



Е. Н. Абраш, Н. В. Братусь, А. А. Лобанов, Ю. С. Лобанова

ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

Киров
2023



А. А. Лобанов, Ю. С. Лобанова, Е. Н. Абраш, Н. В. Братусь

ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

Учебное пособие

Издание 2-е, переработанное и дополненное

Под редакцией А. А. Лобанова

Киров
2023

© АНО ДПО «Межрегиональный центр инновационных технологий в образовании», 2023
© МИРЭА – Российский технологический университет, 2023
© Лобанов А. А., Лобанова Ю. С., Абраш Е. Н., Братусь Н. В., 2023

Авторы:

**Лобанов Александр Анатольевич,
Лобанова Юлия Сергеевна,
Абраш Елизавета Николаевна,
Братусь Надежда Валерьевна**

Рецензенты:

Цыпкин Владимир Николаевич, кандидат технических наук, доцент,
доцент кафедры проблем управления, заместитель директора Института
искусственного интеллекта МИРЭА по менеджменту качества обучения;

Воронов Юрий Александрович, кандидат технических наук,
доцент кафедры микро- и нанoeлектроники НИЯУ МИФИ

Л62 Лобанов, А. А. Проектирование информационных систем [Электронный ресурс]: учебное пособие / А. А. Лобанов, Ю. С. Лобанова, Е. Н. Абраш, Н. В. Братусь; под ред. А. А. Лобанова. – Электрон. текст. дан. (3,2 Мб). – Киров: Изд-во МЦИТО, 2023. – Изд. 2-е, перераб. и доп. – 1 электрон. опт. диск (CD-R). – Систем. требования: PC, Intel 1 ГГц, 512 Мб RAM, 3,2 Мб свобод. диск. пространства; CD-привод; ОС Windows XP и выше, ПО для чтения pdf-файлов. – Загл. с экрана. DOI: 10.52376/978-5-907743-35-9

ISBN 978-5-907743-35-9

Учебное электронное издание

Учебное пособие описывает все темы практических занятий дисциплины «Проектирование информационных систем», читаемой в Российском технологическом университете. Описано моделирование, проектирование и расчет параметров информационных систем. Даны исходные темы, примеры выполнения всех практических работ, параметры для расчета энтропии информационной системы. Во 2-м издании учебного пособия переработан материал, изложение которого, на взгляд авторов, стало более логичным. Добавлено подробное описание процедуры использования CASE-средств для создания функциональной модели и модели потоков данных проектируемой информационной системы. Учебное пособие может быть рекомендовано студентам, обучающимся по направлению профессиональной подготовки высшего профессионального образования 09.03.04 «Программная инженерия», и преподавателям, которые ведут практические занятия по дисциплинам «Проектирование информационных систем», «Проектирование архитектуры информационных систем» и подобным дисциплинам.

Учебное пособие издается в авторской редакции.

ISBN 978-5-907743-35-9

УДК 004.415.53:004.416.2
ББК 32.972

Оглавление

Введение	6
Общие указания по выполнению практических работ по дисциплине «Проектирование информационных систем»	7
Правила начисления дополнительных баллов за активность	11
Процесс выполнения и защиты работы	13
Практическая работа № 1. Формирование требований к системе	14
Цель работы	14
Теоретические сведения	14
Примерный состав требований к проектируемой информационной системе	19
Ход выполнения работы	20
Контрольные вопросы к практической работе № 1	23
Практическая работа № 2. Проектирование диаграммы прецедентов информационной системы в нотации UML	24
Цель работы	24
Ход работы	24
Теоретическая часть	24
Ход выполнения работы	26
Контрольные вопросы к практической работе № 2	29
Практическая работа № 3. Выбор (эскизное проектирование) архитектуры системы	30
Цель работы	30
Ход выполнения работы	30
Создание и оформление пояснительной записки эскизного проекта	35
ГОСТ Р 59795-2021 Пояснительная записка к эскизному проекту на создание автоматизированной системы (пример эскизного проекта)	37
Функциональная структура информационной системы	38
Контрольные вопросы к практической работе № 3	40
Практическая работа № 4. Функциональное проектирование модели информационной системы с использованием методологии SADT	41
Цель работы	41
Теоретические сведения	41
Ход выполнения работы	43
Практическая работа № 5. Проектирование функциональной модели информационной системы в нотации IDEF0	51
Цель работы	51
Ход работы	51
Практическая работа № 6. Проектирование модели потоков данных в нотации DFD	59
Цель работы	59
Ход работы	59
Контрольные вопросы к практическим работам № 4–6	66
Практическая работа № 7. Проектирование структуры данных информационной системы и создание ER-диаграммы	68
Цель работы	68
Теоретические сведения	69

Видеопример выполнения работы	73
Контрольные вопросы к практической работе № 7	76
Практическая работа № 8. Создание диаграммы состояний	78
Цель работы	78
Краткие теоретические сведения	78
Контрольные вопросы к практической работе № 8	81
Практическая работа № 9. Расчет информационной энтропии проектируемой системы	82
Цель работы	82
Краткие теоретические сведения	82
Обзор теории практической работы	83
Ход выполнения работы	83
Контрольные вопросы к практической работе № 9	84
Практическая работа № 10. Создание полного текстового описания, глоссария и расчет параметров проектируемой информационной системы	85
Цель работы	85
Ход выполнения работы	85
Контрольные вопросы к практической работе № 10	86
Список использованных источников	87
Приложение А. Темы для выполнения практических работ	89
Приложение Б. Пример выполнения практической работы № 1	91
Приложение В. Пример выполнения практической работы № 3	100
Приложение Г. Пример выполнения практической работы № 4	120
Приложение Д. Пример выполнения практической работы № 9	123

Введение

Настоящее учебное пособие подготовлено на кафедре инструментального и прикладного программного обеспечения МИРЭА – Российского технологического университета. Пособие представляет собой 2-е издание, дополненное и переработанное. Материалы, изложенные в учебном пособии, предназначены для поддержки процесса обучения. Предлагаемое учебное пособие дополняет учебники по проектированию информационных систем, раскрывая последовательность и правила проектирования информационных систем. В предлагаемом издании переработаны материалы и иллюстрации к практическим работам, пособие дополнено информацией об использовании CASE-средства Ramus Educational.

В учебном пособии доступным для студентов языком изложен порядок и правила формирования технического задания на создание автоматизированной системы. Подробно расписана методика создания функциональных диаграмм, диаграмм потоков данных и ER-диаграмм. Приведены основные положения и даны ссылки на нормативные документы, которые регламентируют порядок и отдельные этапы проектирования информационных и автоматизированных систем.

Последовательно рассматривается вопрос создания технического задания на автоматизированную систему, создание функциональной модели в нотации IDEF0, модели потоков данных в нотации DFD, а затем и логической модели базы данных в нотации ERD.

В данном учебном пособии приведены примеры правильно выполненных практических работ по курсу «Проектирование информационных систем» для обучения студентов по направлению подготовки 09.03.04 «Программная инженерия». Важным достоинством предлагаемого учебного пособия является достаточное подробное описание порядка использования современных свободно распространяемых CASE-средств, в частности Ramus Educational.

Пособие может быть полезно студентам, обучающимся дисциплине «Проектирование информационных систем», преподавателям, ведущим этот курс, аспирантам, а также широкому кругу специалистов, которые интересуются вопросами проектирования информационных систем.

Общие указания по выполнению практических работ по дисциплине «Проектирование информационных систем»

Для грамотного выполнения практических заданий дисциплины «Проектирование информационных систем», в первую очередь необходимо ознакомиться с общими указаниями по выполнению работ, представленными в настоящем пособии.

На первом практическом занятии семестра студент выбирает тему практических работ, которая проходит сквозь все практические работы. Выбранная тема определяет всю будущую информационную систему, которая проектируется по мере создания практических работ. Реальный процесс проектирования информационной системы является непрерывным, а отдельные крупные этапы невозможно согласовать с жесткими рамками практических работ. Поэтому разделение на работы носит, в некоторой степени, условный характер. Учебное пособие построено по следующему принципу: в каждой работе формулируется цель, определяются задачи, приводятся краткие теоретические сведения, которые необходимо знать проектировщику, приступающему к работе. Описывается процесс работы над заданием. В конце, приводится пример выполнения работы. Предложенный подход, на взгляд автора, позволяет создать комплект проектной документации проектируемой информационной системы (ИС) определённого назначения.

Задания выдаются студентам индивидуально (см. прил. А). Возможно использование темы, предложенной студентом, но только после согласования с преподавателем.

В отдельных случаях (при создании сложных систем или при создании информационной системы по заданию кафедры) после согласования с преподавателем, возможно выполнение работ группой студентов (не более двух человек).

Для выполнения проектных заданий в качестве нормативных документов используются государственные стандарты РФ [1, 2], руководящие документы и методические рекомендации по моделированию объектов предметной области [3, 4].

Во время практических занятий, преподаватель выполняет одновременно роль менеджера проекта и представителя заказчика. Как представитель заказчика – руководит постановкой проектных задач. В частности, с ним необходимо согласовывать вопросы содержания проекта. Обязательно согласовываются с преподавателем все вносимые изменения в проект. Как руководитель проекта (менеджер проекта) преподаватель дает рекомендации и согласовывает предложения студента по вопросам выполнения проектных задач. В задачи преподавателя входит проверка и оценка результатов практических работ. Проверке подлежит график выполнения работ. Для этого, в начале семестра, преподаватель самостоятельно формирует график дедлайнов, в зависимости от индивидуального расписания группы. Преподаватель контролирует и оценивает состав и качество решения проектной задачи в рамках практической работы, а также срок сдачи работ. Преподаватель **в праве** провести защиту выполненной работы с докладом студентов о промежуточных результатах.

Роль преподавателя в практических занятиях очень важна. Начальные данные для проектирования, как и в реальных проектах, являются неполными и слабо формализованными. В задачи студента, на начальных стадиях проектирования, входит постоянное уточнение ряда вопросов, связанных с целеполаганием и сбором как функциональных, так и нефункциональных требований у преподавателя, как представителя заказчика. По результатам таких консультаций должны быть уточнены следующие параметры проектируемой информационной системы: состав и содержание производственных функций, состав и содержание информационных потоков, возможные значения реквизитов документов и т. д. Также преподаватель консультирует по вопросам, связанным с методикой и методологией выполнения проектных заданий, принятия проектных решений.

Результат каждой промежуточной практической работы оформляется в виде отчёта (текстового файла, в формате *.doc/.docx или PDF). Отчеты должны быть оформлены в соответствии с ГОСТ 7.32-2017 [1]. Отчет предъявляется

преподавателю для проверки. В отчет необходимо внести исправления, если по результатам проверки преподавателем были сделаны замечания (справедливо для работ сданных до дедлайна). По запросу преподавателя может потребоваться подготовить краткий доклад с презентацией для выступления на очередном занятии. В результате выполнения всех заданий будет получен комплект проектной документации информационной системы, который оформляется как единый документ, с титульным листом, списком использованных источников и приложениями (если требуется). Преподавателем проверяется соответствие оформления отчета требованиям ГОСТ 7.32-2017 «Отчет о научно-исследовательской работе. Структура и правила оформления». Список источников оформляется по ГОСТ Р 7.0.100–2018. Библиографическая запись. Библиографическое описание: общие требования и правила составления [2]. Несоответствие оформления нормативным документам резко снижает оценку работы.

Каждая практическая работа оценивается следующим образом: сначала оцениваются сроки выполнения работ, в соответствии с графиком, установленным преподавателем, затем оценивается соответствие оформления отчёта нормативным документам и качество устного доклада с презентацией (если выполнение презентации и устного доклада, оговорено преподавателем). Выставляется оценка по этим параметрам. После оценки оформления и сроков, преподаватель оценивает качество выполнения задания, насколько технически грамотно решена поставленная инженерная задача. Выставляется вторая оценка. Рекомендуется выставлять две оценки. Дополнительные баллы выставляются только после того, как преподаватель зачет все работы, выполненные студентом.

Максимальное количество баллов за активность по дисциплине равно 25. Для получения баллов за активность по дисциплине в течение учебного семестра необходимо выполнить следующие действия с соблюдением следующих требований (см. табл. 1):

Таблица 1 – Действия и оценка для получения дополнительных баллов

Действие	Мин. балл	Макс. балл	Срок выполнения	Критерии оценки
Своевременная сдача всех практических работ	0	12,5	Сроки сдачи работ определяются преподавателем индивидуально в зависимости от расписания занятий в группе	Глубина проработки материала, раскрытие темы, умения и навыки работы с нормативными документами, понимание методологии. Грамотное техническое описание. Оформление пояснительной записки
Качественное оформление работы согласно требованиям ГОСТ 7.32-2017 и ГОСТ Р 7.0.100-2018	0	12,5	—//—	Соответствие требованиям нормативных документов

Правила начисления дополнительных баллов за активность

Дополнительные баллы за активность начисляются следующим образом: в семестре предусмотрено выполнение 10 практических работ. Каждая работа оценивается по 2 параметрам.

Технологическая реализация (смысловое и научное наполнение работы, т. е. насколько правильно решена поставленная задача) максимум в 10 баллов, при этом:

10 баллов – работа выполнена безукоризненно: все задачи решены, цель работы достигнута.

7 баллов – работа выполнена хорошо: цель работы достигнута, решены все задачи, хотя и не полностью.

5 баллов – работа в принципе выполнена (между хорошо и удовлетворительно): цель можно считать достигнутой при том, что, либо одна задача не решена, либо все задачи решены, но не полностью.

2 баллов – работа выполнена частично. Цель достигнута частично, не решено более одной из поставленных задач.

0 баллов – работа не выполнена.

Возможно выставление промежуточных баллов, для более объективной оценки работы.

Оформление и сроки сдачи каждой работы также оцениваются в 10 баллов, при этом до 5 баллов присваивается за своевременную сдачу работы и 5 баллов за оформление. Оформление работы должно соответствовать требованиям ГОСТ 7.32-2017.

10 баллов – работа выполнена в срок, оформление полностью соответствует ГОСТ 7.32-2017, смысл работы передан предельно ясно;

7 баллов – работа сдана с опозданием не более 3 рабочих дней, а оформление минимально отличается от ГОСТ 7.32-2017, смысл работы передан ясно;

5 баллов – работа сдана с опозданием более 3, но не более 6 рабочих дней, а оформление минимально отличается от ГОСТ 7.32-2017, смысл работы передан ясно;

2 баллов – работа сдана с опозданием более 6, но не более 9 рабочих дней, оформление минимально отличается от ГОСТ 7.32-2017, смысл работы передан ясно;

0 баллов – работа сдана с опозданием в более 9 рабочих дней, оформление минимально отличается от ГОСТ 7.32-2017, смысл работы передан неправильно.

Возможно выставление промежуточных баллов, для более объективной оценки работы. При этом работы, не соответствующие ГОСТ 7.32-2017, смысл которых не ясен, возвращаются на переделку авторам.

Таким образом, за одну работу студент может получить до 10 баллов за технологическую часть выполнения всех работ и до 10 баллов за своевременное выполнение и корректное оформление. Баллы, полученные в течение семестра, умножаются на коэффициент и округляются до 0,1 в большую сторону, но не более чем 12,5. Таким образом за активность в СДО РТУ МИРЭА можно получить дополнительно 12,5 баллов за технологическую часть выполнения всех работ и 12,5 баллов за своевременно корректное оформление всех работ, что в сумме соответствует максимально возможным 25 баллам.

Процесс выполнения и защиты работы

Выполнение каждой работы начинается с повторения теоретических сведений. Преподаватель проверяет готовность студентов к выполнению путем опроса (задавая контрольные вопросы).

Работы выполняются студентом как во время аудиторных занятий, так и самостоятельно. Выполнение работ состоит из следующих этапов:

1. Студенты получают допуск к работе после того, как будет выбрана тема для работ. Преподаватель должен утвердить тему, выбранную студентом.

2. Студент уточняет детали, необходимые для выполнения задания, у преподавателя, который выступает в роли заказчика/представителя заказчика. Выполняется задание итерационным способом, при консультации заказчика.

3. После того, как преподаватель утвердит материалы выполненной работы, студент оформляет отчет о проделанной работе.

4. Выполненная и оформленная работа подлежит защите.

Защита отчёта выполняется в следующем порядке:

1. Преподаватель проверяет оформление отчета на соответствие стандартам и нормативным документам.

2. После проверки оформления преподаватель переходит к проверке структуры и содержания отчёта.

3. Затем проверяется корректность выполненного описания и полученных результатов.

4. После того, как преподаватель убедиться в корректности всех аспектов работы, преподаватель задает контрольные вопросы. В процессе защиты необходимо, чтобы студент – автор работы объяснил основные аспекты работы, обосновав полученные результаты.

Отчет должен включать в себя:

1. Титульный лист (рекомендуется создать единый титульный лист для всех работ).

2. Задание на работу.

3. Содержание отчёта.

4. Описание результатов по каждой части задания.

4. Список использованных источников.

5. Приложения (тексты программ, содержание проектных документов, данные для расчета энтропии и т. п.).

Практическая работа № 1. Формирование требований к системе

Цель работы

Выполнить анализ предметной области создания информационной системы. Сформировать требования к проектируемой информационной системе.

Теоретические сведения

Разработка требований к информационной системе (ИС) – один из самых ответственных этапов среди всех этапов жизненного цикла информационной системы. Ошибки на этапе формирования требований являются наиболее дорогостоящими. Виной тому плохие требования. Плохие требования получаются в результате неправильного, неточного или неполного изучения и описания бизнес-процессов автоматизируемой организации, формальном (поверхностном) изучении источников информации и неправильном выборе самих этих источников. Но главная причина – формальное отношение к решаемой задаче. От полноты, точности и четкости формулирования требований зависит эффективность работы команды проекта и успех всего проекта, в том числе соблюдение сроков внедрения информационной системы и стоимость разработки, и ее качество. Все перечисленные атрибуты принято называть успешностью системы.

Формирование требований к ИС является, по сути, эвристической процедурой, которая плохо поддается формализации и алгоритмизации. Конечно, некоторые типовые проектные решения в области информационных технологий формализованы. Однако и такой подход имеет недостатки. Часто информационные системы, созданные с использованием ТПР (типовых проектных решений) не удовлетворяют заказчиков, что приводит к необходимости реализации большого числа доработок, что, в свою очередь, приводит к решению эвристических задач, а, следовательно, к увеличению стоимости проектируемой информационной системы. Заметим, однако, что в рамках данного учебного пособия рассматривается канонический подход к проектированию информационных систем. Если требования были сформированы верно, протестированы и уточне-

ны, процесс проектирования значительно упрощается. Кроме того, современные CASE-средства позволяют достаточно быстро создавать ИС с использованием готовых требований.

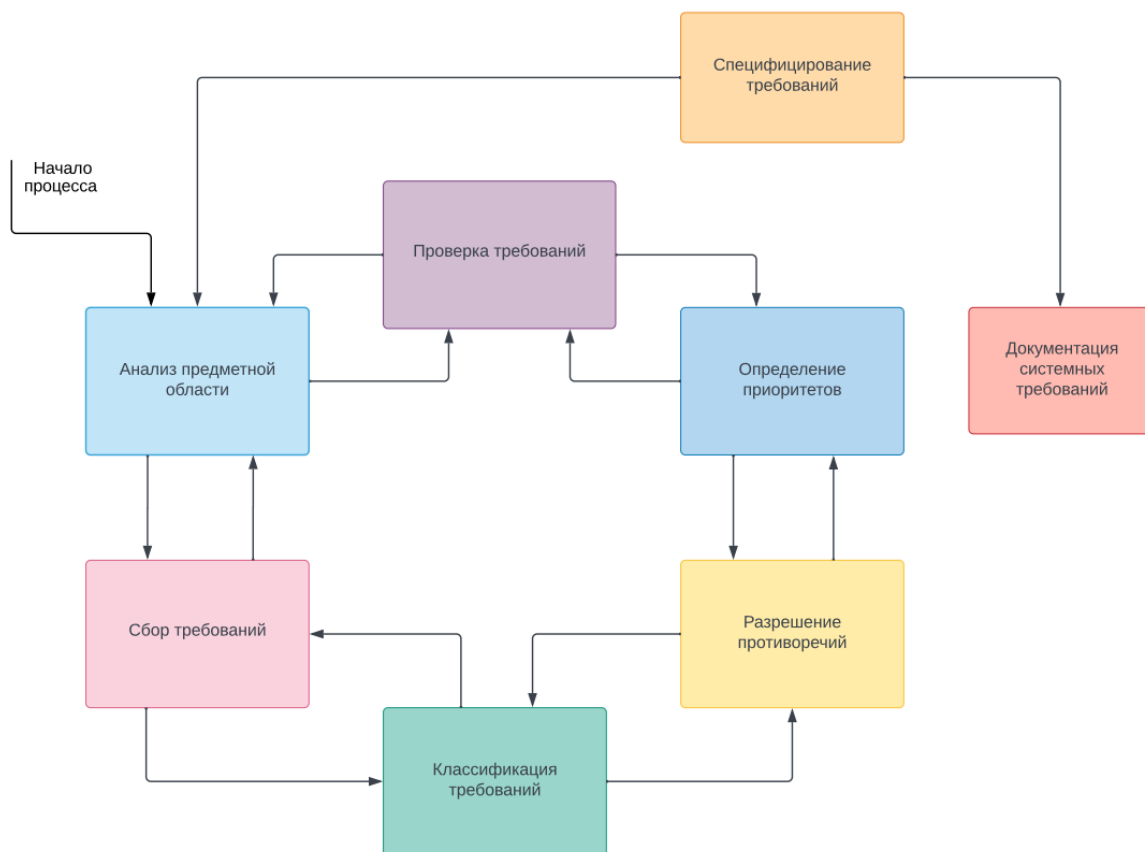


Рисунок 1 – Процесс формирования и анализа требований к создаваемой информационной системе

Этап разработки требований состоит из следующих шагов (см. рис. 1):

- Анализ предметной области необходим для того, чтобы описать внешнюю среду, в которой предполагается эксплуатировать создаваемую ИС. Анализ предметной области необходим для того, чтобы более точно понять и сформулировать требования заказчика.

- Сбор требований. В процессе сбора требований происходит взаимодействие с источниками информации, например, анализом предметной области, нормативными документами, учебной, научной и методической литературой, описывающей и регламентирующей принципы работы отрасли предприятия-заказчика. В процессе сбора требований необходимо взаимодействие со всеми участниками проекта, включая будущих пользователей создаваемой ИС.

– Классификация требований. В процессе классификации требований все собранные на предыдущем шаге требования трансформируются в группы, связанные логически и технологически.

– Разрешение противоречий. После того, как требования были сгруппированы, командой разработчиков выполняется большая работа по поиску зависимости в требованиях. Выделяются в первую очередь взаимозависимые требования. Затем выполняется анализ противоречий в требованиях. Если такие противоречия обнаруживаются, то ведется работа по совершенствованию требований с целью преодоления найденных противоречий. Взаимосвязанные требования отслеживаются на всем протяжении работы, поскольку изменения, внесенные в одно требование, неизбежно ведут к необходимости изменения взаимосвязанных.

– Назначение приоритетов. Требования, определенные и классифицированные ранее, ранжируются по степени влияния на проектируемую ИС. Коллектив проектировщиков следит за приоритетом реализации требований в первой и последующих версиях ИС.

– Проверка полноты требований. После того, как требования собраны, объединены в группы и выполнено ранжирование требований, команда проекта еще раз проверяет требования на полноту, последовательность и непротиворечивость.

Согласно исследованиям, проведенным иностранными учеными, ошибки, допущенные на стадии сбора требований, составляют от 40 до 60% всех дефектов проекта [5]. Пожалуй, выявление требований к информационной системе во всех деталях является наиболее трудной задачей. Детали требований необходимо выяснять в тесном взаимодействии с людьми, с механизмами и иными системами. Специалист, который формирует требования, должен не только хорошо владеть современными информационными технологиями, но и быстро разбираться в особенностях технологических процессов автоматизируемого предприятия или организации. Кроме того, следует уметь находить источники ин-

формации и верно их интерпретировать. Именно неправильно сформулированные требования могут принести больше всего проблем в процессе проектирования и разработки информационной системы.

Техническое задание (ТЗ) является основным документом, который определяет требования и порядок создания (развития или модернизации – далее создания) автоматизированной системы (АС) и в соответствии с которым проводится разработка АС и ее приемка при вводе в действие [4].

Накопленный опыт при разработке информационных систем позволяет утверждать, что в процессе управления требованиями (разработки, ранжирования, верификации и т. д.) разработчики чаще всего сталкиваются со следующими проблемами, приводящими к пересмотру списка требований и изменениям в ИС [6]: недостаточное вовлечение пользователей на всех этапах разработки требований, непонимание заказчиком требований, разработанных командой проекта, разночтения в понимании ключевых терминов и т. д.

Для решения этой проблемы, по мнению специалистов, необходимо: определить и разделить на классы пользователей проектируемой информационной системы; определить основные источники получения информации о потребностях клиентов; отобрать представителей пользователей/заинтересованных лиц из ранее определенных классов/групп и систематически работать с ними над корректностью требований, причем указанные ранее пользователи/заинтересованные лица должны обладать полномочиями принимать решения и действительно разбираться во всех тонкостях организации работ автоматизируемого предприятия. Сторонние приложения и аппаратные компоненты, с которыми будет взаимодействовать проектируемая система, должны быть рассмотрены как дополнительные классы пользователей.

Еще одной проблемой является так называемое «разрастание» требований пользователей. Под этим термином понимают увеличение объема требований, как в параметрах качества, так и количества, в процессе управления требованиями. Такое разрастание требований часто ведет к тому, что проект выходит, как

за рамки времени, так и бюджета, первоначально установленного заказчиком. Кроме того, разрастание требований приводит к пересмотру дизайна, разрыву логических связей внутри проектируемой информационной системы и в конечном счете к ухудшению успешности ИС. Для сохранения качества информационной системы в целом необходима работа по определению взаимозависимых требований для того, чтобы понять, как предполагаемые изменения отразятся на архитектуре и дизайне ИС. Только после этого возможно принятие решения о реализации этих изменений в проекте ИС.

Неоднозначность требований. Важнейшим правилом создания требований к информационной системе является однозначность их толкования. Важно, чтобы и команда проекта, и заказчик говорили на «одном языке». Во многом это связано с легкомысленным отношением как разработчиков, так и заказчиков к нормативным документам, стандартам и методикам в целом. Такое отношение, к сожалению, зарождается еще в пору обучения. Обучающиеся не хотят утруждать себя работой по изучению стандартов и нормативов, создавать такие документы, как список использованных нормативных документов, список принятых сокращений и глоссарий. Все это приводит к созданию некачественных, двусмысленных требований, да и в целом, ТЗ может быть истолковано по-разному. В случае создания неоднозначных требований у команды разработчиков и заказчика возникают разные представления о создаваемой ИС. Каждый живет в своей выдуманной реальности. Крах наступает, когда готовое решение вводится в эксплуатацию. Проблема может быть решена только во время личного общения между командой проекта и заказчиками. Следует также представлять материал заказчику в том виде, который ему понятен.

Для того чтобы требования были действительно качественными, а создаваемая система – успешной, требуется постоянное взаимодействие между командой проекта и представителями заказчика. Необходимо охватить все уровни сотрудников заказчика, от топ-менеджмента до рядовых операторов, которые должны будут ежедневно использовать создаваемую ИС. Чтобы информация

была доступна для понимания и анализа представителями заказчика следует как можно больше использовать функциональные модели и диаграммы, в таких нотациях как IDEF0, DFD, BPMN, UML, FlowChart и др. Следует широко использовать таблицы и другие приемы визуализации требований.

Более подробно про формирование и управление требованиями можно узнать, прочитав пособие Тестирование и отладка программного обеспечения [7], подготовленное на нашей кафедре. Для углубленного изучения вопросов, связанных с разработкой требований к программному обеспечению, можно использовать книгу Карла Вигерса и Джойа Битти [8].

Следует помнить, что позиция некоторых разработчиков типа: «зачем читать ТЗ, если сразу можно писать код» является по сути варварской и несовременной. При таком подходе конечный продукт, в общем случае, не соответствует запросам заинтересованной стороны. Риск возникновения ошибок резко возрастает, а, следовательно, возрастает и цена ошибок. Некоторые исследователи приводят пугающие данные, что обнаружение ошибки пользователем после релиза может увеличить стоимость решения в 21 раз [9]. Поэтому студенты в рамках первой работы должны очень ответственно подойти к разработке требований. Нельзя относиться к этой работе как к проходной и выполнять её только ради сдачи. Следует внимательно и вдумчиво составить требования к проектируемой ИС, детально обсудить их с преподавателем, ведущим практические занятия, внимательно выслушать конструктивную критику и тщательно переработать требования.

Примерный состав требований к проектируемой информационной системе

При разработке требований нужно определить уровень информационной системы. Так как под определение «информационная система», которое приводится в лекционном курсе, подходит и небольшой интернет-магазин, и колоссальные по своей сложности ERP-системы. Если ведется проектирование небольшой информационной системы с числом пользователей менее 1000 и объемом хранимых данных менее 1 Тбайт, можно использовать различные упрощенные способы создания технического задания.

Ход выполнения работы

Главным является вопрос понимания созданного ТЗ заказчиком. Так, например, можно использовать таблицы для описания пользователей и действий пользователей. Пример такой таблицы приведен в таблице 2.

Таблица 2 – Пример оформления функциональных требований для разработки веб-приложения

Целевая аудитория	Задачи	Задействованный функционал	Задействованные страницы
Потенциальные или нерегулярные клиенты	Ознакомиться с общей информацией о компании	->	Главная страница
	Ознакомиться с деятельностью отдельных ферм	->	Страницы ферм
	Отправить заявку на катание на животном	Отправка сообщений через форму обратной связи	Главная страница, страницы ферм
Лояльные клиенты	Получить контактную информацию	->	Все страницы сайта (шапка сайта)

После создания такой таблицы можно придумать некоего «конкретного» пользователя, глазами которого надо просмотреть всю создаваемую систему, описать пользовательскую историю (user story) и предложить заказчику для утверждения. Более подробно про данный подход можно узнать, посмотрев вебинары проекта Школа Вебмастеров, спикер – Алексей Бородин [10, 11, 12].

Интересным является подход к разработке ТЗ для веб-проектов Руслана Раянова. Автор предлагает создавать ТЗ поэтапно. Такой подход согласуется со спиральной моделью жизненного цикла и по утверждению автора является «весьма клиентоориентированным». Действительно, во-первых, первоначально заказчик, который, как правило, не имеет специального образования в области информационных технологий, не может оценить весь огромный документ ТЗ, составленный, например, по ГОСТ. Во-вторых, часто требования возникают уже в процессе эксплуатации ИС. Таким образом, сформулировав ТЗ к первому этапу создания ИС с функционалом, например, 25%, мы не рискуем всем про-

ектом. Более того, после ввода в эксплуатацию объекта с 25% функционалом для заказчика становится более понятна идеология ТЗ. Появляются новые требования, которые ранее не были очевидны для заказчика. ТЗ на второй этап для реализации ИС с 50% функционалом становится более конкретным и объективным. Более подробно с описываемым подходом можно ознакомиться на странице проекта Falcon Space [13]. На сайте проекта также можно скачать шаблон ТЗ по данной методике¹.

Представим далее перечень возможной документации на информационную систему:

1. Общая часть:

- список использованных нормативных документов;
- список терминов и определений;
- глоссарий;
- описание бизнес-ролей.

2. Требования к системе:

- функциональные требования;
- нефункциональные требования;

3. Макет информационной системы.

Общая часть включает материалы и определения, необходимые для анализа и управления требованиями, процесса проектирования, создания тестовых сценариев и т. д. По сути, в этой части определяется философия и методология проектирования ИС.

Функциональные требования – это описание требований пользователей, как правило в виде user-stories (описание, что должна делать система). Функциональные, как правило, формируются с использованием текстовых предписаний, например, «сотрудник склада в процессе использования системы должен формировать приходные и расходные накладные».

¹ Шаблон ТЗ https://docs.google.com/document/d/1ELoKZa4swO9xXXNSTwyYD2X4gfAkUfV_/edit

Кроме этого, функциональные требования можно объединять в use case или, по-русски, варианты использования, сценарии использования или прецеденты использования. Сценарий использования – это некая абстракция, которая включает диаграммы, составленные по определенным правилам, и текстовые документы, поясняющие диаграммы. Сценарий использования позволяет описать поведение системы при взаимодействии с объектами внешней среды. В нотации UML 2.4.1² прецеденты являются средством для определения требуемых использований системы [14]. Поведение информационной системы в целом, или отдельных ее модулей, необходимое заказчику, определяется одним или несколькими вариантами использования, которые задаются в соответствии с потребностями акторов (субъектов в UML).

Нефункциональные требования – определяют внутренние и внешние условия или атрибуты функционирования системы. К. Вигерс [8] выделяет следующие основные группы нефункциональных требований:

- Внешние интерфейсы (External Interfaces).
- Атрибуты качества (Quality Attributes).
- Ограничения (Constraints).

User Interface (сокращенно – UI), или интерфейс пользователя, является наиболее важным по сравнению с другими внешними интерфейсами информационных систем. Помимо этого, можно выделить аппаратные интерфейсы (взаимодействие с внешними устройствами), программные интерфейсы и интерфейсы передачи информации (коммуникационные интерфейсы) [14].

Для формирования атрибутов качества информационных систем: применимость, надежность, производительность, функциональная пригодность, можно использовать модель FURPS+. Более подробно про модель FURPS+ можно узнать из лекционного курса дисциплины «Проектирование информационных систем».

К ограничениям следует отнести такие факторы, как формулировка условий, выбор платформы реализации и/или развертывания, в том числе протоко-

² <http://www.omg.org/spec/UML/2.4.1/>

лы, серверы приложений, баз данных, и т. д., которые, в свою очередь, могут относиться, например, к внешним интерфейсам [7, 8].

При создании сложных информационных систем, включающих в себя большое число разнородных аппаратных средств (не только сервер и аппаратную часть клиента), необходимо создание ТЗ в соответствии с ГОСТ 34.602-2020 [4].

Пример выполнения практической работы №1 приведен в приложении Б.

Контрольные вопросы к практической работе № 1

1. Что такое требование к информационной системе?
2. Какие нормативные документы регламентируют сбор и управление требованиями к информационной системе?
3. Что является важным при формировании нефункциональных требований к информационной системе?
4. Какие средства существуют для формирования функциональных требований к информационной системе?
5. При помощи каких диаграмм можно описать функциональные требования к информационной системе?
6. Для чего выполняется ранжирование требований к информационной системе?
7. Что значит связные требования к информационной системе?
8. Приведите критерии, определяющие качество собранных требований к информационной системе.

Практическая работа № 2. Проектирование диаграммы прецедентов информационной системы в нотации UML

Цель работы

Основная цель работы – создать диаграмму прецедентов (use case) для одного из классов или прецедентов проектируемой информационной системы. В процессе достижения цели студенты получают навыки создания и использования диаграмм UML.

Ход работы

1. Ознакомиться с теоретической частью.
2. Выполнить практическое задание.
3. Оформить отчет.

Теоретическая часть

Диаграмма прецедентов (также use case или диаграмма вариантов использования) создается для описания общих функциональных требований к системе. Углубленное проектирование системы требует более детального описания, которое создается, в частности, с помощью диаграмм вариантов использования и в диаграммах, описывающих поток событий.

Использование диаграммы вариантов использования в процессе проектирования информационной системы позволяет определить: пользователей и границы проектируемой информационной системы; интерфейс системы. С помощью диаграммы use case удобно общаться проектировщикам и разработчикам. С ее помощью можно создавать тесты и пользовательскую документацию. Причем диаграмму прецедентов можно использовать как при объектно-ориентированном, так и при структурном подходе к проектированию.

Основными элементами диаграмм вариантов использования являются «активный субъект» или actor. Actor следует переводить с английского как участник, исполнитель, действующее лицо или действующий субъект, а не актёр! Согласно приведённому переводу, это субъект (человек, функция, модуль ИС, внешняя система, подсистема, организация, класс и т. п.), который взаимо-

действует с проектируемой нами системой. Под взаимодействием понимается любое воздействие субъектов друг на друга, которое изменяет поведение или состояние проектируемой нами системы.

Вариант использования (прецедент или use case) – графическое описание некоторого набора последовательных событий (включая варианты этих событий), выполнение системой которых приводит к тому результату, который наблюдает участник. Важно, что с помощью прецедентов можно описать что происходит в системе. Use cases не отвечают на вопрос «как», т. е. при этом не описывается каким образом достигается результат, а показывается «что» происходит. Графически варианты использования (прецеденты) принято обозначать как эллипс. Внутри эллипса приводят название. Название блока должно отражать суть функций проектируемой системы при взаимодействии между активным субъектом и системой. Каждый прецедент подразумевает определенный поток событий, который происходит по мере выполнения описываемой функции системы. Как было отмечено ранее, описание потока событий должно определять, **ЧТО** должно быть осуществлено, а не то, **КАК** это должно быть осуществлено.

Между actor'ом (активным субъектом) и прецедентом на схеме должна быть показана ассоциативная связь (association relationship). Связь необходима на схеме для того, чтобы показывать взаимодействие субъекта с системой в рамках предлагаемого варианта использования. Направление связи указывает на инициатора взаимодействия. Инициатором может быть, как субъект, так и система. Связи также могут быть установлены между вариантами использования. Диаграммы UML предлагают два типа связей – включающие (inclusive) и расширяющие (extensive). Отношение включения показывает, что некоторый прецедент включает в себя сценарий другого прецедента, причем составные части обязательно входят в состав общего прецедента. Отношение расширения – выборочное отношение включения; составные элементы необязательно входят в общий вариант использования (тем самым расширяя его функционал). Кроме

того, имеется связь обобщения (generalization relationship), которая означает, что действующий субъект (вариант использования) может быть обобщен до другого субъекта (варианта использования). Частный случай имеет тот же функционал, что и общий. Данная связь помогает избежать дублирования вариантов использования и многочисленного пересечения линий отношений.

Ход выполнения работы

Создание use case диаграммы с наименованием «Покупка товара со склада». Можно использовать любые удобные для студента CASE-средства, например, draw.io³.

Анализ активных субъектов, взаимодействие которых с проектируемой системой необходимо отразить на диаграмме.

Создание действующих субъектов. Например, Менеджер, Бухгалтер, Кладовщик и т. п.

Создание прецедентов. Например, Оформление заказа, Оформление счёта, Оформление накладной, Выдача товара и т. д.

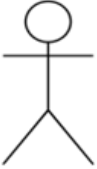

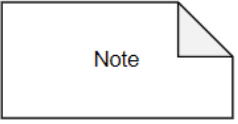

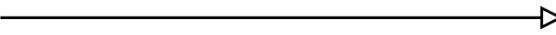
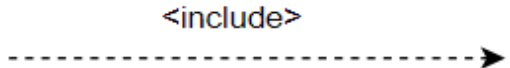
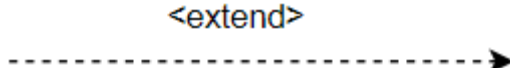
Создание комментариев, призванных пояснять созданные взаимодействия.

Расстановка связей, обозначающих зависимости. В процессе этой работы автор диаграммы должен продумать отношения и зависимости между нарисованными прецедентами.

Ниже, в табл. 3, представлены основные элементы и связи use case диаграмм.

³ <https://app.diagrams.net/>

Таблица 3 – Основные элементы и связи диаграмм прецедентов

Элемент	Описание
 Actor	Actor – участник, действующий субъект
 Use Case	Прецедент (вариант использования) – события, которые могут происходить в системе
 Note	Обозначение комментария к прецеденту
	Ассоциация между субъектом и прецедентом
	Отношение обобщения
 <include>	Включающее отношение
 <extend>	Расширяющее отношение

В draw.io основные элементы для построения диаграммы прецедентов можно найти после открытия проекта слева в разделах «General» и «UML».

В результате должна быть получена диаграмма, похожая на представленную на рис. 2:

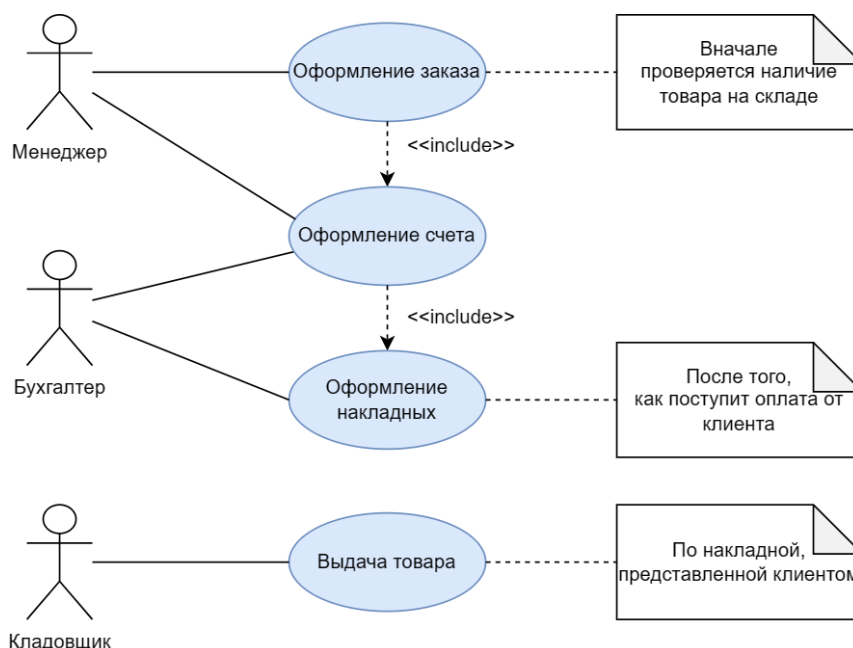


Рисунок 2 – Пример use case диаграммы

Кроме диаграммы должно быть создано подробное текстовое описание процессов, описанных на диаграмме. Необходимые термины должны быть вынесены в общий глоссарий.

Также на рис. 3 представлен пример диаграммы прецедентов системы автоматизации покупки билетов в кинотеатре, построенной в draw.io:

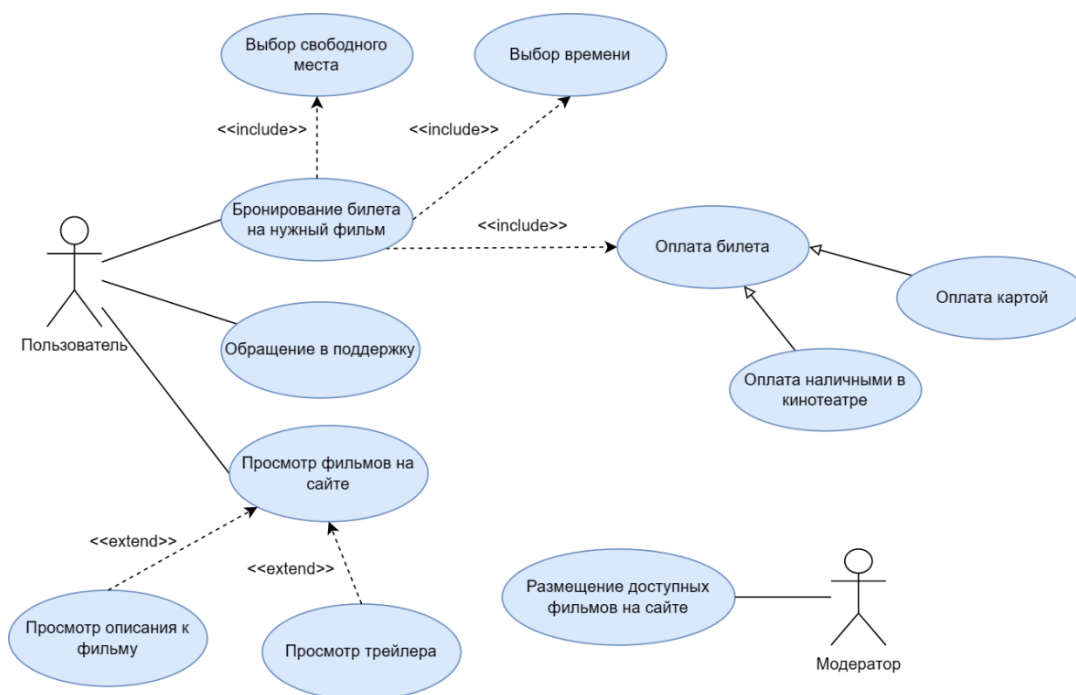


Рисунок 3 – Диаграмма прецедентов системы покупки билетов в кинотеатре

Действующие субъекты: пользователь, модератор.

Прецеденты: просмотр фильмов на сайте, обращение в поддержку, бронирование билета на нужный фильм и т. д.

Существование расширяющих, включающих и обобщающих отношений между прецедентами, а также ассоциаций между субъектом и прецедентом.

Например, прецедент «Просмотр фильмов на сайте» расширяют «Просмотр описания к фильму» и «Просмотр трейлера», а «Бронирование билета на нужный фильм» включает в себя «Выбор свободного места». Вариант «Оплата билета» состоит из оплаты наличными или картой при помощи обобщающих связей.

Контрольные вопросы к практической работе № 2

1. Для чего создается диаграмма прецедентов?
2. Что позволяет определить использование диаграммы прецедентов в процессе проектирования информационной системы?
3. Назовите главный элемент диаграммы прецедентов.
4. Дайте определение, что такое «прецедент» или «вариант использования» в рамках диаграммы прецедентов UML.
5. На какой вопрос должна отвечать созданная диаграмма прецедентов?
6. Для чего необходимы связи на диаграмме прецедентов UML, что они показывают?
7. Назовите два типа связей прецедентов UML.

Практическая работа № 3. Выбор (эскизное проектирование) архитектуры системы

Цель работы

Цель работы – уточнить требования, созданные в предыдущих практических работах, на основании макета информационной системы.

Ход выполнения работы

В задачи работы входит:

1. Создать макет информационной системы по результатам практической работы № 1.
2. Уточнить требования, разработанные в практической работе № 1. Для чего с созданным макетом провести совещание с представителем заказчика (преподавателем, ведущим практические занятия).
3. По результатам выполнения пунктов 1 и 2 создать эскизный проект информационной системы в соответствии со стандартами ГОСТ 34 серии.

Цель эскизного проектирования – создать предварительные варианты макетов проектируемой информационной системы. Макеты нужны для сравнения (в т. ч. и экономических параметров) и выбора по результатам сравнения наилучшего варианта.

Безусловно, одного текста пояснительной записки технического задания (ТЗ) недостаточно для того, чтобы концепцию создаваемой системы понял заказчик. Для этого ТЗ дополняется макетом. Макет информационной системы строится для того, чтобы можно было уточнить требования заказчика, чтобы заказчик понимал содержание ТЗ.

Примерная схема процесса создания ТЗ с использованием макетирования.

Макетирование – удобный, эффективный инструмент для построения моделей информационных систем, включая БД, что входит в понятие «проектирование ИС». Четкое разделение функциональной модели и модели потоков данных создаваемой информационной системы по сути является воплощением информационно-логической методологии проектирования ИС. Такой подход оп-

тимален для создания информационных систем, в том числе и с веб-интерфейсом. Для создания макетов информационных систем можно использовать графические схемы, составленные в различных нотациях, например, IDEF0, BPMN, Flowchart (блок-схемы по ГОСТ 19.701-90 [3]), а также их сочетания. Выбор нотации зависит как от задач, стоящих перед проектировщиком, так и от прозрачности представления результатов заказчику. Пример созданных макетов представлен на рис. 4 и 5. На рис. 4 представлено описание работы бортовой ИС в нотации IDEF0 [15], разработанной в рамках научной работы на кафедре ИиППО РТУ МИРЭА. Рис. 5 представляет макеты, находящиеся в свободном доступе и не использующие какую-либо специфическую нотацию, однако описывающие ИС.

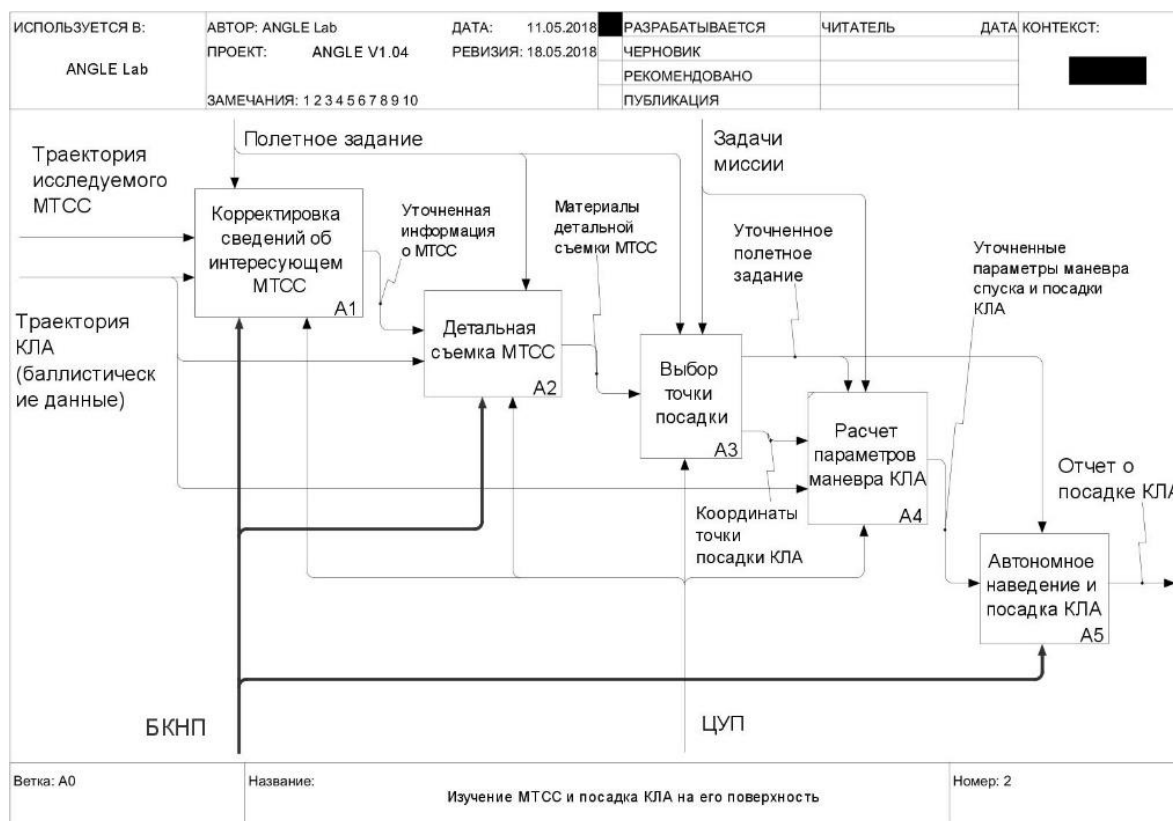


Рисунок 4 – Использование нотации IDEF0 для функционального моделирования информационной системы

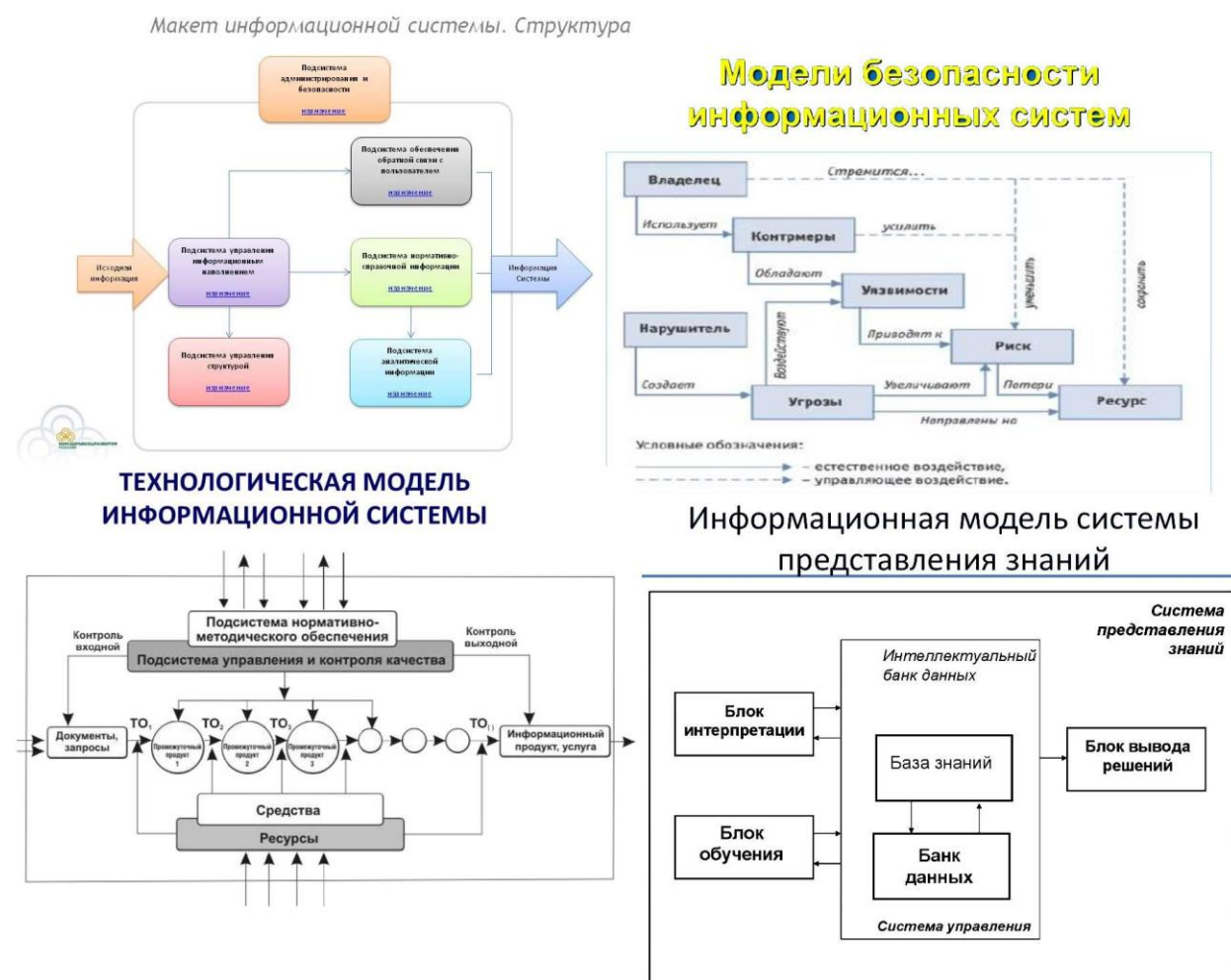


Рисунок 5 – Примеры создания макетов ИС в различных нотациях

Создание сложных систем⁴ возможно проводить в следующем порядке: анализ требований к информационной системе, макетирование, создание детальных спецификаций.

Макетирование (прототипирование) – процесс создания модели требуемого продукта. В процессе макетирования студенты должны стремиться к тому, чтобы в проекте не осталось неопределенностей. В первую очередь надо снять возможные неоднозначности в требованиях заказчика.

В процессе макетирования могут быть созданы следующие формы:

– графический макет (схема человеко-машинного диалога, созданная с использованием CASE-средств, графических выразительных средств и т. д.);

⁴ Классификация информационных систем <https://pandia.ru/text/78/380/1029-3.php>

- динамический макет – модель создаваемой информационной системы, с помощью которой возможно исполнение требуемых функций;
- существующая программа – программное обеспечение, названные характеристики которого должны быть оптимизированы.

Макетирование по сути является итерационным процессом. Структура этого процесса показана на рис. 6. Многократное прорабатывание требований с заказчиком, включая обсуждения графических и динамических макетов, позволяет убедиться во взаимопонимании между заказчиком и проектировщиком, найти все требования, а также повысить качество этих требований.

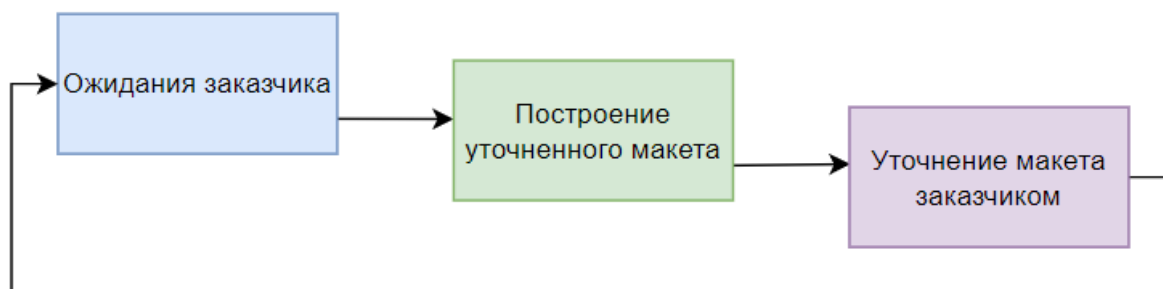


Рисунок 6 – Последовательность итеративного макетирования создаваемой информационной системы

Блок «Построение уточненного макета», как правило, выполняется по алгоритму. Алгоритм создания макета или уточнения, если это второй и последующие итерации, представлен на рис. 7. Процесс создания или уточнения макета ИС начинается со сбора требований заказчика к информационной системе. Эта работа была выполнена в практической работе № 1 данного учебного курса. При необходимости можно обратиться к конспекту лекций и специальной литературе, ссылки на которую приводились ранее в тексте данного учебного пособия. Проектировщик (студент) совместно с представителем заказчика (преподаватель) уточняют и обсуждают требования, полученные на предыдущем этапе. Затем проектировщик создает макет информационной системы. Макет оценивается заказчиком в рамках практических занятий. Заказчик (преподаватель) критически оценивает созданный макет и дает ценные указания проек-

тировщику, уточняет те или иные функции и особенности предложенного макета. В это же время уточняются требования (см. практическую работу № 1) к проектируемой информационной системе. Уточнение макета продолжается до тех пор, пока не будут установлены и технически верно интерпретированы все требования заказчика. Одновременно с этим требования должны быть понятны разработчику. Разработчик и проектировщик должны однозначно интерпретировать макет.

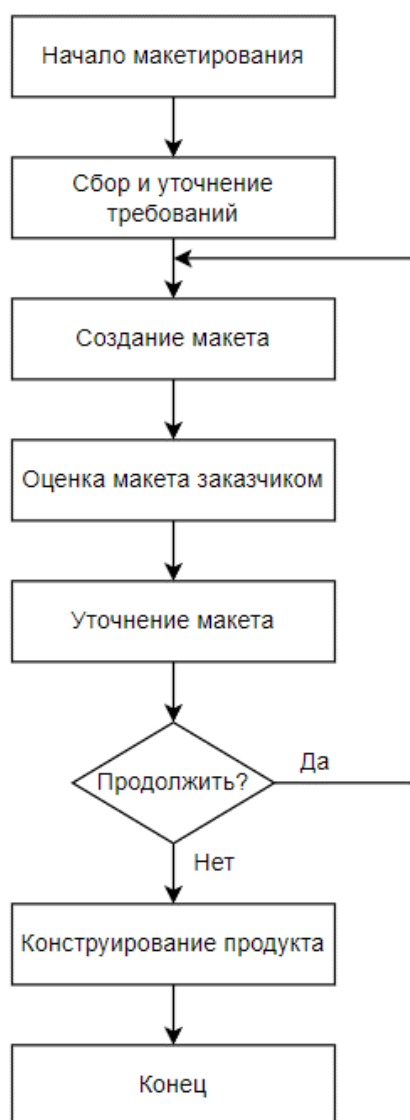


Рисунок 7 – Последовательность действий при макетировании программного обеспечения

К достоинствам итеративного макетирования можно отнести то, что с помощью последовательно уточняемых макетов можно определить полные достоверные требования к ИС. К сожалению, недостатки также имеются. Во-

первых, в процессе уточнения и заказчик и разработчик могут начать воспринимать макет как конечный продукт. Возможно, описанный недостаток не очевиден для обучающихся. Поясним, что именно подразумевается под этим.

При виде модели проектируемой ИС, у заказчика может создаться опасная иллюзия, что уже решены все вопросы, относящиеся, например, к качеству и сопровождаемости информационной системы. Это заблуждение, в особо опасном случае, может привести к тому, что заказчик будет требовать от разработчика незамедлительной трансформации макета в готовый продукт. Процесс проектирования и разработки информационной системы может непоправимо пострадать от таких опрометчивых действий заказчика. Дело в том, что в процессе создания макета проектировщик часто идет на некоторые компромиссы и упрощения. Иногда это делается сознательно. Например, некоторые функции могут быть опущены или упрощены с учетом уточнения в последующих версиях макета. Иногда это происходит из-за недостатка информации, в частности, может быть предложен не соответствующий задаче язык программирования. Возможно использование не оптимальной для данного проекта базы данных. Часто в рамках сжатых сроков для демонстрации заказчику выбирают не оптимальный, а хорошо известный или отработанный алгоритм, который, однако, не совсем подходит для реализации проекта. Возможны и другие подобные неоптимальные решения. В процессе разработки проектировщик может забыть о том, что вышеперечисленные допущения необходимо проанализировать и в случае необходимости заменить на более подходящие. В результате далеко не лучший выбранный вариант интегрируется в систему. Однако, это преодолимо, если проектировщик правильно организует работу с заказчиком и разработчиком.

Создание и оформление пояснительной записки эскизного проекта

Макет, который прошел несколько итераций и удовлетворяет требованиям заказчика, а кроме того, является прозрачным для разработчика, преобразуется в эскизный проект. Эскизный проект оформляется как самостоятельный документ «Пояснительная записка к эскизному проекту». Пояснительная записка оформляется в соответствии с ГОСТ 19.105-78 и ГОСТ 19.404-79.

В рамках эскизного проектирования согласно ГОСТ 34.601-90 выполняются два основных этапа: 1) разработка предварительных проектных решений по системе и ее частям; 2) разработка документации на информационную систему и ее части. Более подробно создание эскизного проекта раскрыто в ГОСТ 34.201-2020. По результатам этих этапов должны быть оформлены следующие основные документы:

1) ведомость, в которой систематизировано перечисляются объекты, предметы и т. д., входящие в информационную систему (например, модули, подмодули, аппаратное обеспечение и т. д.);

2) схема, графически изображающая формы документов, частей, элементов системы и связей между ними в виде условных обозначений;

3) инструкция – в инструкции последовательно излагается состав и правила выполнения действий при создании информационной системы;

4) обоснование, в котором изложены сведения, подтверждающие целесообразность принимаемых решений;

5) описание – документы, в которых поясняется назначение проектируемой информационной системы в целом, назначения ее отдельных частей, описываются принципы действия этих элементов, их взаимодействие и в каких условиях они должны функционировать;

6) конструкторская документация, которая оформляется по ГОСТ 2.102;

7) программный продукт, оформленный по ГОСТ 19.101⁵.

После утверждения всеми полномочными лицами, пояснительная записка становится программным документом, дублировать, вести учет и хранить который необходимо в соответствии с ГОСТ 19.601-78 «Общие правила дублирования, обращения, учета и хранения» и ГОСТ 19.602-78 «Правила дублирования, учета и хранения программных документов, выполненных печатным способом». При выполнении последующих этапов может быть обнаружено, что необходимо внести изменения в эскизный проект (инкрементная модели или

⁵ Более подробно можно ознакомиться с процессом в ГОСТ 34.201-2020 «Виды, комплектность и обозначение документов при создании автоматизированных систем»

иначе – поэтапная модель жизненного цикла с обратной связью). Тогда такие изменения необходимо отразить в пояснительной записке в соответствии с ГОСТ 19.603-78 «Общие правила внесения изменений в программные документы» и ГОСТ 19.602-78 «Правила внесения изменений в программные документы, выполненные печатным способом».

Далее, как пример, представлена часть расширенного перечня работ стадии эскизного проекта [15]:

- разработка плана совместных работ на разработку ПО;
- разработка и обоснование математической модели системы на ЭВМ и описание результатов моделирования;
- разработка и обоснование алгоритмов и временных графиков функционирования программного обеспечения (ПО) по всем режимам работы;
- разработка и обоснование ресурсов памяти для реализации алгоритмов;
- разработка перечня документов на ПО;
- разработка и обоснование структуры базы данных (БД), внешних входных и выходных данных;
- разработка и обоснование алгоритмов информационного обеспечения;
- определение взаимосвязей между видами программ;
- разработка и обоснование набора тестов для проверки ПО;
- разработка и обоснование организации наращивания и развития ПО;
- оформление пояснительной записки и ведомости эскизного проекта ПО (в соответствии с ГОСТ 19.105-78, ГОСТ 19.404-79 и ГОСТ 2.106-68 «ЕСКД. Текстовые документы»).

ГОСТ Р 59795-2021 Пояснительная записка к эскизному проекту на создание автоматизированной системы (пример эскизного проекта)

«Пояснительная записка к эскизному проекту на создание автоматизированной системы», как проектный документ, создается по правилам, описанным в ГОСТ Р 59795-2021 «Автоматизированные системы. Требования к содержанию документов».

«Пояснительная записка» (далее ПЗ) создается командой разработчиков на этапе эскизного проектирования информационной системы.

В каждом разделе пояснительной записки эскизного проекта кратко описываются требования к содержанию. Пояснительная записка оформляется в соответствии с ГОСТ. Далее приведем типовые примеры разделов пояснительной записки:

1. Общие положения.

2. Основные технические решения:

– Решения по структуре системы, подсистем, средствам и способам связи для информационного обмена между компонентами системы.

– Решения по взаимосвязям АС со смежными системами, обеспечению ее совместимости.

– Решения по режимам функционирования, диагностированию работы системы.

– Решения по персоналу и режимам его работы.

– Сведения об обеспечении заданных в техническом задании потребительских характеристик системы, определяющих ее качество.

– Состав функций, комплексов задач, реализуемых системой.

– Состав и размещение комплексов технических средств.

– Решения по составу информации, объему, способам ее организации, видам машинных носителей, входным и выходным документам и сообщениям, последовательности обработки информации и другим компонентам.

– Методы и средства разработки.

3. Мероприятия по подготовке объекта автоматизации к вводу системы в действие.

Функциональная структура информационной системы

В рамках этого раздела формируется схема функциональной структуры проектируемой информационной системы. Большой прямоугольник изображает информационную систему вообще. Внутри этого прямоугольника размещаются

подсистемы, которые обозначаются прямоугольниками меньших размеров. Внутри каждого прямоугольника-подсистемы, в виде списка, указываются функции, выполняемые данной подсистемой. Список подсистем и функций, выполняемых ими, содержится в разделе технического задания, который называется «Требования к функциям, выполняемым системой».

Затем, на диаграмме функциональной структуры отрисовывают связи как между подсистемами внутри системы, так и с внешними информационными системами. Важно также отразить связи с пользователями. Связи изображают так, как это описано в пункте технического задания, который называется: «Требования к информационному обмену между компонентами системы». Схематически изображают администратора рядом с обслуживаемой им подсистемой (см. рис. 8).

Пример выполнения практической работы №3 приведен в приложении В.

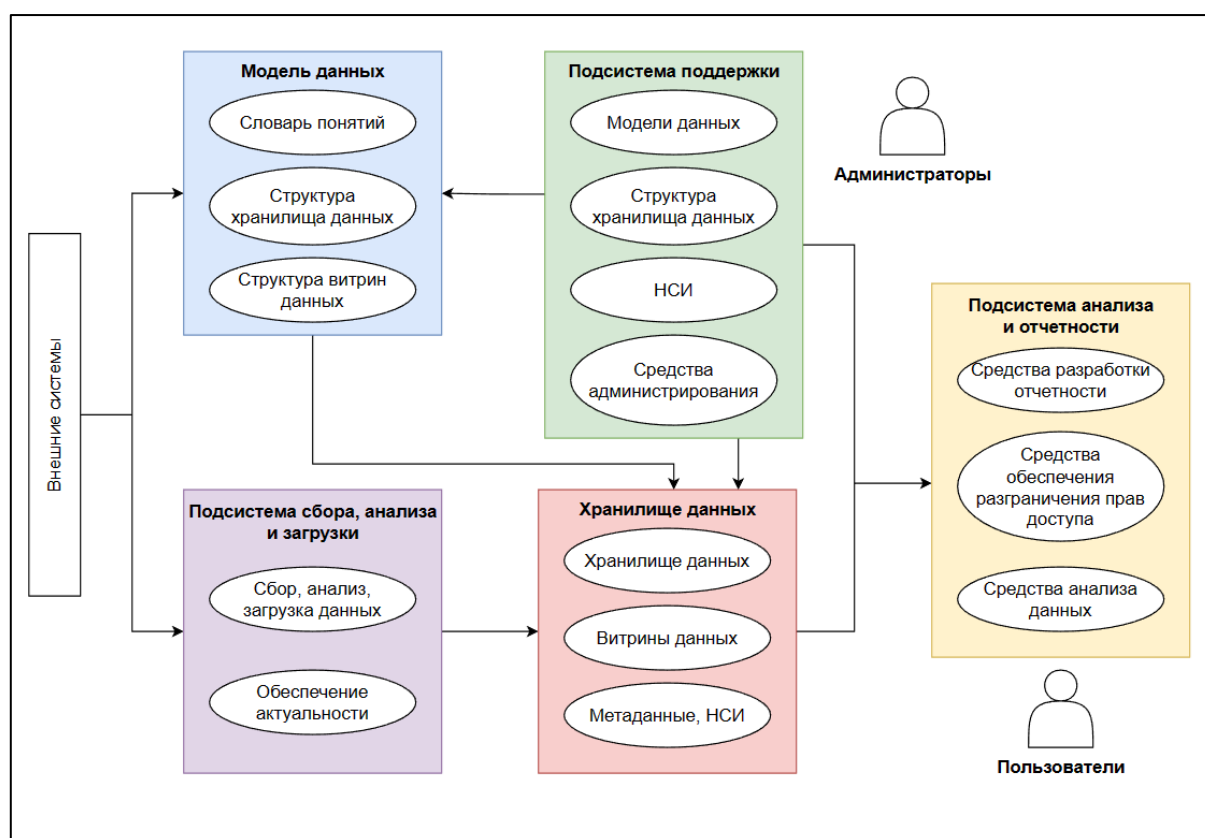


Рисунок 8 – Макет модели информационного обмена между компонентами системы

Контрольные вопросы к практической работе № 3

1. В чем состоит цель эскизного проектирования информационной системы?
2. Для чего создаётся макет информационной системы?
3. Дайте определение термину «Макетирование информационной системы».
4. Назовите основные особенности, на которые необходимо обратить внимание при создании макета информационной системы.
5. Какие нотации можно использовать для создания макета информационной системы?
6. Какие формы создаются в процессе макетирования информационной системы?
7. Приведите структуру и дайте описание процесса макетирования информационной системы.
8. Приведите последовательность действий при макетировании программного обеспечения.
9. Приведите достоинства и недостатки итеративного макетирования.
10. Опишите процесс создания и оформления пояснительной записки эскизного проекта информационной системы.
11. Какие нормативные документы регламентируют процесс создания и оформления эскизного проекта?
12. Какие действия должны быть выполнены, чтобы пояснительная записка эскизного проекта информационной системы стала программным документом?
13. Какие нотации можно использовать для описания в графическом виде функциональной модели информационной системы?

Практическая работа № 4. Функциональное проектирование модели информационной системы с использованием методологии SADT

Цель работы

Целью данной практической работы является выбор и проектирование функциональной модели информационной системы в нотации IDEF0, составление краткого описания ИС, включая цель, способ и средства её создания. В рамках данной работы выполняется моделирование диаграммы контекстного уровня А-0. Работа должна быть выполнена с соблюдением требований руководящего документа РД IDEF0 – 2000 Методология функционального моделирования IDEF0 [15].

Теоретические сведения

SADT (акроним от англ. STRUCTURED ANALYSIS AND DESIGN TECHNIQUE) – методология структурного анализа и проектирования, интегрирующая процесс моделирования, управление конфигурацией проекта, использование дополнительных языковых средств и руководство проектом со своим графическим языком. В качестве такого графического языка используют нотацию IDEF0 для создания функциональной модели информационной системы.

Функциональная модель информационной системы – абстрактная графическая модель, которая отражает функции, выполняемые информационной системой, их связь. Функциональная модель также должна отражать направление движения материальных потоков и потоков информации, преобразуемых системой. Функциональная модель в нотации IDEF0 строится иерархически. На самом верхнем уровне располагается диаграмма контекста или контекстная диаграмма А-0 (читается: «А минус ноль»). Контекстная диаграмма содержит только один функциональный блок. Важно это помнить, только один блок! Хорошие CASE-средства просто не позволят вам вставить в контекстную диаграмму два или больше блоков. Самое важное на диаграмме А-0 – это название функционального блока. Название должно отражать цель функционирования

всей системы, например, «Автоматизировать начисление заработной платы» или «Автоматизация бухгалтерского учета предприятия». Обратите внимание, что второй вариант отступает от требований РД IDEF0 – 2000, однако, главное – ясность изложения цели для всех читающих диаграмму. Варианты «Обработка данных» или «Система поиска информации» – неприемлемы, потому что из названия блока непонятно, как выполняется обработка данных и каких именно данных и т. д.

ВАЖНО: схему только тогда можно считать диаграммой IDEF0, когда присутствует рамка и штампы (верхний и нижний, см. рис. 4). Без рамки и штампов диаграмма не более, чем рисунок. Работы, в которых диаграммы оформлены без рамки и штампов, преподавателем не проверяются и подлежат исправлению.

Основная сложность создания контекстной диаграммы состоит в определении оптимального числа стрелок по входу, выходу, управлению и механизмам. Стрелок не должно быть много (одна из рекомендаций РД IDEF0 – 2000 – всего одна стрелка!!! Эту рекомендацию, конечно же, следует считать сильно идеализированной). Увлечаться сокращением числа стрелок также не стоит. Их должно быть столько, чтобы другие участники проекта информационной системы могли четко понять и разграничить потоки, связанные с главным функциональным блоком. Традиционно рекомендуют использовать не более 6 стрелок. Тем не менее, сколь бы не была совершенна созданная контекстная диаграмма, ей требуется текстовое пояснение, а термины, использованные в процессе создания и описания контекстной диаграммы, должны быть отражены в ГЛОССАРИИ.

Задание: создание краткого описания объекта автоматизации, постановка задачи процесса автоматизации, описание основных параметров проектируемой информационной системы, описание путей достижения цели. Начать проектирование функциональной модели выбранной информационной системы и создать контекстную диаграмму в нотации IDEF0, добавить текстовое описание разработанной модели.⁶

⁶ По согласованию с преподавателем, ведущим практические занятия, можно использовать другие нотации для создания, допускается создавать с использованием других нотаций, например, BPMN или UML. Однако следует учитывать, что при этом объем работ увеличивается.

Ход выполнения работы

Каждый студент должен выбрать объект автоматизации в качестве базы для реализации программного комплекса (это должно быть сделано в начале семестра, см. практическую работу № 1). Варианты объектов для автоматизации даны в приложении А. Кроме того, по согласованию с руководителем студент вправе предложить собственный тип объекта автоматизации. В случае проектирования сложных аппаратно-программных комплексов, которые невозможно целиком реализовать за время изучения дисциплины, допускается бригадная работа студентов, при условии, что каждый студент будет выполнять проектирование собственного модуля ИС.

Работу следует оформлять в соответствии с ГОСТ 7.32-2017 «Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления» (введен в действие Приказом Росстандарта от 24.10.2017 N 1494-ст). Для проектирования контекстной диаграммы функциональной схемы ИС в нотации IDEF0 необходимо использовать РД IDEF0 – 2000. Диаграммы должны иметь рамку и штамп, как указано в нормативе [15].

Пример выполнения работы будет производиться в программе Ramus Educational. Программа относится к CASE-средствам и является удобным инструментом для проектирования информационных систем. В программе реализованы такие методологии, как IDEF0 и DFD. Ramus обладает интуитивно понятным пользовательским интерфейсом, что позволяет быстро обучиться построению моделей.

Каждая работа в Ramus начинается с создания проекта. В первом диалоговом окне программа предложит создать новый файл или открыть существующий. При создании нового проекта открывается следующее диалоговое окно, называемое *Мастер «Свойства проекта»*. На 1 этапе свойств следует ввести запрашиваемые данные (автор, название проекта, название модели), а также выбрать нотацию для последующего моделирования модели. Стоит отметить,

что в созданной модели **НОТАЦИЮ ИЗМЕНИТЬ НЕВОЗМОЖНО!** Ниже на рис. 9 представлен пример заполнения:

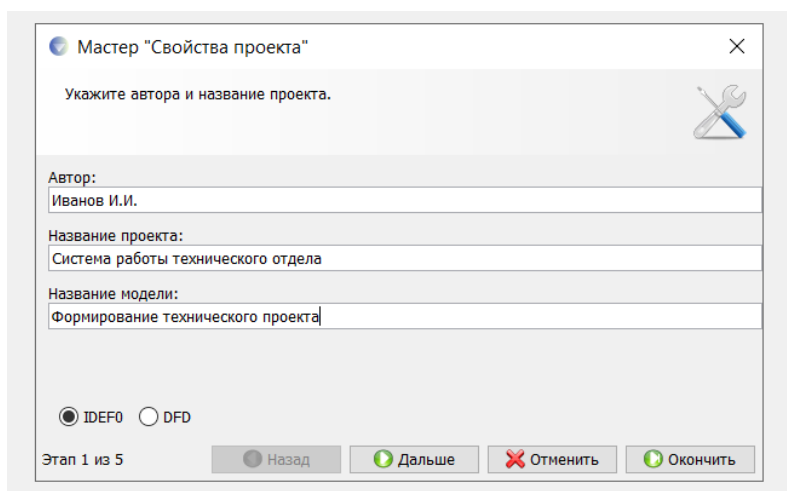


Рисунок 9 – Заполнение необходимых данных проекта

В рамках учебного курса заполненных данных будет достаточно, однако стоит упомянуть и другие этапы свойств проекта:

- 2 этап (*Используется в*) – указание предприятия или организации, где используется проект.

- 3 этап (*Описание*) – дается краткое описание проекта.

- 4 этап (*Классификаторы*) – указание классификаторов, используемых в модели. С помощью классификаторов создается единая система упорядочения объектов системы (документы, хранилища данных, персонал и т. д.). Создание классификаторов будет показано в практической работе №6.

- 5 этап (*Собственники классификаторов*) – указание классификатора, элементы которого попадают в список собственников процесса.

Перемещение между этапами производится с помощью кнопок «Назад» и «Дальше». Приветствуется заполнение 2 и 3 этапов проекта. Чтобы завершить ввод различных данных проекта и приступить непосредственно к моделированию, следует нажать кнопку «Окончить». На рис. 10 представлено рабочее пространство программы:

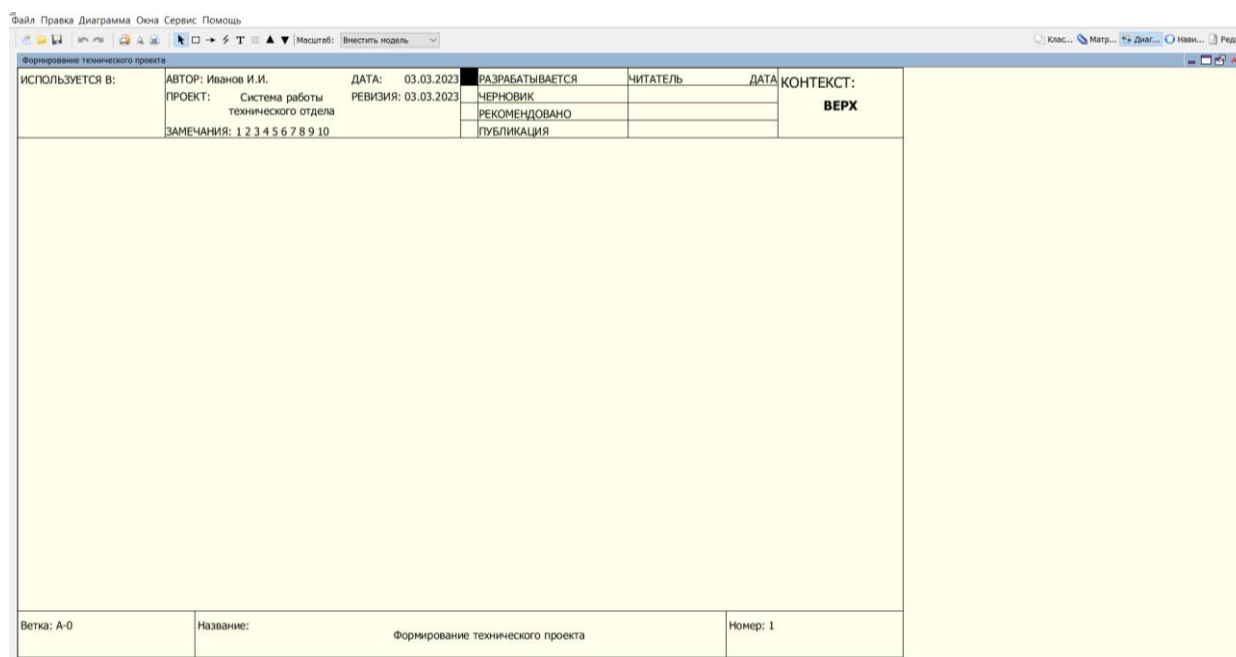


Рисунок 10 – Рабочее пространство программы

Строка меню обеспечивает основной функционал Ramus:

- Файл – функции, связанные с целым проектом (создание нового проекта, открытие существующего, сохранение, предварительный просмотр, печать).
- Правка – отмена и повтор действий в проекте.
- Диаграмма – функции, связанные с созданием и редактированием диаграммы, а также просмотр свойств модели (диаграммы) и экспорт диаграммы.
- Окна – формирование рабочего пространства под личные требования.
- Сервис – вспомогательные функции, такие как открытие проекта в браузере, просмотр свойств проекта и программы.
- Помощь – справка и информация о программе.

Теперь необходимо изучить основную панель приложения (см. рис. 11).

Панель повторяет функционал из вкладок «Файл», «Правка» и «Диаграмма» для быстрого и удобного доступа.



Рисунок 11 – Интерфейс программы Ramus Educational

Описание интерфейса приложения:

1. Режим курсора – базовые режим. С помощью данного режима можно перемещать и масштабировать функциональные блоки и потоки данных.

2. Добавление функциональных блоков.

3. Добавление стрелок (потоков данных).

4. Добавление тильд для упрощения понимания сложных диаграмм.

Тильда прикрепляет к стрелке ее название.

5. Добавление дополнительного текста к стрелкам.

6. Переход к родительской диаграмме (уровню выше).

7. Переход к дочерней диаграмме (уровню ниже).

Пункты 6-7 будут полезны в следующей практической работе.

Перед работой в программе всегда можно настроить ее свойства. Необходимо перейти в «Сервис» --> «Свойства программы» --> «Диаграмма» (см. рис. 12). В целях личного удобства приветствуется изменение свойств.

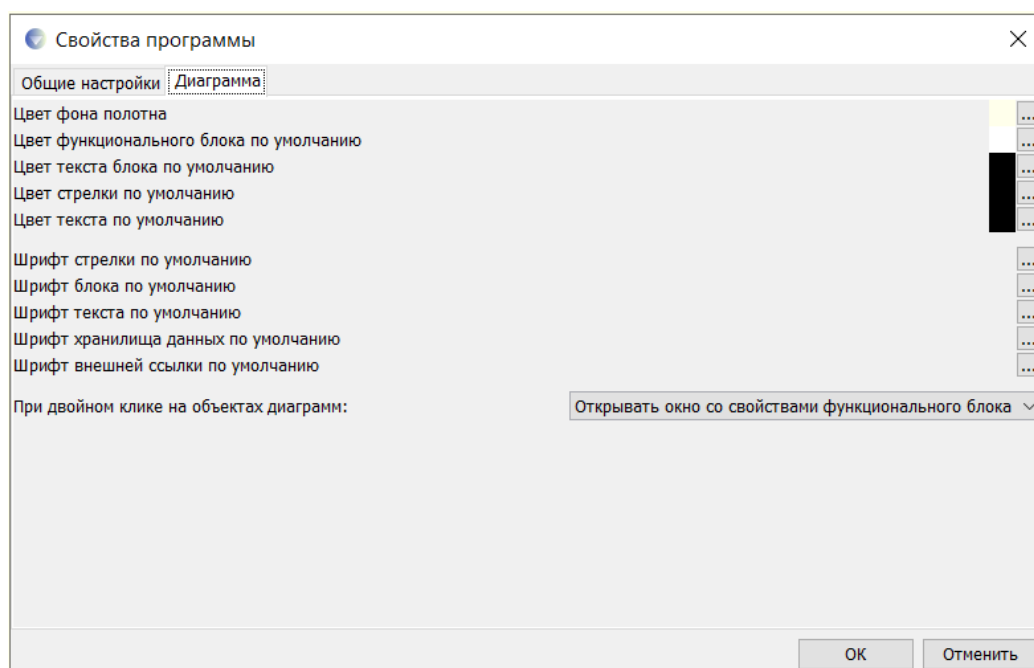


Рисунок 12 – Окно настройки свойств программы

Так как в работе проектируется контекстная диаграмма, на модели будет расположен только один функциональный блок. Осталось определить его название, а также стрелки входов, выходов, механизмов и управления.

В качестве учебного примера рассмотрим систему автоматизации доставки пиццы из ресторана. Название главного функционального блока будет звучать как «Автоматизация доставки пиццы из ресторана».

Стрелки входов будут определять, какие потоки информации необходимы, чтобы начать выполнение функции. В данном случае можно добавить такие стрелки, как «Данные пользователя», «Список товаров».

Стрелки выходов определяют, что получается после отработки функционального блока. Стрелки могут идти как и в следующий процесс, так и являться граничными (как в данной работе). Здесь это «Документы на заказ», «Доставленная пицца», «Статистика продаж».

Стрелки управления определяют условия, необходимые для достижения правильного выхода. Это могут быть различные правила, регламенты, законы и т. д. Например, это «Рецепт приготовления пиццы», «Законодательство», «Политика организации» и т. п.

Стрелки механизмов определяют, с помощью кого или чего будет проходить весь процесс. Это может быть «Транспорт для доставки пиццы», «Сотрудники доставки пиццерии», «Платежная система» и т. п.

После создания проекта в Ramus добавим главный функциональный блок на диаграмму (кнопка 2 интерфейса), а затем присоединим стрелки в зависимости от их назначения (кнопка 3 интерфейса). Стрелки присоединяются по следующему принципу (см. рис. 13):



Рисунок 13 – Построение диаграммы в нотации IDEF0

При нажатии на блок или стрелку появляются их свойства. Можно определить название, а также цвет, шрифт, тип блока и т. д. (см. рис. 14).

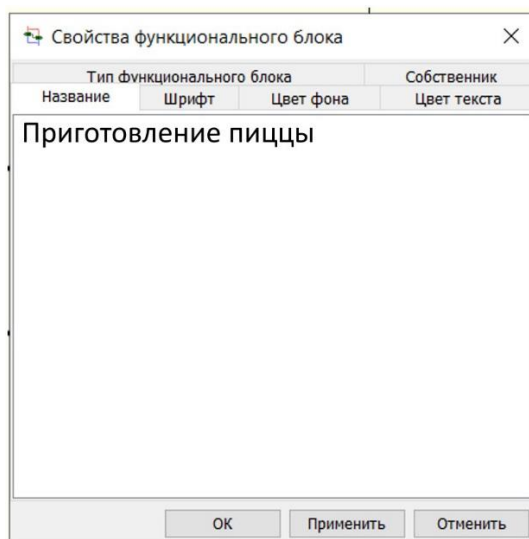


Рисунок 14 – Окно свойств функционального блока

Нередким случаем бывает подчеркивание всего текста красным, хотя все написано в соответствии с правилами русского языка. Для решения проблемы необходимо перейти в «Сервис» --> «Свойства проекта» --> «Правописание» и изменить язык на русский. Установка таких настроек позволит отслеживать ошибки в написании названий функциональных блоков и стрелок.

Для установки стрелок к функциональному блоку необходимо выбрать соответствующий режим и нажать сначала на край рабочего пространства, а затем на нужную сторону блока до появления черного треугольника. Если это стрелка выхода, необходимо нажать на правую сторону функционального блока, затем на правый край рабочего пространства.

Удобной функцией программы является центрирование стрелок относительно блока. Она позволяет снизить визуальную нагрузку на диаграмму. Для выполнения действия нажмите правой кнопкой мыши на функциональный блок, а затем выберите «Центровать присоединённые стрелки». Однако этой функцией надо пользоваться с осторожностью. Самое главное – наглядность и читаемость диаграммы. Кроме того, всплывающее меню позволяет перейти к редактированию активного элемента, выбрать тип функционального блока или вовсе удалить его. Меню при нажатии правой кнопкой на стрелку позволяет редактировать или удалить ее и добавить тильду.

Также необходимо следить за тем, чтобы названия стрелок не заходили на сами стрелки. Тильды к стрелкам не являются обязательной частью диаграммы, однако всегда приветствуются для дополнительного понимания процесса. Чтобы добавить тильду на диаграмму, необходимо выбрать режим тильд (кнопка 4 интерфейса) и нажать на стрелку, к которой нужно добавить тильду (название автоматически свяжется со стрелкой). Теперь можно передвигать название в свободное пространство на диаграмме.

Спроектированные диаграммы в Ramus Educational необходимо добавлять в отчеты по практическим работам в формате экспортированных рисунков в высоком разрешении, а не скриншотов. Для этого следует перейти во вкладку «Диаграмма» и выбрать пункт «Экспортировать как рисунки». Далее, выбрав необходимый(-ые) уровень(-ни) диаграммы для экспорта, высокое разрешение, тип рисунков .jpg или .png и папку для сохранения всех диаграмм проекта, выполнить сохранение, нажав на кнопку «ОК». Пример формата выгруженной диаграммы представлен на рис. 15.

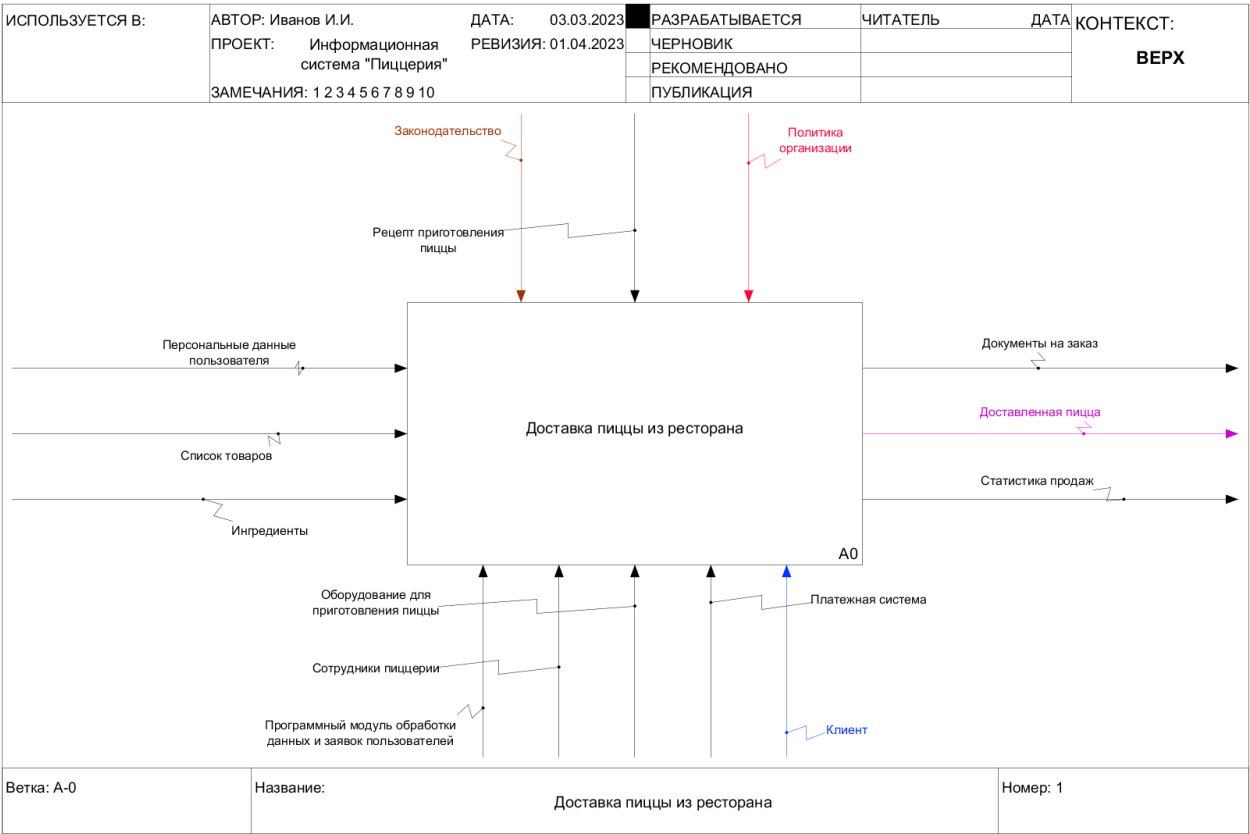


Рисунок 15 – Пример создания контекстной диаграммы в нотации IDEF0

Пример спроектированной контекстной диаграммы с пояснительной запиской представлен в приложении Г.

Практическая работа № 5. Проектирование функциональной модели информационной системы в нотации IDEF0

Задание: декомпозировать функциональную модель проектируемой системы в нотации IDEF0, создать декомпозиции как минимум двух уровней. Допустимо декомпозировать один, наиболее значимый для проектируемой системы, функциональный блок на каждом уровне. Добавить описание функциональных блоков и потоков данных, а также выводы к работе.

Цель работы

Целью данной практической работы является продолжение проектирования функциональной модели выбранной информационной системы, моделирование как минимум двух уровней декомпозиции в нотации IDEF0 и составление текстового описания проектируемых модулей на уровнях декомпозиции. Работу следует оформлять в соответствии с ГОСТ 7.32-2017 «Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления» (введен в действие Приказом Росстандарта от 24.10.2017 N 1494-ст). Для проектирования уровней декомпозиции диаграммы функциональной схемы ИС в нотации IDEF0 необходимо использовать РД IDEF0 – 2000. Методология функционального моделирования IDEF0 Руководящий документ [15].

Ход работы

Используя контекстную диаграмму проекта информационной системы, полученную по итогам выполнения предыдущей практической работы, выполнить её декомпозицию (уровень A0) и выбрать и декомпозировать как минимум один блок. Декомпозируемый блок необходимо согласовать с преподавателем.

Чтобы показать, как декомпозировать диаграмму в программе Ramus Educational, возьмем прошлую работу.

Полная функциональная модель имеет иерархическую структуру, где каждый блок имеет одну родительскую и несколько дочерних диаграмм (кроме контекстной диаграммы). Иерархия производится с помощью номера, состоящий из

префикса и числа. Нумерация на определенном уровне осуществляется слева направо по расположению блоков. Контекстная диаграмма имеет номер А-0, а ее декомпозиция А0; функциональные блоки на уровне А0 будут иметь нумерацию А1, А2, А3 и т. д. Если нужно декомпонировать определенный блок, то дочерние элементы будут нумероваться вместе с родительским номером (например, декомпозиция блока А2 приведет к нумерации А21, А22, А23 и т. д.).

Перейдем непосредственно к самой декомпозиции контекстной диаграммы. Декомпозиция предназначена для детализации процессов, протекающих в системе. Для того, чтобы добавить новый уровень декомпозиции, необходимо выбрать декомпозируемый функциональный блок, и затем нажать на кнопку перехода к дочерним диаграммам (см. рис. 10 предыдущей практической работы). Если дочерняя диаграмма не была уже создана, появится окно создания новой. Ниже на рис. 16 изображено данное окно:

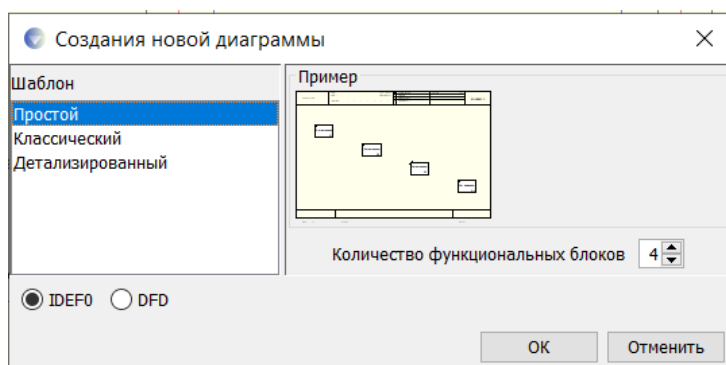


Рисунок 16 – Окно создания диаграммы декомпозиции

Шаблон лучше выбирать простой, так как при выборе классического или детализированного шаблона есть риск запутаться в стрелках, а также получить в работе туннелированные стрелки. Туннелированные стрелки изображаются первоначально в квадратных скобках. Квадратные скобки в стрелка показывают проектировщику, что на предыдущем или последующем уровне произошло удаление этой стрелки. Если автор намеренно отправил стрелку в туннель, необходимо выбрать во всплывающем контекстном меню соответствующий пункт и скобки из квадратных превратятся в круглые, что правильно. В против-

ном случае необходимо разобраться в причине появления квадратных скобок и устранить ее.

Важно! В практических работах использовать туннелирование можно только с разрешения преподавателя, а лучше и вовсе не использовать. Необходимости туннелирования в работах нет. Всего два уровня декомпозиции – не перегружены стрелками, а следовательно, туннелирование, наоборот приведет к ухудшению читаемости диаграмм.

Если же по договоренности с руководителем, все же принято решение о туннелировании, тогда следует учитывать следующие моменты.

Туннелированные стрелки можно использовать в следующих случаях:

- Необходимо изобразить малозначимую стрелку именно на ДАННОМ уровне декомпозиции (скобки в начале стрелки).
- Использование стрелок на нижних уровнях диаграммы происходит одинаково во всех процессах декомпозиции (скобки в конце стрелки).

Заметим, что для их обозначения нужно поменять скобки на круглые, тем самым обозначив, что в работе нет ошибок и данная стрелка нужна для лучшего понимания процессов. Для этого необходимо нажать по квадратным скобкам правой кнопкой мыши, из контекстного меню выбрать «Тоннель» --> «Обозначить тоннель круглыми скобками». Функция «Тоннель» доступна только для стрелок с квадратными скобками.

Чтобы лучше понять использование туннелированных стрелок, вернемся к контекстной диаграмме на рис. 15. В качестве стрелок управления у нас выступают «Законодательство» и «Политика организации», которые используются во всех процессах и уровнях модели. В самом начале их можно туннелировать (скобки в конце стрелки), тем самым обозначив, что они не будут отображаться на дочерних диаграммах, но будут использоваться во всех процессах. Чтобы появились квадратные скобки на необходимых потоках данных, необходимо удалить стрелки на уровне декомпозиции.

Перейдем к выбору количества блоков. Количество функциональных блоков на декомпозиции желательно обдумать заранее, но, если вдруг не хватило блоков или оказались лишние, их можно добавить с помощью специального режима добавления блоков или удалить с помощью выпадающего меню при нажатии правой кнопкой мыши на соответствующий функциональный блок соответственно.

На диаграммах декомпозиции должно быть как минимум 3 функциональных блока! Декомпозиция диаграммы с меньшим числом блоков не будет нести особого смысла. С другой стороны, диаграммы с большим количеством блоков (более 6) плохо читаются и их смысл тоже теряется. Рекомендуется использовать от трех до шести функциональных блоков на одном уровне декомпозиции.

После успешной декомпозиции контекстной диаграммы появятся созданные блоки и стрелки, указанные на родительском уровне. Расположение функциональных блоков обычно находится с верхнего левого угла по правый нижний.

Для построения диаграммы следует руководствоваться теми же правилами, что и в предыдущей работе. Однако стрелки уже присутствуют, но не принадлежат ни одному процессу. После обозначения основных процессов декомпозиции, нужно присоединить стрелки в пределах контекста. Для этого наведите курсор на начало или конец стрелки. Когда интересующая нас стрелка будет выделена (ширина стрелки увеличиться), нажмите и присоедините к соответствующему блоку (появится черный треугольник). Обязательно, все вышеописанные действия проводить в режиме курсора!

Стоит обратить внимание на внутренние стрелки между блоками. В режиме создания стрелок нажмите на сторону выхода функционального блока и подведите к стороне входа другого блока. Путем внутренних стрелок осуществляется взаимосвязь элементов диаграммы декомпозиции.

При декомпозиции контекстной диаграммы «Доставка пиццы из ресторана» были спроектированы следующие функциональные блоки:

1. Регистрация пользователя в системе (A1);
2. Оформление заказа (A2);
3. Приготовление пиццы (A3);
4. Доставка пиццы (A4).

Функциональный блок «Регистрация пользователя в системе». Первым этапом заказа пиццы из ресторана идет регистрация пользователя (потенциального клиента) на сайте. Для того, чтобы данный процесс смог начать свою работу, необходимо подать на вход персональные данные пользователя. Данный процесс на протяжении всей своей работы будет проходить на основе законодательства и политики организации. Процесс помогут произвести программный модуль обработки данных и заявок пользователей и клиент. В результате обработки процесса получатся запросы к сайту, отправляемые пользователем после регистрации.

Функциональный блок «Оформление заказа». В данном блоке будет проходить формирование корзины товаров и оформление заказа на ее основе. На вход будут подаваться запросы к системе и список доступных товаров. Данный процесс также будет проходить на основе законодательства и политики организации. Клиент, ПМ обработки данных и заявок пользователей и платежная система будут помогать успешному прохождению процесса. На выходе получатся документы на заказ и сформированная заявка клиента.

Функциональный блок «Приготовление пиццы». На данном этапе будет происходить приготовление пиццы в зависимости от заказа клиента. На вход данного процесса будут поступать заявка клиента и необходимые ингредиенты для приготовления. Процесс проходит также на основе законодательства и политики организации; добавляется рецепт приготовления пиццы. Процесс проводят с помощью сотрудников пиццерии и оборудования. По окончании формируется собранный заказ.

Функциональный блок «Доставка пиццы». В последнем процессе происходит доставка заказа рестораном до клиента. На вход поступает собранный заказ.

Процессом управляют законодательство и политика организации. Для правильного выхода процессу помогают клиент, сотрудники пиццерии; программный модуль обрабатывает статус доставки на протяжении всего процесса. В результате заказ доставляется до клиента и собирается на этом статистика продаж.

Важными действиями в проектировании диаграммы декомпозиции являются разветвляющиеся стрелки (активно используются в управлении и механизмах). Чтобы создать ветвление стрелки, необходимо выбрать режим добавления стрелок, навести на начало стрелки до появления его выделения от остальной части, нажать на стрелку и довести до нужного блока. Создать такие стрелки довольно просто, но вот сделать так, чтобы диаграмма была более читаемой – сложнее. Следует отслеживать и недопускать ненужные пересечения стрелок для снижения визуальной нагрузки на диаграмму.

Если сегменты (линии стрелки, которые начинаются или заканчиваются на стороне блока, в точке ветвления или на границе) стрелки идут в более чем 2 блока, нужно использовать следующий принцип:

1. Доведите основную стрелку до блока, в котором она впервые встречается.
2. Сделайте ветвление стрелки до последнего блока, в котором она встречается.
3. В результате получится довольно большое ветвление, от которого можно вести другие сегменты к блокам, находящимся между началом и концом. Для их реализации необходимо нажать на самом длинном сегменте, реализованном в пункте 2 (при наведении выделяется жирным), и довести до нужного блока; процесс проходит в режиме создания стрелок.

Старайтесь разные разветвляющиеся стрелки «вкладывать друг в друга», создавая таким образом меньшие пересечения. Не забывайте о центрировании стрелок, добавлении тильд, а также смены цвета для разветвляющихся стрелок.

На рис. 17 представлен первый уровень декомпозиции в нотации IDEF0.

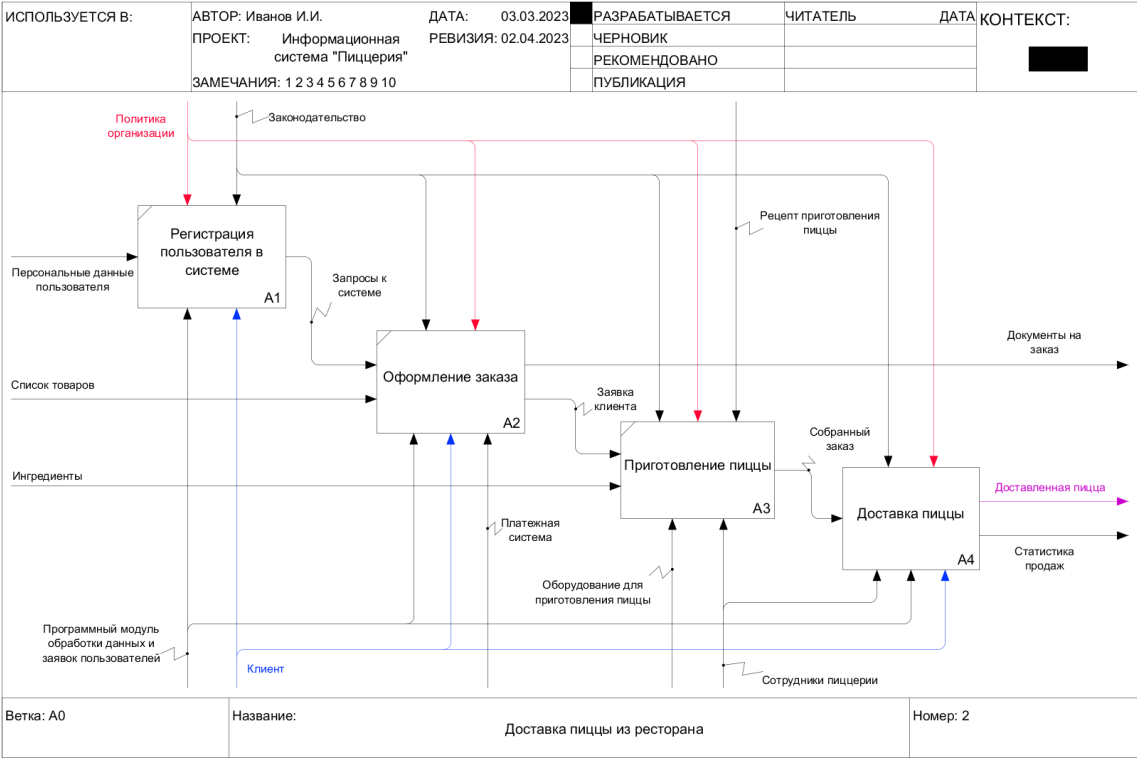


Рисунок 17 – Декомпозиция контекстной диаграммы

Далее произведем декомпозицию функционального блока «Доставка пиццы». Перемещаться между уровнями диаграммы нам помогают те же стрелки, расположенные сверху на линии инструментов. В результате получены следующие функциональные блоки:

1. Получение заявки на доставку (A41).
2. Доставка пиццы клиенту (A42).
3. Подтверждение доставки заказа (A43).
4. Добавление заказа в отчетность по продажам (A44).

Хотим отметить, что на диаграмме появится туннелированная стрелка «Курьерская служба» (на верхних уровнях она малозначима и может быть показана лишь на данном уровне модели).

Функциональный блок «Получение заявки на доставку». На данном этапе сотрудники пиццерии передают заказ курьерской службе для доставки. Вход блока – собранный заказ, механизмы – сотрудники пиццерии, курьерская служба, управление – законодательство, политика организации. На выходе получается заявка на доставку.

На этапе «Доставка пиццы клиенту» курьер доставляет заказ до пункта назначения. На вход подают заявку, выполняют процесс курьерская служба и

клиент, руководствуясь законодательством и политикой организации. В результате заказ будет доставлен.

Этап «Подтверждение доставки заказа» проходит с помощью курьерской службы и модулем обработки заявок пользователей. Курьер должен подтвердить доставленный заказ, руководствуясь политикой организации и законодательством. В последний этап переходит статус доставки.

Последний этап доставки «Добавление заказа в отчетность по продажам». После получения окончательного статуса доставки программный модуль добавляет заказ в отчетность по продажам и формирует финальную статистику продаж.

На рис. 18 представлена полученная диаграмма:

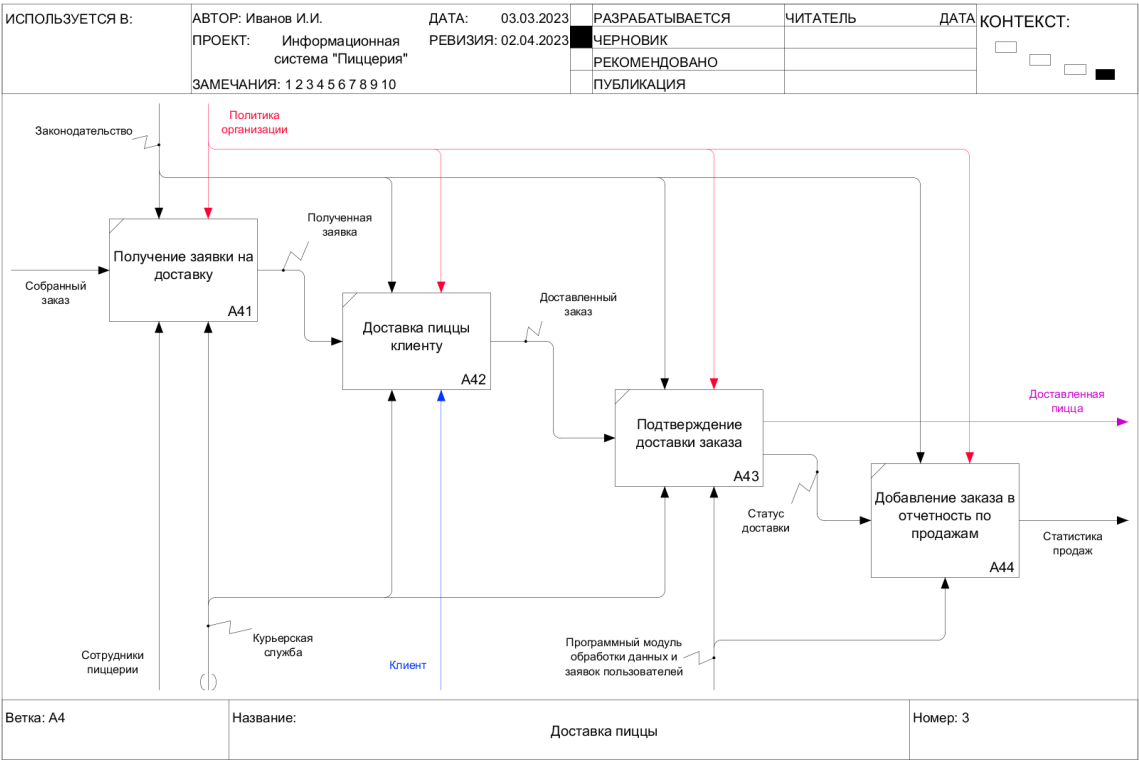


Рисунок 18 – Диаграмма декомпозиции процесса «Доставка пиццы»

По окончанию построения какой-либо диаграммы, можно изменить ее статус проектирования. На рис. 18 можно увидеть в шапке статус «Черновик». Для изменения статуса на определенном уровне модели выберите «Диаграмма» --> «Свойства диаграммы» --> «Статус» и нажмите на необходимый статус.

В работе также необходимо добавить текстовое описание функциональных блоков с их входящими и исходящими объектами (данными).

Практическая работа № 6. Проектирование модели потоков данных в нотации DFD

Задание: выбрать наиболее значимый функциональный блок нижнего уровня декомпозиции из предыдущей практической работы и выполнить его декомпозицию в нотации DFD. Декомпозиция выполняется как отдельный файл в нотации диаграммы потоков данных. Необходимо создать двухуровневое описание потоков данных. Добавить текстовое описание потоков и блоков, сделать выводы к работе.

Цель работы

Целью данной практической работы является продолжение создания описания проектируемой информационной системы, которое заключается в выборе наиболее значимого функционального блока нижнего уровня декомпозиции из предыдущей практической работы и проектировании его декомпозиции в нотации DFD (диаграмма потоков данных), а также подготовка к выполнению диаграммы сущность – связь проектирования баз данных ИС.

Ход работы

Для декомпозиции в нотации DFD проектируемой системы выбрать один из блоков декомпозиции, полученных в предыдущей практической работе, и выполнить дальнейшую декомпозицию (как минимум два уровня) в нотации DFD. Работу следует оформлять в соответствии с ГОСТ 7.32-2017 «Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления» (введен в действие Приказом Росстандарта от 24.10.2017 N 1494-ст).

Для примера возьмем систему автоматизации образования в России. Данная система является очень обширным объектом, поэтому ее можно детализировать большое количество раз. Остановимся на системе обучения в университете для демонстрации построения DFD-диаграммы.

Синтаксис данной нотации может применяться в двух вариантах: Йордана Де Марко и Гейна-Сарсона. В программе Ramus Educational применяется синтаксис Гейна-Сарсона, поэтому работа будет рассматриваться с данным вариантом.

В табл. 4 представлены основные элементы диаграммы в нотации DFD по Гейну-Сарсону (иногда могут отличаться из-за разных версий программы):

Таблица 4 – Элементы DFD-диаграммы по Гейну-Сарсону

Элемент	Описание
	Потоки данных
	Функциональный блок (процесс)
	Хранилище данных (базы данных, таблицы)
	Внешняя сущность (не входят в систему, но могут отправлять или получать информацию)

Процесс декомпозиции диаграммы рассмотрен в практической работе №5; различием является лишь установка флажка на нотацию DFD (см. рис. 19).

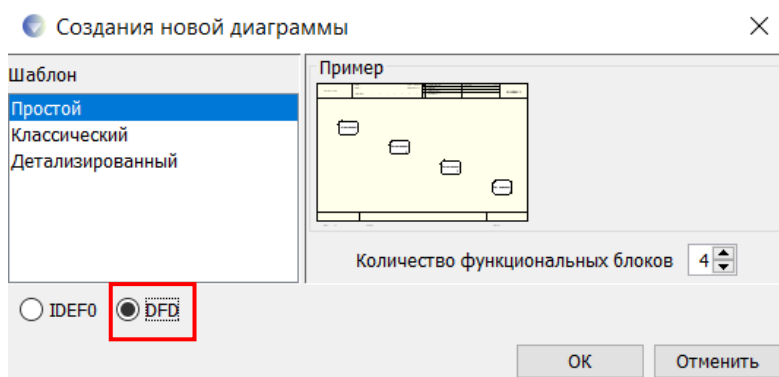


Рисунок 19 – Декомпозиция диаграммы в нотации DFD

С установкой новой нотации меняется также и интерфейс программы, который представлен на рис. 20.



Рисунок 20 – Интерфейс приложения в нотации DFD

Список измененных элементов в интерфейсе:

1. Функциональный блок в нотации DFD.
2. Режим добавления внешних ссылок (сущностей).
3. Режим добавления хранилищ данных.

Остальные необходимые инструменты программы упоминались в практической работе № 4.

Синтаксис DFD не предполагает наличие граничных стрелок, однако при декомпозиции диаграммы IDEF0 в DFD программа Ramus позволяет создавать граничные стрелки и не считает это за ошибку. Можно некоторые граничные стрелки (в основном управление и механизмы) удалить. Следует заметить, что в принципе использование граничных стрелок на DFD диаграммах, хоть и не является ошибкой, все же не рекомендуется.

В нотации DFD стрелки (потoki данных) равнозначны и могут присоединяться к любой стороне процессов, а также могут быть двунаправлены. У каждого процесса необходим как минимум один входящий и один выходящий по-

ток! При построении диаграммы в нотации DFD не забывайте, что процессы должны активно взаимодействовать либо друг с другом, либо через хранилища данных, тем самым передавая информацию. При необходимости обозначайте, кто получает или передает информацию (внешние сущности) и через что это происходит (документы, хранилища данных).

Модель, полностью сделанная в нотации DFD, обычно предполагает внешние сущности как входы и выходы в системе.

Продолжим наше обучение с построения декомпозиции диаграммы. Необходимо обозначить начальные и выходные данные декомпозируемого процесса «Обучение в университете». На входе данного процесса заходит поток информации об учащихся, а на выходе получается поток сведений об успеваемости студентов. Теперь можно декомпонировать процесс.

На новом уровне диаграммы будут проходить три функциональных блока (здесь, как и в IDEF0, обязательно присутствие не менее 3 функциональных блоков на уровне): «Согласовать список учащихся», «Согласовать предметы», «Проводить занятия».

Функциональный блок «Согласовать список учащихся». На вход поступает начальная информация об учащихся (например, из учебного отдела), и происходит согласование списков курсов и групп студентов. Для получения дополнительной информации можно обратиться к Списку студентов (хранилище данных). На выходе процесса мы получим структурированный список студентов университета.

Функциональный блок «Согласовать предметы». На текущем этапе согласовываются предметы групп студентов, обращаясь к Списку доступных преподаваемых предметов. На выходе получаем список предметов для групп студентов.

Функциональный блок «Проводить занятия». Процесс проходит с помощью преподавателей (внешняя сущность). В результате учитывается информация об успеваемости студентов, которая уходит дальше в учебный.

На рис. 21 представлена полученная декомпозиция:

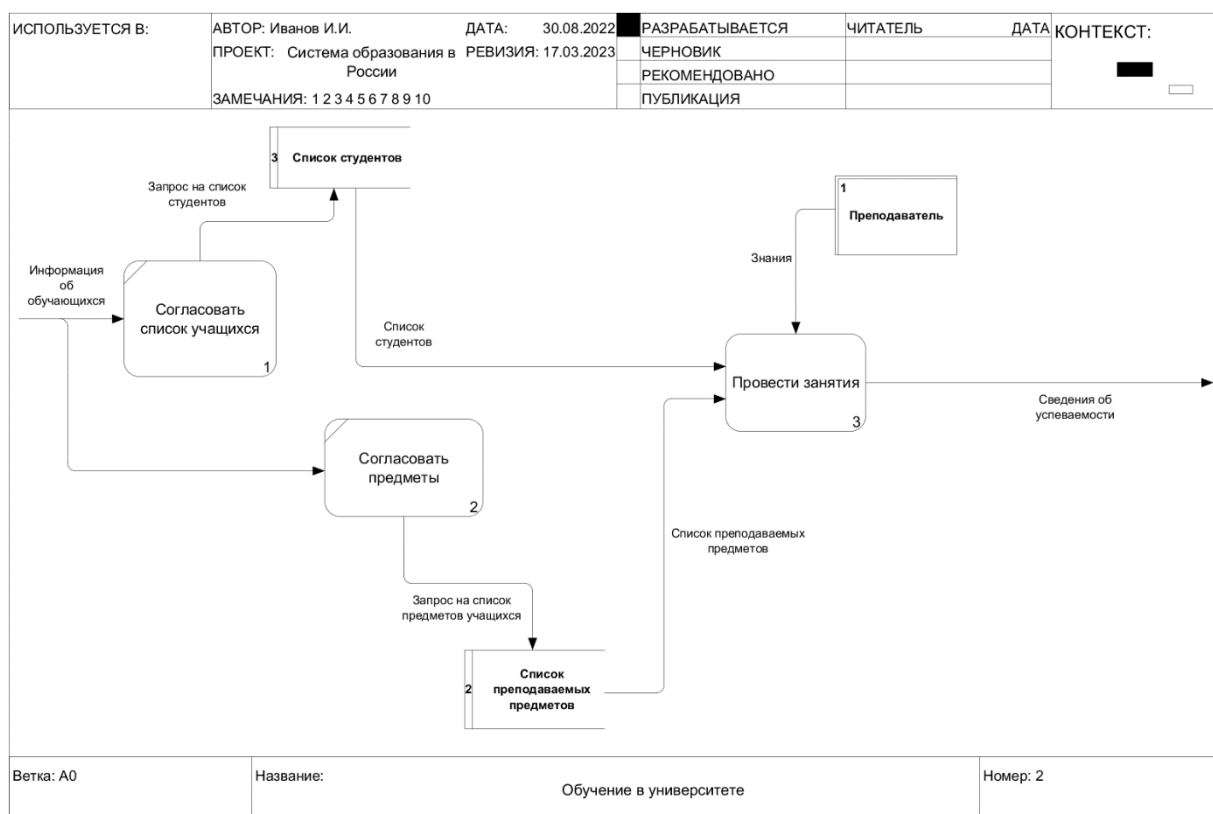


Рисунок 21 – Декомпозиция процесса в нотации DFD

Так как заданием объявлено реализовать двухуровневую диаграмму в нотации DFD (по согласованию с преподавателем возможно проектирование 1 уровня), продолжим декомпозицию процесса «Проводить занятия».

Процесс «Проверить присутствие студентов». Проводится проверка присутствия студентов по списку группы, внося статус явки студента в список присутствующих.

Процесс «Ознакомить с темой предмета». На текущем этапе проходит устное обучение об основных моментах тематики предмета преподавателем. Данную информацию получают лишь присутствующие студенты на занятии.

Процесс «Провести практическую часть обучения». После получения основных знаний о проходящей теме, студенты начинают самостоятельно обучаться и выполнять практические задания, выданные преподавателем. Обучение проходит с помощью базы знаний (хранилище данных).

Процесс «Оценить успеваемость». Преподаватель оценивает качество выполнения работ студентов и отправляет информацию об их успеваемости. Студенты тем временем получают свои знания.

На рис. 22 представлен второй уровень проектируемой DFD-диаграммы:

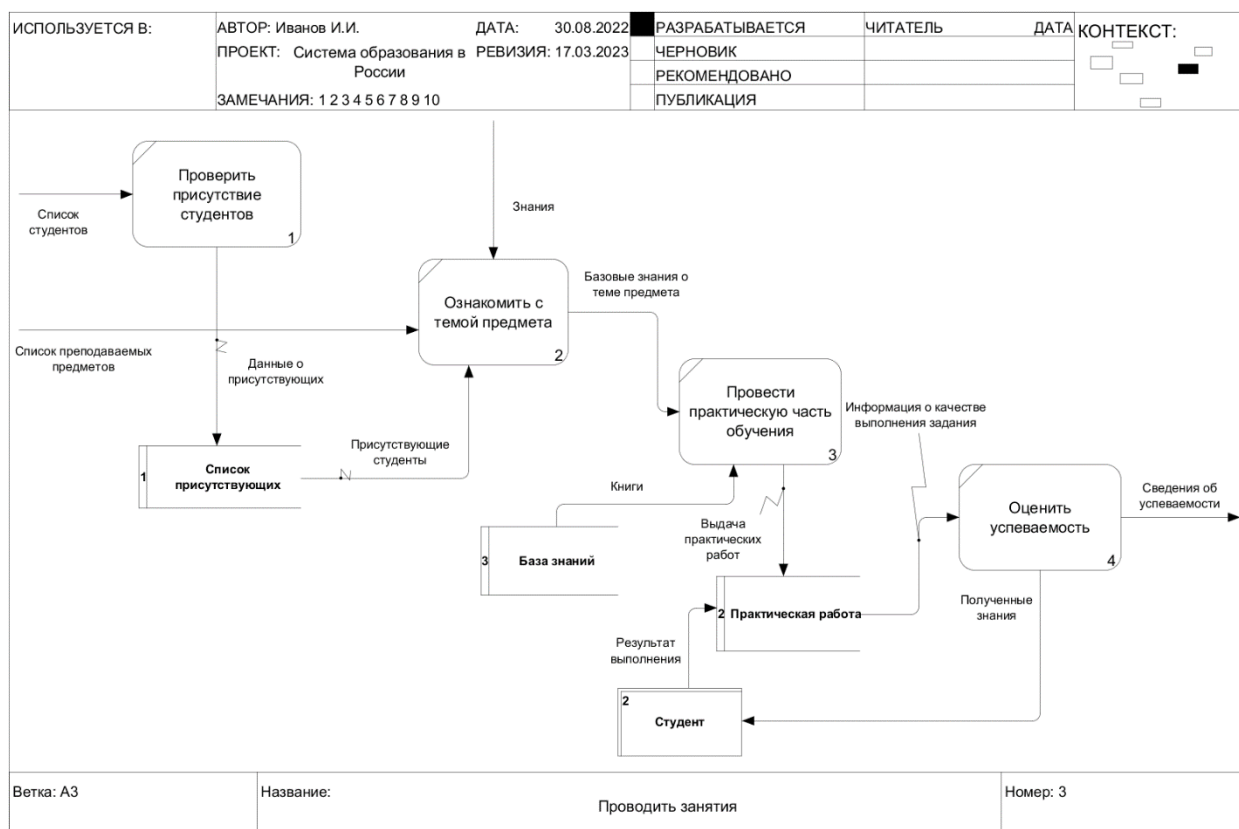


Рисунок 22 – Второй уровень проектируемой диаграммы в нотации DFD

Обратите внимание на процесс добавления внешних сущностей и хранилищ данных на диаграмму! Они связаны с материальными объектами, находящимися за пределами границ анализируемой системы. Для того, чтобы без ошибок обозначить их в проектируемой системе, необходимо добавить классификаторы.

Чтобы открыть окно работы с классификаторами, необходимо в правой верхней части рабочей панели выбрать Классификаторы, а затем выбрать «Окна» --> «Показать окно» --> «Классификаторы». Также работу с классификаторами можно произвести, непосредственно нажав на недавно добавленные внешнюю сущность или хранилище данных, и затем выбрать «Задать DFD-объект». В начале работы окно работы с классификаторами будет пустым.

Процесс добавления DFD-объекта:

1. Нажать правой кнопкой мыши на левое белое пространство окна. Выберете «Создать элемент».
2. Перейти в созданный классификатор и нажмите на зеленый плюсики в верхней панели инструментов. Создан DFD-объект.
3. Нажать на новый объект дважды и дать ему название. Далее нажать на выбор этого объекта в качестве действующего.

Классификаторы помогают правильно распределить внешние объекты системы (классификация может производиться в зависимости от требований). В представленном случае объекты были разделены на хранилища данных (список студентов, база знаний), внешние сущности (студент, преподаватель) и необходимые документы для проведения процесса (практическая работа, список присутствующих).

Так как классификаторы имеют иерархическую структуру, при открытии контекстного меню можно свернуть и развернуть дерево.

Несколько слов о панели редактирования классификаторов. В левом окне указаны созданные классификаторы, в правом окне принадлежащие к ним объекты. Имеется возможность редактирования, как и классификаторов, так и их объектов. Изменять классификаторы можно только из контекстного меню, которое появляется при нажатии правой кнопкой мыши. Объекты можно редактировать с помощью верхней панели или появляющегося контекстного меню. Не забывайте выбирать элемент, который хотите изменить! В меню имеются такие функции, как добавить или удалить элемент, вставить иконку элемента, сортировать по названию. Редактирование объектов также дополняется следующими функциями: объединить элементы, переместить элементы в другой классификатор.

Ниже на рис. 23 показан пример реализованных классификаторов для текущего задания:

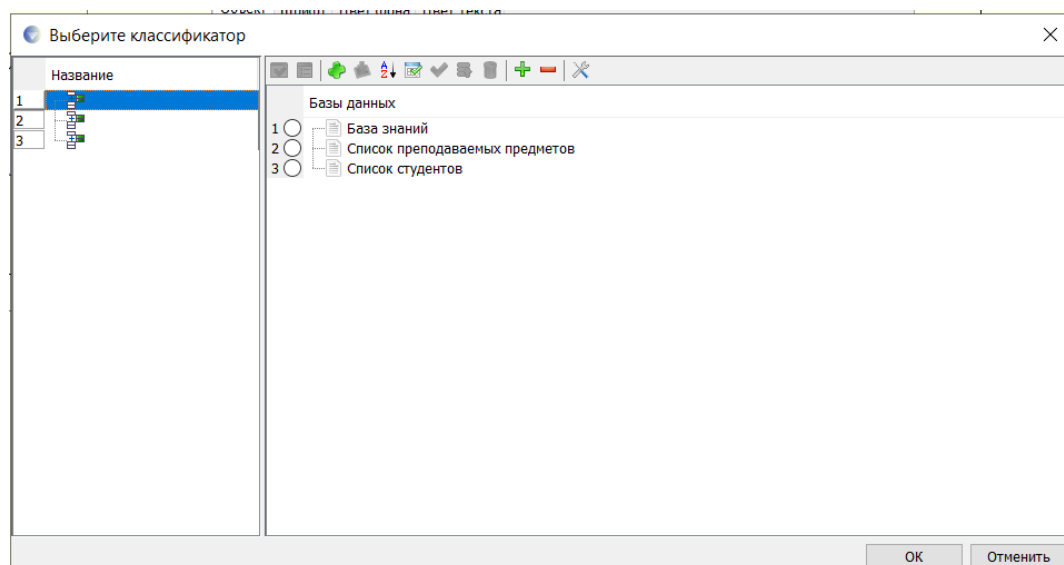


Рисунок 23 – Созданные классификаторы для DFD-диаграммы

Контрольные вопросы к практическим работам № 4–6

1. Что такое функциональная модель информационной системы?
2. Какие нотации можно использовать для создания функциональной модели информационной системы?
3. Как строится функциональная модель информационной системы в нотации IDEF0?
4. Что является самым важным при создании диаграммы IDEF0 уровня А-0?
5. Какие нормативные документы регламентируют создание диаграмм IDEF0?
6. Приведите основные требования к названию функционального блока диаграммы IDEF0 уровня А-0.
7. Приведите основные требования к названиям функциональных блоков диаграммы IDEF0.
8. Какие атрибуты отличают диаграмму IDEF0, отвечающую требованиям нормативных документов, от простой картинки?
9. Приведите требования нормативных документов по определению числа стрелок на диаграмме IDEF0. Являются ли эти требования выполнимыми? Как правильно интерпретировать эти требования?

10. Что необходимо для однозначного и полного понимания диаграммы IDEF0?
11. Что подразумевается под декомпозицией диаграмм в нотации IDEF0? Как можно осуществлять поиск функций-потомков и функций-родителей в нотации IDEF0? Какие CASE-средства поддерживают правильную нумерацию диаграмм и функциональных блоков IDEF0?
12. Назовите основные правила декомпозиции функциональных моделей согласно методологии SADT.
13. Какие требования предъявляются к стрелкам, входящим в блок функциональной модели IDEF0?
14. Какие процессы отражает диаграмма в нотации DFD?
15. На каком уровне декомпозиции можно использовать нотацию DFD?
16. Сколько уровней описания в нотации DFD рекомендуется выполнить в работе?
17. Как рекомендуется создавать DFD-диаграмму с использованием CASE-средства Ramus Educational?
18. Как расшифровывается аббревиатура DFD?
19. Как расшифровывается аббревиатура IDEF0?
20. Как расшифровывается аббревиатура SADT?
21. Назовите ограничения нотации методологии SADT.
22. Назовите достоинства методологии SADT.
23. Поясните связь между методологией SADT и нотациями IDEF0 и DFD.

Практическая работа № 7. Проектирование структуры данных информационной системы и создание ER-диаграммы

Цель работы

Создание модели «сущность – связь» в нотации ERD.

Основу проекта информационной системы составляют функциональная модель и модель взаимосвязей сущностей системы. В процессе создания модели «сущность – связь» студенты закрепляют знания о базах данных, полученные ранее. Формируют навыки проектирования баз данных. Выполняется анализ и формирование структуры базы данных.

В процессе выполнения работы необходимо решить задачу анализа модели потоков данных в нотации DFD и формирования ER-диаграммы. Вариант индивидуального задания определяет предметную область для разработки проекта базы данных некоторой информационной системы. Последовательность выполнения работы представлена далее.

1. Создание плана разработки проекта БД ИС. В процессе необходимо рассматривать БД как составную часть проектируемой ИС.

2. Создание текстового описания информационных объектов, сущностей и связей Проектируемой БД. Для этой цели повторно выполнить анализ предметной области создания ИС с целью выявления сущностей и связей. В процессе полезно выполнить описание типовых запросов проектируемой БД.

3. Построение концептуальной модели данных, создание ER-диаграммы «сущность – связь».

4. Проверка полноты и корректности созданной ER-диаграммы с использованием языка SQL. Для проверки необходимо создать типовые запросы, например, поиск и анализ данных.

5. Оформление отчета о выполненной работе. Отчет должен содержать: план разработки модели БД, выдержки из анализа предметной области, касающиеся определения сущностей БД и связей между ними, ER-диаграмму, примеры тестовых запросов SQL.

Теоретические сведения

База данных (БД) – важнейший компонент информационной системы. База данных – набор данных, структурированный особым образом в рамках выбранной модели.

Модель данных – это концептуальное представление данных (в частности сущностей и связей между ними) в проектируемой БД, выраженное графически (в виде схемы) и/или описательно (текстовое представление). Модель данных создается для того, чтобы разработчик понимал концепцию организации хранения и обработки данных.

Наиболее часто используют семантическое моделирование для описания модели данных. Как инструмент семантического моделирования используют различные варианты диаграмм «сущность-связь» (от английского Entity-Relationship).

Понятия «сущность» и «связь» являются базовыми понятиями ER-диаграмм. Сущность, как объект диаграммы entity-relation, описывает или представляет некоторые объекты или наборы объектов, как реальных, так и абстрактных. В качестве таких объектов могут выступать сотрудники или люди, которые выполняют определенные действия, подлежащие автоматизации. Объекты описываются или группируются в виде сущности по характерному признаку или характеристике, которые позволяют обособить данный объект от других. В нотации, предложенной Гордоном Эверестом, на ER-диаграмме сущности представляются в виде прямоугольников. Внутри прямоугольника ~~нахо-~~
~~дится~~ название этой сущности. Название пишется в единственном числе, именительном падеже.

«Экземпляр» сущность – её конкретный представитель. Также используются термины: «запись» и «кортеж», они являются эквивалентом экземпляра. Признак или характеристика, использованные для выделения объекта в сущность, называется – «атрибут сущности».

Связь – определенная ассоциация (или соответствие, взаимозависимость) между двумя сущностями. Связь может соединять не только две сущности, но сущность «саму с собой». **Сущность** необходимы для поиска взаимосвязей внутри БД.

Связи имеют несколько разных типов.

Тип один-к-одному. При таком типе связи только одна запись первой сущности связана только с одной записью в другой сущности. Например, у одного преподавателя на кафедре есть один компьютер.

Один-ко-многим. Такой тип связи встречается наиболее часто. При таком типе связей несколько строк из дочерней таблицы зависят от одной строки в родительской таблице. Например, в одной группе учатся несколько студентов. В этом случае таблица группы является родительской, а таблица студентов – дочерней.

При использовании типа связи «многие-ко-многим» подразумевается, что множественным записям из одной таблицы соответствуют множественные записи из другой таблицы. Например, у студентов групп предметы ведут разные преподаватели.

Порядок выполнения работы

Изучив предметную область (образовательное учреждение), переходят к изучению нормативных документов, регулирующих деятельность организации – заказчика. По результатам этого этапа выделяют цель и задачи системы (см. практические работы № 1–3).

Для примера рассмотрим систему сопровождения учебного процесса в высшем учебном заведении. Задача системы – сбор и обработка информации об основных участниках учебного процесса. Основные участники процесса – студенты и преподаватели. Система должна формировать печатные формы документов, необходимых для обеспечения непрерывности учебного процесса, в том числе: ведомости, допуски, сводные отчёты по результатам промежуточной аттестации. Система использует данные и документы от приёмной комиссии,

учебных отделов институтов, управления кадров. Информация об успеваемости студентов хранится в течение всего периода обучения, а часть (зачетная книга, студенческий билет, личный листок, учебная карточка) хранится 75 лет после завершения обучения.

Таким образом, проектируемая система должна выполнять следующие действия:

1. Хранить данные студентов и их успеваемость.
2. Генерировать и выводить на печать ведомости и допуски в рамках учебных отделов институтов.

Выделим все существительные в этих предложениях и проанализируем их: **Студент** – явная сущность. **Успеваемость** – явная сущность. **Институт** – сущность. **Специальность** – сущность, но, требуется уточнение, сколько специальностей существует в рамках одного института. **Предмет** – сущность.

1. На первоначальном этапе моделирования данных информационной системы явно выделены две основные сущности: **Студент** и **Успеваемость**.

2. Критерием успеваемости является наличие/отсутствие отметки о сдаче экзаменов и зачетов.

3. Сразу возникает очевидная связь **один-ко-многим** – «студент сдаёт несколько экзаменов» и «экзамены сдаются каждым студентом». Можно ее представить следующим образом (см. рис. 24)



Рисунок 24 – Пример связи один-ко-многим в различных нотациях

Студенты числятся за конкретными институтами, обучаются по определённым специальностям и сдают экзамены по дисциплинам (предметам) согласно программе подготовки данной специальности. Учитывая это, в **ER-модель** необходимо добавить дополнительные сущности. Тогда получим диаграмму, представленную на рис. 25.

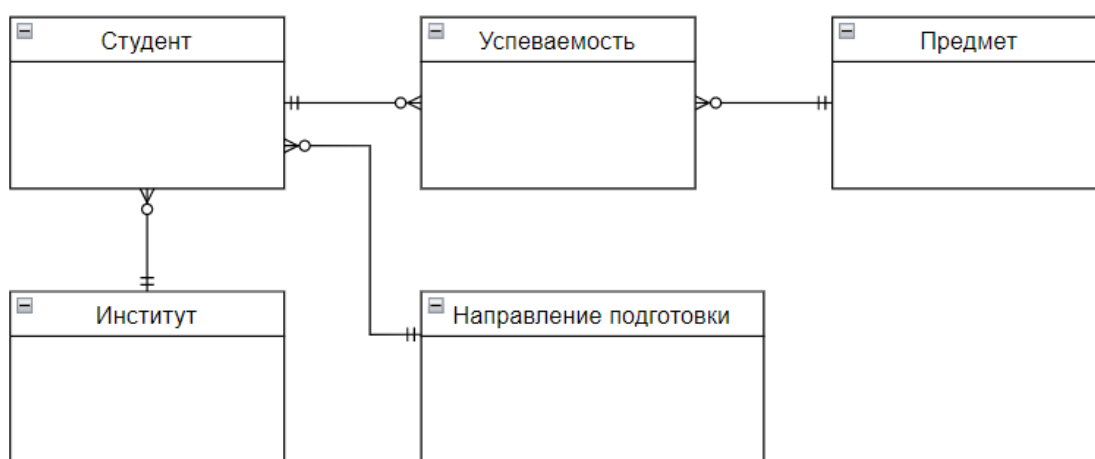


Рисунок 25 – Пример построения ER-диаграммы

Следующий этап проектирования модели требует внести атрибуты сущностей. После этого диаграмма примет законченный вид, представленный на рис. 26.

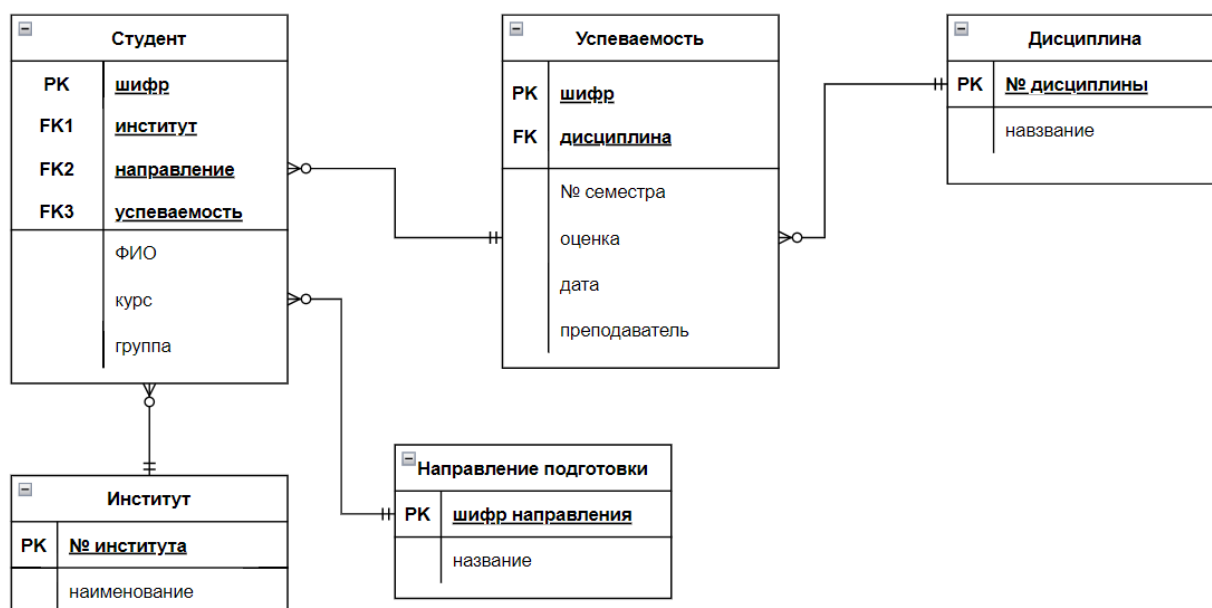


Рисунок 26 – ER-диаграмма, дополненная атрибутами сущностей

Отметим, что предложенные этапы моделирования являются условными и нацелены на формирование общих представлений о процессе моделирования. Разработанная ER-диаграмма – лишь концепция, которая не учитывает особенности конкретной СУБД. Однако физическую диаграмму, учитывающую особенности СУБД, можно построить, для этого надо учитывать такие параметры, как допустимые типы, наименования полей и таблиц, ограничения целостности и т. п.

Видеопример выполнения работы

Для выполнения работы рекомендуется использовать бесплатный онлайн-сервис draw.io <https://rt.draw.io/>. Процесс выполнения можно посмотреть по ссылке: <https://youtu.be/uKImrwjOKTU>.

Дополнительный материал по работе

В разделе будет описан дополнительный материал по проектированию ER-диаграмм в сервисе draw.io (элементы построения можно найти в разделе «сущность-связь»).

Для примера возьмем систему доставки пиццы. Можно выделить такие основные сущности, как Пицца, Ингредиент, Заказ, Клиент, Служба доставки (все сущности на диаграмме обозначаются в единственном числе).

На первом этапе построим связь между сущностями Пицца и Ингредиент. В данном случае образуется связь **многое-ко-многим** – «пицца содержит много ингредиентов» и «один ингредиент может содержаться в разных пиццах». На практическом уровне данную связь не советуют использовать, так как в дальнейшем могут возникнуть проблемы и ошибки в разработанной базе данных. Чтобы упростить связь, создадим вспомогательную сущность «Состав» и свяжем ее с остальными таблицами со связью один-ко-многим. Получится таблица, которая будет содержать уникальный номер пиццы и ингредиента (дополнительным атрибутом получится количество ингредиентов) (см. рис. 27).

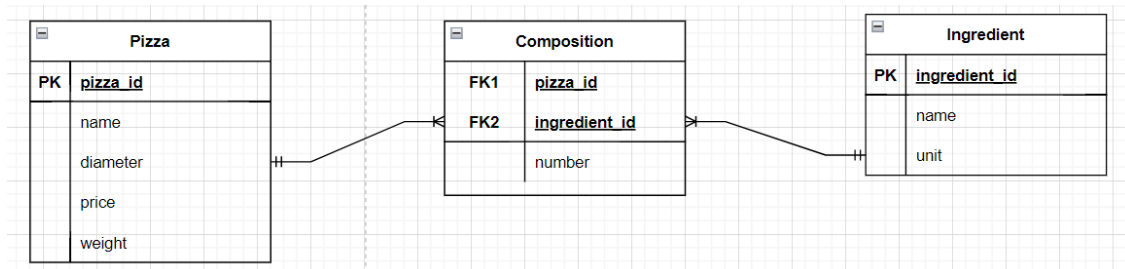


Рисунок 27 – Проектирование сущностей Пицца и Ингредиент и связей между ними

У каждой сущности есть свой уникальный номер (например, pizza_id или ingredient_id), а также ряд дополнительных атрибутов. У сущности Пицца – название, диаметр, цена, вес; у сущности Ингредиент – название, единица измерения.

Чтобы произвести доставку пиццы на дом, для начала нужно оформить Заказ, а выполняет его Клиент. Создадим данные две сущности и установим между ними связь **один-ко-многим** – «клиент может иметь много заказов, но сам заказ относится к одному клиенту». Далее возникает еще одна связь много-ко-многим между Заказом и Пиццей, исправим ее тем же способом, что показан выше. В результате к нашей ER-диаграмме прибавляются еще 3 таблицы (см. рис. 28).

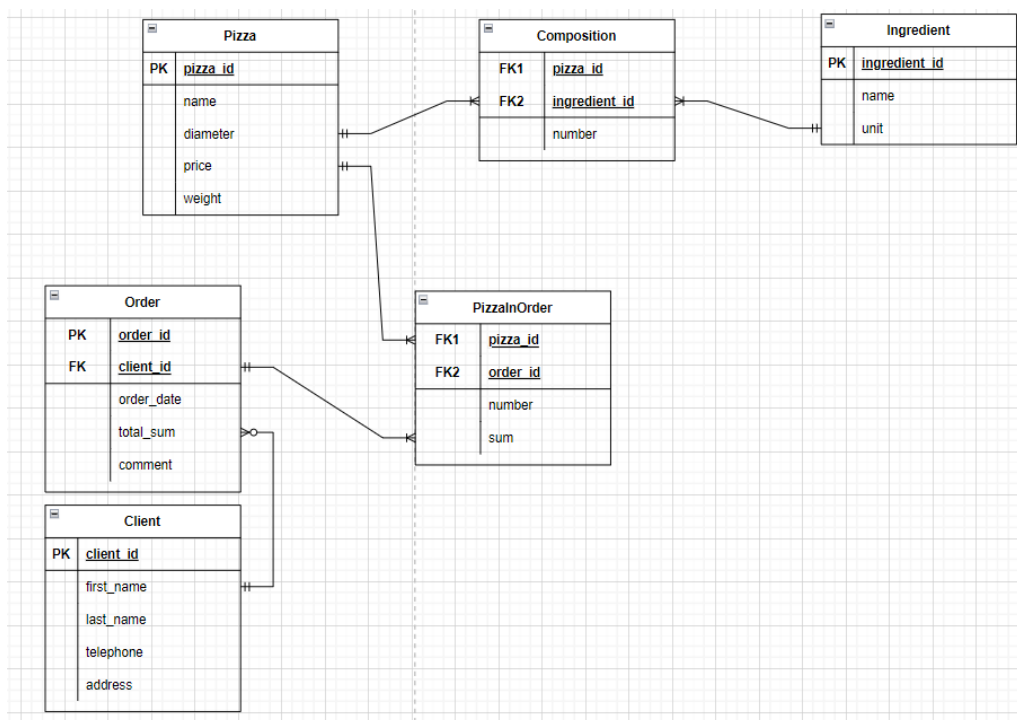


Рисунок 28 – Проектирование сущностей Клиент и Заказ

В вспомогательной таблице «Пицца в заказе» присутствуют уникальные номера пиццы и заказа, а также дополнительные атрибуты – количество и сумма.

Для оформления заказа от клиента всегда требуется его имя и фамилия, телефон. В самом заказе помимо добавленных пицц необходимо знать дату заказа, итоговую сумму, комментарий к заказу, адрес доставки.

Последним этапом проектирования является процесс доставки пиццы. У каждой пиццерии, доставляющей пиццу, есть служба доставки. В самой службе всегда есть курьеры. Из данной информации можно определить, что у сущностей Служба доставки и Курьер будет связь **один-ко-многим** – «у службы есть много работников, но курьер работает лишь в одной».

После получения нового заказа, его номер пересылается в службу доставки и назначается курьер. При доставке курьер должен знать информацию о заказе и о клиенте, а клиент в свою очередь должен быть уведомлен, какой курьер доставляет ему пиццу. Из данной информации получились следующие сущности и связи (см. рис. 29):

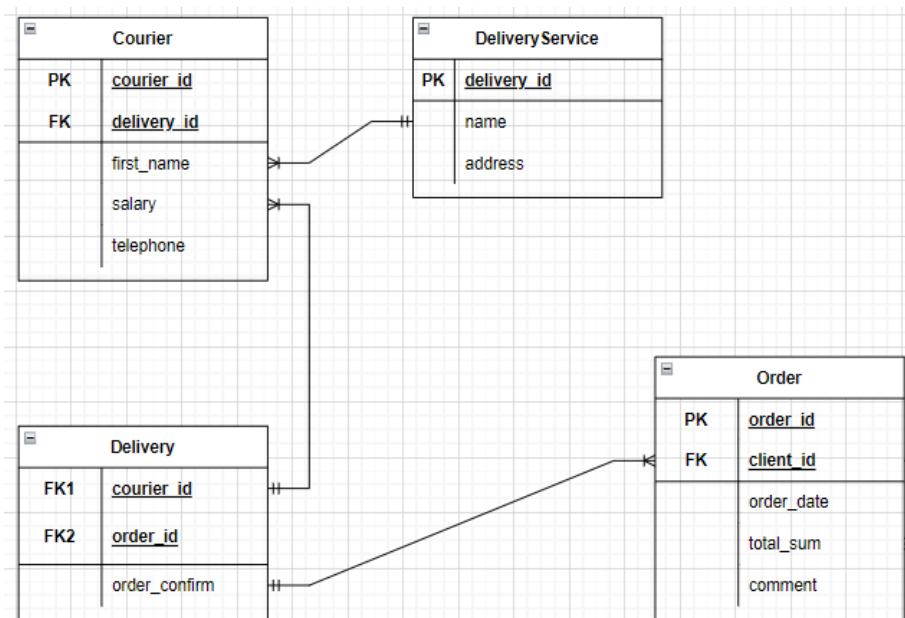


Рисунок 29 – Проектирование сущностей Служба доставки, Курьер и Доставка

Кратко рассмотрим атрибуты сущностей. У Службы доставки – название и местоположение, у Курьера – имя, зарплата, телефон, у Доставки – подтверждение заказа.

Для наглядности на рис. 30 представлена полная ER-диаграмма системы доставки пиццы:

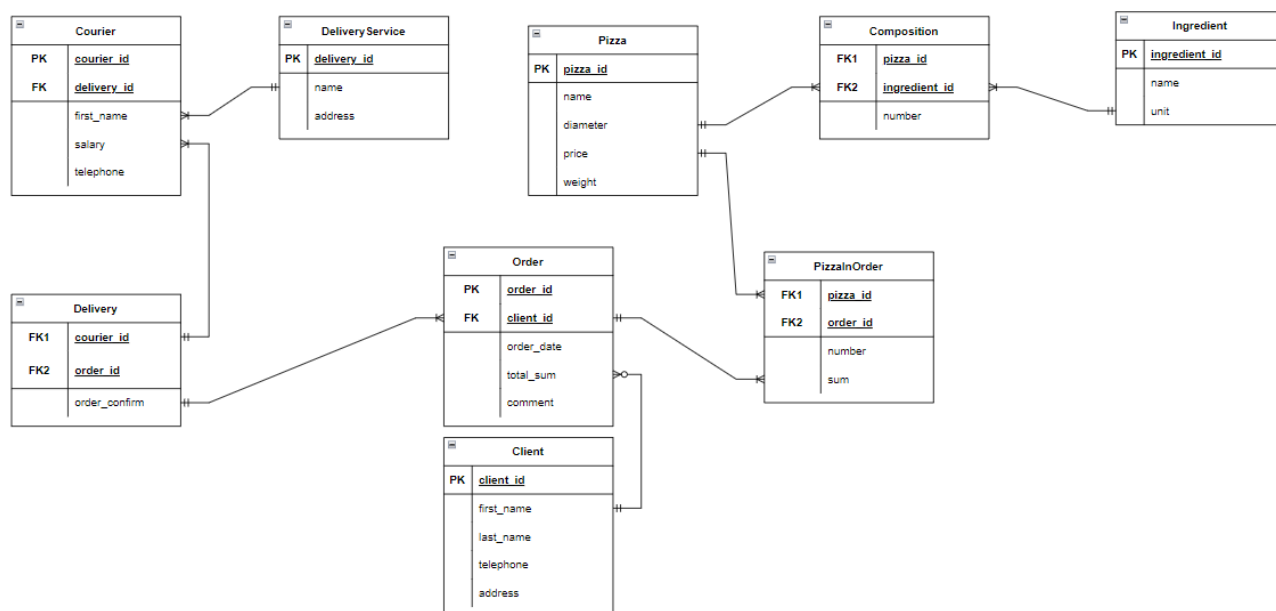


Рисунок 30 – ER-диаграмма системы доставки пиццы

Контрольные вопросы к практической работе № 7

1. Что такое база данных информационной системы? Дайте определение.
2. Что такое модель данных? Дайте определение.
3. Что такое «сущность» в рамках ERD-моделирования?
4. Что такое «связь» в рамках ERD-моделирования?
5. Какую роль играет ER-диаграмма в проекте информационной системы?
6. Какую задачу надо решить в процессе выполнения работы?
7. В каком аспекте рассматривается БД информационной **системе** в рамках данной работы?
8. Каким образом определяются сущности в процессе создания ER-диаграммы?
9. Кто из авторов предложил наиболее часто используемую нотацию ERD-моделирования?
10. Что такое «экземпляр сущности»? Дайте определение.
11. Каков принцип группировки объектов в общую сущность ERD?

12. Какие элементы входят в модель ERD?
13. Какие CASE-средства рекомендованы для разработки ER-диаграммы?
14. Какие типы связей доступны при создании ER-диаграммы?
15. Назовите правила и приведите пример использования связей один-ко-многим ER-диаграммы?
16. Назовите правила и приведите пример использования связей один-к-одному ER-диаграммы?
17. Назовите правила и приведите пример использования связей многие-ко-многим ER-диаграммы?

Практическая работа № 8. Создание диаграммы состояний

Цель работы

Создать диаграмму состояний проектируемой информационной системы для одного из ранее разработанных классов или прецедентов.

Краткие теоретические сведения

Диаграммы введены в язык UML для моделирования динамических аспектов системы. По сути, диаграмма состояний соответствует таковой в теории автоматов. Полезно использовать диаграмму состояний для моделирования жизненного цикла информационной системы или ее частей. При моделировании с помощью диаграммы состояний проектировщик может описать процесс изменения состояний только одного экземпляра определенного класса – одного объекта. Моделирование объекта выполняется как описание реакцией этого объекта на некоторые внешние события. В качестве моделируемого объекта выступают, как правило, классы, прецеденты или система в целом. Еще раз отметим, что эти объекты описываются через их реакции на некоторые внешние события.

Диаграмма в языке моделирования UML – наглядное представление некоей совокупности элементов модели системы в виде графа, на котором дуги (отношения) связывают вершины (сущности). В своём графическом виде различные виды диаграмм UML (диаграммы классов, компонентов, объектов и др.) применяются для визуализации разных аспектов устройства или поведения моделируемой системы.

Диаграмма не принадлежит к семантическим элементам языка UML и её значение не зависит от того, каким образом она представлена. Кроме редких исключений основная часть несомой ею информации содержится не в размере или расположении её элементов, а в её топологической конфигурации. Чаще всего применяются три основных типа визуальных отношений между элементами: соединение (в виде линий от одной двумерной формы к другой), включение (в виде помещения одних форм внутрь других) и визуальное подкрепление



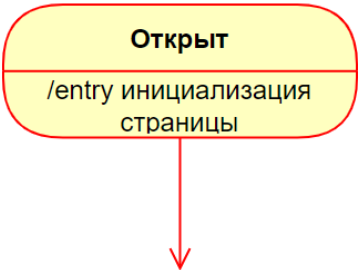


(в виде размещения одних символов недалеко от других). Возможно также представить данную нотацию и в аналитической форме как совокупность связей узлов графа.

Возьмем для построения диаграммы состояний онлайн-сервис draw.io. В качестве примера будет выступать система онлайн-магазина одежды.

Главная цель данной информационной системы – автоматизировать продажу одежды, поэтому разрабатывать диаграмму состояний следует на основе выбора одежды, оформления заказа и дальнейшей покупки.

Перед проектированием диаграммы изучим основные элементы (в сервисе draw.io они расположены в разделе «UML»), указанные в табл. 5.

Таблица 5 – Элементы диаграммы состояний и их описание

Элемент	Описание
	Старт жизненного цикла системы или ее части, определенных процессов
	Элемент состояния системы
	Элемент состояния системы с подзаголовками: 1) entry/ – операция перед наступлением состояния; 2) do/ – операции во время данного состояния; 3) exit/ – операция после перехода в другое состояние
	Соединение/развилка – может соединять несколько переходов или наоборот разветвлять
	Конец жизненного цикла системы, ее частей, определенных процессов

Первым состоянием жизненного цикла онлайн-магазина одежды будет его инициализация. В текущем состоянии выполняется инициализация подсистемы магазина «USER».

При инициализации роли процесс переходит к следующему состоянию «Открыт». Происходит открытие страницы магазина с преждевременной инициализацией списка товаров. Пользователь выбирает ~~неправившейся~~ товар и добавляет его в корзину.

После выбора всех интересующих товаров процесс переходит в состояние «Корзина». До взаимодействия с состоянием инициализируется список товаров в корзине. В самой корзине пользователь может изменить количество товара или вовсе удалить его.

Когда корзина товара полностью удовлетворяет клиента, система переходит в состояние «Оформление заказа». При нахождении в данном состоянии пользователь заполняет всю необходимую для доставки информацию, а также выбирает способ оплаты.

При выполнении этих действий заказ считается оформленным, но не подтвержденным, и процесс покупки переходит в состояние «Оплата заказа». Пользователь оплачивает товары, и по окончании проходит запись статуса оплаты.

Далее остановимся на развилке состояний. Бывают случаи, когда из одного состояния диаграммы можно перейти сразу в несколько. Например, при дополнительной детализации диаграммы могла бы получиться развилка в виде «Выбор раздела» --> «Каталог», «Личный кабинет», «Корзина» и т. д. Обратным действием (соединение) может являться переход из всех разделов на главную страницу.

Текущая диаграмма упрощена в целях большего понимания и получении навыков именно в проектировании диаграммы состояний.

Продолжим проектирование второй части диаграммы. Здесь у нас встречается развилка в виде успешного или провального статуса оплаты. Если оплата успешно совершена, происходит переход на «Подтверждение заказа». В этом состоянии система автоматически должна отправить трек отслеживания и чек на почту. В случае ошибки операции система в состоянии «Отмена заказа» отправляет уведомление о возникшей проблеме. После всех действий процесс системы заканчивается. Полную диаграмму можно рассмотреть на рис. 31.

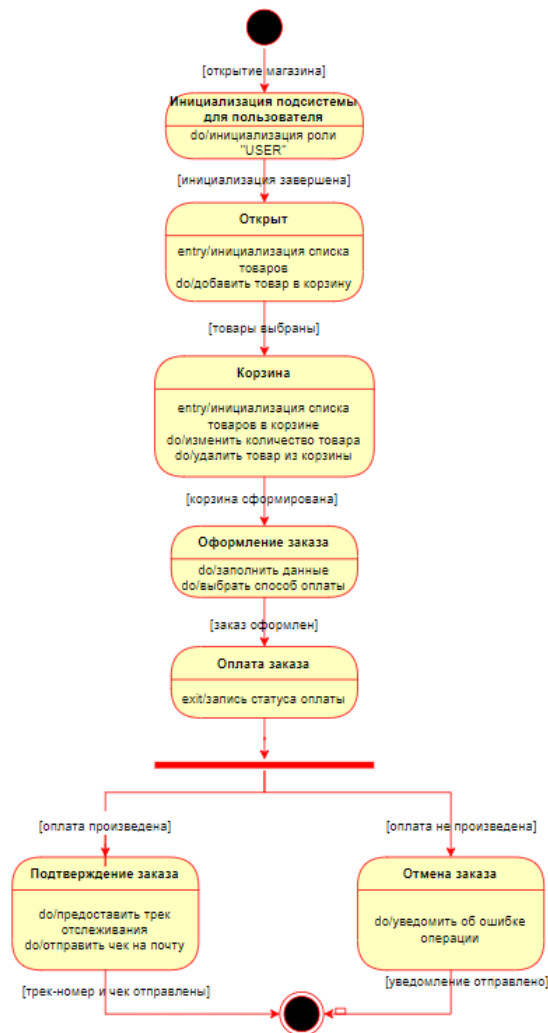


Рисунок 31 – Диаграмма **состояний** системы онлайн-магазина одежды

Контрольные вопросы к практической работе № 8

1. Для чего в язык UML введена диаграмма состояний?
2. Для каких целей используется диаграмма состояний UML?
3. Что выступает в качестве объекта моделирования диаграммы состояний UML?
4. Каким образом передается основная информация, заложенная в диаграмме состояний UML?
5. Какие типы отношений чаще всего применяются в диаграмме состояний UML?
6. Как можно представить диаграмму состояний UML в аналитическом виде?

Практическая работа № 9. Расчет информационной энтропии проектируемой системы

Цель работы

Целями выполнения работы являются:

1. Закрепление имеющихся знаний о параметрах ИС. Изучение методологии расчета требуемых параметров проектируемой информационной системы.
2. Приобретение навыков анализа и формализованного описания заданной предметной области.
3. Приобретение навыков расчета параметров информационной системы.

В процессе выполнения работы решаются следующие задачи:

1. Выполняется системный анализ заданной предметной области. Составляется формализованное описание информационных объектов предметной области.
2. Выполняется расчет параметров проектируемой информационной системы.

Краткие теоретические сведения

Информация – сообщение, которое имеет ценность, значимость для субъекта. Информация, не обладающая ценностью, называется тривиальной.

Энтропия – степень неопределенности информации, в том числе в информационной системе. Энтропия чем-то схожа с дисперсией, но в отличие от последней, нам не обязательно знать закон распределения.

Статистически информация – сообщение о состоянии системы, уменьшающее неопределенность знаний о ней. Для измерения информации вводятся параметры:

- количество информации I ;
- объем данных VD .

Меры информации:

- синтаксическая, семантическая, **прагматическая** объем данных – VD ;
- ценность использования – I_c ;
- количество информации – I .

Синтаксическая мера информации. Объем данных V_D в сообщении измеряется количеством символов (разрядов) в этом сообщении. Двоичная система счисления: единица измерения – бит (bit – binary digit – двоичный разряд), байт – 8 бит. Десятичная система счисления: единица измерения – дит (десятичный разряд).

Количество информации I на синтаксическом уровне определим с помощью понятия неопределенность состояния системы – энтропия системы – не-которая система $H()$ – мера неосведомленности (неопределенности) о системе $H()$ – неопределенность состояния системы после получения сообщения $I()$ – количество информации о системе, полученной в сообщении (уменьшение неопределенности состояния системы) $I() = H() - H()$.

Энтропия системы может рассматриваться как мера недостающей информации. Энтропия системы $H()$, имеющая N возможных состояний по формуле Шеннона P_i – вероятность того, что система находится в i -м состоянии.

$$H(x) = - \sum_{i=1}^{i=n} [p_i \cdot \log_a p_i]$$

Семантическая мера измерения информации. Для измерения смыслового содержания информации используется понятие тезаурус пользователя.

Обзор теории практической работы

Дополнительная информация по работе представлена в следующем видео: https://youtu.be/Ds56e6_EVFQ

Ход выполнения работы

Вариант индивидуального задания определяет предметную область для разработки проекта базы данных некоторой информационной системы.

В процессе выполнения работы необходимо:

1. Составить таблицу данных, оперируемых в проектируемой информационной системе, рассматривая эту таблицу, как диапазон возможных значений данных, хранящихся в будущей информационной системе, автоматизирующей бизнес-процессы некоторой организации.

2. Выполнить анализ заданной предметной области. Сформулировать словесное описание информационных объектов. Описать типовые запросы для поиска и анализа информации об объектах предметной области.

3. Вычислить необходимые промежуточные параметры для вычисления энтропии системы, а также количественные параметры информации в проектируемой ИС.

4. Проверить полноту и корректность выполненных вычислений.

5. Составить результирующую таблицу и выполнить выводы по работе.

Задание: выполнить расчет, по крайней мере, одного, наиболее важного, параметра проектируемой информационной системы. Параметр рекомендуется согласовать с преподавателем, ведущим занятия. По умолчанию предлагается выполнить расчет энтропии системы.

Пример выполнения практической работы приведен в приложении Д.

Контрольные вопросы к практической работе № 9

1. Что такое энтропия информационной системы?
2. Дайте определение термину «Семантическая информация».
3. Что такое энтропия Шеннона?
4. Что такое дисперсия информации?
5. Взаимосвязь реляционных данных, корреляция данных.
6. Что такое модели данных?
7. Что такое мера информации?
8. Как выполнить анализ параметров проектируемой информационной системы?

Практическая работа № 10. Создание полного текстового описания, глоссария и расчет параметров проектируемой информационной системы

Цель работы

Оформить все выполненные задания в едином файле. Для оформления использовать требования ГОСТ 7.32-2017 и ГОСТ Р 7.0.100-2018.

Ход выполнения работы

Задание: создать полное текстовое описание всех процессов и потоков, отображенных в процессе создания и декомпозиции функциональной модели и диаграмм потоков данных проектируемой информационной системы. Вычлени понятия, используемые в полном текстовом описании, и создать Глоссарий, дав необходимые определения используемых понятий. Выполнить расчет одного, наиболее важного, параметра информационной системы. Параметр рекомендуется согласовать с преподавателем, ведущим занятия. По умолчанию предлагается выполнить расчет энтропии системы.

Работа выполняется как во время аудиторных занятий, так и в виде самостоятельной внеаудиторной работы. Выполнение каждой работы состоит из трёх этапов:

1. Подготовка и получение допуска к работе.
2. Получение индивидуального задания и выполнение основной части работы (задание взять из практической работы №1).
3. Оформление и защита отчёта о проделанной работе.

В начале каждой работы выполняется повторение теоретического материала и проверка готовности к выполнению работы с помощью контрольных вопросов. После получения допуска к выполнению работы выдаётся индивидуальный вариант задания для самостоятельной работы.

На заключительном этапе оформляется отчёт о проделанной работе с описанием полученных результатов и выполняется процедура защиты отчёта.

Процедура защиты отчёта заключается в проверке:

1. Правильности структуры, содержания и оформления отчёта.
2. Корректности полученных результатов и полноты их описания.
3. Способности дать объяснение и необходимое обоснование полученным результатам.

Отчет должен включать в себя:

1. Титульный лист.
2. Задание к выполнению работы.
3. Список используемых нормативных документов.
4. Список используемых сокращений.
5. Содержание отчёта.
6. Описание результатов по каждой части задания.
7. Глоссарий.
8. Приложение (таблицы данных).

Контрольные вопросы к практической работе № 10

1. Что входит в полное текстовое описание работ?
2. Какие нормативные документы определяют оформление полного текстового описания?
3. Какие параметры страницы должны быть в отчете в соответствии с нормативными документами?
4. Какое форматирование текста считается правильным согласно нормативным документам?
5. Укажите правильное оформление рисунков, таблиц, формул и ссылок на источники в тексте согласно нормативным документам.
6. Что такое Глоссарий? Дайте определение.

Список использованных источников

1. ГОСТ 7.32–2017. Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления": введен в действие Приказом Росстандарта от 24.10.2017 N 1494-ст: дата введения 2018-01-07. – URL: http://www.consultant.ru/document/cons_doc_LAW_292293/ (дата обращения: ~~20.04.2021~~). – Текст: электронный.

2. ГОСТ Р 7.0.100–2018. Национальный стандарт Российской Федерации. Библиографическая запись. Библиографическое описание: общие требования и правила составления: Приказом Федерального агентства по техническому регулированию и метрологии (Росстандартом) от 03 декабря 2018 N 1050-ст: дата введения 2019-01-07. – М.: Стандартинформ, 2018. – 128 с.

3. ГОСТ 19.701–90 ЕСПД (ИСО5807–85). Межгосударственный стандарт. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения: Постановлением Государственного комитета СССР по управлению качеством продукции и стандартам N 3294: дата введения 1990-01-01. – М.: Стандартинформ, 2010 – 158 с.

4. ГОСТ 34.602–2020. Межгосударственный стандарт. Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы: Приказом Федерального агентства по техническому регулированию и метрологии № 1522-ст от 19 ноября 2021 г.: дата введения 2022-01-01. – М.: Российский институт стандартизации, 2021. – 12 с.

5. Жизненный цикл информационной системы. Этап формирования требований к ИС: сайт. – URL: <https://www.sites.google.com/site/metodsybd/blok-5-etapy-ziznennogo-cikla/5-1-etap-formirovania-trebovanij-k-is> (дата обращения: 22.08.2022). – Текст: электронный.

6. Жизненный цикл информационной системы. Проектирование ИС: сайт. – URL: <https://www.sites.google.com/site/metodsybd/blok-5-etapy-ziznennogo-cikla/5-2-proektirovanie-is> (дата обращения: 22.08.2022). – Текст: электронный.

7. Алпатов, А.Н. Тестирование и отладка программного обеспечения: учебное пособие / А. Н. Алпатов, А. А. Лобанов, Ю. С. Лобанова. – Киров:

Изд-во МЦИТО, 2021. – 58 с. – URL: <https://mcito.ru/publishing/epub/tutorials?view=345> (дата обращения: 22.08.2022). – Текст: электронный. ISBN: 978-5-907419-18-6 <https://doi.org/10.52376/978-5-907419-18-6>

8. Вигерс, К. Разработка требований к программному обеспечению. 3-е изд., дополненное. Пер. с англ. / И. К. Вигерс, Дж. Битти. – М.: Издательство «Русская редакция»; СПб.: БХВ-Петербург, 2014 – 736 с.

9. Мартин, Р. Чистая архитектура. Искусство разработки программного обеспечения / Р. Мартин – СПб.: Питер, 2019 – 352с.

10. Ликбез по техническому заданию: сайт. – URL: <https://habr.com/ru/post/490006/> (дата обращения: 23.08.2022). – Текст: электронный.

11. Как составить техническое задание на разработку интернет-магазина – планируем сайт правильно: сайт. – URL: <https://www.insales.ru/blogs/university/kak-sostavit-tekhnicheskoe-zadanie> (дата обращения: 23.08.2022). – Текст: электронный.

12. Корпоративные хранилища данных. Интеграция систем. Проектная документация Проектный опыт. Аналитика и статьи. Примеры и шаблоны документов. ГОСТы: сайт. – URL: https://www.prj-exp.ru/patterns/pattern_tech_task.php (дата обращения: 23.08.2022). – Текст: электронный.

13. Раянов, Р. Техническое задание на разработку сайта. Шаблон ТЗ / Р. Раянов. – Текст: электронный // Falcon space: интернет-портал. – URL: <https://falcon.web-automation.ru/blog/tz-na-sozдание-sistemy-ili-sayta---prosto-o-slozhnom> (дата обращения: 24.08.2022).

14. Маглинец, Ю. Анализ требований к автоматизированным информационным системам. Лекция 2: Понятие требования. Классификации требований / Ю. Маглинец <https://intuit.ru/studies/courses/2188/174/lecture/4714?page=2> (дата обращения: 24.08.2022) – Текст: электронный.

15. Грекул, В.И. Проектирование информационных систем. Практикум: Учебное пособие / В.И. Грекул, Н.Л. Коровкина, Ю.В. Куприянов – М.: Национальный Открытый Университет «ИНТУИТ» 2012. – 187с.

Приложение А. Темы для выполнения практических работ

Примерные темы для выполнения практических работ по дисциплине «Проектирование информационных систем»:

1. Информационная система «Партнер».
2. Информационная система «Шелкографическое производство».
3. Информационная система «Электронный решебник задач естественных наук».
4. Информационная система «Интернет-магазин портативной электроники».
5. Информационная система «Репозиторий 3D-моделей».
6. Информационная система «Электронный сборник лабораторных работ».
7. Информационная система «Интернет-магазин игровой валюты».
8. Информационная система «Интернет-магазин одежды».
9. Информационная система «Электронный деканат».
10. Информационная система «Математический справочник школьника».
11. Информационная система «Электронные визитки».
12. Информационная система «Электронный дневник школьника».
13. Информационная система «Электронный помощник юриста».
14. Информационная система «Проектирование сети».
15. Информационная система «Интернет-магазин музыкальных инструментов».
16. Информационная-справочная система по фотографической технике с поддержкой удаленных пользователей.
17. Информационная система управления бизнес-процессами в сфере розничной торговли.
18. Информационная система-конфигуратор «Проектировщик сети».
19. Информационная система защиты интернет-ориентированных информационных ресурсов от атак класса «отказ в обслуживании».
20. Информационная система мониторинга активного сетевого оборудования.
21. Информационная медиа-система с удаленным доступом, поддерживающая тематику «Качество, сертификация и стандартизация программных средств» с учетом гармонизации и нормирования контента.

22. Информационная подсистема функционала бизнес-процессов прототипирования приложения Docs.

23. Информационная система – справочник с использованием когнитивной оценки контента.

24. Специализированный вертикальный аддитивный микропортал, предназначенный для контроля знаний учащихся в рамках школьного курса математики.

25. Информационная система «Интернет-портал по программированию с модулем экспертной оценки».

26. Информационная система поддержки фриланс-занятости в мультимедиа сфере.

27. Информационная система контроля успеваемости обучающихся.

28. Информационная система диспетчеризации инженерных подсистем жизнеобеспечения центра обработки данных.

29. Информационная система «Вариативный микропортал поддержки учебно-методического комплекса».

30. Информационная система «Начисление заработной платы сотрудникам фирмы».

31. Интернет-портал для дистанционного обучения английскому языку.

32. Информационная образовательная система с поддержкой дистанционного обучения на основе CMS Moodle.

33. Геоинформационная система на базе 3D-модели РТУ МИРЭА.

34. Информационная система контроля и слежения за автомобилем.

35. Типизируемая информационная система балльно-рейтинговой оценки успеваемости студентов.

36. Информационно-аналитическая система поддержки бизнес-процессов сервисного центра по ремонту компьютерного и периферийного оборудования.

37. Информационная система с экспертной оценкой неисправности портативного компьютера.

Приложение Б. Пример выполнения практической работы № 1

Введение

В 2021 году по Москве открыто большое количество музеев разных направленностей и их число только увеличивается. Музеи приносят большую пользу людям: обеспечивают культурное просвещение и духовное развитие, предоставляют возможность изучать историю по картинам, а также историю развития искусства, вдохновляют, позволяют найти новых знакомых и обладают другими полезными свойствами. Поэтому появляются интернет-порталы со всеми имеющимися музеями в городе, чтобы каждый мог выбрать для себя тот, который ему захотелось бы посетить. Интернет-порталы музеев помогают посетителям таких порталов найти график работы музея, анонсы выставок, описание достопримечательностей и другую ценную информацию.

Информационная система «Интернет-портал музеев Москвы» спроектирована, чтобы уменьшить время поиска сведений по музеям, пообщаться с единомышленниками и быть в курсе последних новостей о временных выставках, новых музеях и новых экспонатах.

Целью практической работы является формирование требований к описанной выше системе. Задачей практической работы является описание объекта автоматизации, формулировка основных задач автоматизации объекта, описание основных параметров проектируемой информационной системы, описание путей достижения целей. Кроме того, необходимо сформулировать требования к информационной системе.

1 Общие сведения

1.1 Список терминов и определений

Альбом – фотографии пользователей, сделанные в музеях.

Афиша – новости о выставках, галереях, выставочных залах, новых музеях, скидках в музей и о конференциях, проходящих в музеях.

БД (База Данных) – представленная в объективной форме совокупность самостоятельных материалов, систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ).

ИП (Интернет-Портал) – многофункциональная площадка с разнообразным интерактивным сервисом, включающая в себя обширные возможности и услуги, в том числе путём предоставления пользователям ссылок на другие сайты.

ИС (Информационная Система) – система, предназначенная для хранения, поиска и обработки информации, и соответствующие организационные ресурсы (человеческие, технические, финансовые и т. д.), которые обеспечивают и распространяют информацию.

Музеи – каталог музеев Москвы, с описанием, местонахождением и графиком работы.

Открытый музейный форум – раздел на сайте, предназначенный для общения пользователей, обмена знаниями и проведения дискуссий.

СУБД (Система Управления Базами Данных) – совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.

CSS (Cascading Style Sheets) – формальный язык описания внешнего вида документа, написанного с использованием языка разметки.

HTML (Hyper Text Markup Language) – стандартизированный язык разметки веб-страниц во Всемирной паутине.

MS (Microsoft) – одна из крупнейших транснациональных компаний по производству проприетарного программного обеспечения для различного рода вычислительной техники.

PHP (Hypertext Preprocessor) – скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений.

URL (Uniform Resource Locator) – система унифицированных адресов электронных ресурсов, или единообразный определитель местонахождения ресурса (файла).

W3C (World Wide Web Consortium) – организация, разрабатывающая и внедряющая технологические стандарты для Всемирной паутины.

1.2 Описание бизнес-ролей

Пользователь – человек, имеющий доступ ко всем возможностям портала и пользующийся ими.

Гость – человек, посетивший интернет-портал музеев Москвы и/или совершивший на нем какое-либо действие.

Администратор – специалист, который отвечает за поддержание работы сайта.

2 Требования к системе

2.1 Требования к системе в целом

2.1.1 Требования к структуре и функционированию системы

Система имеет модульную структуру, включающую в себя следующие модули:

- модуль раздела «Музеи»;
- модуль раздела «Афиша»;
- модуль раздела «Альбом»;
- модуль раздела «Личный кабинет»;
- модуль работы с базой данных;
- модуль раздела «Форум»;
- модуль раздела «Новости».

Система должна выполнять следующие функции:

- осуществление общепортального поиска;
- мониторинг посещений портала;
- обработка трафика большого объема;
- поддержка пользователей в чате тех. поддержки;
- информирование о сбоях.

2.1.2 Требования к численности и квалификации персонала системы и режиму его работы

Для поддержания сайта и эксплуатации веб-интерфейса системы управления сайтом от персонала не должно требоваться специальных технических навыков, знания технологий или программных продуктов, за исключением общих навыков работы с персональным компьютером и стандартным веб-браузером (например, MS Internet Explorer 7.0 или выше).

Режим работы администраторов зависит от работы организации, использующей Систему, за исключением работы по устранению ошибок ПО, которые были обнаружены в период экспериментальной эксплуатации в нерабочее время.

Режим работы других пользователей не ограничен.

2.1.3 Показатели назначения

Подсистемы, разработанные и доработанные в рамках данного раздела, обязательно должны отвечать следующим требованиям:

1. Время на полный запуск (или перезапуск) системы и компонентов системы должно составлять не более 15 минут.
2. Коэффициент юзабилити не менее 85%.
3. Коэффициент интерактивности не менее 88%.
4. Коэффициент достоверности информации не менее 92%.
5. Ответ тех. поддержки на вопрос пользователя не более 30 минут.
6. Модерация форума раз в 12 часов.
7. REST API подсистемы администрирования: 50 запросов в минуту при времени отклика не более трёх секунд.
8. В режиме отправки/приемки сообщений, подсистема должна поддерживать интенсивность минимум 100 запросов в секунду при среднем размере конверта 300 Кб. Интенсивность должна быть обеспечена разработанным SDK.

Требования к аппаратной части и масштабированию для обеспечения перечисленных показателей должны быть определены на этапе технического проектирования.

2.1.4 Требования к надежности

Программное обеспечение не должно выходить из строя более чем на 3 минуты.

Для устойчивости к потере данных необходимо регулярно производить выгрузку хранимой информации.

Надежность требуемого уровня достигается путем комплексного применения организационных и организационно-технических мероприятий. При этом необходимо использовать соответствующие требованиям программно-аппаратные средств. В частности, можно использовать следующие базовые подходы:

- системное и базовое ПО и технические средства, соответствующие классу решаемой задачи;
- четкое соблюдение правил эксплуатации, а также регламентных сроков обслуживания используемых программно-аппаратных средств;
- допуск к информационной системе только пользователей, прошедших предварительное обучение.

2.1.5 Требования к безопасности

Безопасность данных пользователей должна обеспечиваться шифрованием, а также обеспечением устойчивости программно-технических средств к возможным кибератакам.

2.1.6 Требования к эргономике и технической эстетике

Взаимодействие пользователей с прикладным программным обеспечением, входящим в состав системы должно осуществляться посредством визуального графического интерфейса (GUI). Интерфейс системы должен быть понятным и удобным, не должен быть перегружен графическими элементами и должен обеспечивать быстрое отображение экранных форм.

2.1.7 Требования к транспортабельности для подвижных АС

Требования к транспортабельности не предъявляются.

2.1.8 Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы

Техническим обслуживанием, ремонтом и хранением сервера АС занимаются сетевые инженеры-техники, специалисты по серверным и сетевым технологиям, а также мастера по ремонту компьютерного и другого технического оборудования.

2.1.9 Требования к защите информации от несанкционированного доступа

При работе с системой необходимо, чтобы данные могли быть восстановлены в случае потери, информация компании и пользователей была защищена от доступа или модификации несанкционированными лицами.

2.1.10 Требования по сохранности информации при авариях

Серверное программное обеспечение системы должно восстанавливать свое функционирование при перезапуске аппаратных средств. Для обеспечения сохранности данных требуется предусмотреть резервное копирование.

2.1.11 Требования к защите от влияния внешних воздействий

Требование к защите от влияния внешних воздействий не предъявляются.

2.1.12 Требования к патентной чистоте

Требования к патентной чистоте не предъявляются.

2.1.13 Требования по стандартизации и унификации

Для реализации статических страниц и шаблонов должны использоваться языки HTML и CSS. Исходный код должен разрабатываться в соответствии со стандартами W3C (HTML 5). Для реализации интерактивных элементов клиентской части должны использоваться языки JavaScript. Для реализации динамических страниц должен использоваться язык PHP.

2.1.14 Дополнительные требования

Дополнительные требования не предъявляются.

2.2 Требования к функциям (задачам), выполняемым системой

Таблица 2.1 – Требования к функциям, выполняемым системой

Функция	Задача
Обработка трафика большого объема	Запись данных в БД
	Выгрузка данных в оперативную память
	Графическое отображение данных
Информирование о сбоях	Отправление уведомлений о сбое
Работа с пользователями	Регистрация пользователей
	Авторизация пользователей
	Аутентификация пользователей
	Использование личного кабинета
	Техническая поддержка
Функция поиска и просмотр	Возможность поиска определенного музея
	Возможность просмотра всех доступных музеев
Обработка, хранение и поддержка БД	Создание резервных копий в соответствии с графиком

2.3 Требования к видам обеспечения

2.3.1 Требования к математическому обеспечению системы

Математическое обеспечение системы должно обеспечивать реализацию перечисленных в данном ТЗ функций, а также выполнение операций конфигурирования, программирования, управления базами данных и документирования. Алгоритмы должны быть разработаны с учетом возможности получения некорректной входной информации и предусматривать соответствующую реакцию на такие события.

2.3.2 Требования к информационному обеспечению системы

Состав, структура и способы организации данных в системе должны быть определены на этапе технического проектирования.

Данные, используемые системой, должны храниться в реляционной СУБД. Структура базы данных определяется с учетом особенностей внутренней модели системы принятия решений.

Информационный обмен между серверной и клиентской частями системы должен осуществляться по протоколу HTTP.

2.3.3 Требования к лингвистическому обеспечению системы

Интернет-портал музеев Москвы должен быть реализован на русском и английском языках. Должна быть предусмотрена возможность переключения между русским и английским языками через настройки внутри системы. Система ввода-вывода должна поддерживать английский и русский языки.

2.3.4 Требования к программному обеспечению системы

Программное обеспечение клиентской части должно удовлетворять следующим требованиям:

- веб-браузер: Internet Explorer 10.0 и выше, или Firefox 10.0 и выше, или Opera 12 и выше, или Safari 14 и выше, или Chrome 88 и выше;
- включенная поддержка JavaScript и cookies.

2.3.5 Требования к техническому обеспечению системы

Платформа, на которой будет развернута серверная часть системы, должна удовлетворять следующим минимальным требованиям:

- не менее 4 GB оперативной памяти;
- не менее 500 GB свободного места на жестком диске;
- ОС на базе Linux или ОС Windows;
- поддерживаемый протокол передачи данных HTTP / HTTPS, скорость передачи данных 20 Мбит/с;
- процессор с тактовой частотой не менее 4.6 GHz.

2.3.6 Требования к метрологическому обеспечению системы

Требования к метрологическому обеспечению не предъявляются.

2.3.7 Требования к организационному обеспечению системы

Требования к организационному обеспечению не предъявляются.

2.3.8 Требования к методическому обеспечению системы

Необходимо разработать несколько типов руководств:

- руководство пользователя для администраторов ресурса;
- руководство пользователя для клиентов сервиса.

Приложение В. Пример выполнения практической работы № 3

Введение

В 2021 году по Москве открыто большое количество музеев разных направленностей и их число только увеличивается. Музеи приносят большую пользу людям: обеспечивают культурное просвещение и духовное развитие, предоставляют возможность изучать историю по картинам, а также историю развития искусства, вдохновляют, позволяют найти новых знакомых и обладают другими полезными свойствами. Поэтому появляются интернет-порталы со всеми имеющимися музеями в городе, чтобы каждый мог выбрать для себя тот, который ему захотелось бы посетить. Интернет-порталы музеев помогают посетителям таких порталов найти график работы музея, анонсы выставок, описание достопримечательностей и другую ценную информацию.

Информационная система «Интернет-портал музеев Москвы» спроектирована, чтобы уменьшить время поиска сведений по музеям, пообщаться с единомышленниками и быть в курсе последних новостей о временных выставках, новых музеях и новых экспонатах.

Целью практической работы является формирование требований к описанной выше системе. Задаaniem практической работы является описание объекта автоматизации, формулировка основных задач автоматизации объекта, описание основных параметров проектируемой информационной системы, описание путей достижения целей. Кроме того, необходимо сформулировать требования к информационной системе.

1 Общие сведения

1.1 Полное наименование системы и ее условное обозначение

Наименование системы: Moscow museums.

Условное обозначение: ММ.

1.2 Номер договора

Шифр темы: АИС-ММ.

Номер контракта: №1/11-11-11-001 от 09.02.2023.

1.3 Наименование организаций – Заказчика и Разработчика

Заказчиком системы является РТУ МИРЭА.

Адрес заказчика: Проспект Вернадского, д. 78

Разработчиком системы является ООО “Садовая”.

1.4 Основания для разработки системы

Работа по созданию системы популяризации культурного наследия и обеспечения доступа к информации по музеям Москвы.

1.5 Плановые сроки начала и окончания работы по созданию системы

Плановый срок начала работ по созданию системы ИП музеев Москвы – 16 февраля 2023 года.

Плановый срок окончания работ по созданию системы ИП музеев Москвы – 25 мая 2023 года.

1.6 Источники и порядок финансирования работ

Собственные средства разработчика.

1.7 Порядок оформления и предъявления заказчику результатов работ по созданию системы

Результаты работ передаются Заказчику в порядке, определенном контрактом в соответствии с Календарным планом работ контракта на основании Актов сдачи-приемки выполненных работ (этапа работ).

Документация ММ передается на бумажных (два экземпляра, один экземпляр после подписания Заказчиком должен быть возвращен Исполнителю) и на

машинных носителях (DVD) (в двух экземплярах). Текстовые документы, передаваемые на машинных носителях, должны быть представлены в форматах PDF.

Все материалы передаются с сопроводительными документами Исполнителя.

1.8 Перечень нормативно-технических документов, методических материалов, использованных при разработке ТЗ

При разработке автоматизированной системы и создании проектно-эксплуатационной документации Исполнитель должен руководствоваться требованиями следующих нормативных документов:

– ГОСТ 19.106-78. Единая система программной документации. Требования к программным документам, выполненным печатным способом.

– ГОСТ 34.602 – 2020 Техническое задание на создание автоматизированной системы

– ГОСТ Р 59793-2021. Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.

– ГОСТ 34.201–2020. Информационные технологии. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем.

– ГОСТ Р 59795-2021. Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов.

1.9 Определения, обозначения и сокращения

Альбом – фотографии пользователей, сделанные в музеях.

Афиша – новости о выставках, галереях, выставочных залах, новых музеях, скидках в музей и о конференциях, проходящих в музеях.

БД (База Данных) – представленная в объективной форме совокупность самостоятельных материалов, систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ).

ИП (Интернет-Портал) – многофункциональная площадка с разнообразным интерактивным сервисом, включающая в себя обширные возможности и услуги, в том числе путём предоставления пользователям ссылок на другие сайты.

ИС (Информационная Система) – система, предназначенная для хранения, поиска и обработки информации, и соответствующие организационные ресурсы (человеческие, технические, финансовые и т. д.), которые обеспечивают и распространяют информацию.

Музеи – каталог музеев Москвы, с описанием, местонахождением и графиком работы.

Открытый музейный форум – раздел на сайте, предназначенный для общения пользователей, обмена знаниями и проведения дискуссий.

СУБД (Система Управления Базами Данных) – совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.

CSS (Cascading Style Sheets) – формальный язык описания внешнего вида документа, написанного с использованием языка разметки.

HTML (Hyper Text Markup Language) – стандартизированный язык разметки веб-страниц во Всемирной паутине.

MS (Microsoft) – одна из крупнейших транснациональных компаний по производству проприетарного программного обеспечения для различного рода вычислительной техники.

PHP (Hypertext Preprocessor) – скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений.

URL (Uniform Resource Locator) – система унифицированных адресов электронных ресурсов, или единообразный определитель местонахождения ресурса (файла).

W3C (World Wide Web Consortium) – организация, разрабатывающая и внедряющая технологические стандарты для Всемирной паутины.

1.10 Описание бизнес-ролей

Пользователь – человек, имеющий доступ ко всем возможностям портала и пользующийся ими.

Гость – человек, посетивший интернет-портал музеев Москвы и/или совершивший на нем какое-либо действие.

Администратор – специалист, который отвечает за поддержание работы сайта.

2 Назначение и цели создания (развития) системы

2.1. Назначение системы

Система интернет-портала музеев Москвы предназначена для популяризации культурного наследия и культурного просвещения.

2.2 Цели создания системы

Основными целями создания ИС являются:

- популяризация культурного наследия;
- культурное просвещение общества;
- информирование о последних событиях, связанных с музеями;
- предоставление возможности пользователям общаться, обмениваться опытом и обсуждать выставки и музеи.

3 Характеристика объекта автоматизации

3.1 Краткие сведения об объекте автоматизации

Объектом автоматизации является ИП музеев Москвы. В независимости от рода занятий пользователя.

3.2 Сведения об условиях эксплуатации объекта автоматизации

Условия эксплуатации комплекса технических средств Системы должны соответствовать условиям эксплуатации группы 2 ГОСТ 21552-84 «Средства вычислительной техники. Общие технические требования, приемка, методы испытаний, маркировка, упаковка, транспортировка, хранение».

Условия эксплуатации персональных компьютеров Системы соответствуют Гигиеническим требованиям к видео-дисплейным терминалам, персональным электронно-вычислительным машинам и организации работы (Санитарные правила и нормы. СанПиН 2.2.2.542-96).

Исполнитель должен проверить соблюдение условий эксплуатации комплекса технических средств на этапе технического проектирования.

4 Требования к системе

4.1 Требования к системе в целом

4.1.1 Требования к структуре и функционированию системы

Система имеет модульную структуру, включающую в себя следующие модули:

- модуль раздела «Музеи»;
- модуль раздела «Афиша»;
- модуль раздела «Альбом»;
- модуль раздела «Личный кабинет»;
- модуль работы с базой данных;
- модуль раздела «Форум»;
- модуль раздела «Новости».

Система должна выполнять следующие функции:

- осуществление общепортального поиска;
- мониторинг посещений портала;
- обработка трафика большого объема;
- поддержка пользователей в чате тех. поддержки;
- информирование о сбоях.

4.1.2 Требования к численности и квалификации персонала системы и режиму его работы

Для поддержания сайта и эксплуатации веб-интерфейса системы управления сайтом от персонала не должно требоваться специальных технических навыков, знания технологий или программных продуктов, за исключением общих навыков работы с персональным компьютером и стандартным веб-браузером (например, MS Internet Explorer 7.0 или выше).

Режим работы администраторов зависит от работы организации, использующей проектируемую систему, за исключением работы по устранению ошибок ПО, которые были обнаружены в период экспериментальной эксплуатации в нерабочее время.

Режим работы других пользователей не ограничен.

4.1.3 Показатели назначения

Доработанные и разработанные подсистемы должны обеспечивать следующие показатели назначения:

1. Время на полный запуск (или перезапуск) системы и компонентов системы должно составлять не более 15 минут.
2. Коэффициент юзабилити не менее 85%.
3. Коэффициент интерактивности не менее 88%.
4. Коэффициент достоверности информации не менее 92%.
5. Ответ тех. поддержки на вопрос пользователя не более 30 минут.
6. Модерация форума раз в 12 часов.
7. REST API подсистемы администрирования: 50 запросов в минуту при времени отклика не более трёх секунд.
8. Режим отправки / приёмки сообщений пользователя на основе разработанного SDK с интенсивностью не менее 100 запросов в секунду при среднем размере конверта 300 Кб.

Требования к аппаратной части и масштабированию для обеспечения перечисленных показателей должны быть определены на этапе технического проектирования.

4.1.4 Требования к надежности

Программное обеспечение не должно выходить из строя более чем на 3 минуты.

Для устойчивости к потере данных необходимо регулярно производить выгрузку хранимой информации.

Уровень надежности должен достигаться согласованным применением организационных, организационно-технических мероприятий и программно-аппаратных средств.

Надежность должна обеспечиваться за счет:

- применения технических средств, системного и базового программного обеспечения, соответствующих классу решаемых задач;

- соблюдения правил эксплуатации и технического обслуживания программно-аппаратных средств;
- предварительного обучения пользователей.

4.1.5 Требования к безопасности

Безопасность данных пользователей должна обеспечиваться шифрованием, а также обеспечением устойчивости программно-технических средств к возможным кибератакам.

4.1.6 Требования к эргономике и технической эстетике

Взаимодействие пользователей с прикладным программным обеспечением, входящим в состав системы должно осуществляться посредством визуального графического интерфейса (GUI). Интерфейс системы должен быть понятным и удобным, не должен быть перегружен графическими элементами и должен обеспечивать быстрое отображение экранных форм.

4.1.7 Требования к транспортабельности для подвижных АС

Требования к транспортабельности не предъявляются.

4.1.8 Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы

Техническим обслуживанием, ремонтом и хранением сервера АС занимаются сетевые инженеры-техники, специалисты по серверным и сетевым технологиям, а также мастера по ремонту компьютерного и другого технического оборудования.

4.1.9 Требования к защите информации от несанкционированного доступа

При работе с системой необходимо, чтобы данные могли быть восстановлены в случае потери, информация компании и пользователей была защищена от доступа или модификации несанкционированными лицами.

4.1.10 Требования по сохранности информации при авариях

Серверное программное обеспечение системы должно восстанавливать свое функционирование при перезапуске аппаратных средств. Для обеспечения сохранности данных требуется предусмотреть резервное копирование.

4.1.11 Требования к защите от влияния внешних воздействий

Требование к защите от влияния внешних воздействий не предъявляются.

4.1.12 Требования к патентной чистоте

Требования к патентной чистоте не предъявляются.

4.1.13 Требования по стандартизации и унификации

Для реализации статических страниц и шаблонов должны использоваться языки HTML и CSS. Исходный код должен разрабатываться в соответствии со стандартами W3C (HTML 5). Для реализации интерактивных элементов клиентской части должны использоваться языки JavaScript. Для реализации динамических страниц должен использоваться язык PHP.

4.1.14 Дополнительные требования

Дополнительные требования не предъявляются.

4.2 Требования к функциям (задачам), выполняемым системой

Таблица 4.1 – Требования к функциям, выполняемым системой

Функция	Задача
Обработка трафика большого объема	Запись данных в БД
	Выгрузка данных в оперативную память
	Графическое отображение данных
Информирование о сбоях	Отправление уведомлений о сбое
Работа с пользователями	Регистрация пользователей
	Авторизация пользователей
	Аутентификация пользователей
	Использование личного кабинета
	Техническая поддержка
Функция поиска и просмотр	Возможность поиска определенного музея
	Возможность просмотра всех доступных музеев
Обработка, хранение и поддержка БД	Создание резервных копий в соответствии с графиком

4.3 Функциональная структура системы

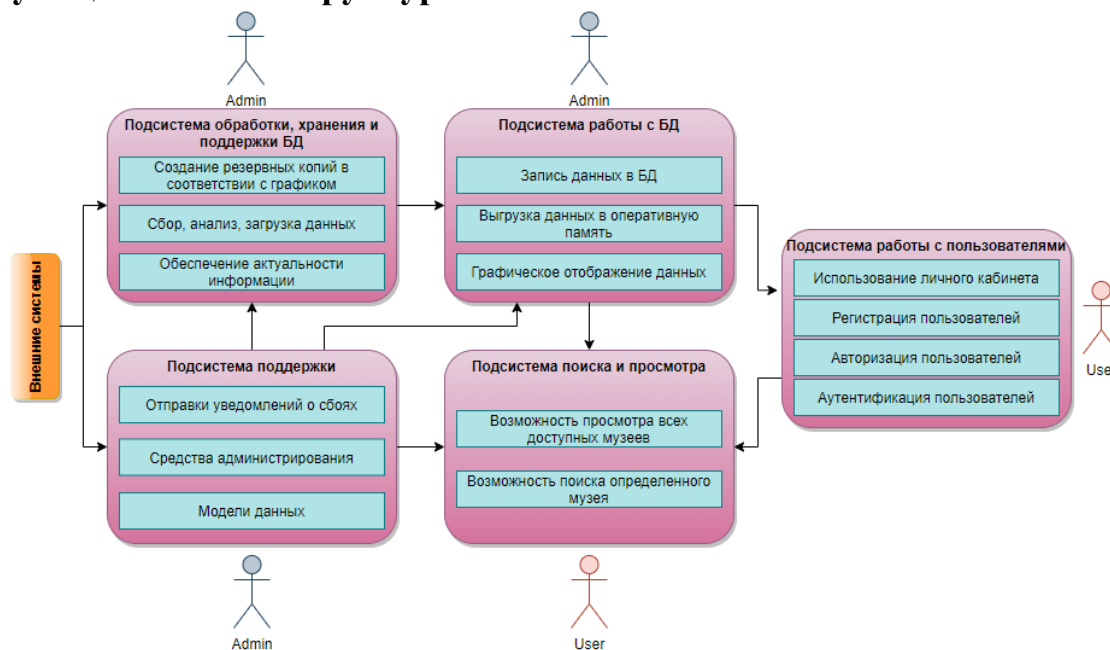


Рисунок 1 – Структурная диаграмма

Связь «Подсистема поддержки – Подсистема поиска и просмотра» определяет процесс отслеживания администратором отображения на портале графических элементов, обеспечения релевантного поиска по музеям.

Связь «Подсистема работы с пользователями – Подсистема поиска и просмотра» определяет доступность для пользователей определенных действий на портале.

Связь «Подсистема работы с БД – Подсистема работы с пользователями» определяет процесс добавления / изменения данных в БД при регистрации, авторизации пользователей и изменения ими информации в личном кабинете.

Связь «Подсистема обработки, хранения и поддержки БД – Подсистема работы с БД» определяет работу администратора с данными в БД по всем пользователям на уровне сервера.

Связь «Подсистема поддержки – Подсистема обработки, хранения и поддержки БД» определяет работу администратора при сбоях в БД.

Связь «Подсистема работы с БД – Подсистема поиска и просмотра» определяет скорость загрузки БД при поиске информации пользователем.

Связь «Подсистема поддержки – Подсистема работы с БД» определяет процесс передачи уведомлений о сбоях в БД и дальнейшее устранение неполадок администратором.

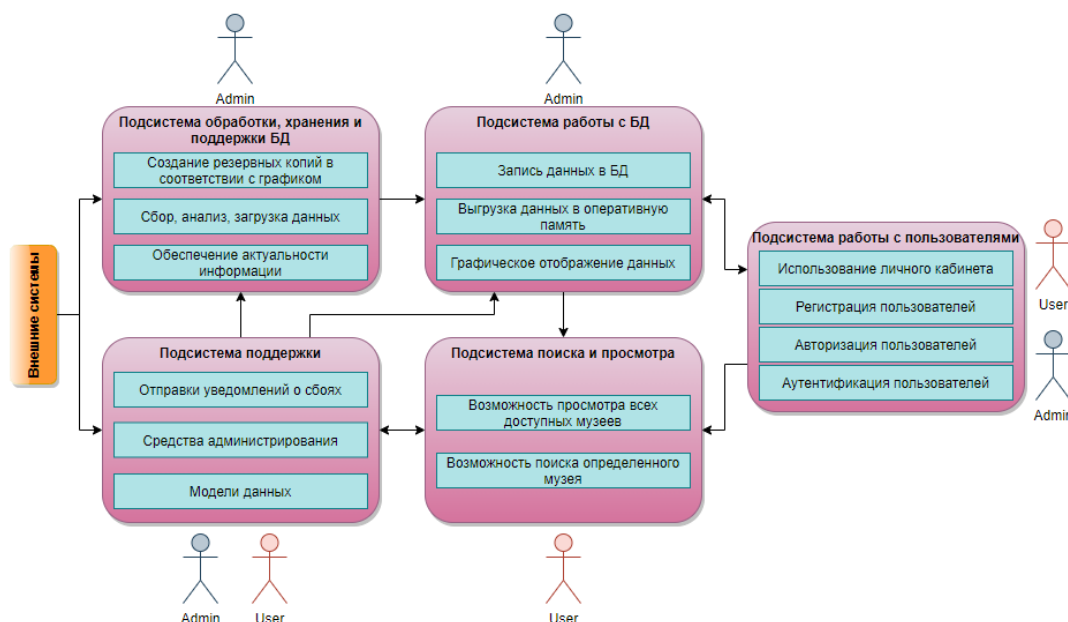


Рисунок 2 – Альтернативная структурная диаграмма

Отличие второй функциональной структуры системы (рис. 2) от первой (рис. 1) заключается в том, что в подсистеме работы с пользователями появляется администратор и в подсистеме поддержки – пользователь.

При этом образуются двунаправленные связи между подсистемами «поддержки» и «поиска и просмотра», а также подсистемами «работы с БД» и «работы с пользователями».

Во втором варианте структурной диаграммы пользователь при неудачном поиске музея или некорректной работе ИП в подсистеме поддержки имеет возможность отправлять уведомления о сбоях, минуя администратора. Администратор же теперь не только отслеживает данные в БД, но и может регистрировать / удалять пользователей по необходимости. Затраты на систему увеличатся, так как у администратора появятся новые обязанности, но данная функция увеличит количество посетителей на ИП.

4.4 Требования к видам обеспечения

4.4.1 Требования к математическому обеспечению системы

Математическое обеспечение системы должно обеспечивать реализацию перечисленных в данном ТЗ функций, а также выполнение операций конфигурирования, программирования, управления базами данных и документирования. Алгоритмы должны быть разработаны с учетом возможности получения некорректной входной информации и предусматривать соответствующую реакцию на такие события.

4.4.2 Требования к информационному обеспечению системы

Состав, структура и способы организации данных в системе должны быть определены на этапе технического проектирования.

Данные, используемые системой, должны храниться в реляционной СУБД. Структура базы данных определяется с учетом особенностей внутренней модели системы принятия решений.

Информационный обмен между серверной и клиентской частями системы должен осуществляться по протоколу HTTP.

4.4.3 Требования к лингвистическому обеспечению системы

Интернет-портал музеев Москвы должен быть реализован на русском и английском языках. Должна быть предусмотрена возможность переключения между русским и английским языками через настройки внутри системы. Система ввода-вывода должна поддерживать английский и русский языки.

4.4.4 Требования к программному обеспечению системы

Программное обеспечение клиентской части должно удовлетворять следующим требованиям:

- веб-браузер: Internet Explorer 10.0 и выше, или Firefox 10.0 и выше, или Opera 12 и выше, или Safari 14 и выше, или Chrome 88 и выше;
- включенная поддержка JavaScript и cookies.

4.4.5 Требования к техническому обеспечению системы

Платформа, на которой будет развернута серверная часть системы, должна удовлетворять следующим минимальным требованиям:

- не менее 4 GB оперативной памяти;
- не менее 500 GB свободного места на жестком диске;
- ОС на базе Linux или ОС Windows;
- поддерживаемый протокол передачи данных HTTP / HTTPS, скорость передачи данных 20 Мбит/с;
- процессор с тактовой частотой не менее 4.6 GHz.

4.4.6 Требования к метрологическому обеспечению системы

Требования к метрологическому обеспечению не предъявляются.

4.4.7 Требования к организационному обеспечению системы

Требования к организационному обеспечению не предъявляются.

4.4.8 Требования к методическому обеспечению системы

Необходимо разработать несколько типов руководств:

- руководство пользователя для администраторов ресурса;
- руководство пользователя для клиентов сервиса.

5 Состав и содержание работ по созданию (развитию) системы

Разработка системы предполагается по укрупненному календарному плану, приведенному в таблице 5.1.

Таблица 5.1 – Календарный план работ по созданию АС ММ

Этапы работ	Содержание работ	Сроки
1. Исследование и обоснование создания АС	1.1. Обследование (сбор и анализ данных) автоматизированного объекта, включая сбор сведений о зарубежных и отечественных аналогах	16.02.2023- 23.02.2023
2. Составление технического задания	2.1. Разработка функциональных и нефункциональных требований к системе	24.02.2023- 28.02.2023
3. Эскизное проектирование	3.1. Разработка предварительных решений по выбранному варианту АС и отдельным видам обеспечения	01.03.2023- 09.03.2023
4. Техническое проектирование	4.1. Разработка диаграмм	10.03.2023- 17.03.2023
	4.2. Разработка макетов интерфейса	18.03.2023- 31.03.2023
5. Разработка программной части	5.1. Разработка модуля «Музеи»	01.04.2023- 25.04.2023
	5.2. Разработка модуля «Афиша»	
	5.3. Разработка модуля «Альбом»	
	5.4. Разработка модуля «Личный кабинет»	
	5.5. Разработка модуля работы с БД	
	5.6. Разработка модуля «Форум»	
	5.7. Разработка модуля «Новости»	
6. Предварительные комплексные испытания	6.1. Проверка работоспособности системы в условиях, приближенных к реальным	26.04.2023- 03.05.2023
7. Опытная эксплуатация	7.1. Эксплуатация с привлечением небольшого количества участников	04.05.2023- 10.05.2023
	7.2. Устранение замечаний, выявленных при эксплуатации, АС	11.05.2023- 15.05.2023
8. Ввод в промышленную эксплуатацию	8.1. Приемка АС в промышленную эксплуатацию (внедрение АС)	16.05.2023- 25.05.2023

6 Порядок контроля и приемки системы

В соответствии с разделом 5 необходимо на каждой стадии создания системы установить контроль и приемку результатов работ.

На стадии 5 происходит прием готовой версии программного продукта (модели), а остальные результаты работ представляются в виде документов согласно таблице 5.1.

Приемка этапа включает в себя рассмотрение и оценку объема работ и предоставленной технической документации в соответствии с требованиями технического задания.

Организацию и проведение приемки системы должен осуществлять заказчик, а приемка системы должна производиться только после того, как будут выполнены все задачи системы.

Заказчик обязан предоставить материальную часть (технические средства), проектную документацию и специально выделенный персонал.

Последним этапом при приемке системы является составление акта приемки.

7 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие

Для обеспечения готовности объекта к вводу системы в действие провести комплекс мероприятий:

- приобрести компоненты программного обеспечения, заключить договора на их лицензионное использование;
- завершить работы по установке технических средств;
- провести диагностику устойчивости сети к нагрузкам;
- провести обучение сотрудников.

7.1 Приведение поступающей в систему информации к виду, пригодному для обработки с помощью ЭВМ

Информация вводится пользователем в разработанные экранные формы компонентов системы.

7.2 Изменения, которые необходимо осуществить в объекте автоматизации

Изменений не требуется.

7.3 Создание условий функционирования объекта автоматизации, при которых гарантируется соответствие создаваемой системы требованиям, содержащимся в ТЗ

Для функционирования создаваемой системы требуется платформа, технические характеристики которой соответствуют предъявленным.

7.4 Создание необходимых для функционирования системы подразделений и служб

Для функционирования системы не требуется дополнительных подразделений и служб.

7.5 Сроки и порядок комплектования штатов и обучения персонала

Комплектование штатов подразделений и служб, необходимых для функционирования системы, а также подготовка их сотрудников должны быть завершены до начала опытной эксплуатации системы.

8 Требования к документированию

Проектная документация должна быть разработана в соответствии с ГОСТ 34.201-2020 и ГОСТ 7.32-2017.

Отчетные материалы должны включать в себя текстовые материалы (представленные в виде бумажной копии и на цифровом носителе в формате MS Word) и графические материалы.

Предоставить документы:

- 1) схема функциональной структуры автоматизируемой деятельности;
- 2) описание технологического процесса обработки данных;
- 3) описание информационного обеспечения;
- 4) описание программного обеспечения АС;
- 5) схема логической структуры БД;
- 6) руководство пользователя;
- 7) описание контрольного примера (по ГОСТ 24.102);
- 8) протокол испытаний (по ГОСТ 24.102).

9 Источники разработки

- ГОСТ 34.602-2020. Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.
- ГОСТ Р 59793-2021. Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.
- ГОСТ 34.201-2020. Информационные технологии. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем.
- ГОСТ Р 59795-2021. Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов.
- ГОСТ 19.106-78. Единая система программной документации. Требования к программным документам, выполненным печатным способом.
- ГОСТ 19.105-78. Единая система программной документации. Общие требования к программным документам.

Приложение Г. Пример выполнения практической работы № 4

Помимо контекстной диаграммы необходимо также создать краткое описание объекта автоматизации (см. задание данной практической работы) и описание функциональных блоков диаграммы. Ниже будет представлен пример описания объекта автоматизации интернет-портала музеев Москвы.

Введение

Для проектирования была выбрана информационная система автоматизации поиска интересующих потребителя музеев, обсуждения исторических тем, временных выставок, новых музеев и новых экспонатов, добавления новостей и/или фотографии в Альбом, а также возможности просматривать цены и покупать билеты в музеи. Название системы «Интернет-портал музеев Москвы». Система создается для популяризации культурного наследия и культурного просвещения.

Цель создания ИС

Целью создания ИС «Интернет-портал музеев Москвы» является увеличение заинтересованности общества в культурном просвещении.

По определению ИС: «Информационная система – это сложный программный комплекс, который способен собирать, сохранять, обрабатывать и выдавать по запросу пользователя информацию». Проектируемая ИС полностью удовлетворяет всему перечню требований, указанных в определении, т.к.:

1. Сайт собирает информацию о музеях из Интернета, проверяя достоверность информации. Также системой собираются персональные данные участников платформы.
2. Хранит полученную информацию в базе данных.
3. Информация из подпунктов выше обрабатывается, на основе чего при помощи специальных алгоритмов пользователь при каждом последующем поиске получает более релевантный ответ.
4. Доступ пользователей к огромному количеству информации на сайте (музеям, альбому, новостям и т. д.).

Краткое описание

ИС «Интернет-портал музеев Москвы» представлена в виде сайта. Сайт является удобным интернет-сервисом, представляющим информацию о музеях, выставках и других тематиках, связанных с музеями. Для комфортного и круг-

лосуточного доступа сайт также адаптирован для мобильных устройств и представлен на русском и английском языках.

Одно из важных достоинств проектируемой ИС – большой функционал для зарегистрированных пользователей. Авторизованные пользователи могут писать на форуме, выкладывать фотографии, публиковать новости. Это даст дополнительный интерес пользователям к регистрации на платформе, которая позволит им получить полный доступ к функциям портала.

Кроме того, платформа позволяет пользователю просматривать цены и покупать билеты в музеи, при условии того, что данный музей принял правила платформы и подключен к системе.

Способ создания ИС

В качестве способа определения требований была выбрана методология «последовательных приближений», которая основана на том, что все расчеты и графические построения, связанные с определением основных элементов, разбиваются на несколько более мелких элементов, в которых происходит их уточнение. Данный метод также хорошо сочетается с нотацией IDEF0, которая основана на декомпозиции каждого блока на более мелких с уточнением деталей.

Средства создания ИС

В качестве средств создания ИС были использованы языки программирования JavaScript, HTML, CSS, СУБД MySQL и сервис для развёртывания сервера Apache HTTP Server. Для моделирования проектируемой ИС будет использоваться нотация IDEF0 в CASE-средстве Ramus Educational.

Проектирование контекстной диаграммы функциональной модели ИС

Была спроектирована контекстная диаграмма А-0 в нотации IDEF0.

В качестве управления были выбраны следующие нормативные и правовые документы:

1. Закон о персональных данных.
2. Политика сайта.
3. Алгоритмы для обработки информации и выдачи релевантного ответа.

В качестве входящих информационных потоков, которые подлежат обработке и преобразованию в процессе работы ИС, были указаны:

1. Персональные данные пользователя.

2. Запрос пользователя.

В качестве механизмов (ресурсов, выполняющих работу) были выделены:

1. Администратор.
2. Гость (не имеющий полного доступа к функциям портала).
3. Авторизованный пользователь сайта.

В качестве выходов получены следующие информационные элементы:

1. Информация о доступных билетах и ценах.
2. Информация о музеях (график работы, расположение и т. д.).

На рисунке 1 представлена контекстная диаграмма проектируемой информационной системы.

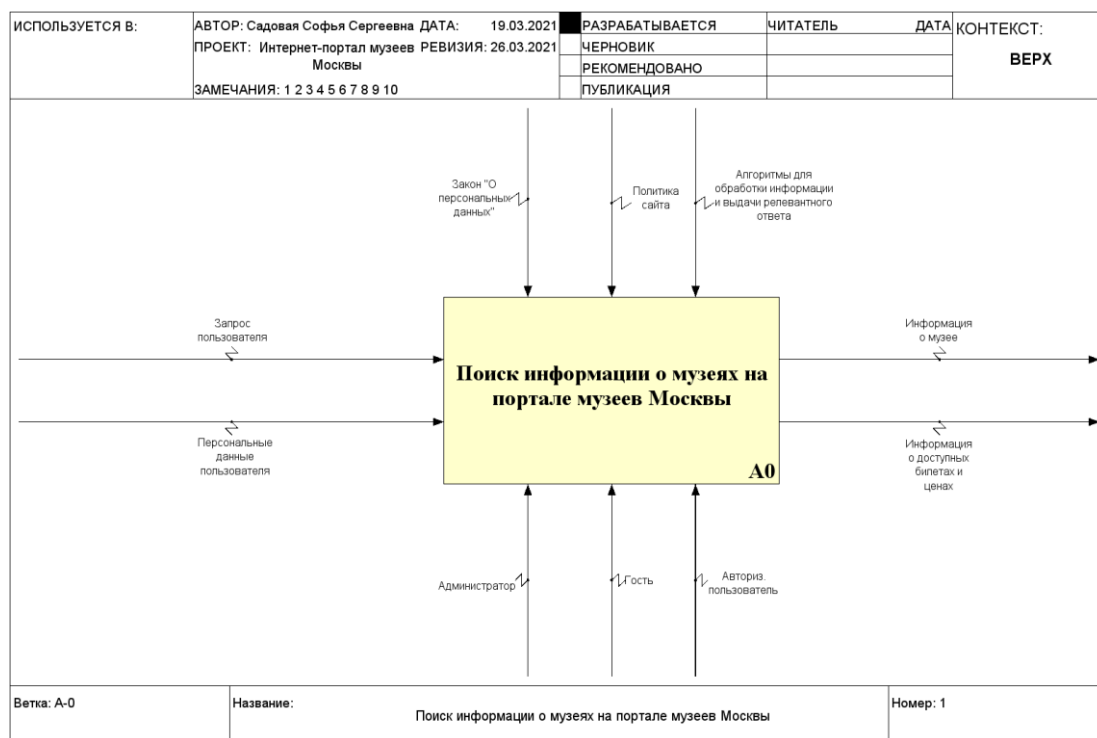


Рисунок 1 – Контекстная диаграмма процесса поиска информации о музеях на портале музеев Москвы

Вывод

В результате выполнения данной практической работы определена цель, способ и средства создания ИС, составлено краткое описание, а также смоделирована контекстная диаграмма A-0 в нотации IDEF0.

Приложение Д. Пример выполнения практической работы № 9

Описание ЭСЕ

Элементарная семантическая единица (ЭСЕ) – неделимая единица информации, использующаяся в ИС. ЭСЕ представляет собой завершенную контекстную конструкцию, вызываемую в результате поиска по различным атрибутам или в результате тех или иных команд в виде отклика или отчета. В случае исследования настоящей системы за элементарную семантическую единицу была выбрана одна из характеристик поиска, а именно сертификатов, возвращаемых на запрос. В нашем примере эта величина меняется случайным образом в пределах от 100000 до 200000 [сертификатов].

Наполнение системы

Проектируемая информационная система может быть наполнена практически любым количеством элементов базы данных. Их количество ограничиваются только параметрами сервера.

В рамках данной система была наполнена работы Система была наполнена 100 ЭСЕ. В рамках ограничений объема данной работы, невозможно привести полный перечень всех записей ЭСЕ, поэтому пример первых десяти записей приведен в таблице 1.

Структуризация ведется по количеству сертификатов, возвращаемых на запрос.

Таблица 1 – Список элементарных семантических единиц

Наименование	Параметр
Курс	154248
Курс	177237
Курс	108191
Курс	178588
Курс	190708
Курс	176805
Курс	182143
Курс	163235
Курс	159315
Курс	164788

Математические расчеты

Для дальнейшего исследования проектируемой ИС необходимо рассчитать вероятности, с которыми ЭСЕ принимает то или иное значение. Для оценки этих вероятностей было принято решение разбить весь диапазон значений на 10 дискретных величин с шагом в 9721,4. Расчеты ведутся с помощью формулы $P(\xi)=n/N$, где n – благоприятное число исходов (в данном случае число сертификатов, попадающих в данный диапазон), а N – общее число исходов. В таблице 2 приведены возможные значения, принимаемые ЭСЕ и их вероятности.

Таблица 2 – Ряд распределения

№	x	P(x)
1	105649,7	14/100=0.14
2	115371,1	11/100=0.11
3	125092,5	11/100=0.11
4	134813,9	6/100=0.06
5	144535,3	6/100=0.06
6	154256,7	8/100=0.08
7	163978,1	12/100=0.12
8	173699,5	8/100=0.08
9	183420,9	17/100=0.17
10	193142,3	7/100=0.07

Расчет математического ожидания информационного блока системы

Математическим ожиданием случайной величины называется сумма произведений всех возможных значений случайной величины на вероятности этих значений. Рассчитаем математическое ожидание для нашей системы, взяв за случайную величину число сертификатов. Расчёт математического ожидания информационного блока на примере 10 записей выполняется по формуле 1:

$$Mx_i = \sum_{i=0}^n [p_i \cdot x_i] \quad (1)$$

Используя данные, полученные в таблице 2, получаем:

$M(10) = 148618,3$ [сертификатов], следовательно, наиболее вероятное количество сертификатов на запрос находится в районе 148618,3 [сертификатов].

Расчет дисперсии информационного блока системы

$$Dx_i = \sum_{i=0}^n [p_i \cdot (x_i)^2] - [\sum_{i=0}^n (p_i \cdot x_i)]^2 \quad (2)$$

Используя данные, полученные в таблице 2, получаем:

$$D(10) = 879242467,3 \text{ [сертификатов}^2\text{]}$$

Расчет среднеквадратического отклонения

$$\sigma_{xi} = \sqrt{Dxi} = \sqrt{879242467,3}$$

$$\sigma_{xi} = 29652,0 \text{ [сертификатов]}$$

Расчет энтропии системы

Энтропия системы – это сумма произведений вероятностей различных состояний системы на логарифмы этих вероятностей, взятая с обратным знаком (формула 3).

$$H(x) = -\sum_{i=1}^n [p_i \cdot \log_a p_i] \quad (3)$$

За основание логарифма **a** возьмем двоичную систему счисления.

Энтропия фрагмента информационного наполнения в размере 10 ЭСЕ:

Используя данные, полученные в таблице 3, получаем:

$$H(x) = 3,238 \text{ [бит]}$$

Порядок выполнения работы с использованием табличного процессора Excel можно посмотреть в видео: https://youtu.be/o4_OT7IHUYKE

Вывод

В данной практической работе был осуществлен расчет основных характеристик проектируемой ИС, и получены следующие результаты (см. таблицу 3):

Таблица 3 – Параметры проектируемой ИС

Параметр	Значение
Математическое ожидание информационного блока	148618,3 [сертификатов]
Допустимый разброс значений смысловых информационных блоков (дисперсия)	879242467,3 [сертификатов ²]
Среднеквадратическое отклонение	29652,0 [сертификатов]
Энтропия информационного наполнения	3,238 [бит]

Оформление и верстка Ю. Болдырева
Дата подписания к использованию: 14.07.2023
Объем издания: 3,2 Мб. Комплектация: 1 электрон. опт. диск (CD-R)
Тираж 10 экз.



Издательство АНО ДПО «Межрегиональный центр
инновационных технологий в образовании»
610047, г. Киров, ул. Свердлова, 32а, пом. 1003
Тел.: 8-800-222-30-98
<https://mcito.ru/publishing>; e-mail: book@mcito.ru

ISBN 978-5-907743-35-9



9 785907 743359 >