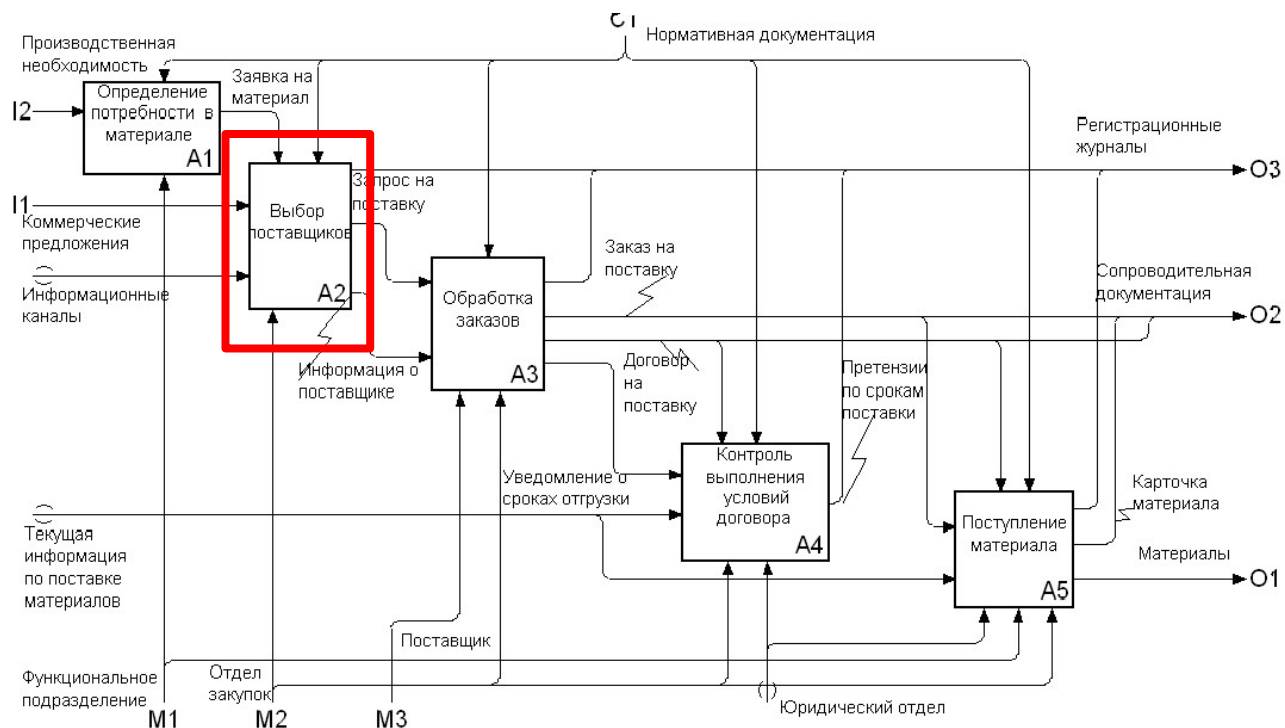


Рисунок 3. – Перечень задач функциональной подсистемы

На следующем уровне декомпозиции рассматриваются логически взаимосвязанные элементарные технологические операции, детально описывающие выполнение бизнес-функции (решение исследуемой задачи). Одной из нотаций функционального моделирования, наиболее удобной для детального представления решения задачи, является нотация ЕРС (рис. 4).

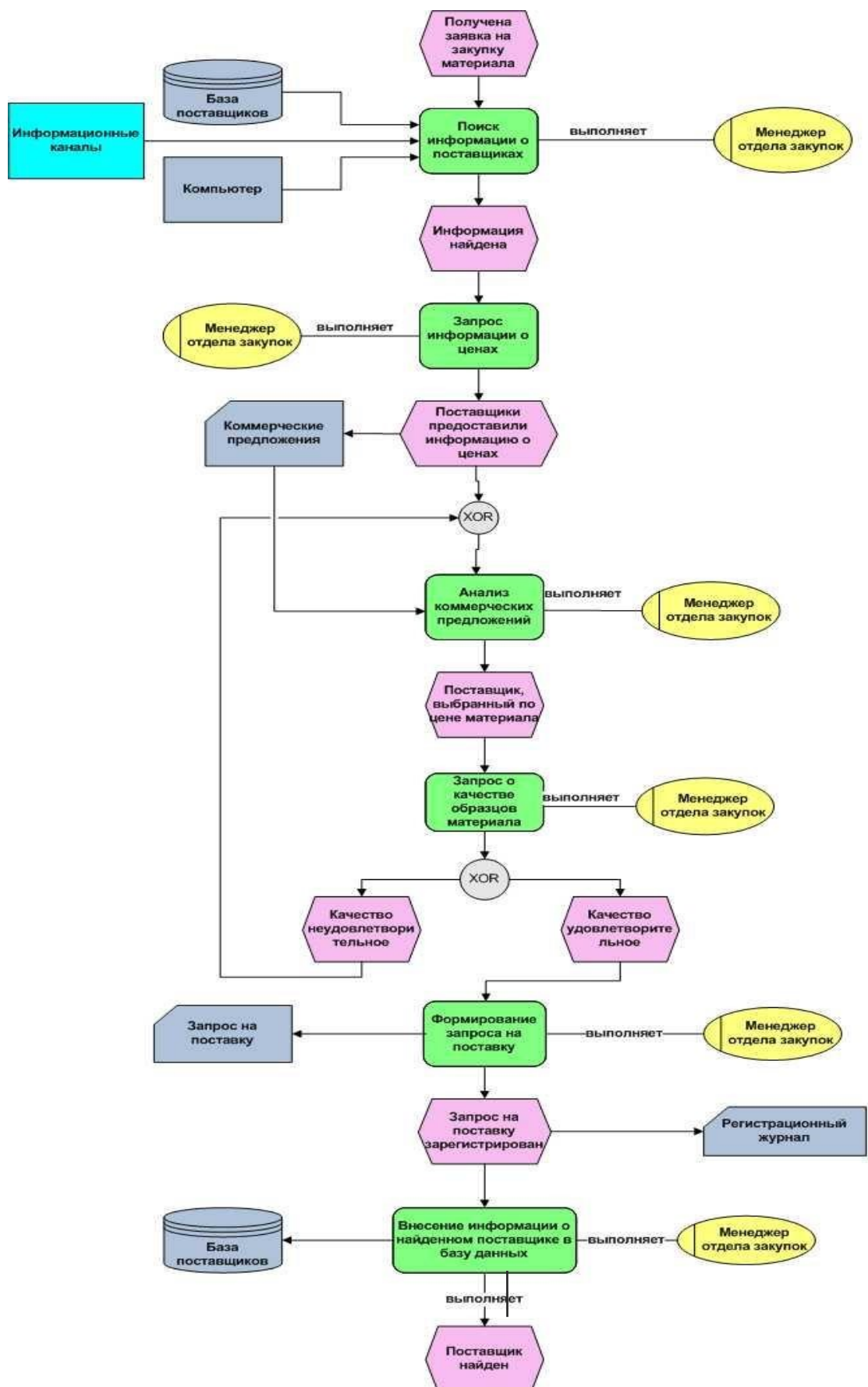


Рисунок. 4 – Задача «Выбор поставщиков» в нотации EPC *Жизненный цикл ИС.*

Одним из базовых понятий методологии проектирования ИС является понятие жизненного цикла. **Жизненный цикл** – период создания и использования ИС, охватывающий ее различные состояния, начиная с момента возникновения необходимости в данной автоматизированной информационной системе и заканчивая моментом ее полного выхода из употребления у пользователей.

Существует целый ряд стандартов, регламентирующих жизненный цикл информационных систем, программного обеспечения, а также процессы их разработки. Среди наиболее известных стандартов можно выделить следующие:

ISO/IEC 12207 (*Standard for Information Technology — Software Life Cycle*

Processes – «Процессы жизненного цикла ПО»);

ISO/IEC 15288 «Системная инженерия – Процессы жизненного цикла систем» (табл. 1);

ГОСТ 34.601-90 «Автоматизированные системы. Стадии создания».

Таблица 1 Стадии создания системы по стандарту ISO/IEC 15288

Стадия	Описание
Формирование концепции (стадия замысла)	Анализ потребностей, выбор концепции и проектных решений
Разработка	Проектирование системы
Реализация (производство)	Изготовление системы: Разработка программного обеспечения ИС и/или адаптация существующей ИС и/или доработка функционала существующей ИС
Эксплуатация	Ввод в эксплуатацию и использование системы
Поддержка	Обеспечение функционирования системы
Снятие с эксплуатации	Прекращение использования, демонтаж, архивирование системы

От качества проектных работ зависит эффективность функционирования системы. Поэтому каждая стадия разделяется на несколько этапов, в течение которых выполняется определенная последовательность работ. Результаты работ фиксируются в соответствующих документах.

В рамках курсовой работы должны быть рассмотрены:

- Стадия формирования концепции. На этой стадии проводится анализ предметной области. Требуется выделить функциональную подсистему ИС и описать исследуемый бизнес-процесс. Для некоторых тем курсовой работы, по указанию руководителя, на данной стадии необходимо выполнить разработку **технического задания** на создание системы управления бизнес-процессом.
- Стадия проектирования ИС на примере выделенной на предыдущем этапе функциональной подсистемы, поддерживающей выполнение одного или нескольких рассматриваемых бизнес-процессов. На данной стадии необходимо выполнить **информационное моделирование исследуемого бизнес-процесса**.
- Стадия реализации на примере **автоматизации одной из задач бизнес-процесса**, связанной с необходимостью автоматизации ее элементов. На данной стадии рассматриваются разработка (или адаптация) программного обеспечения для проектных решений функциональной подсистемы системы управления бизнес-процессами. Данная стадия является заключительным этапом курсовой работы

3.3 План и методика выполнения курсовой работы

3.3.1 Стадия формирования концепции ИС

Основная задача обследования – оценка реального объема проекта по созданию ИС, ее целей и задач, определение состава функциональных подсистем и возможностей реализации проекта. При изучении объекта предметной области необходимо проанализировать организационную и функциональную структуры, технико-экономические характеристики основных бизнес-процессов, материальные и информационные потоки между подразделениями и внутри них, методы планирования и учета.

При проведении анализа предметной области могут быть использованы следующие методы сбора информации.

Устный опрос производится по заранее составленному вопроснику на рабочем месте специалиста с записью ответов и позволяет в форме собеседования понять технологию работы и опыт опрашиваемого. Достоинства метода заключаются в его простоте и в том, что можно приступить к подготовке нового решения уже на стадии анализа. К недостаткам этого метода относятся затруднения психологического порядка и разнородность результатов опроса.

Анкетирование с помощью перечня вопросов дает полную и основательную информацию при условии готовности опрашиваемых к

правдивым ответам. Чтобы повысить качество анкетирования, можно предлагать варианты ответов на каждый вопрос. Существенное влияние на качество результатов оказывают четкость, недвусмысленность вопросов, поэтому разработка перечня вопросов предполагает знание предметной области.

Наблюдение, измерение и оценка. С помощью этих методов собираются сведения о параметрах, признаках и объектах в соответствующей предметной области. Важные для изучения параметры, признаки и объекты точно оцениваются сотрудниками и регистрируются в карточках или в формулярах (например, по частоте, количеству, продолжительности, затратам).

Самостоятельное описание требований проводится путем изучения нормативных документов, регламентирующих выполнение бизнес-процессов. Это один из способов получить первоначальное представление об исследуемом объекте и сформулировать вопросы к экспертам. Недостаток этого подхода – опасность пропуска знаний, специфичных для объекта исследования, неформализованных знаний, эмпирических правил и процедур, широко используемых на практике, но не вошедших в документы.

3.3.1.1 Общая характеристика объекта проектирования

При проведении предпроектного обследования в первую очередь нужно охарактеризовать исследуемый объект, где планируется внедрение или реинжиниринг информационной системы.

Общую характеристику исследуемого объекта рекомендуется составлять, придерживаясь следующего плана (в скобках указаны документы, являющиеся источниками информации).

Описание общих принципов функционирования исследуемого объекта (документы, определяющие функционирование организации в целом, например, Устав, Миссия организации).

Описание структуры исследуемого объекта (иерархическая организационная структура). Данный пункт является обязательным в случае сбора информации о бизнес-процессе на конкретном предприятии. Если предприятие является очень крупным, можно ограничиться описанием организационной структуры подразделения, сотрудники которого являются основными исполнителями или владельцами рассматриваемого бизнес-процесса. Впоследствии при описании бизнес-процесса средствами функционального моделирования, а также с использованием мнемосхем следует использовать организационную структуру в качестве основы для выбора структурных подразделений и сотрудников, участвующих в рассматриваемом процессе.

Описание направлений деятельности исследуемого объекта (документы, определяющие направления деятельности, например, стандарты, Положения). Такие сведения могут также содержаться в Уставе предприятия.

Перечень основных бизнес-процессов. При перечислении основных бизнес-процессов можно пользоваться классификацией бизнес-процессов, приведенной в Приложении 2, по мере необходимости корректируя ее в зависимости от сферы деятельности исследуемого объекта.

Если курсовая работа выполняется без привязки к конкретному предприятию, то **в качестве исследуемого объекта можно рассматривать типовые экономические, организационные, производственные процессы** (например, процессы кадрового учета, процессы налогового учета и т.д.). В этом случае структура исследуемого объекта не приводится, процессы характеризуются в общем виде со ссылкой на нормативные документы, законодательные акты федерального или регионального уровня (например, Налоговый кодекс, Трудовой кодекс, Регламент работы с обращениями граждан в местных органах государственной власти и управления и т.д.).

3.3.1.2 Выделение функциональной подсистемы ИС и описание бизнес-процесса

Выделение функциональных подсистем, как правило, производится на основе иерархической организационной структуры исследуемого объекта, построенной по функциональному признаку. Иными словами, структура органов управления исследуемым объектом совпадает с набором функциональных подсистем ИС. Например, на производственном предприятии существует служба логистики, которая решает вопросы снабжения комплектующими деталями и сбыта готовой продукции. Ей соответствует своя функциональная подсистема управления снабжением и сбытом, специфическим объектом управления которой являются процессы снабжения и сбыта. Внешние для этой подсистемы входы и выходы связывают ее как с внешней средой, так и с другими функциональными подсистемами ИС. В службе логистики существует своя сравнительно замкнутая информационная система, выполняющая основные функции управления бизнес-процессами снабжения и сбыта (прогнозирование, планирование, анализ, учет, контроль, регулирование).

Выделение функциональной подсистемы ИС производится путем обоснованного выбора структурного подразделения или бизнес-процесса, рассматриваемого в качестве объекта исследования в данной курсовой работе.

Рекомендуемый план работы по выделению функциональной подсистемы ИС:

Обоснование выбора бизнес-процесса – необходимо указать, почему именно этот бизнес-процесс выбран в качестве объекта исследования. Основанием для выбора бизнес-процесса могут служить его важность для предприятия, отнесение его к определенному классу бизнес-процессов (Приложение 2), количество сотрудников, задействованных в его выполнении, сложность его выполнения, отсутствие автоматизированной обработки информации.

Описание бизнес-процесса в виде мнемосхемы, разрабатываемой в любом графическом редакторе, включая MS Office Visio. Мнемосхема отражает взаимодействие участников процесса, дает общее представление о его сути.

Детальное обследование бизнес-процесса. Результатом предварительного обследования бизнес-процесса является его вербальное (словесное) описание. К этому описанию рекомендуется приложить заполненную форму обследования бизнес-процесса (Приложение 3).

Разработка функциональной модели существующего бизнес-процесса. Функциональная модель существующего бизнес-процесса (модель «как есть») разрабатывается в стандарте SADT с использованием программного обеспечения, поддерживающего нотацию функционального моделирования IDEF0 для верхних уровней процессной модели (рис. 2, 3) и нотацию EPC для нижнего уровня процессной модели (рис. 4).

В ходе функционального моделирования в нотации IDEF0 чаще всего встречаются следующие ошибки, снижающие качество модели:

- неправильное именование интерфейсных дуг, например, глаголом или отглагольным существительным, обозначающим действие;
- неправильное именование функциональных блоков, например, отсутствие действия в его названии;
- отсутствие иерархической нумерации функциональных блоков;
- отсутствие связи между диаграммами (диаграмма нижнего уровня не является декомпозицией диаграммы верхнего уровня);
- необоснованное отсутствие у функциональных блоков одной или нескольких интерфейсных дуг;
- неописанные интерфейсные дуги;
- необоснованное использование синонимов и сокращений в названиях интерфейсных дуг, допускающее неоднозначное толкование одних и тех же информационных элементов (например, «Нач.» и «Начальник», «Руководитель» и «Начальник»);
- менее 3 и более 9 функциональных блоков на одной диаграмме.

- При построении модели в нотации ЕРС следует придерживаться следующих основных правил:
- диаграмма должна начинаться и заканчиваться описанием события;
- выполнение функции обязательно заканчивается событием;
- функция описывается активным глаголом или отглагольным существительным («Выбор поставщика», «Зарегистрировать счет»);
- событие констатирует свершившийся факт («Поставщик выбран», «Счет зарегистрирован»).

Если объектом исследования служит конкретное предприятие, рекомендуется наряду с изучением регламентных, нормативных и организационно-распорядительных документов использовать анкетирование и опрос сотрудников для сбора наиболее полной и точной информации о порядке выполнения бизнес-процессов.

3.3.1.3 Разработка технического задания на создание системы управления бизнес-процессом

В соответствии с ГОСТ 34.602-89 «Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы» этот документ является основным, определяющим требования и порядок создания автоматизированной системы, в соответствии с которым проводится разработка ИС и ее приемка при вводе в действие. Техническое задание (ТЗ) на ИС разрабатывают на основании исходных данных, полученных в ходе предпроектного обследования. **ТЗ рекомендуется составлять в соответствии со следующим планом:**

- Описание основных целей создания подсистемы, выделение критериев эффективности ее функционирования с указанием перечня и значений основных показателей, которые должны быть достигнуты в условиях автоматизированного управления исследуемым объектом (например, время формирования результатных документов, скорость обслуживания клиентов и т.д.).
- Описание требований к подсистеме в целом и требований к выполняемым функциям. Требования к подсистеме в целом формируются в контексте бизнестребований, а требования к функциям – в контексте требований пользователей.
- Требования к программному, информационному, организационному и техническому обеспечению.

- Состав и содержание работ по созданию функциональной подсистемы ИС. Следует описывать тот набор работ, который выполняется в рамках курсовой работы (т.е., не включая этап ввода в эксплуатацию).

В Приложении 4 приведены выдержки из ГОСТ 34.602-89 «Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы». При выполнении курсовой работы рекомендуется использовать это Приложение в качестве основы для разработки ТЗ. Нумерация разделов в Приложении 4 соответствует нумерации разделов ГОСТ 34.602-89. В курсовой работе нумерация разделов технического задания должна соответствовать нумерации, принятой в курсовой работе.

3.3.2 Стадия проектирования ИС

3.3.2.1 Постановка задачи

Постановка задачи – это описание **исследуемой задачи** по определенным правилам, которое дает исчерпывающее представление о ее сущности, логике преобразования информации для получения результата. На основе постановки задачи проектировщик должен представить логику ее решения и рекомендовать стандартные программные средства или типовые проектные решения, пригодные для ее реализации.

В соответствии с процессным подходом под задачей понимается четвертый уровень процессной модели (бизнес-функции). При выборе задачи для автоматизации следует руководствоваться декомпозицией контекстной диаграммы. Один из функциональных блоков этого уровня, как правило, представляет собой задачу, решаемую в одном из структурных подразделений предприятия одним или несколькими сотрудниками (рис. 3, 4).

Для постановки задачи используются сведения, необходимые и достаточные для полного представления ее логической и информационной сущности. Такими сведениями располагает специалист, осуществляющий решение задачи в условиях ручной обработки или с использованием компьютерной техники. При постановке задачи необходимо, прежде всего, описать информационное обеспечение, алгоритмы ее решения.

При описании постановки задачи обращается внимание на ее объемновременные характеристики. Они отражают объемы входной и выходной информации (количество документов, строк, знаков, обрабатываемых в единицу времени), временные особенности поступления, обработки и выдачи информации.

В процессе описания постановки задачи важной является выверка точности и полноты названий всех информационных единиц. Четкость

наименований информационных элементов и их идентификация, устранение синонимов и омонимов в названиях экономических показателей обеспечивают более высокое качество результатов обработки. Для каждого вида информации дается описание всех ее элементов в виде таблицы (табл. 2).

Таблица 2 Пример описания информационных элементов для документа «Заявка»

Наименование реквизита	Идентификатор	Тип данных	Разрядность
Номер заявки	Zayavka	Числовой	4
Дата создания заявки	Date_zayavka	Дата	8
Название подразделения	Name_otdel	Текстовый	20
Код материала	Kod_mat	Числовой	8
Название материала	Name_mat	Текстовый	20
Количество материала	Kol_mat	Числовой	3
Единицы измерения	Ed_mat	Текстовый	5
ФИО сотрудника	FIO_sotrudnik	Текстовый	30

Наименование реквизита должно соответствовать документу или вытекать из него. Не допускаются даже мелкие по словари информационной системы (тезаурус).

Идентификатор представляет собой условное обозначение, с помощью которого можно оперировать значением реквизита. Идентификатор может строиться по мнемоническому принципу, использоваться для записи алгоритма и представлять собой сокращенное обозначение полного наименования реквизита.

Разрядность реквизита необходима для подсчета объема занимаемой памяти.

Постановка задачи выполняется в соответствии с планом:

1. Административно-организационная сущность задачи:

- наименование задачи, место ее решения (указать исследуемый объект или класс исследуемых объектов ИС);
- цель решения (формулируется с точки зрения важности задачи для выполнения бизнес-процесса в целом, вне зависимости от средств ее решения);
- назначение (для каких подразделений исследуемого объекта и пользователей она предназначена);
- периодичность и сроки решения задачи;
- источники и способы поступления данных;

- потребители результатной информации и способы ее отправки. **2.**

Описание исходной (входной) информации:

- перечень исходной информации (документы или наборы данных);
- формы представления по каждой позиции перечня;
- объем информации (для бумажных документов – в страницах или строках, для электронных – в Кб);
- описание структурных единиц информации по каждому из представленных документов или наборов данных (см. табл. 2).

3. Описание результатной (выходной) информации (по аналогии с исходными данными):

- перечень результатной информации;
- формы представления;
- периодичность и сроки представления;
- объем информации;
- перечень пользователей результатной информации (подразделение и должность); описание структурных единиц информации.

4. Описание используемой условно-постоянной информации:

- перечень условно-постоянной информации (классификаторов, справочников с указанием их полных наименований);
- формы представления (бумажный или электронный);
- описание структурных единиц информации (по аналогии с исходными данными).

5. Описание алгоритма решения задачи - описание способов формирования результатной информации с указанием последовательности выполнения логических и алгоритмических действий (может быть использована ссылка на диаграмму в нотации ЕРС, если она достаточно подробно описывает решение задачи с точки зрения выполняющего ее сотрудника).

3.3.2.2 Разработка локального классификатора информации

Прим.: данный раздел, не является обязательным и может быть включен в курсовую работу только по указанию руководителя.

Информационное обеспечение (ИО) – это совокупность информации, характеризующей состояние управляемого объекта и являющейся основой для принятия управленческих решений. Информационное обеспечение делится на немашинное и внутримашинное.

Немашинное ИО включает:

- систему исследуемых показателей предметной области (например, показатели бухучета, финансовой деятельности и т.д.);

- систему классификации и кодирования;
- документацию;
- информационные потоки.

Внутримашинное ИО – система специальным образом организованных данных, подлежащих автоматизированной обработке, накоплению, хранению, поиску и передаче в виде, удобном для восприятия техническими средствами.

Оно включает информационные массивы, базы и банки данных, базы знаний.

В данной курсовой работе из всего перечня компонентов ИО необходимо разработать классификатор технико-экономической информации, используемой в проектируемой функциональной подсистеме, и результатный документ, формируемый с использованием выбранного программного обеспечения.

Классификаторы технико-экономической информации.

Систематизация экономической информации вызывает необходимость применения различных классификаторов:

- общегосударственные классификаторы (ОК), создаваемые в централизованном порядке и являющиеся едиными для всей страны;
- отраслевые, единые для отрасли национальной экономики;
- региональные, единые для данной территории;
- локальные, составляемые на номенклатуры, характерные для данного предприятия, организации, фирмы.

Для минимизации количества справочников, предназначенных для систематизации технико-экономической информации, необходимо сначала составить перечень классификаторов всех категорий, применяемых в процессе решения задачи.

Если при постановке задачи была выявлена информация, являющаяся довольно устойчивой (неизменной) во времени, носящая справочный характер и отсутствующая в общегосударственном, отраслевом или региональном классификаторе, то для удобства использования такой информации в ИС следует разработать **локальный классификатор**.

Разработка локального классификатора происходит в 4 этапа:

Установление перечня и количества объектов, подлежащих классификации и кодированию. Сначала устанавливаются номенклатуры (структурные подразделения, поставщики, материалы, оборудование, готовые изделия, операции технологического процесса и т.д.), затем по каждой номенклатуре определяется полный список всех позиций, подлежащих кодированию.

Систематизация объектов по определенным классификационным признакам (выбор метода классификации).

При построении классификаторов используют **иерархический** или **фасетный методы классификации**.

Иерархический метод заключается в последовательном делении заданного множества на подчиненные множества, каждое из которых в свою очередь делится на подчиненные ему подмножества. В общем виде иерархическую классификационную схему можно представить в следующем виде (рис. 5):

0 уровень



1 уровень



2 уровень



3 уровень

Рисунок 5. – Иерархическая система классификации

Классификационные схемы, построенные на основе иерархического принципа, имеют неограниченную ёмкость, величина которой зависит от числа ступеней деления и количества объектов классификации, которое можно расположить на каждой ступени. Количество объектов на каждой ступени определяется основанием кода, т.е. числом знаков в алфавите кода.

Иерархический метод классификации более предпочтителен для объектов с относительно стабильными признаками и для решения стабильного комплекса задач.

Фасетный метод заключается в делении заданного множества на группировки независимо, по различным признакам классификации. При этом методе классификации заранее жесткой классификационной схемы и конечных группировок не создается. Разрабатывается лишь система таблиц признаков объектов классификации, называемых фасетами. При необходимости создания классификационной группировки для решения конкретной задачи осуществляется выборка необходимых признаков из фасетов и их объединение в определенной последовательности. В общем виде

фасетную классификационную схему можно представить в следующем виде (рис. 6):

	Фасеты				
	Ф1	Ф2	Ф3	...	Фn
Значения фасетов					
1	*	*			
2	*		*		*
...				*	
k		*			*

Рисунок 6. – Фасетная система классификации.

Такой принцип построения классификационных группировок делает классификатор на основе фасетного метода классификации очень гибким, хорошо приспособленным для использования в условиях большой динамичности решаемых задач.

Определение правил обозначения объектов кодирования (выбор метода кодирования) на основании системы классификации.

Выбор метода кодирования зависит от количества выделяемых признаков, числа позиций в каждом признаке и степени устойчивости номенклатуры. Методы кодирования могут носить самостоятельный характер – регистрационные методы кодирования, или быть основанными на предварительной классификации объектов – классификационные методы кодирования.

Классификационные методы кодирования бывают двух видов: последовательный и параллельный.

При **последовательном методе** кодирования код объекта классификации образуется с использованием кодов последовательно расположенных подчиненных группировок, полученных при иерархическом методе классификации. В этом случае код нижестоящей группировки образуется путем добавления соответствующего количества разрядов к коду вышестоящей группировки.

Последовательный метод применяется в такой последовательности:

- определяется число группировочных признаков и их зависимость;
- устанавливается число позиций в каждом группировочном признаке;
- производится кодирование порядковыми номерами сначала старшего признака, затем следующих признаков внутри старших, каждый раз начиная с 1, 01, 001 в зависимости от значности младшего признака в пределах его старшего признака.

При *параллельном (позиционном) методе* код объекта классификации образует с использованием независимых группировок, полученных при фасетном методе классификации. При этом методе кодирования признаки объекта кодируются независимо друг от друга. Для параллельного метода кодирования возможны два варианта записи кодов объекта.

Каждый фасет и признак внутри фасета имеют свои коды, которые включаются в состав кода объекта. Такой способ записи удобно применять тогда, когда объекты характеризуются неодинаковым набором признаков и различны их числом. При формировании кода какого-либо объекта берутся только необходимые признаки.

Для определения групп объектов выделяется фиксированный набор признаков и устанавливается стабильный порядок их следования, то есть устанавливается фасетная формула. В этом случае не надо каждый раз указывать, значение какого признака приведено в определенных разрядах кода объекта.

Разработка кодовых обозначений и положений по их ведению и внесению в них изменений.

На этом этапе осуществляется непосредственное присвоение разработанных кодов всем позициям номенклатуры. Заканчивается этот этап составлением классификатора, который оформляется в виде справочника, содержащего условное кодовое обозначение объекта и его полное наименование.

Код объекта классификации формируется по схеме, учитывающей порядок следования признаков объекта (рис. 7).

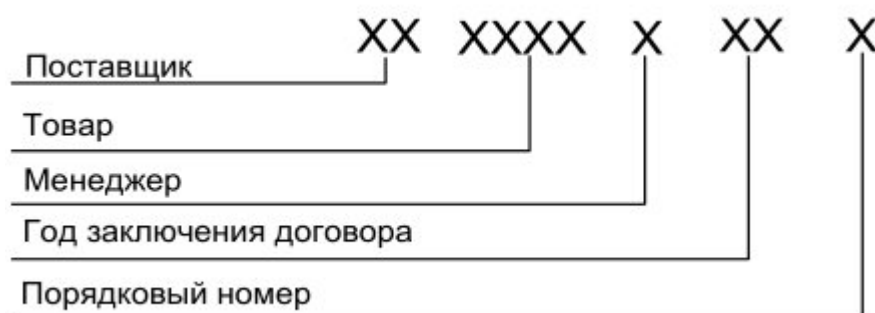


Рисунок 7. - Пример кодирования договоров на поставку продукции

При разработке классификаторов и систем кодирования следует соблюдать следующие основные требования. **Коды должны:** охватывать все объекты, подлежащие кодированию, и давать им

однозначное обозначение; учитывать перспективы развития системы и расширение списка

кодируемых объектов; обладать максимальной информативностью при минимизации длины кода; отличаться логичностью формирования, удобством восприятия и

запоминания; быть едиными для разных задач внутри одного исследуемого объекта (например, коды материалов, подразделений должны быть едиными для задач бухгалтерского учета и технической подготовки производства); учитывать существующие системы кодирования и общепринятые

обозначения во взаимодействующих АИС.

3.3.2.3 Информационное моделирование бизнес-процесса

С целью соблюдения одного из принципов создания ИС (принципа однократного ввода данных) информационная модель подсистемы разрабатывается на основе функциональной модели предлагаемого бизнеспроцесса.

Для этого в любом CASE-средстве моделирования присутствует механизм построения отчетов по интерфейсным дугам, используемым в функциональных диаграммах. Например, в программном продукте StarUML в меню **Отчеты** нужно выбрать вариант **Стрелки процессов SADT**. Этот отчет предоставляет информацию о совокупности данных, используемых в процессе решения задачи. Следовательно, эти же данные должны присутствовать и в информационной модели. Фрагмент отчета по стрелкам процессов SADT представлен в таблице 3.

Таблица 3

Фрагмент отчета по стрелкам процессов SADT

№	Наименование стрелки	Исходит из		Входит в	
		Процесс	Тип стрелки	Процесс	Тип стрелки
	Подготовленный контракт	A1.2 Подготовка и оформление заявки на материал	Выход	A1.3 Согласование заявки на материал	Вход
	Подразделения предприятия	{Туннель}	Выход	A4.1 Получить требование на продукцию	Механизм
	Проверенная документация	A2 Выбор поставщиков	Выход	{Граница}	Вход

		A2.2 Проверить сопроводительную документацию	Выход	A2.3 Проверить продукцию	Вход
	Проверенная продукция	A2.3 Проверить продукцию	Выход	A2.4 Провести маркировку продукции	Вход
	Продукция	{Граница}	Выход	{Туннель}	Вход
		A4.3 Произвести отпуск продукции	Выход	A1 Определение потребности в материале	Вход
	Продукция от поставщика	{Туннель}	Выход	{Туннель}	Вход
	Сопроводительная документация	{Граница}	Выход	A0 процесс закупка	Вход
	Требование	Заказчик	Выход	A4.1 Получить требование на продукцию	Вход
№	Наименование стрелки	Исходит из		Входит в	
		Процесс	Тип стрелки	Процесс	Тип стрелки
	Требование на продукцию	A4.1 Получить требование на продукцию	Выход	A4.2 Оформить лимитнозаборную карту на продукцию	Вход
	Юрист	{Туннель}	Выход	A1.3 Согласование заявки на материал	Механизм

Для построения информационной модели сначала необходимо провести **анализ отчета** по следующей схеме:

- выявить ошибки построения функциональной модели, связанные с нарушением стандарта проектирования SADT:

- неописанные стрелки – в функциональной модели присутствуют необозначенные стрелки, как правило, это стрелки, соединяющие два последовательно расположенных функциональных блока;

- несвязанные элементы функциональной модели – некоторые функциональные блоки не имеют входов, выходов или механизмов управления;

- неправильное именование стрелок, например, глаголом или отглагольным существительным, обозначающим действие;

- стрелки, названные по-разному, но обозначающие один и тот же объект, например, «Нач.» и «Начальник», «Док-т» и «Документ»;

- исправить ошибки проектирования путем возврата на указанные диаграммы функциональной модели, внесения изменений и повторным созданием отчета;

- выявить и удалить «лишние» данные, служащие только для пояснения функциональной модели и не несущие смысловой нагрузки в информационной модели (к этой группе относятся также стрелки-механизмы, обозначающие ИС);

- выделить данные, означающие движение информационного элемента по его жизненному циклу, например, «проект приказа», «зарегистрированный приказ», «подписанный приказ» или «продукция», «продукция от поставщика», «проверенная продукция» и т.д. В этом случае необходимо оставить только один, исходный информационный элемент – «приказ», «продукция»;

- оставшиеся данные распределить по следующим группам:
сущности, отражающие нормативную или справочную информацию (к ним относятся ГОСТы, стандарты предприятия, классификаторы, справочники и т.д.);

- сущности, являющиеся документами (это основные информационные элементы, в которых отражается суть решаемой задачи или смысл бизнес-процесса – приказы, акты, заявления и т.д.);

- сущности, отражающие информацию о материальных объектах, фигурирующих в бизнес-процессе и не являющихся документами (товар, продукция, деталь и т.д.);

- сущности, отражающие информацию о сотрудниках или организационной структуре (это стрелки-механизмы, указывающие отношение сотрудников или структурных подразделений к обработке того или иного документа);

- атрибуты вышеперечисленных сущностей, являющиеся реквизитами документов или сотрудников («номер», «дата», «подпись», «ФИО» и т.д.).

Атрибуты сущностей информационной модели должны совпадать со структурными единицами информации в постановке задачи.

После анализа отчета можно приступить к проектированию информационной модели, используя нотацию IDEF1X [12]. Информацию об отношениях между сущностями можно получить из функциональной модели. В результате должна быть построена информационная модель, отображающая взаимодействие информационных элементов в ходе выполнения бизнеспроцесса с использованием ИС (рис. 8).

В модели могут быть использованы все виды отношений между сущностями (идентифицирующие, неидентифицирующие), включая отношения категоризации.

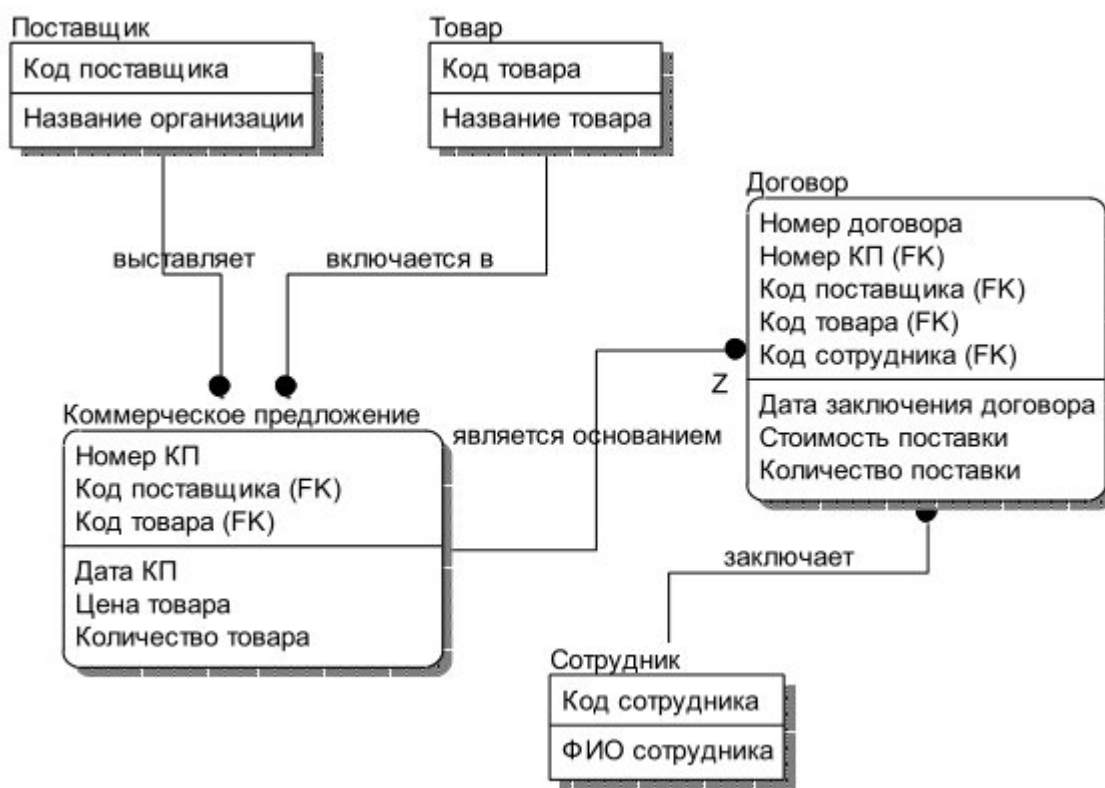


Рисунок 8. - Пример информационной модели в нотации IDEF1X

3.3.2.4 Разработка динамической модели работы подсистемы

Алгоритм работы спроектированной подсистемы должен соответствовать постановке задачи и учитывать общий и все частные случаи решения задачи, если таковые имеются. Под частными случаями понимается реакция системы на наличие проблемных (нештатных) ситуаций в процессе организационного управления.

Динамическая модель разрабатывается на основе BPMN-технологии, что позволяет отражать роль каждого пользователя в ходе выполнения бизнеспроцесса с использованием ИС. Такой подход в дальнейшем

существенно облегчает задачу технического писателя при разработке технологических инструкций для пользователей ИС.

Пример динамической модели работы функциональной подсистемы приведен на рисунке 9, где показан фрагмент процесса обработки распорядительного документа с использованием системы электронного документооборота.

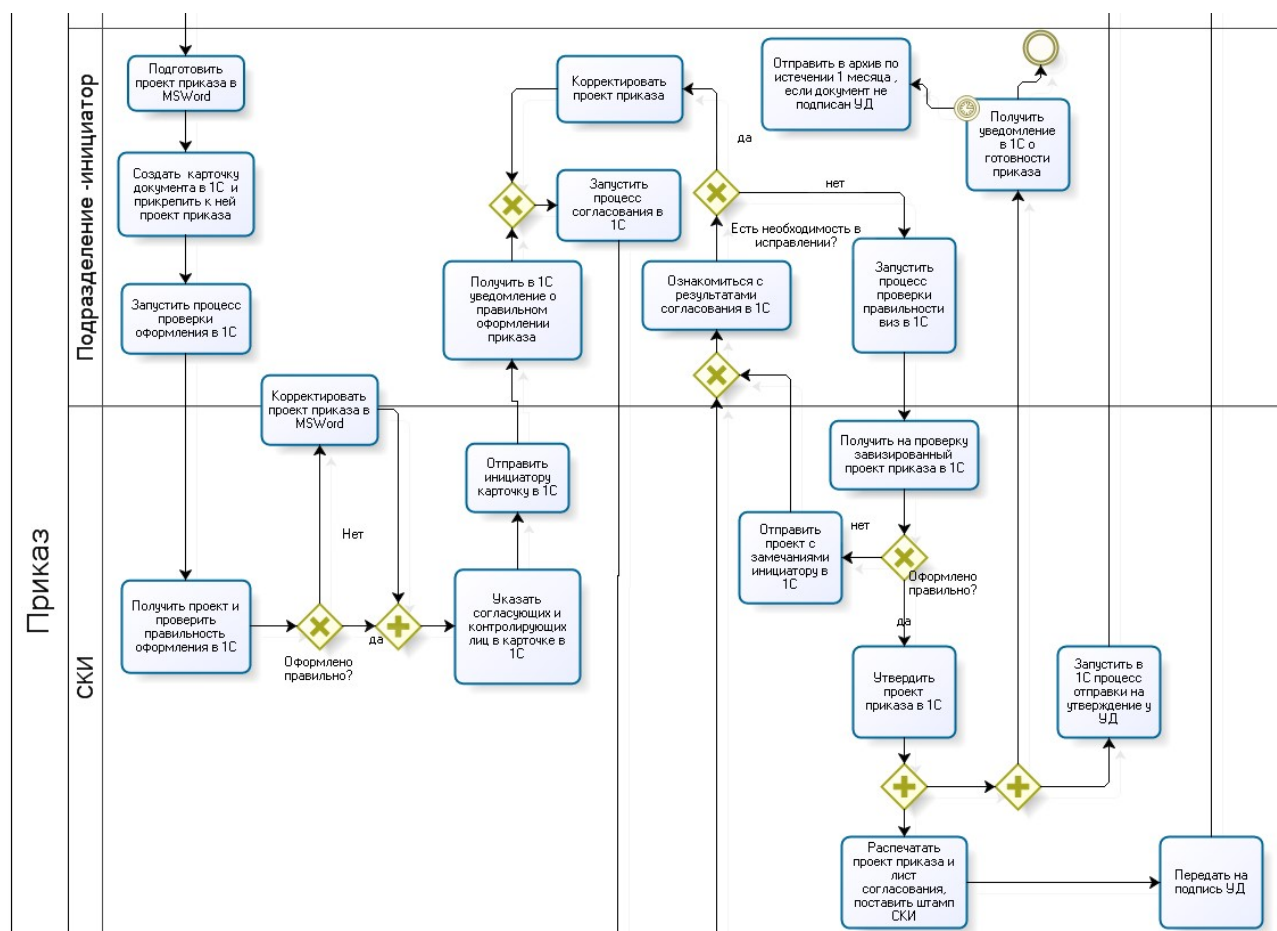


Рисунок 9. - Пример динамической модели в нотации BPMN

3.3.3 Стадия реализации проектных решений

Заключительным этапом курсовой работы является разработка (или адаптация) программного обеспечения для функциональной подсистемы системы управления бизнес-процессами. Этот этап выполняется по следующему плану:

Обоснование выбора программного обеспечения для реализации функциональной подсистемы.

Разработка пользовательского интерфейса с применением выбранного CASE-средства или адаптация типового проектного решения (готовой ИС) к особенностям предметной области.

Разработка технологической инструкции пользователя по работе с подсистемой с приведением экранных форм, отражающих последовательность получения результатной информации.

3.3.3.1 Выбор ПО для реализации функциональной подсистемы

В данной курсовой работе для реализации функциональной подсистемы может быть использована одна из двух технологий проектирования - технология оригинального проектирования или технология типового проектирования.

В случае выбора технологии оригинального проектирования необходимо разработать пользовательский интерфейс автоматизации конкретного бизнеспроцесса. Для разработки пользовательского интерфейса подсистемы целесообразно использовать соответствующие CASE-средства, имеющиеся на рынке программного обеспечения. Анализ рынка CASE-средств выполняется с целью выбора CASE-средства, максимально удовлетворяющего потребностям и возможностям разработчика.

В случае использования технологии типового проектирования необходимо выбрать на рынке программного обеспечения готовую информационную систему и провести ее адаптацию к особенностям предметной области. Анализ рынка ИС выполняется с целью выбора системы, максимально удовлетворяющей потребностям ее пользователей. Потребности пользователей ИС должны соразмеряться с реальной ситуацией на рынке, т.е., необходимо использовать реальные данные о характеристиках сравниваемых информационных систем.

И в том, и в другом случае для обоснованного выбора средства реализации принятых проектных решений необходима консультация руководителя курсовой работы. Ниже, в качестве примера, представлен ***метод анализа иерархий***, который состоит в декомпозиции проблемы выбора на все более простые составляющие части и дальнейшей обработке последовательности суждений по парным сравнениям (рис. 10).

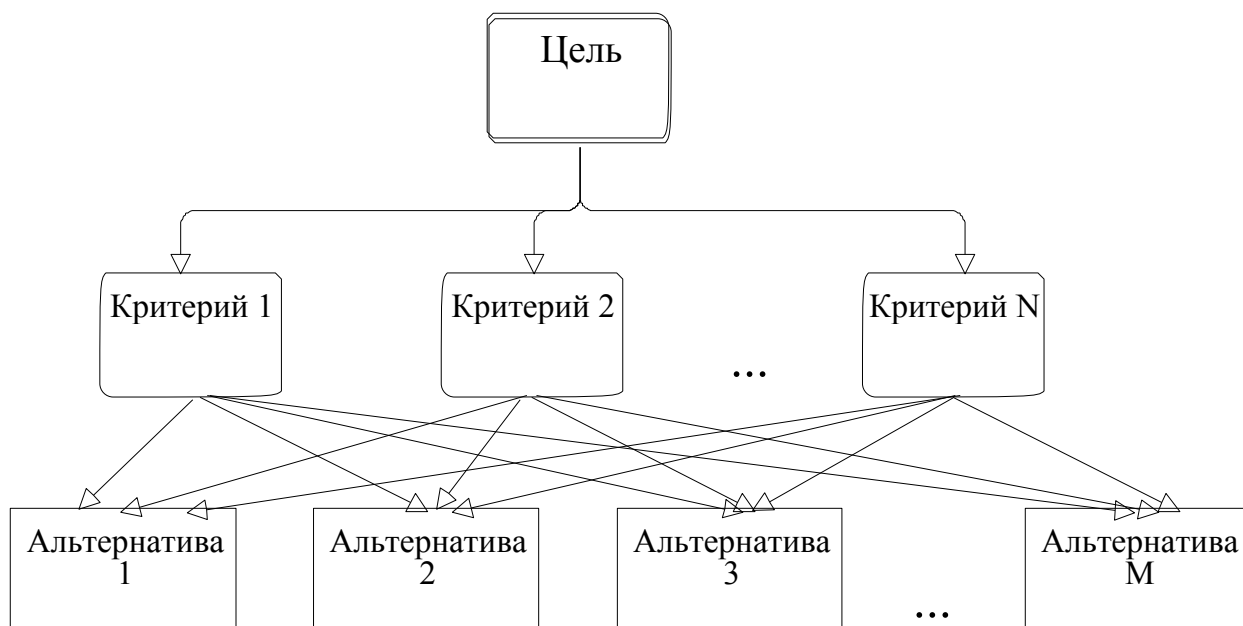


Рисунок 10. – Схема метода анализа иерархий

Метод анализа иерархий применяется при необходимости принятия решения по выбору одного объекта из нескольких альтернатив. При этом необходимым условием применения этого метода является наличие общих характеристик совокупности сравниваемых объектов. Естественно, что значения этих характеристик для каждого из сравниваемых объектов будут различными. Эти характеристики служат в конечном итоге критериями выбора наилучшей альтернативы. В наиболее элементарном виде иерархия строится с вершины (цели), через промежуточные уровни (характеристики или критерии) к самому низкому уровню (сравниваемые объекты или альтернативы).

В данном случае целью является выбор наиболее подходящего CASE-средства или готовой информационной системы, критериями могут служить характеристики программного обеспечения или проекта по его внедрению, приведенные в таблице 4 или 5 соответственно, в качестве альтернатив выступает ПО, близкое по классу.

Таблица 4
Критерии, применяемые для оценки CASE-средств

Критерий	Определение
Минимум трудоемкости создания ПО	Количество человеко-месяцев, затрачиваемых на создание ПО с использованием CASE-средств

Максимум продуктивности	Объем работы (измеряемый в количестве строк кода или функциональных точек), приходящийся на единицу трудоемкости (человеко-месяц) при использовании данного CASE-средства
Максимум качества создаваемого ПО	Количество дефектов в рабочих продуктах при использовании данного CASE-средства
Требования к оборудованию	Минимальные требования к характеристикам оборудования, необходимого для использования CASE-средств

Таблица 5 Критерии, применяемые для оценки ИС

Критерий	Определение
Минимум времени внедрения ИС	Временной интервал от начала внедрения ИС до начала безошибочной работы всех пользователей ИС
Минимум затрат на внедрение ИС	Суммарная стоимость приобретения, обучения пользователей и сопровождения ИС
Удобство использования	Наличие пользовательской документации, интуитивно понятный дружественный интерфейс, время реакции системы
Удобство сопровождения	Наличие разнообразных средств поддержки поставщика, история присутствия на рынке ПО

После иерархического воспроизведения проблемы возникает необходимость установления приоритетов критериев и оценки каждой из альтернатив по проранжированным критериям. При определении численных предпочтений используется заданная шкала предпочтений одного сравниваемого объекта другому (табл. 6). В данном случае под объектами понимаются критерии и альтернативы.

Таблица 6
Шкала предпочтений объектов по методу анализа иерархий

Степень превосходства	Определение
0	Объекты не сравнимы
1	Объекты одинаково важны
3	Умеренное превосходство одного над другим

5	Существенное превосходство одного над другим
7	Значительное превосходство одного над другим
9	Абсолютное превосходство одного над другим
2,4,6,8	Промежуточные значения степени превосходства

В методе анализа иерархий сначала сравниваются попарно критерии для определения их приоритетов, поскольку на конечный выбор альтернативы критерии влияют по-разному в зависимости от их значимости. Затем аналогичным образом сравниваются и альтернативы по отношению к каждому из критериев.

Для получения экспертных оценок значимости критериев строится матрица с перечнем объектов сравнения. Эксперт должен попарно сравнить критерии по отношению к поставленной цели и заполнить предложенную матрицу парных сравнений (табл. 7).

Таблица 7 Матрица парных сравнений критериев

	Критерий 1	...	Критерий j	...	Критерий N
Критерий 1	1	...	a_{1j}	...	a_{1n}
...	...	...	...	...	...
Критерий i	a_{i1}	...	a_{ij}	...	a_{in}
...	...	...	...	...	...
Критерий N	a_{n1}	...	a_{nj}	...	1

При этом ячейки матрицы заполняются значениями шкалы предпочтений (табл. 3) следующим образом: если при сравнении i-го критерия с j-м критерием получено $a_{ij}=b$ (где b – важность), то при сравнении j-го критерия с i-м критерием получаем $a_{ji}=1/b$, т.е., если $a_{ji}=3$, то $a_{ij}=1/3$. Таким образом, данная матрица является диагональной и обратно симметричной.

Для каждой такой матрицы ищется вектор значений критериев $y_1, \dots, y_n$ по формуле:

$$y_i = \sqrt[N]{\prod_{j=1}^N a_{ij}}, \quad (1)$$

Далее этот вектор нормализуется:

$$y_{in} = \frac{y_i}{\sum_{i=1}^N y_i}, \quad (2)$$

где y_{in} - нормализованный коэффициент значимости, показывающий вклад

каждого критерия в достижение цели.

Здесь необходимо отметить следующее. Из линейной алгебры известно, что у положительно определенной, обратно симметричной матрицы, имеющей ранг, равный 1, максимальное собственное число равно размерности этой матрицы (т.е. n). При проведении сравнений в реальной ситуации вычисленное максимальное собственное число матрицы λ_{max} будет отличаться от соответствующего собственного числа для идеальной матрицы. Это различие характеризует рассогласованность реальной матрицы и степень доверия к полученным результатам. Чем больше это отличие, тем меньше доверие. В данном методе используется индекс согласованности (ИндС), который дает информацию о степени нарушения согласованности.

Индекс согласованности вычисляется по формуле:

$$\text{ИндС} = \frac{\lambda_{max} - n}{n - 1} \quad (3)$$

где
$$\lambda_{max} = \sum_{i=1}^n \lambda_i \quad (4)$$

где
$$\lambda_i = \sum_{j=1}^n a_{ij} \times y_{in} \quad (5)$$

Вместе с матрицей парных сравнений мы имеем меру оценки степени отклонения от согласованности. Если такие отклонения превышают установленные пределы, то тому, кто проводит суждения, следует перепроверить их в матрице.

Далее эта величина сравнивается с той, которая получилась бы при случайном выборе количественных суждений из нашей шкалы, и образовании обратно симметричной матрицы. Значения случайной согласованности (СС) для идеальных матриц разного размера приведены в таблице 8.

Таблица 8

Значения случайной согласованности для матриц разного размера

Размер матрицы	1	2	3	4	5	6	7	8	9	10
СС	0	0	0,58	0,9	1,12	1,24	1,32	1,41	1,45	1,49

Отношение согласованности (ОС), дающее представление о верности сделанных суждений, вычисляется по формуле:

$$\text{ОС} = \text{ИндС}/\text{СС} \quad (6)$$

При этом, если:

{	ОС ≤ 0,1	матрица, безусловно, согласованна;
	0,1 < ОС ≤ 0,2	согласованность матрицы приемлема;
	ОС > 0,2	согласованность матрицы не приемлема.

Если матрица согласована, следовательно, веса критериев соответствуют нормализованному вектору приоритетов (2). Это значит, что каждый критерий влияет на выбор той или иной альтернативы с указанным весом.

Теперь необходимо провести аналогичное попарное сравнение альтернатив по каждому критерию в отдельности. Для этого для каждого критерия нужно заполнить матрицу альтернатив (табл. 9):

Таблица 9

Матрица парных сравнений альтернатив по i-му критерию

	Альтернатива 1	...	Альтернат. j	...	Альтернатива М
Альтернатива 1	1	...	a _{1j}	...	a _{1m}
...	...	...	...	...	...
Альтернатива i	a _{i1}	...	a _{ij}	...	a _{im}
...	...	...	...	...	...
Альтернатива М	a _{m1}	...	a _{mj}	...	1

Для ранжирования альтернатив по i-му критерию и проверки согласованности матриц следует провести вычисления по формулам (1)-(6). В результате для каждого i-го критерия будет получен нормализованный вектор приоритетов альтернатив $x_{1in}, \dots, x_{min}$.

Заключительным этапом выбора оптимального проектного решения является определение глобального приоритета альтернатив с учетом весов всех критериев. Для этого необходимо заполнить таблицу 10.

Таблица 10

Определение глобальных приоритетов альтернатив

	Критерий 1	Критерий 2	...	Критерий j	...	Критерий N
Нормализ. вектор приоритетов критериев	y_{1H}	y_{2H}	...	y_{jH}	...	y_{nH}
Альтерн. 1	x_{11H}	x_{12H}	...	x_{1jH}	...	x_{1nH}
Альтерн. 2	x_{21H}	x_{22H}	...	x_{2jH}	...	x_{2nH}
...	...	...	...	...	...	...
Альтерн. M	x_{m1H}	x_{m2H}	...	x_{mjH}	...	x_{mnH}

Глобальный приоритет для каждой альтернативы вычисляется как сумма произведений соответствующих значений нормализованного вектора приоритетов для критерия и значения нормализованного вектора приоритетов этой альтернативы в отношении данного критерия по формуле:

$$A_i = \sum_{j=1}^n (x_{ijH} \times y_{jH}) \quad (7)$$

Наилучшим проектным решением будет считаться альтернатива с максимальным значением глобального приоритета.

3.3.3.2 Реализация интерфейса подсистемы

Реализация интерфейса подсистемы заключается в разработке элементов прикладного программного обеспечения в виде форм электронных документов, соответствующих информационной модели. Документ, получаемый в ходе выполнения бизнес-процесса и содержащий информацию о результатах его выполнения, должен соответствовать постановке задачи. В качестве средства разработки форм электронных документов предполагается использование CASE-средства. При этом требуется представить скрины экранных форм.

Если в качестве средства реализации проектного решения была выбрана готовая информационная система, необходимо показать, как выглядят электронные формы, сопровождающие выполнение рассматриваемого бизнес-процесса (рис. 11).

ГК "Меркурий" / Документооборот 8 КОРП. редакция 2.1 (1С:Предприятие)

Администрат

Запись журнала передачи

Записать и закрыть Записать

Еще ?

Документ: Договор подряда строительных работ для Клауст ЗАО (№ 1-ДП от 27.04.2012) (Вну ...)

Тип экземпляра: Оригинал № экземпляра: 1

Кому передан: Мишин С.А.

Дата передачи: 01.10.2018

Срок возврата: 09.10.2018

☒ Возвращен: 08.10.2018

Комментарий:

Как работать с Журналом передачи

Любой факт передачи бумажной версии документа или его копии обязательно отражается в Журнале передачи.

Заполните поля карточки:

- документ (выберите вид документа из списка),
- тип экземпляра (выберите из предложенного списка),
- кому передан (выберите сотрудника организации или контрагента),
- срок возврата.

После возвращения документа поставьте в карточке соответствующую отметку.

Контроль Журнала передачи выполняет секретарь. При появлении документов в списке "Невозвращенные" секретарь направляет напоминание о необходимости вернуть документ.

Рисунок 11. - Заполненная карточка документа в системе 1С:
Документооборот

3.3.3.3 Разработка инструкции для пользователя функциональной подсистемы

Технологическая инструкция – один из эксплуатационных документов, предусмотренных ГОСТ серии 34. Задача технологической инструкции – соотнести функциональность программного обеспечения автоматизированной системы с практикой, описать работу пользователей ИС по существу.

Технологическая инструкция относится к определенной функциональной роли и представляет собой сборник описаний операций. В описании операции обязательно присутствуют следующие графы или рубрики:

триггер, т.е. событие или обстоятельство, при наступлении которого пользователь выполняет операцию. Триггером может быть какое-либо внешнее по отношению к пользователю событие (обращение контрагента, другого сотрудника, распоряжение начальника, поступление какого-то документа или сообщения), наступление определенного момента по установленному расписанию или собственное решение; **результат операции**, т.е. существенное изменение ситуации, которое происходит после ее успешного выполнения. Результат не следует путать с выходными данными или документами; пошаговое описание выполнения операции. Каждый шаг

описания касается какой-то одной задачи, решаемой посредством программного обеспечения.

В курсовой работе технологическая инструкция должна быть разработана для пользователя, указанного в качестве участника процесса в динамической модели работы функциональной подсистемы.

4. ОФОРМЛЕНИЕ КУРСОВОЙ РАБОТЫ

Курсовая работа выполняется в электронной форме и размещается обучающимся в личном кабинете в виде отдельных файлов:

- курсовая работа с именем файла ФИО_КР_Проектирование ИС.docx (пример: «Иванов И.И._КР_ Проектирование ИС.docx»)
- файлы с исходным программным кодом, например:
 - файл конфигурации Ivanov_AS.dt (для проектов на 1С:Предприятие)
 - файл Ivanov_AS.py (для проектов на Python)
 - и т.п.
- сопутствующие файлы (например, файл базы данных)

Курсовая работа должна быть представлена в формате редактируемого файла, например: .odt, .docx, .rtf. **Нельзя предоставлять курсовую работу в виде неотредактируемого pdf-файла.** Имя файла должно иметь следующий вид:

«ФИО_КР_Проектирование ИС.odt»

Например: «Иванов И.И._КР_ Проектирование ИС.docx» или «Иванов И.И._КР_ Проектирование ИС_Отчет.rtf»

Текст на листе А4 должен иметь книжную ориентацию, альбомная ориентация допускается только для таблиц и схем приложений. Поля страницы должны иметь следующие размеры: левое – 30 мм, правое - 15 мм, верхнее – 20 мм, нижнее – 20 мм. Текст печатается шрифтом Times New Roman, 14 кегль (для сносок 10 кегль), междустрочный интервал - 1,5. Абзац – 1,25 см. Формулы необходимо представлять с помощью редактора формул.

Номера страниц размещаются в нижнем правом углу. Применяется сквозная нумерация листов, начиная с титульного листа и включая приложения. Номер листа на титульном листе не проставляют. Второй лист курсовой работы – оглавление. Оглавление, введение, главы, заключение, приложение начинаются с новой страницы. Заголовки выделяется полужирным шрифтом. Таблицы и рисунки могут располагаться как непосредственно в тексте, так и в приложениях. Таблицы и рисунки должны содержать заголовки, достаточно полно отражающие их содержание и специфику. Общий объем курсовой работы – не более 30 страниц формата А4.

Текст глав курсовой работы должен распределяться на параграфы. Главы и параграфы должны быть пронумерованы в пределах всей работы. «Введение», «Заключение» и «Список литературы» не нумеруются. Номер параграфа должен состоять из номера главы и номера параграфа, разделенный

точкой, например, «2.1.» (первый параграф второй главы). Каждую главу следует начинать с нового листа (страницы), а параграфы продолжать, отступив от предыдущего текста два межстрочных интервала.

Таблицы следует располагать в работе непосредственно после текста, в котором они упоминаются впервые, или на следующей странице. На все таблицы должны быть ссылки в тексте. Все таблицы, если их несколько, имеют сквозную нумерацию в пределах всего текста. Название и номер таблицы пишется перед таблицей с выравниванием по правому краю, например:

Таблица 1 – Объекты конфигурации

Если информация, приводимая в таблице, заимствована из каких-либо источников, то после названия таблицы необходимо поставить ссылку. Данные в таблице должны быть представлены шрифтом 12 размера и одинарным межстрочным интервалом без отступа (красной строки).

Если таблица имеет большой размер, то ее необходимо поместить в приложение. Если все же размещение таблицы в тексте признано более целесообразным, то она переносится на следующие страницы с автопереносом шапки таблицы (для этого в текстовых редакторах используется команда «Повторять как заголовок на каждой странице» в «Свойствах таблицы»).

Иллюстрации (чертежи, графики, схемы, диаграммы и др.) следует располагать непосредственно после текста, в котором они упоминаются впервые, или на следующей странице. На все иллюстрации должны быть ссылки в тексте. Иллюстрации – схемы и графики, именуемые рисунками, нумеруются сквозной нумерацией по всей работе. Название и номер указывается после иллюстрации с выравниванием по центру, например:

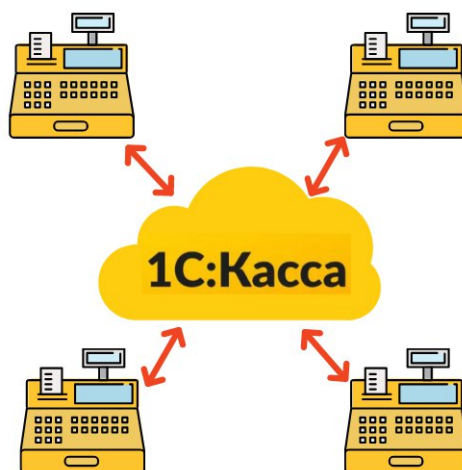


Рисунок 1 – 1С:Касса

Схемы в работе должны быть сгруппированы в единый объект.

Ссылки на иллюстрации не следует оформлять как самостоятельные фразы, в которых лишь повторяется то, что содержится в подписи. В том месте, где речь идет о теме, связанной с иллюстрацией, помещают ссылку либо в виде заключенного в скобки выражения «(Рис. 3)», либо в виде оборота типа: «...как это показано на рис. 3» или «... как это следует из рис. 3».

Каждую иллюстрацию необходимо снабжать подрисуночной подписью, которая должна соответствовать основному тексту и самой иллюстрации. Подпись под иллюстрацией имеет следующие основные элементы:

тематический заголовок иллюстрации, содержащий текст с характеристикой изображаемого объекта в наиболее краткой форме;

ссылка на источник, откуда взят рисунок, если это необходимо.

Следует помнить, что все таблицы, рисунки, схемы, иллюстрации и т.п. выполняются обучающимся самостоятельно с использованием соответствующих инструментов текстовых и графических редакторов.

После «Заключения» в курсовой работе приводится список литературы. Описание источников осуществляется в соответствии с ГОСТ Р 7.0.5-2008.

Общее количество источников в курсовой работе должно быть не менее 20 наименований.

При выполнении курсовой работы, обучающийся постоянно сверяет выполненные элементы работы с индикаторами оценивания курсовой работы.

5. ПОРЯДОК АТТЕСТАЦИИ И ЗАЩИТЫ КУРСОВОЙ РАБОТЫ

Законченная и полностью оформленная курсовая работа не позднее чем за две недели до начала экзаменационной сессии, загружается в Электронный университет для проверки руководителем и предварительной оценки. Обучающиеся заочной формы обучения (кроме групп выходного дня) представляют курсовую работу не позднее дня начала очередной сессии.

Руководитель проверяет работу на объем заимствований и при условии законченного правильного оформления и положительной оценки содержания, допускает работу к защите. Работа, не отвечающая установленным требованиям, возвращается для доработки с учетом сделанных замечаний и повторно загружается в Электронный университет в срок, указанный руководителем, но до начала экзаменационной сессии (по заочной форме – до зачета/экзамена по соответствующей дисциплине).

Курсовая работа подлежит проверке на наличие заимствований в установленном порядке. Также Обучающийся должен сделать запись на последней странице работы: «Настоящим подтверждаю, что курсовая работа

выполнена мною самостоятельно, заимствования находятся в допустимых пределах», а затем подписаться. В представленной курсовой работе допускается заимствование в объеме не более 30%. **Оригинальность текста курсовой работы должна составлять не менее 60%.** При уменьшении объема заимствования, ниже установленного, допускается увеличение объема цитирования. При этом оригинальность курсовой работы должна соответствовать установленным требованиям.

Защита курсовых работ проводится как на учебных занятиях, так и в ходе консультаций или специально отведенное в расписании время. На публичной защите необходимо показать не только знание темы и наличие работоспособной конфигурации, но и степень овладения научным методом мышления, логическим, динамичным, структурным и/или статистическим анализом исследуемых проблем, способность к самостоятельному научному труду, умение четко и ясно излагать свои мысли и выводы. Также важна способность показать сформированное и обоснованное личное мнение как результат проведенного авторского исследования.

Формой аттестации по курсовой работе является дифференцированный зачет («отлично», «хорошо», «удовлетворительно», «неудовлетворительно»). Оценка за курсовую работу выставляется руководителем по результатам защиты в ведомость.

~ При оценке курсовой работы учитываются:

- ~ содержание устного доклада;
- ~ актуальность темы
- ~ теоретический уровень и самостоятельность в постановке вопроса;
- ~ связь вопросов теории с практикой;
- ~ наличие работоспособной конфигурации;
- ~ наличие минимум 5 выполненных заданий курсовой работы;
- ~ обоснование выводов, рекомендаций, предложений и т.п.;
- ~ качество оформления работы;
- ~ использование в работе новейшей литературы, зарубежного опыта;
- ~ обоснованность ответов на вопросы преподавателя.

Обучающийся, не представивший в установленный срок курсовую работу или не защитивший ее по неуважительной причине, считается имеющим академическую задолженность.

ИНДИКАТОРЫ ОЦЕНИВАНИЯ КУРСОВОЙ РАБОТЫ:

Критерий	Показатель соответствия	Нормативные значения (баллы)	Оценка руководителя (баллы)
Соответствие оформления курсовой работы ГОСТ 7.32.-2017, ГОСТ 7.0.100-2018, ГОСТ Р 7.0.5-2008	полностью соответствует	10	
	соответствует с незначительными отклонениями	5	
	соответствует с существенными отклонениями	0	
Выполнение стадии формирования концепции ИС соответствует методике выполнения КР	полностью соответствует	20	
	соответствует с незначительными отклонениями	10	
	соответствует с существенными отклонениями	5	
	не соответствует	0	
Выполнение стадии проектирования ИС соответствует методике выполнения КР	полностью соответствует	30	
	соответствует с незначительными отклонениями	20	
	соответствует с существенными отклонениями	10	
	не соответствует	0	
Выполнение стадии реализации проектных решений соответствует методике выполнения КР	полностью соответствует	30	
	соответствует с незначительными отклонениями	20	
	соответствует с существенными отклонениями или не соответствует	10	
	не соответствует	0	
Приведен полный вывод по возможности использования конфигурации для автоматизации предметной области. Цель работы в заключении КР достигнута и подтверждена	полностью соответствует	10	
	соответствует с незначительными отклонениями	5	
	соответствует с существенными отклонениями или не соответствует	0	

Критерии оценивания

Оценка	Диапазон критерия оценивания
Отлично	от 85 до 100 баллов.
Хорошо	от 66 до 84 баллов;
Удовлетворительно	от 50 до 65 баллов;
Неудовлетворительно	49 баллов и менее

ВАРИАНТЫ ПРЕДМЕТНЫХ ОБЛАСТЕЙ

1. Проектирование информационной системы финансового контроля деятельности спортивно-развлекательного комплекса.
2. Проектирование информационной системы контроля использования государственных инвестиций.
3. Проектирование информационной системы банковских услуг нового розничного бизнеса.
4. Проектирование информационной системы бюджетирования
5. Проектирование информационной системы расчетов хозяйственных операций.
6. Проектирование информационной системы учета внутренних клиентских расчетов.
7. Проектирование информационной системы обработки заявок клиентов web-представительства.
8. Проектирование информационной системы учета продаж.
9. Проектирование информационной системы финансового анализа средствами «1С: бухгалтерия».
10. Проектирование информационной системы бухгалтерского учета имущества: амортизация и инвентаризация имущества.
11. Проектирование информационной системы управления учебным процессом для общеобразовательного учреждения.
12. Проектирование информационной системы бюджетного планирования в коммерческом банке.
13. Проектирование информационной системы абонентского финансового обслуживания клиентов.
14. Проектирование информационной системы мониторинга управления документооборотом.
15. Проектирование информационной системы регистрации и коммуникации в среде Internet участников конференций, семинаров.
16. Проектирование информационной системы автоматизированного рабочего места менеджера отдела продаж.
17. Проектирование информационной системы расчета заработной платы промышленной организации.
18. Проектирование информационной системы расчетов показателей деятельности предприятия.
19. Проектирование информационной системы расчетов финансово-экономических показателей.
20. Проектирование информационной системы формирования интегрированных электронных учебных курсов.
21. Проектирование информационной системы управленческого учета.

22. Проектирование информационной системы учета движения денежных средств.
23. Проектирование информационной системы учета движения клиентуры в стоматологии.
24. Проектирование информационной системы учета изготовления оконных и дверных блоков.
25. Проектирование информационной системы учета логистических операций.
26. Проектирование информационной системы учета рабочего времени и расчета заработной платы.
27. Проектирование информационной системы учета труда и начисления заработной платы.
28. Проектирование информационной системы формирования контекстной интернет-рекламы.
29. Проектирование информационной системы формирования отчетности в операционных кассах.
30. Проектирование информационной системы продаж спортивных товаров.
31. Проектирование информационной системы торговли.
32. Проектирование информационной системы ведения регламентных работ.
33. Проектирование информационной системы учета товаров.
34. Проектирование информационной подсистемы фокусировки баннерного показа портала.
35. Проектирование информационной системы формирования отчетности модуля анализа пассивов.
36. Проектирование информационной системы сбора и обработки оперативной финансовой информации.
37. Проектирование информационной системы управления приема заказов на продукцию через web-представительство.
38. Проектирование информационной системы мониторинга образовательного процесса.
39. Проектирование информационной системы мониторинга технического помещения и контроля затопления
40. Проектирование информационной системы учета расходов электроэнергии на устранение затоплений технических помещений



МОСКОВСКИЙ УНИВЕРСИТЕТ ИМ. С.Ю.ВИТТЕ

Факультет Информационных технологий

Кафедра Кафедра информационных систем

Направление подготовки/Специальность Бизнес-информатика

КУРСОВОЙ ПРОЕКТ

по дисциплине Проектирование информационных систем

Студента _____
(фамилия, имя, отчество)

на тему: Проектирование информационной системы для решения задач профессиональной деятельности
(тема работы)

Руководитель работы _____
(ученая степень, звание, инициалы и фамилия)

г.Москва 20 ____ г.

ПРИМЕР ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

Оглавление

Введение	_____
1. Формирование концепции информационной системы	_____
1.1. Общая характеристика объекта проектирования	_____
1.2. Функциональная подсистема и описание бизнес-процесса информационной системы	_____
1.3. Техническое задание на создание системы автоматизации исследуемого бизнес-процесса	_____
2. Проектирование информационной системы	_____
2.1. Постановка задачи	_____
2.2. Информационное моделирование бизнес-процесса	_____
2.3. Разработка динамической модели работы подсистемы	_____
3. Разработка проектных решений	_____
3.1. Выбор программного обеспечения для реализации функциональной подсистемы	_____
3.2. Реализация интерфейса подсистемы	_____
3.3. Разработка инструкций для пользователя функциональной подсистемы	_____
Выводы	_____
Список использованных источников	_____
Приложения	_____

Введение

Игры являются неотъемлемой частью уже сформировавшегося общества. Еще с самого детства человек начинает познавать мир с помощью игры так как процесс игр запоминается гораздо проще, чем обычное заучивание чего-либо.

Игра может не надоедать часами, следовательно, усидчивость человека возрастает в разы, а вместе с ней и усвоение материала. Человеку проще ассоциировать полученную информацию с каким-то действием. Именно поэтому обучение в игровой форме является наиболее эффективным методом обучения так как сочетает в себе игру и образовательный подтекст.

Игры учат не только определенным вещам или к примеру, точным наукам. Через игры человек может познавать мир, путешествовать и окунуться в ситуации, которые в реальной жизни сложно представить.

Объектом модернизации является тестирование, проводящееся в Личном кабинете Университета. Предметом исследования являются инструменты проведения тестирования.

1. Формирование концепции информационной системы

1.2. Общая характеристика объекта проектирования

Одной из главных составляющих любого образовательного учреждения, является проверка знаний студентов, а также их закрепление. Так и в университете представлена система для подобной оценки, а именно тесты, что представлены в личном кабинете у каждого обучающегося. Данное тестирование проводится для всех форм обучения включительно. На выполнение заданий дается ровно 45 минут, и, если студент справляется неудовлетворительно у него в запасе имеется вторая попытка, которую можно использовать сразу после окончания тестирования.

У теста имеются несколько видов вопросов, такие как:

- Вопросы с одним вариантом ответа;
- Вопросы с одним или несколькими вариантами ответа;
- Вопросы на определение последовательности;
- Вопросы на сопоставление;
- Вопросы с пропущенным словом или словосочетанием;
- Вопросы с развернутым ответом.

На базе этих вопросов выстроено тестирование для дальнейшего использования в качестве проверки знаний студентов по отведенной ему дисциплине в зависимости от направления студента.

Для удовлетворительного прохождения тестирования студенту нужно набрать более 50 баллов за тестирование, тогда этот результат можно считать допуском к сессии.

На основе полученных баллов система может рекомендовать поставить положительную оценку, если набранные баллы превышают отметку «75», тогда система в рекомендательных целях предложит преподавателю выставить оценку «хорошо» или в рамках зачета «зачтено». Если баллы студенты превышают отметку в «85» баллов соответственно, тогда система может рекомендовать оценку «отлично» за пройденную дисциплину.

Перечислим основные функции, которые выполняет данная система:

- Проверка знаний студента, на основе созданных преподавателем дисциплины тестов, которые в должной мере способны отобразить, тот, багаж знаний, который студент получил в ходе обучения в университете;
- Оценка знаний студента, то есть, возможность визуализировать подготовленность студента по тем или иным дисциплинам;
- Осуществляет контроль знаний студента, так как преподаватель может видеть корректность написания тестовых работ и тем самым имеет возможность корректировать программу обучения студентов в зависимости от результатов тестирования;
- Обучение студентов и развитие способности к быстрому поиску информации;
- Ведение учета.

1.3. Функциональная подсистема и описание бизнес-процесса

Система оценки знаний студентов имеет свои собственные процессы, которые по итогу выдают результат в виде балльной оценки знаний студента.

Поподробнее рассмотрим процессы, представленные в данной системе:

- Создание и последующая загрузка преподавателем или администраторам университета заданий в общую базу данных, где впоследствии выдается доступ на прохождение тестов студентам в их личных кабинетах;
- Подготовка и решение тестовых заданий студентом;
- Выполнение проверки выполненных заданий и последующее выставление баллов за каждое из них;

- Вывод результата оценивания знаний в личный кабинет студента.

Рассмотрим существующую технологию на примере диаграммы IDEF0.

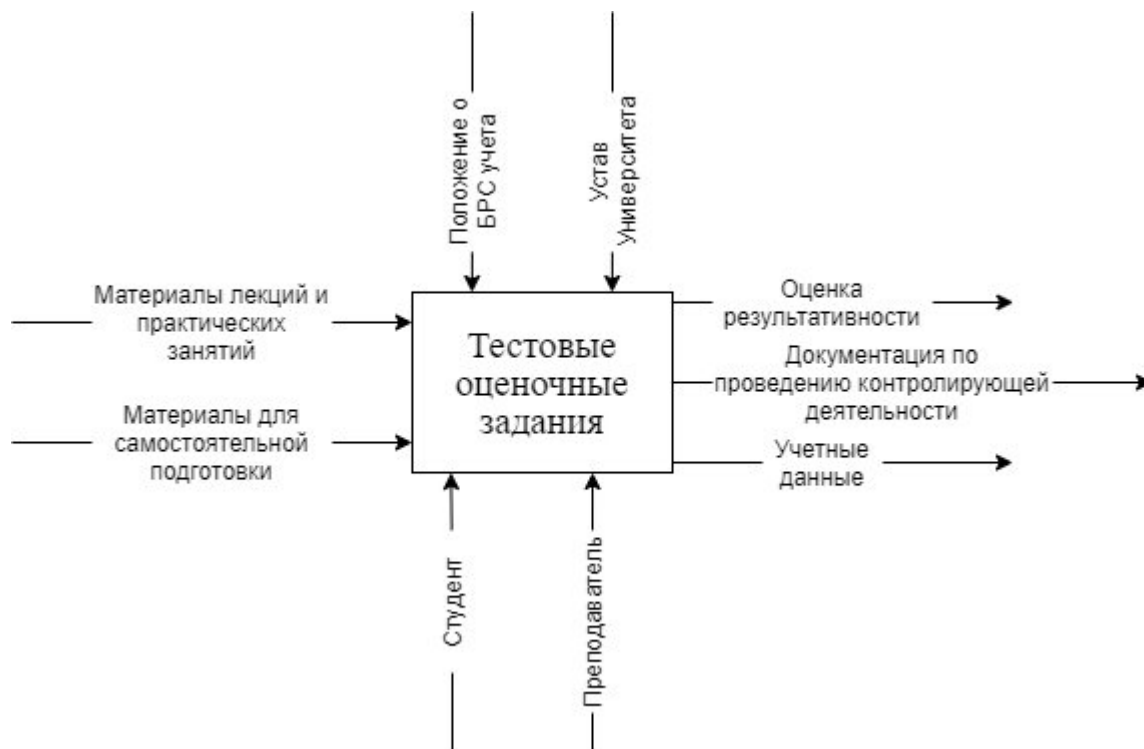


Схема 1. IDEF0 – Тестовые оценочные задания.

На данной схеме можно наблюдать, то, что управляющее воздействие исходит от:

- Положения о БРС учете;
- Устава Университета.

Ресурсами представленной технологии являются:

- Материалы лекций и семинарских занятий;
- Материалы для самостоятельной подготовки.

В виде механизма в данной системе выступает студент и преподаватель, как движущие ее части, без которых система не могла существовать.

В качестве продукта или результата представленной системы является:

- Оценка результативности студента;
- Документация по проведению контролирующей деятельности;
- Учетные данные.

1.4. Техническое задание на создание системы

Техническое задание (далее - ТЗ) на осуществление работ по созданию системы подготовки студентов к дисциплинарному и итоговому тестированиям (далее – Система). Задача предполагает разработку

программного продукта с возможностью для Игрока прочитать подготовительный материал, предусмотренный дисциплиной; возможностью прохождения теста и получения результата после каждого ответа на вопрос; возможностью получения удовольствия от игрового процесса и процесса обучения.

1.1. Вид и цели выполнения работ

Разработка и внедрение системы для обучения конечных пользователей.

1.2. Методические обязанности

При введении в эксплуатацию Системы следует следовать следующим документационным рекомендациям:

- ~ Приказ Министерства просвещения РФ от 20 февраля 2020 г. № 59 “Об утверждении Порядка проведения социально-психологического тестирования обучающихся в общеобразовательных организациях и профессиональных образовательных организациях”
- ~ Федеральный закон "Об образовании в Российской Федерации"

Результаты работы должны соответствовать следующим стандартам:

- ~ ISO/IEC 14102:2008 Оценка и выбор CASE- средств;
- ~ ISO 12119:1994 (ГОСТ Р – 2000 г) Требования к качеству и тестирование;
- ~ ISO 14764: 1999 (ГОСТ Р – 2002) Сопровождение программных средств;
- ~ ISO/IEC 15940:2013 Разработка систем и программ. Службы средств поддержки программных разработок;
- ~ ГОСТ Р 51904 – 2002 Программное обеспечение встроенных систем. Общие требования к разработке и документированию.

1.3. Полное и краткое наименование Системы

Полное наименование: Система подготовки студентов к дисциплинарному и итоговому тестированиям в Университете.

Краткое наименование: Система подготовки к тестированию.

1.4. Сроки выполнения работ

Работы должны быть выполнены не позднее 19.12.2021

1.5. Порядок оформления и предъявления заказчику результатов работ

Система передается в виде функционирующего комплекса на базе средств вычислительной техники Студентов в сроки, установленные в п. 1.4 настоящего ТЗ. Приемка результатов работ осуществляется комиссией

в составе уполномоченных представителей Заказчика и Исполнителя на территории Университета.

Порядок предъявления результатов работ, их испытаний и окончательной приемки определены в п. 6 настоящего ТЗ. Совместно с предъявлением результатов работ производится сдача разработанного Исполнителем комплекта документации согласно п. 7 настоящего ТЗ.

1.6. Заказчик

Московский университет имени С. Ю. Витте

1.7. По льзо ва те ли

Студенты Университета и лица, заинтересованные в подготовке к тестированию на базе Университета.

2. На зна че ние и цели создания системы

2.1. Цели внедрения Системы

Внедрение подобной системы преследует такие цели, как:

- Прохождение тестов в развлекательной форме;
- Возможность прохождения тестов в любой точке мира без подключения к сети интернет;
- Возможное поднятие настроения и последующая подготовка и настройка, направленная для решения тестов;
- Упрощенная подготовка студентов к тестированию.

2.2. За да чи, ре ша е мые при внедрении Системы

Система должна обеспечить решение следующих задач:

- Система должна в должной мере ознакомливать пользователей с тестами из личного кабинета студента;
- Система должна быть доступной на мобильных устройствах в любой точке мира;
- Система должна предоставлять пользователям информацию о корректности их знаний.

1.

2.

3. Требования к системе

3.1. Общие требования к Системе

Программа должна соответствовать следующим принципам:

- 1) Информационное единство Системы – вся информация должна находиться на одном мобильном устройстве.
- 2) Доступность получения системы – возможность скачать его на Сайте университета или в официальном магазине мобильных устройств на ОС Android – Play Маркет.

3.1.1. Требование к структуре Системы

Разрабатываемая система должна выполнять следующие функции: внесение данных ТОЗ и подготовительного материала; редактирование этих данных; возможность показать результат пройденного тестирования.

3.1.2. Требования к функционированию Системы

Для системы необходимо обеспечить возможность круглосуточного функционирования. Нахождение базы данных непосредственно на устройстве способно поддержать такой режим работы.

3.1.3. Требования к эксплуатации, техническому обслуживанию

Условия эксплуатации должны соответствовать общим техническим требованиям ГОСТ 21552-84.

3.1.4. Требования к защите информации от несанкционированного доступа

Для обеспечения юридической защиты персональных данных и для предотвращения утечки конфиденциальной информации, содержащейся в информационной базе, все элементы системы должны быть полностью совместимы с антивирусными средствами, средствами защиты от несанкционированного доступа и иными средствами защиты, используемыми у Заказчика и Пользователей Системы и штатно функционировать на устройствах, где они установлены.

3.1.5. Требования по стандартизации и унификации

Основой для унификации и стандартизации должны служить требования, установленные ГОСТ 34.201-89, ГОСТ 34.601-90, ГОСТ 34.602-89, ГОСТ 34.603-92, РД 50-34.698-90, ГОСТ Р 51583-2000.

3.2. Общие требования к функционированию Системы

Система должна соответствовать следующим системным требованиям:

- Работа на мобильном устройстве с ОС Android, подключенного к сети Интернет.
- Работа пользователей с текстовыми файлами на базе мобильного ПО.
- Одновременное выполнение операций лишь одним пользователем.
- Сохранение и вывод информации на экран монитора при нажатии на кнопку «Результаты пройденных тестов»

3.2.1. Требования к техническому и программному обеспечению

Клиентская часть Системы должна обеспечивать надлежащую работоспособность на имеющихся в распоряжении Заказчика и пользователей Системы ПЭВМ со следующими характеристиками аппаратного обеспечения, указанными в таблице 2.

Наименование параметра, ед. изм.	Значение
----------------------------------	----------

Процессор	Helio G25, 2.0 ГГц, аналогичный или выше
Оперативная память	1 Гб
Встроенная память	До 1 Гб свободного места
Операционная система	Android 5.0 и выше

Таблица 1. Характеристики мобильного телефона.

4. Состав и содержание работ

4.1. Требования к проведению обучения сотрудников

Программа обучения должна быть подготовлена Исполнителем и согласована с Заказчиком не позднее, чем за 10 рабочих дней до даты проведения обучения.

Исполнитель должен провести обучение сотрудников Заказчика. Количество слушателей определяется Заказчиком по согласованию с Исполнителем.

4.2. Техническая поддержка, гарантийное обслуживание прикладного ПО

Исполнитель гарантирует полную работоспособность Системы и исправление не выявленных ошибок.

4.3. Требования к результатам выполнения работ

4.3.1. Результат выполнения работ

Результатом выполнения работ должна быть система, работающая на устройствах Студентов согласно данного ТЗ.

Ввод Системы в эксплуатацию должен осуществляться в соответствии с ПНСТ 277-2018.

5. Порядок контроля и приемки системы

Дата сдачи-приемки выполненных работ назначается Заказчиком в течение 3 рабочих дней со дня получения письменного уведомления Подрядчика о готовности к сдаче результата выполненных работ и не может быть позднее чем через 7 рабочих дней со дня получения письменного уведомления Подрядчика.

Сдача-приёмка работ производится в соответствии с настоящим Техническим заданием и Программой, и методикой проведения испытаний, составленной в соответствии с ГОСТ 34.603.

6. Требования к документированию

Исполнитель должен разработать и передать Заказчику не позднее начала проведения приёмочных испытаний необходимую документацию на систему.

Документация на систему должна включать в себя: руководство пользователя; руководство администратора. Вся документация должна быть на русском языке.

2. Проектирование информационной системы

2.2. Постановка задачи

Постановка задачи осуществляется с помощью use-case диаграммы представленной на схеме 2.

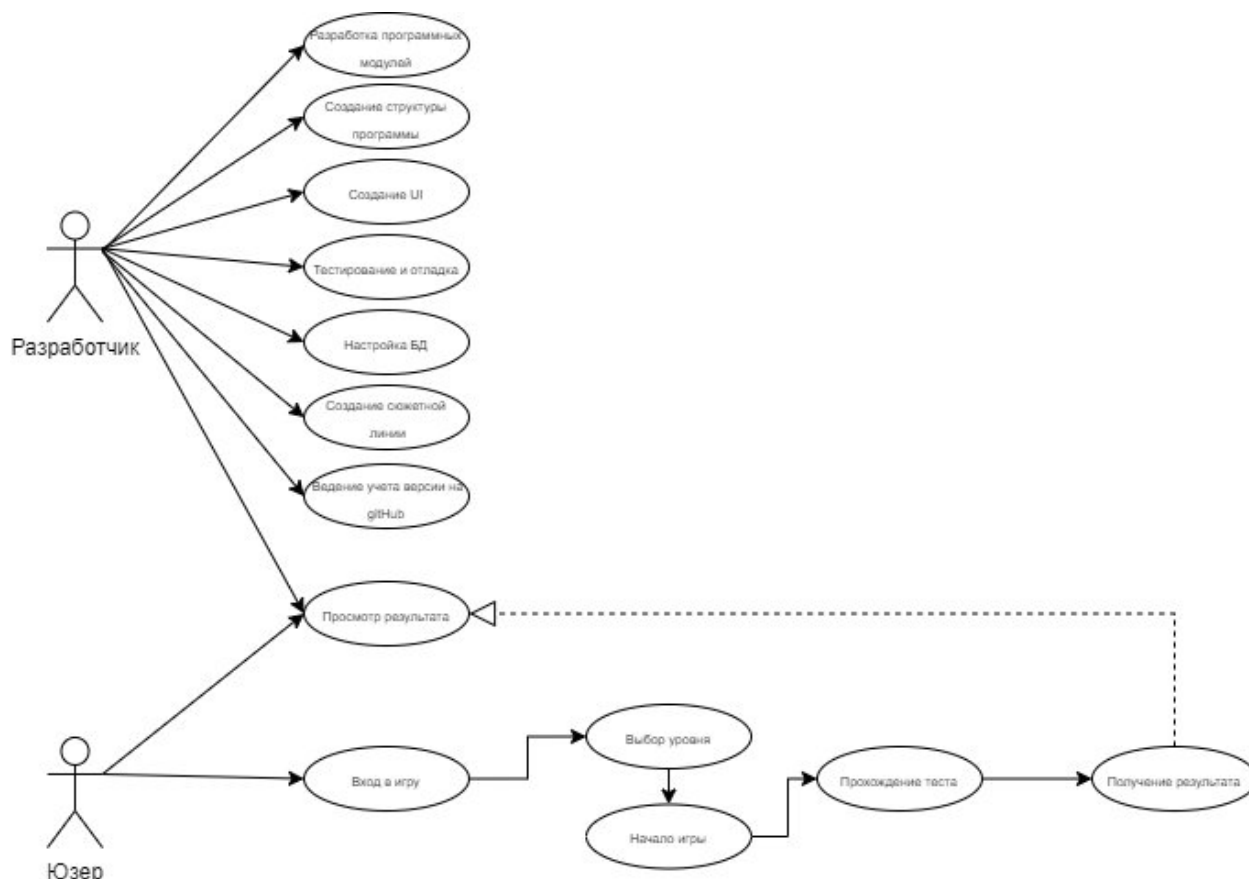


Схема 2. Use-case Диаграмма Системы.

На схеме продемонстрированы субъекты, которые учувствуют как в разработке, так и в непосредственном использовании разработанной системы, такие как разработчик и пользователь.

Разработчик по сути занимается разработкой и отладкой создаваемого проекта, а пользователь является в данной среде тестировщиком так как от него можно получить обратную связь и разработать решение для возникшей проблемы, если таковые будут появляться.

Возможные пути использования системы от лица разработчика:

- ~ Разработка программных модулей. Так как игра делится на два основных модуля, то и процесс разработки разделен на несколько частей. Один из разработчиков отвечает за основной модуль с игрой, а другой отвечает за модуль тестирования пользователей и за последующее его внедрение в основной модуль игры;

- ~ Создание структуры программы. Создание структуры программы лежит также на плечах разработчиков, но здесь уже нет отчетливого разделения на модули так как нужно разработать общую структуру программы для правильного, а главное эффективного использования программных модулей внутри проекта;
- ~ Создание UI. Данный процесс также не закреплен за каким-либо разработчиком системы так как оформление модуля может отличаться по мере его изменения в системе;
- ~ Тестирование и отладка. Тестированием и отладкой программных модулей занимается вся команда разработчиков, чтобы отменить факт человеческого фактора. Потому чем больше людей задействовано в тестировании проекта, тем меньше шанс проглядеть неисправность;
- ~ Настройка БД. Единственные данные, которые хранятся в игре – это банк вопросов, который осуществлен как объект в среде разработки для более простого доступа, а также редактирования и использования в ходе разработки;
- ~ Создание сюжетной линии. Так как это игра, то у нее должна быть какая-то заложенная идея. В данном случае, она направлена больше на развлекательный характер, чтобы конечный пользователь системы мог расслабиться и по возможности настроиться на решение тестовых заданий;
- ~ Ведение учета версий на GitHub. Для правильного хранения, а также для того, чтобы не потерять в случае непредвиденных обстоятельств сделанную работу, командой разработчиков было принято решение о выгрузки работы на сторонний сервис учета версий под названием GitHub.

Возможные пути использования системы от лица пользователя:

- ~ Вход в игру. Возможность открыть готовый проект и в дальнейшем использовать его по своему усмотрению;
- ~ Выбор уровня. Если входе разработки появится множество уровней, то пользователю будет дана возможность его выбора;
- ~ Начало игры. Пользователь нажимает кнопку, отвечающую за открытие главной сцены и переносится в саму игру;
- ~ Прохождение теста. После прохождения игрового уровня, появится интерактивная часть с тестом, где пользователю предлагается пройти его для дальнейшего прохождения игры;
- ~ Получение результата. После ответа на каждый из вопросов, пользователя уведомляют о том, правильно ли он ответил или нет.

В зависимости от ответа пользователя появляется конкретная всплывающая подсказка.

Общая возможность как пользователя, так и разработчика системы – это возможность просмотра результатов. Данная «use-case» диаграмма показывает постановку задачи для разрабатываемой системы, а также ее возможный функционал.

2.3. Информационное моделирование бизнес-процесса

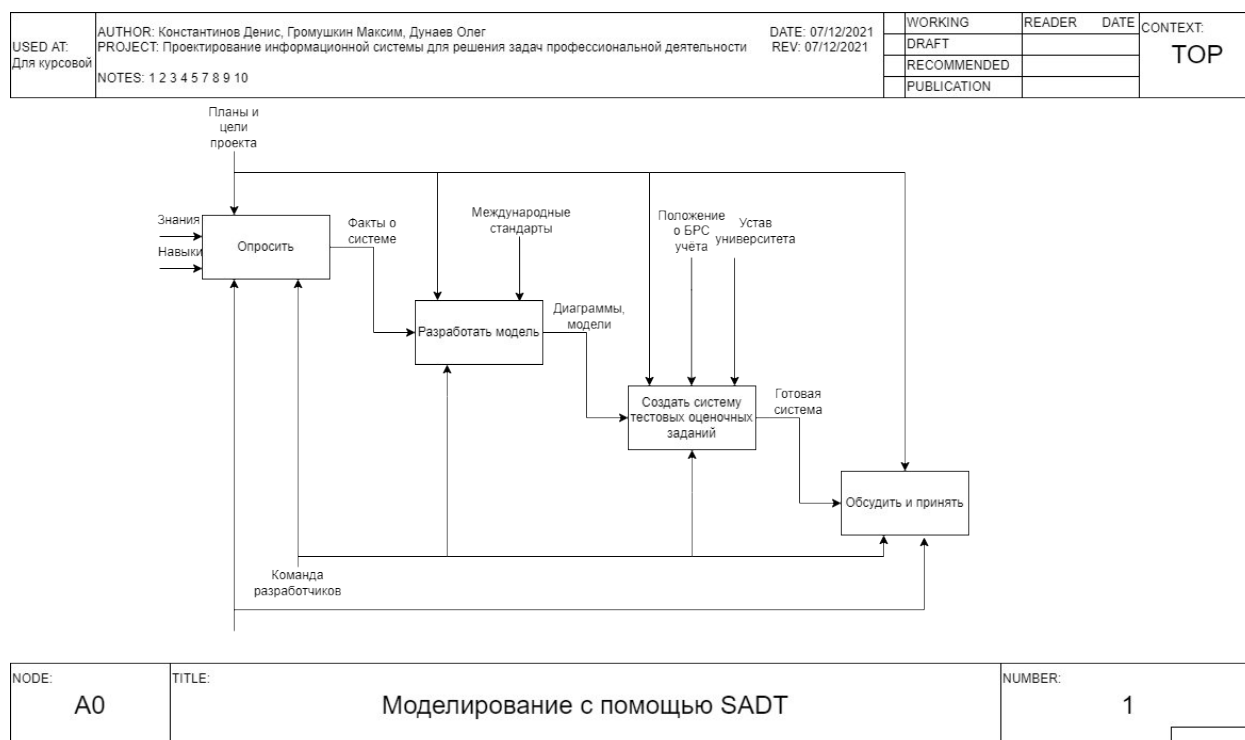


Схема 3. Моделирование с помощью SADT.

На схеме 3 представлен SADT подход на котором можно рассмотреть работу системы. Данную схему можно разбить на 4 функциональных блока, такие как:

1. Опросить;
2. Разработать модель;
3. Создать систему тестовых оценочных заданий;
4. Обсудить и принять.

Главное управляющее воздействие – это планы и цели проекта, ведь именно с них начинается построение и разработка проекта.

Рассмотрим подробнее каждый из функциональных блоков. Начнем с самого первого блока под названием «Опросить». В представленном блоке в виде входных данных принимаются:

~ Знания;

~ Навыки.

Они выступают наиболее важными входными данными так как с них и начинается оценочная составляющая проекта.

В виде механизма на каждом из функциональных блоков, выступают как команда разработчиков, так и студенты, либо преподавательский состав.

И результатом первого блока является факты о системе. Данный результат в свою очередь является входными данным для следующего функционального блока.

Второй блок, который несет название «Разработать систему» довольно схож с первым по параметрам механизма и не много по управляющему воздействию. Дополнением к нему идут международные стандарты, которые выступают ключевым звеном в разработке системе.

Результатом разработки системы являются:

~ Диаграммы;
~ Модели.

Полученный результат в свою очередь, является входными данными для третьего функционального блока под названием «Создать систему тестовых оценочных заданий». Для данного блока в управляющее воздействие добавляется такие пункты, как:

~ Положение о БРС учета;
~ Устав университета.

Результатом данного блока является уже готовая система, которая передается в четвертый по счет функциональный блок и поддается обсуждению и последующими принятию к работе.

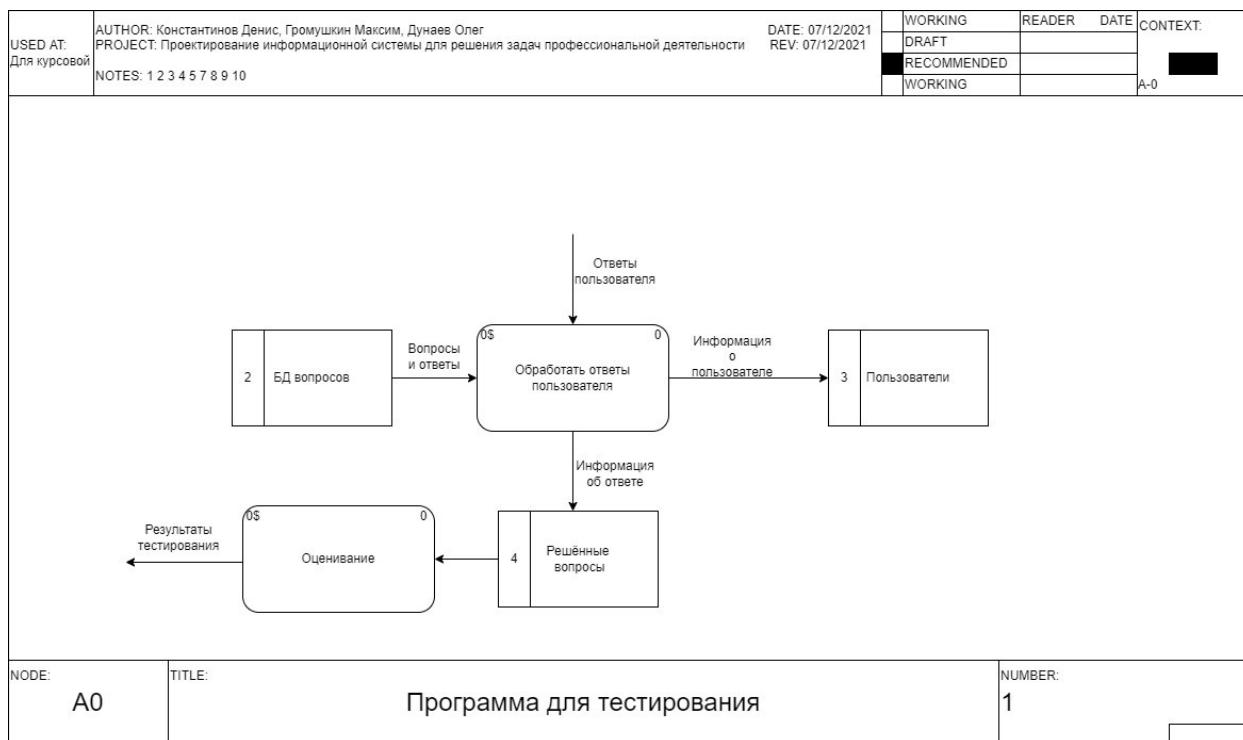


Схема 4. DFD декомпозиция «Программа для тестирования».

Программа для тестирования представляет собой отдельно разработанный модуль, который позволяет студенту пройти тестирование и получить возможный результат оценки своих знаний. В свою очередь это помогает ему не задействовать попытку в личном кабинете на сайте вуза, что в значительной мере помогает ему подготовиться к предстоящему тестированию.

Данная схема показывает обработку ответа пользователя, где ответ сравнивается с правильным ответом в базе данных. Затем если результат удовлетворителен, то пользователь уведомляется о том, что он ответил правильно, если пользователь имеет неудовлетворительный результат, то он уведомляется о том, что выбран неправильный вариант.

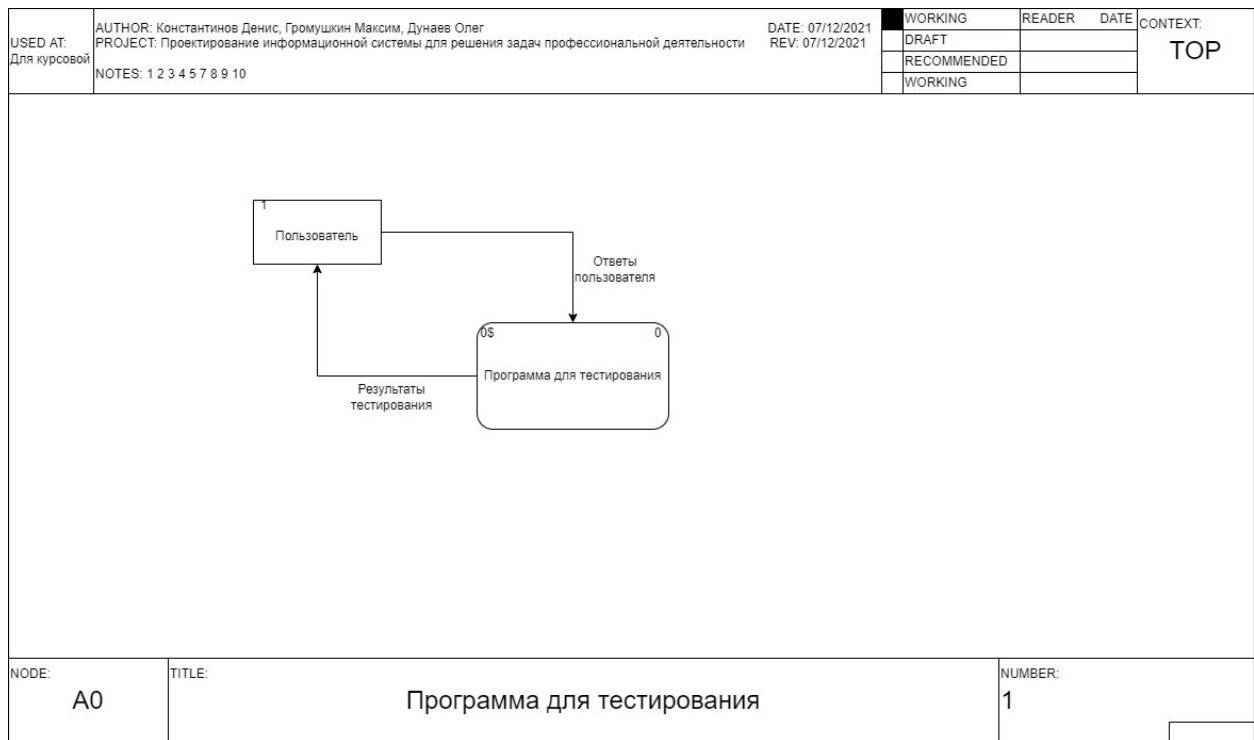


Схема 5. DFD «Программа для тестирования» верхний уровень.

На данной схеме представлено взаимодействие пользователя и самого модуля для тестирования. Взаимодействие происходит в такой последовательности:

1. Пользователь читает вопрос;
2. Пользователь отвечает на вопрос;
3. Программа для тестирования фиксирует ответ;
4. Сравнивает с правильным ответом в базе данных;
5. Фиксирует результат;
6. Показывает результат пользователю.

В зависимости от ответа пользователь может либо принять результат, либо продолжить тестирование, если ответ его не удовлетворяет.

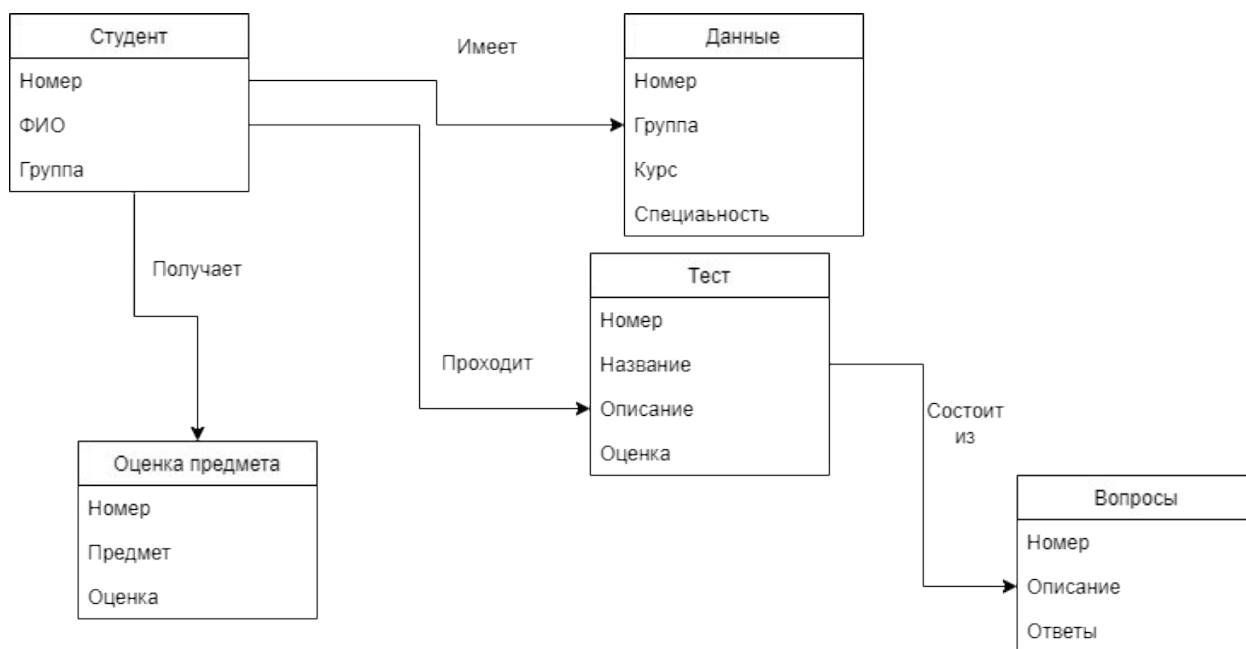


Схема 6. ER-Диаграмма системы тестирования.

Представленная диаграмма описывает взаимосвязь сущностей в программе, а также то, что имеют отдельные объекты.

3. Разработка проектных решений

3.2. Выбор программного обеспечения для реализации функциональной системы

Для возможности внедрения «квиз» части в игру, следует использовать тот же игровой движок, что и использует основа игры. При разработке игры выбор на такую платформа как «Unity», ввиду ее простоты использования, а также бесплатного распространения.

Логика игра прописывается на языке C#.

Для более удобной работы с кодом была использована такая среда разработки как Visual Studio.

3.3. Реализация интерфейса системы

Разработку системы можно разбить на несколько частей:

- Создание класса «Вопрос»;
- Создание системы хранения для класса «Вопрос»;
- Возможность присваивания значений в реальном времени.

Для начала разработаем класс «Вопрос».

Код данного класса можно рассмотреть на рисунке 1. Так называемый вопрос будет объектом системы Unity для последующего удобства в

редактировании и хранении. Сам объект будет содержать в себе такие редактируемые поля как:

- Вопрос;
- Массив с ответами;
- Правильный вариант ответа.

Также содержит логическое выражение «Asked». Это поле отвечает за то, был ли получен ответ на этот вопрос или нет.

И также имеется для проверки общего количества правильных ответов, так как вопросы не повторяются, то в какой-то моменты они могут закончиться. Поэтому метод проверяет количество правильных ответов и если это количество превышает максимальное число вопросов, то система «перезапускается» и появляется возможность повторного ответа на вопросы.

```
using UnityEditor;
using UnityEngine;

[CreateAssetMenuAttribute]
public class QuizQuestion : ScriptableObject
{
    [SerializeField]
    private string question;

    [SerializeField]
    private string[] answers;

    [SerializeField]
    private int correctAnswer;

    — ссылки
    public string Question { get { return question; } }
    — ссылки
    public string[] Answers { get { return answers; } }
    — ссылки
    public int CorrectAnswer { get { return correctAnswer; } }

    — ссылки
    public bool Asked { get; internal set; }

    Сообщение Unity | — ссылки
    private void OnValidate()
    {
        if (correctAnswer > answers.Length)
        {
            correctAnswer = 0;
        }
    }
}
```

Рисунок 1. Код класса «QuizQuestion».

Далее идет хранение всех вопросов и последующая загрузка их в систему для решения. Так как вопрос реализован в виде объекта, то хранится он в отдельной папке «Questions», которая расположена в ресурсах проекта.

Для загрузки был реализован класс, который может подгружать в себя все вопросы и затем манипулировать ими. При загрузке класс загружает все вопросы в отдельную переменную с помощью метода «LoadAllQuestions», в которой можно с легкостью взаимодействовать с ними.

Затем идет метод проверки результатом которого является все вопросы, на которые пользователь еще не дал ответ. Название метода «GetUnaskedQuestion».

Последние два метода в классе отвечают за повторное использование вопросов, если пользователь ответил на все вопросы, которые были ему предоставлены, то программа просто обновляет их для дальнейшего использования. Код данного класса представлен на рисунке 2.

```
using System.Linq;
using UnityEngine;

public class QuestionCollection : MonoBehaviour
{
    private QuizQuestion[] allQuestions;

    private void Awake()
    {
        LoadAllQuestions();
    }

    ссылка: 1
    private void LoadAllQuestions()
    {
        allQuestions = Resources.LoadAll<QuizQuestion>("Questions");
    }

    ссылка: 1
    public QuizQuestion GetUnaskedQuestion()
    {
        ResetQuestionsIfAllHaveBeenAsked();

        var question = allQuestions
            .Where(t => t.Asked == false)
            .OrderBy(t => UnityEngine.Random.Range(0, int.MaxValue))
            .FirstOrDefault();

        question.Asked = true;
        return question;
    }

    ссылка: 1
    private void ResetQuestionsIfAllHaveBeenAsked()
    {
        if (allQuestions.Any(t => t.Asked == false) == false)
        {
            ResetQuestions();
        }
    }

    ссылка: 1
    private void ResetQuestions()
    {
        foreach (var question in allQuestions)
            question.Asked = false;
    }
}
```

Рисунок 2. Код класса «QuestionCollection».

За подгрузку вопроса и его ответов на сцену отвечает класс «UIController». Его можно рассмотреть в приложении.

За логику программы отвечает класс «QuizController». На запуске программы инициализируются такие объекты как:

- Вопросы;

- UI элементы.

На старте выставляется вопрос с помощью метода «PresentQuestion».

И последний метод под названием «SubmitAnswer», который проверяет правильность ответа. Код представленного класса можно увидеть на Рисунке 3.

```
using System.Collections;
using UnityEngine;
using UnityEngine.UI;

[ScriptableObject]
public class QuizController : MonoBehaviour
{
    private QuestionCollection questionCollection;
    private QuizQuestion currentQuestion;
    private UIController uiController;
    [SerializeField]
    private float delayBetweenQuestions = 3f;

    [SerializeField]
    private void Awake()
    {
        questionCollection = FindObjectOfType<QuestionCollection>();
        uiController = FindObjectOfType<UIController>();
    }

    [SerializeField]
    private void Start()
    {
        PresentQuestion();
    }

    [SerializeField]
    private void PresentQuestion()
    {
        currentQuestion = questionCollection.GetUnaskedQuestion();
        uiController.SetupUIForQuestion(currentQuestion);
    }

    [SerializeField]
    public void SubmitAnswer(int answerNumber)
    {
        bool isCorrect = answerNumber == currentQuestion.CorrectAnswer;
        uiController.HandleSubmittedAnswer(isCorrect);

        StartCoroutine>ShowNextQuestionAfterDelay();
    }

    [SerializeField]
    private IEnumerator ShowNextQuestionAfterDelay()
    {
        yield return new WaitForSeconds(delayBetweenQuestions);
        PresentQuestion();
    }
}
```

Рисунок 3. Код класса «QuizController».

Также для автоматизированного преобразования вопросов в объекты Unity, был разработан скрипт, который автоматически вытаскивает вопросы из таблицы Excel и создает на основе шаблона и полученных данных новый объект. Код программы представлен ниже.

```
import os
```

```
import xlrd
```

```

fileName = input('Введите название файл с расширением → ')
def GetPattern():
    # Create an array.
    data = []
    # Open pattern for questions.
    pattern = open('Pattern.txt', 'r')
    # Fill array with data from pattern.
    for p in pattern:
        data.append(p)
    return data
def GetQuestions():
    # Create an array.
    data = []
    # Reading file with questions.
    wb = xlrd.open_workbook(fileName)
    sheet = wb.sheet_by_index(0)
    sheet.cell_value(0, 0)
    # Fill array.
    for i in range(0, sheet.nrows):
        data.append(sheet.row_values(i))
    return data
def SaveAsset(data, name):
    name = name[:-1]
    if name[-1] == (':' or '?'):
        name = name[:-1]
    try:
        path = os.getcwd() + '\\' + fileName.split('.')[0]
        os.mkdir(path)

```

```

except:
    pass

with open(os.path.join(path, name) + '.asset', 'w', encoding='utf-8') as asset:
    for d in data:
        asset.write(d)

def SetData(questions):
    for i in range(0, len(questions)):
        pattern = GetPattern()
        data = []
        for j in range(0, len(pattern)):
            if 'FILE_NAME' in pattern[j]:
                try:
                    pattern[j] = ' ' + pattern[j].split(' ')[2] + ' ' +
pattern[j].split(' ')[3].replace('FILE_NAME', questions[i][0].split(':')[0])
                except:
                    print('FILE_NAME\n' + questions[i][0])

            elif 'QUESTION_NAME' in pattern[j]:
                try:
                    pattern[j] = ' ' + pattern[j].split(' ')[2] + ' ' +
pattern[j].split(' ')[3].replace('QUESTION_NAME', questions[i][0].split(':')[0])
                except:
                    print('QUESTION_NAME\n' + questions[i][0])

            elif 'OPTION_1' in pattern[j]:
                try:
                    pattern[j] = ' ' + pattern[j].split(' ')[2] + ' ' +
pattern[j].split(' ')[3].replace('OPTION_1', (questions[i][1].split(' ~~~ ')[0]))
                except:

```

```

        print('OPTION_1\n' + questions[i][0])

    elif 'OPTION_2' in pattern[j]:
        try:
            pattern[j] = ' ' + pattern[j].split(' ')[2] + ' ' +
pattern[j].split(' ')[3].replace('OPTION_2', (questions[i][1].split(' ~~~ ')[1]))
        except:
            print('OPTION_2\n' + questions[i][0])

    elif 'OPTION_3' in pattern[j]:
        try:
            pattern[j] = ' ' + pattern[j].split(' ')[2] + ' ' +
pattern[j].split(' ')[3].replace('OPTION_3', (questions[i][1].split(' ~~~ ')[2]))
        except:
            print('OPTION_3\n' + questions[i][0])

    elif 'OPTION_4' in pattern[j]:
        try:
            pattern[j] = ' ' + pattern[j].split(' ')[2] + ' ' +
pattern[j].split(' ')[3].replace('OPTION_4', (questions[i][1].split(' ~~~ ')[3]))
        except:
            print('OPTION_4\n' + questions[i][0])

    elif 'CORRECT_ANSWER' in pattern[j]:
        try:
            pattern[j] = ' ' + pattern[j].split(' ')[2] + ' ' +
pattern[j].split(' ')[3].replace('CORRECT_ANSWER', str(questions[i][2]))
        except:
            print('CORRECT_ANSWER\n' + questions[i][0])

    data.append(pattern[j])

```

```

        SaveAsset(data, questions[i][0])

def main():
    SetData(GetQuestions())

if __name__ == '__main__':
    main()

```

Представленная программа работает с шаблоном, который изображен на рисунке 4.

```

%YAML 1.1
%TAG !u! tag:unity3d.com,2011:
--- !u!114 &11400000
MonoBehaviour:
  m_ObjectHideFlags: 0
  m_CorrespondingSourceObject: {fileID: 0}
  m_PrefabInstance: {fileID: 0}
  m_PrefabAsset: {fileID: 0}
  m_GameObject: {fileID: 0}
  m_Enabled: 1
  m_EditorHideFlags: 0
  m_Script: {fileID: 11500000, guid: b3b33ff31784a834f827054a2cce7f7b, type: 3}
  m_Name: FILE_NAME
  m_EditorClassIdentifier:
    question: QUESTION_NAME
    answers:
    - OPTION_1
    - OPTION_2
    - OPTION_3
    - OPTION_4
    correctAnswer: CORRECT_ANSWER

```

Рисунок 4. Шаблон Unity объекта.

Квиз система обладает набор таких UI элементами как:

- Текст;
- Кнопки;

В поле с текстом выводится вопрос, на который в последствие нужно будет ответить пользователю.

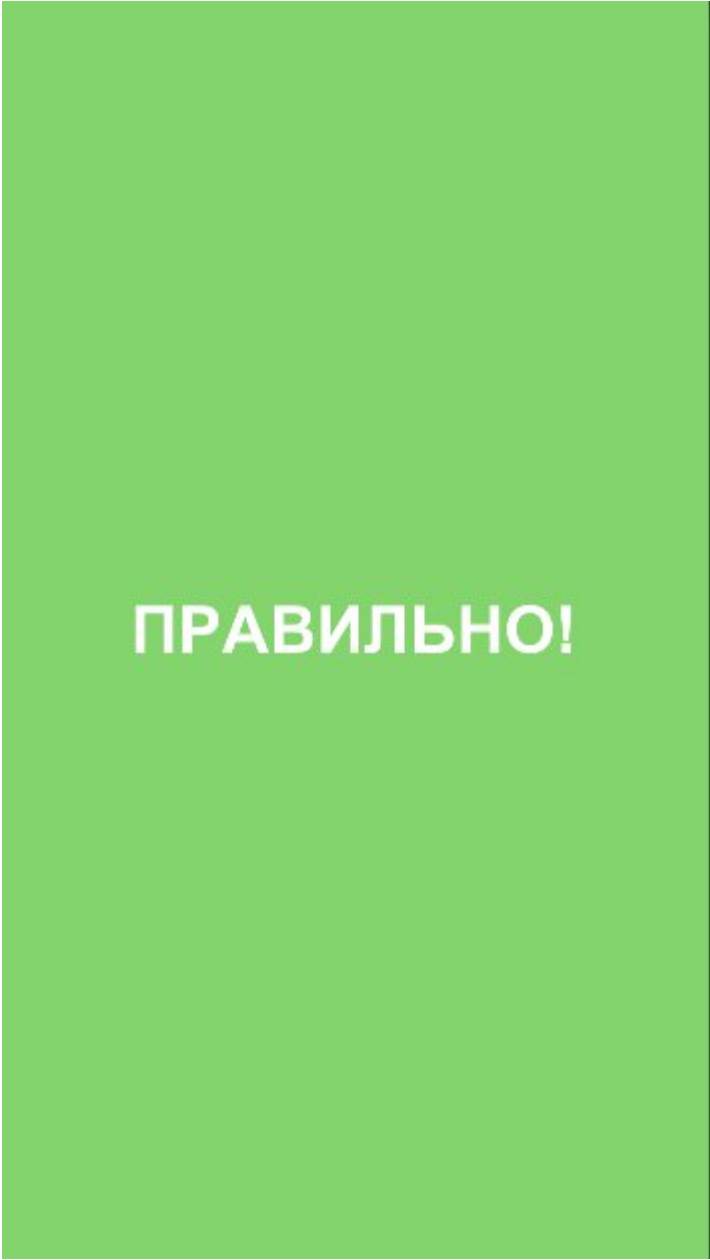
На самих кнопках, также содержится текст, на который воспроизводятся предполагаемые ответы на вопрос. Предполагаемый внешний вид модуля можно увидеть ниже на рисунках 5, 6 и 7 соответственно.

**Как иначе называвются
сведения, рассматриваемые
в каком либо контексте,
которые отличаются
новизной и полезностью для
пользователя?**



Рисунок 5. Внешний вид модуля «квиз».

На рисунке 5 можно заметить описанные ранее элементы UI, заполняются автоматически.



ПРАВИЛЬНО!

Рисунок 6. Правильный ответ.

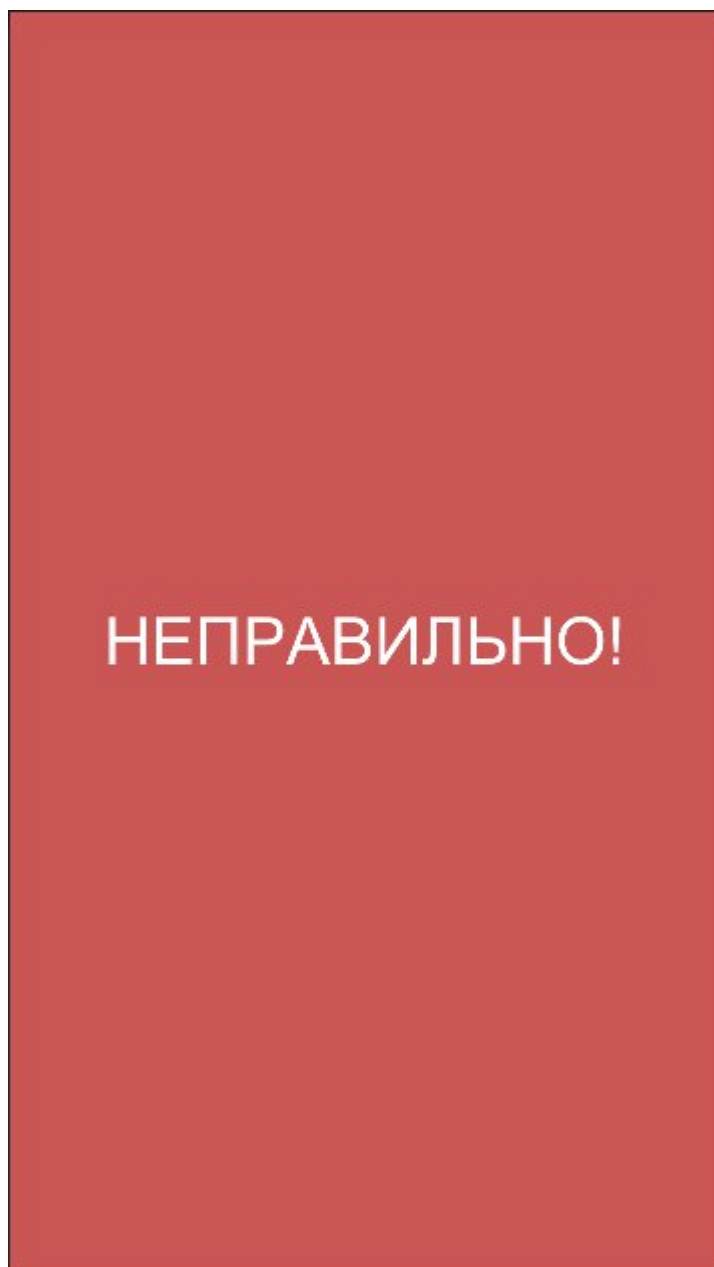


Рисунок 7. Неправильный ответ.

На рисунках 6 и 7, показаны всплывающие окна, которые уведомляют пользователя о том, как он ответил на вопрос – положительно или отрицательно. Данные окна всплывают на 3 секунды, затем исчезают, за это время вопрос и ответы успевают смениться на новые, которые пользователь еще не видел и не решал, но только в том, случае если пользователь не решил все вопросы из банка. Если же пользователь решил всё, то они могут повторяться.

1.1. Внедрение «квиз» модуля в игровую среду

Во внедрении данного модуля не возникло никаких проблем так как код позволяет быть внедренным в любую систему. Для осуществления данного

действия была создана новая сцена, где и располагался новый модуль для игры.

С помощью пустого объекта, названного «Core» были реализованы написанные скрипты путем простого помещения их в данный объект. На рисунке 8 можно увидеть, какие именно скрипты были задействованы в качестве «активных».

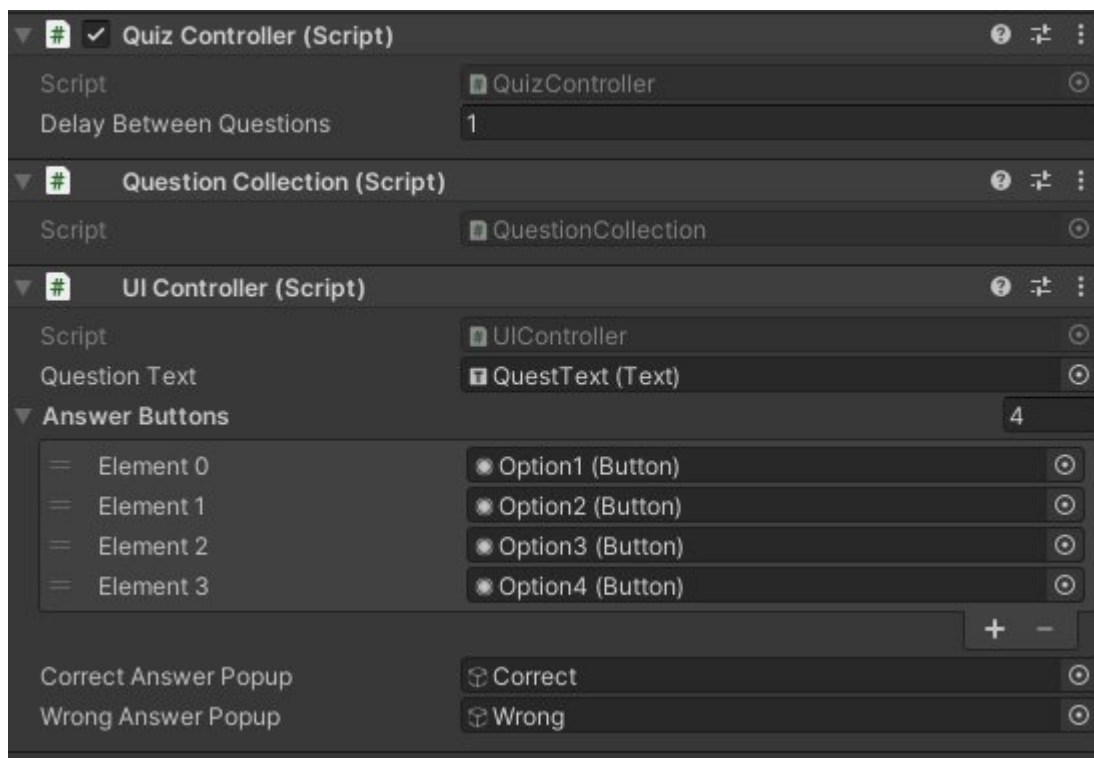


Рисунок 8. Скрипты объекта «Core».

Затем в поля помещаются объекты нужных типов. Ведь именно с ними будет работать система. Также была произведена настройка вопросов так как они хранятся как объекты Unity, то для них была создана отдельная папка с вопросами. Путь к папке с вопросами: «Resources\Questions». На этом внедрение программного модуля было закончено после чего последовало тестирование, в ходе которого не было выявлено никаких ошибок.

Тестирование и отладка модуля «квиз» осуществляется по таким действиям, как:

- ~ Вывод текста на экран в поле для вопроса;
- ~ Вывод текста на экран в полях доступных для вывода ответов;
- ~ Кликабельность кнопок для выбора ответов;
- ~ Работоспособность кнопок с выбором ответа;
- ~ Сохранения результата пользователя в рамках запущенной сессии;

- ~ Появление всплывающего экрана, который уведомляет пользователя о правильности выбранного ответа;
- ~ Появление всплывающего экрана, который уведомляет пользователя о неправильности выбранного ответа.

Для тестирования создадим тест-case таблицу, которую в свою очередь будем использовать для занесения результатов, полученных в ходе тестирования.

Действие	Ожидаемый результат	Результат теста
Вывод текста на экран в поле для вопроса	В поле с текстом для вопроса, появляется текст взятый из исходных данных вопроса	Passed
Вывод текста на экран в полях доступных для вывода ответов	В поля с текстом для ответов, появляется текст взятый из исходных данных вопроса	Passed
Кликабельность кнопок для выбора ответов	Возможность нажать на представленную кнопку	Passed
Работоспособность кнопок с выбором ответа	Сравнение выбранного ответа с правильным ответом и последующее оповещение пользователя о корректности его выбора	Passed
Сохранения результата пользователя в рамках запущенной сессии	Сохранение статистики по правильно отвеченным пользователем ответов в отдельную функцию, которая обнуляется при каждом перезапуске	Passed
Появление всплывающего экрана, который уведомляет пользователя о правильности выбранного ответа	При выборе правильного варианта ответа перед пользователем возникает окно, где его оповещают о том, что он ответил правильно на поставленный вопрос	Passed
Появление всплывающего экрана, который уведомляет пользователя о неправильности	При выборе правильного варианта ответа перед пользователем возникает окно, где его оповещают о том, что он ответил	Passed

выбранного ответа	неправильно на поставленный вопрос	
-------------------	------------------------------------	--

Таблица 2. Тест – case

В ходе тестирования не было замечено отказов в работоспособности модуля, но был выявлен неприятный баг с наложением текста друг на друга. При каждом новом вопросе поле текста не обновлялось, а текст просто отрисовывалось поверх старого.

Данная проблема была исправлена путем замены цвета текста. Благодаря этому теперь текст отображался правильно.

3.4. Разработка инструкций для пользователя системы

В ходе игры перед игроком предстает тест. Для дальнейшего прохождения игры, игроку нужно решить его.

После открытия теста у игрока на экране появляется: вопрос и 4 варианта ответов. Из которых ему предлагается выбрать наиболее правильный вариант ответа.

Дальнейшие действия игрока:

- ~ Прочитать вопрос;
- ~ Выбрать один из предложенных ответов.

После прохождения теста, результат игрока высвечивается на экране и в зависимости от правильности решения, игроку позволяется пройти дальше.

Заключение

В результате курсовой работы была спроектирована система подготовки студентов к дисциплинарному и итоговому тестированиям в Университете. Система позволяет в простой игровой форме подготовиться к оценке знаний.

Была проанализирована система тестов в личном кабинете университета и на ее основе составлен проект и разработан модуль «квиз», который содержит в себе некоторые вопросы из личного кабинета. Данные вопросы были преобразованы в специальный формат для более удобного применения в среде разработки Unity, а также для более удобного использования в самом модуле. Затем сам модуль был протестирован и внедрен в основной модуль игры.

Разработанная система имеет потенциал к усовершенствованию и переработке для более широкого круга пользователей, так как при необходимости материалы для подготовки и База данных с тестовыми заданиями может быть изменена или дополнена.

Приложение упрощает подготовку к тестированию студентов, так как этот процесс становится менее стрессовым. Система отвечает всем поставленным требованиям и имеет понятный пользовательский интерфейс

Таким образом, в ходе курсового проектирования были решены все поставленные задачи по разработке и внедрению.

Список использованной литературы

1. Албахари Дж. С# 6.0. Справочник. Полное описание языка / Дж. Албахари, Б. Албахари — 6-е изд. — Москва: Вильямс, 2016. — 1040 с.
2. Златопольский Д.М. Основы программирования на языке Python. — М.: ДМК Пресс, 2017. — 284 с.
3. Кубенский, А. А. Функциональное программирование: учебник и практикум для академического бакалавриата / А. А. Кубенский. — М.: Издательство Юрайт, 2019. — 348 с.
4. Кудрина, Е. В. Основы алгоритмизации и программирования на языке с#: учеб. пособие для СПО / Е. В. Кудрина, М. В. Огнева. — М.: Издательство Юрайт, 2019. — 322 с.
5. Свейгарт, Эл. Автоматизация рутинных задач с помощью Python: практическое руководство для начинающих. Пер. с англ. — М.: Вильямс, 2016. — 592 с.
6. Трофимов, В. В. Основы алгоритмизации и программирования: учебник для СПО / В. В. Трофимов, Т. А. Павловская; под ред. В. В. Трофимова. — М.: Издательство Юрайт, 2019. — 137 с.
7. Федоров, Д. Ю. Программирование на языке высокого уровня python: учеб. пособие для прикладного бакалавриата / Д. Ю. Федоров. — 2-е изд., перераб. и доп. — М.: Издательство Юрайт, 2019. — 161 с.
8. Федоров, Д. Ю. Программирование на языке высокого уровня python: учеб. пособие для СПО / Д. Ю. Федоров. — М.: Издательство Юрайт, 2019. — 126 с.
9. Unity в действии. Мультиплатформенная разработка на С#. - М.: Питер, 2018. - 608 с.



МОСКОВСКИЙ УНИВЕРСИТЕТ ИМЕНИ С.Ю. ВИТТЕ

Заведующему кафедрой
информационных систем

от студента (ки) _____ курса
факультета информационных
технологий

Фамилия _____

Имя _____

Отчество _____

ЗАЯВЛЕНИЕ

Прошу Вас утвердить мне тему курсовой работы _____

по дисциплине «Проектирование информационных систем» _____

Контактный телефон: _____

Электронная почта: _____

Дата _____

Подпись _____

«СОГЛАСОВАНО»

Заведующий кафедрой информационных систем _____ ФИО

Научный руководитель _____

« ____ » _____ 202__ г.