

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
ГОУ ВПО «ЧЕРЕПОВЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт информационных технологий

Е.В. Ершов, В.В. Селивановских, О.Л. Селяничев, Л.Н. Виноградова

**ТЕХНОЛОГИЯ РАЗРАБОТКИ
ДИПЛОМНОГО ПРОЕКТА
С ИСПОЛЬЗОВАНИЕМ CASE-СРЕДСТВ**

Учебное пособие

Череповец
2010

УДК 004.4
ББК 32.973.26-018.1я73
Т38

Рассмотрено на заседании кафедры
программного обеспечения ЭВМ, протокол
№ 4 от 30.12.09 г.

Одобрено на заседании учебно-мето-
дического совета ГОУ ВПО ЧГУ, протокол
№ 6 от 18.02.10 г.

Т38

Технология разработки дипломного проекта с использованием CASE-средств: Учеб. пособие / Е.В. Ершов, В.В. Селивановских, О.Л. Селяничев, Л.Н. Виноградова. – Череповец: ГОУ ВПО ЧГУ, 2010. – 90 с. ISBN 978 – 5 – 85341 – 381 – 8

Данное учебное пособие содержит теоретический материал и комплекс вопросов по разработке дипломного проекта с помощью современных CASE-средств, таких как ВРwin, ERwin, Rational Rose, а также по технико-экономическому обоснованию программного обеспечения и основам безопасности жизнедеятельности.

Описываются цели и задачи выпускной квалификационной работы, раскрывается содержание дипломного проектирования, приводятся требования к оформлению расчетно-пояснительной записки и графической части выпускной квалификационной работы, основные обязанности руководителя проекта и дипломника, график работы над проектом, порядок допуска к защите и описание публичной защиты выпускной квалификационной работы.

Предназначено для студентов направления 230100 «Информатика и вычислительная техника».

Рецензенты: *А.Л. Смылова* – канд. техн. наук, доцент (ГОУ ВПО ЧГУ); *Е.В. Королева* – канд. техн. наук, инженер-программист (ЗАО «Фирма «Стоик»)

© Е.В. Ершов, В.В. Селивановских,
О.Л. Селяничев, Л.Н. Виноградова, 2010
© ГОУ ВПО «Череповецкий государст-
венный университет», 2010

ISBN 978 – 5 – 85341 – 381 – 8

ПРЕДИСЛОВИЕ

Данное учебное пособие содержит описание и порядок выполнения выпускной квалификационной работы (ВКР), которая является завершающим этапом подготовки специалиста. Успешная защита ВКР служит основанием для присвоения выпускнику квалификации инженера по направлению подготовки дипломированного специалиста 654600 «Информатика и вычислительная техника» (специальность 230105 «Программное обеспечение вычислительной техники и автоматизированных систем»).

В процессе выполнения ВКР студент должен проявить самостоятельность, умение обосновывать принимаемые решения и способность решать поставленные перед ним научно-технические задачи. В этом ему помогут современные средства для разработки спецификаций: Erwin, ВРwin и Rational Rose, которые описаны в данном пособии.

Для дипломированного специалиста предусмотрены два вида ВКР:

- дипломный проект;
- дипломная работа.

Раздел 1

НАЗНАЧЕНИЕ И ТЕМАТИКА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

1.1. Цели и задачи дипломного проектирования

Перед студентами в процессе дипломного проектирования стоят следующие цели и задачи:

- систематизация и закрепление теоретических знаний по дисциплинам специальности, применение этих знаний при разработке программного обеспечения;

- развитие навыков самостоятельной работы с учебно-методической и научной литературой, законодательными и нормативными документами;

- овладение студентами методикой исследования, обобщения и логического изложения материала.

В ходе дипломного проектирования студент обязан:

- правильно применять теоретические положения естественнонаучных, общепрофессиональных и специальных дисциплин;

- уметь самостоятельно выявлять и анализировать аналоги и прототипы разрабатываемого программного обеспечения;

- проявлять организаторские способности и навыки делового общения для преодоления реальных трудностей, возникающих при разработке и внедрении новых технических решений.

1.2. Определение тематики выпускной квалификационной работы

Темы ВКР определяются выпускающей кафедрой с учетом современных тенденций развития технологии разработки программного обеспечения в России и за рубежом, предложений специалистов предприятий, а также на основе тематики научно-исследовательских работ, проводимых на профильных кафедрах.

рах. Название темы ВКР должно быть кратким, отражающим направление разработки, а также, как правило, указывается объект (предприятие, акционерное общество, банк и т.д.), где выполнялась работа.

Темы ВКР доводятся до сведения студентов-выпускников, которые имеют право выбора или могут предложить свою тему с обоснованием целесообразности ее разработки. Темы ВКР в большинстве случаев являются продолжением курсовых работ (проектов), выполненных студентом в процессе освоения образовательной программы.

Закрепление за выпускником темы ВКР, научного руководителя, рецензента и консультантов (если они есть) оформляется приказом ректора перед направлением студента на преддипломную практику.

Раздел 2

СТРУКТУРА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

ВКР содержит следующие разделы:

- титульный лист;
- задание на ВКР;
- аннотацию;
- оглавление;
- введение;
- основную часть;
- технико-экономическое обоснование разработки;
- анализ мероприятий по безопасности жизнедеятельности и обеспечению экологической чистоты;
- заключение;
- приложения;
- графический материал.

2.1. Титульный лист

Титульный лист должен содержать: наименование учебного заведения и структурного подразделения (институт, кафедра), в котором обучается студент, шифр и наименование специальности, название темы ВКР, Ф. И. О. студента и его подпись, согласование допуска к защите с директором института, зав. выпускающей кафедрой, руководителем и консультантами, нормоконтролёром, название города и год окончания учебного заведения, где выполнялась работа.

Форма титульного листа представлена в приложении 1 данного учебного пособия.

2.2. Задание на ВКР

После утверждения темы студент совместно с руководителем составляет задание на выполнение дипломного проекта (работы) в двух экземплярах на бланке, форма которого приведена в приложении 2 данного учебного пособия. Один экземпляр студент прикладывает к расчетно-пояснительной записке, второй – остается на кафедре и является основанием для включения студента, темы ВКР, руководителя, рецензента и консультантов в приказ на дипломное проектирование. Задание подписывается студентом, руководителем, консультантами и утверждается заведующим кафедрой.

2.3. Аннотация

В аннотации в краткой форме отражаются цель и объект дипломного проектирования, полученные результаты и их новизна, особенности разрабатываемого программного продукта, перечень основных терминов и определений, количество иллюстраций, таблиц, приложений и использованных первоисточников. Объем аннотации – не более 1 страницы. Бланк аннотации представлен в приложении 7 данного учебного пособия.

2.4. Оглавление

Разделы ВКР оформляются в виде отдельных частей, содержание которых определяется целью и задачами дипломной работы. Оглавление включает в себя: введение, наименование всех разделов и подразделов, заключение, список литературы, перечень приложений с указанием номеров страниц. Пример оглавления приведен в приложении 8 (для структурного подхода) и в приложении 9 (для объектного подхода).

2.5. Введение

Во введении обосновывается актуальность разрабатываемого программного продукта, определяются цель работы и задачи для ее достижения, указываются используемые методы разработки и источники информации. Примерный объем введения 1-2 страницы.

2.6. Основная часть

Основная часть содержит, как правило, 7 подразделов (1.1 - 1.7), которые могут иметь пункты (например, 1.5.3) и подпункты (например, 1.5.3.1). В них содержатся необходимые обоснования, математические модели, модульная структура системы, функциональные схемы, диаграммы, описания и др. Каждый подраздел заканчивается выводами, облегчающими понимание замысла, логики изложения и результатов исследования.

В подразделе «Сравнительный анализ отечественных и зарубежных аналогов проектируемого объекта» проводится обзор и анализ известных проектных решений по тематике ВКР с указанием их достоинств и недостатков, делаются выводы о целесообразности и необходимости разработки.

В подразделе «Выбор технологии, среды и языка программирования» дается обоснование выбора модели жизненного цикла программного продукта, подхода к разработке, а также выбора инструментальных средств.

В подразделе «Анализ и уточнение требований к программному продукту» проводится анализ процесса обработки информации, выбор структур данных для ее хранения, разрабатываются методы и алгоритмы решения задачи. Данный подраздел уточняет требования к ПО, сформулированные в техническом задании.

В подразделе «Разработка спецификаций программного обеспечения» дается полное и точное описание функций и ограничений программного продукта, при этом используются схемы и/или диаграммы. Каждое графическое представление должно сопровождаться соответствующими подписями и пояснениями.

В подразделе «Проектирование программного обеспечения» приводятся детальные логическая и физическая модели разрабатываемого ПО (структурная и функциональная схемы, модульная структура ПО с описанием всех модулей системы с точки зрения их назначения, карты Константайна, диаграммы классов, алгоритмы функций (методов) и описание межмодульных интерфейсов). Пример заполнения спецификации на программное изделие приведен в приложении 10.

В подразделе «Проектирование интерфейса пользователя» дается описание разработки интерактивного графического интерфейса.

В подразделе «Выбор стратегии тестирования, разработка тестов, программа и методика испытаний» определяются тестируемые модули или подсистемы, окружение при тестировании (аппаратное и программное обеспечение и используемые ресурсы) и результаты тестирования, которые могут оформляться в виде табл. 1:

Результаты тестирования

Дата и время тестирования	Тестируемый модуль и подсистемы	Кто проводил тестирование	Способ тестирования	Результаты тестирования
		<i>Разработчик или тестировщик</i>	<i>Автоматически или вручную</i>	<i>«Успех» или описание неполадки</i>

Таблица должна содержать описание всех тестов, их название, ответственного за тестирование, способ реализации теста, требование, проверяемое данным тестом, а также результат, полученный после выполнения теста. При необходимости в нее могут быть внесены дополнительные поля, а после таблицы – пояснения.

Наборы тестовых данных могут быть оформлены в виде табл. 2 (столбцы можно добавлять свои).

Наборы тестовых данных

Тестовое значение	Реакция подсистемы	Пояснения

2.7. Техничко-экономическое обоснование разработки

Задание на эту (вторую) часть дипломного проекта (работы) выдается консультантом по экономике, назначаемым выпускающей кафедрой. Бланк задания по технико-экономическому обоснованию разработки приведен в приложении 3.

Примерный объем – 10 – 15 % от общего объема пояснительной записки.

В этом разделе дипломник рассчитывает следующие экономические показатели:

- себестоимость программного продукта;
- цена программного продукта;
- годовой экономический эффект;
- срок окупаемости.

Экономическая часть ВКР завершается выводами о целесообразности и экономической эффективности применения разработанного программного продукта.

2.7.1. Расчет себестоимости программного продукта

В себестоимость разработки входят: основная заработная плата разработчиков, дополнительная заработная плата, единый социальный налог, затраты на использование машинного времени, затраты на носители информации, затраты на текущий и профилактический ремонт вычислительной техники, прочие эксплуатационные расходы.

$$C_{п.п} = \text{ФЗП} + \text{ЕСН} + Z_{\text{м.вр}} + Z_{\text{н.и}} + Z_{\text{рем}} + Z_{\text{пр}},$$

где $C_{п.п}$ – себестоимость программного продукта, руб.; ФЗП – фонд заработной платы (сумма основной и дополнительной заработной платы), руб.; ЕСН – единый социальный налог, руб.; $Z_{\text{м.вр}}$ – затраты на использование машинного времени, руб.;

$Z_{н.и}$ – затраты на носители информации, руб.; $Z_{рем}$ – затраты на текущий и профилактический ремонт вычислительной техники, руб.; $Z_{пр}$ – прочие эксплуатационные расходы, руб.

Состав разработчиков (примерный): программист-дипломник, техник, руководитель дипломной работы.

Основная заработная плата каждого разработчика рассчитывается по формуле:

$$Z_{осн} = (C_T + K_p) \cdot V_{р},$$

где $Z_{осн}$ – основная заработная плата, руб.; C_T – ставка заработной платы, руб.; K_p – районный коэффициент, руб.; $V_{р}$ – время разработки, мес.

Районный коэффициент равен 25 % ставки заработной платы.

Например, ставка инженера-программиста равна 4580 руб., срок разработки программного продукта – 3 месяца. Тогда

$$Z_{осн. пр} = (4580 + 1145) \cdot 3 = 17175 \text{ руб.}$$

Дополнительная заработная плата рассчитывается как 12 % от основной заработной платы каждого разработчика.

Сумма основной заработной платы и дополнительной заработной платы всех разработчиков составляет фонд заработной платы.

В течение 2010 г. страховые взносы будут взиматься в размерах ставки ЕСН – 26 %, из них: 20 % – в Пенсионный фонд РФ; 2,9 % – в фонд социального страхования; 3,1 % – в фонд медицинского страхования. С 1 января 2011 г. страховые взносы возрастут до 34 %, из них: 26 % – в Пенсионный фонд РФ; 3,0 % – в фонд социального страхования; 5,0 % – в фонд медицинского страхования.

Все данные по заработной плате сводятся в табл. 3.

Данные по заработной плате

Должность разработчика	Разряд	Время работы, мес.	Ст, руб.	З _{осн} , руб.	З _{доп} , руб.	ФЗП, руб.	Страховые взносы, руб.
1							
...							
Итого:							

Затраты на использование машинного времени вычисляются по формуле:

$$Z_{\text{м.вр}} = C_{\text{м.вр}} \cdot V_{\text{в.т}},$$

где $Z_{\text{м.вр}}$ – затраты на использование машинного времени, руб.;
 $C_{\text{м.вр}}$ – стоимость одного часа машинного времени, руб./ч;
 $V_{\text{в.т}}$ – время использования вычислительной техники, ч.

Стоимость одного часа машинного времени рассчитывается по формуле:

$$C_{\text{м.вр}} = \frac{Ц_{\text{к}}}{C_{\text{сл}} \cdot K_{\text{р.д}} \cdot V_{\text{р.с}}} + C_{\text{т.э}} \cdot M_{\text{в.с}},$$

где $C_{\text{м.вр}}$ – стоимость одного часа машинного времени, руб./ч;
 $Ц_{\text{к}}$ – покупная цена компьютера, руб.; $C_{\text{сл}}$ – срок службы компьютера, год; $K_{\text{р.д}}$ – количество рабочих дней в году; $V_{\text{р.с}}$ – время работы компьютера в течение суток, ч; $C_{\text{т.э}}$ – стоимость одного кВт*ч электроэнергии, руб.; $M_{\text{в.с}}$ – мощность вычислительной системы, кВт.

Время использования вычислительной техники рассчитывается по следующей формуле:

$$V_{\text{в.т}} = K_{\text{д.р}} \cdot V_{\text{р.с}},$$

где $V_{р.т}$ – время использования вычислительной техники, ч;
 $K_{д.р}$ – количество дней разработки ПО; $V_{р.с}$ – время работы компьютера в течение суток, ч.

Затраты на носители информации принимаются в размере 2 % от цены вычислительной техники.

Затраты на текущий и профилактический ремонт принимаются в размере 4 % от цены вычислительной техники.

Прочие эксплуатационные расходы включают в себя затраты на освещение, отопление, охрану, уборку и текущий ремонт помещений. Они принимаются в размере 10 % от стоимости помещения (или его аренды), где происходит разработка программного продукта.

2.7.2. Расчет цены программного продукта

Для определения минимальной цены, ниже которой разработчику будет невыгодно продавать программный продукт, используется следующая формула:

$$C_{п.п} = C_{п.п} \cdot (1 + N_{пр}),$$

где $C_{п.п}$ – цена программного продукта, руб.; $C_{п.п}$ – себестоимость программного продукта, руб.; $N_{пр}$ – норматив прибыли (5 %, в формуле $N_{пр} = 0,05$).

2.7.3. Расчет экономической эффективности

В расчет экономической эффективности входит определение эксплуатационных расходов и капитальных затрат потребителя, годовой экономии эксплуатационных расходов у одного потребителя, срока окупаемости программного продукта, годового экономического эффекта.

Расходы потребителя, связанные с эксплуатацией программы, определяются по следующей формуле:

$$P_{\text{э.п}} = Vp_{\text{п.п}} \cdot C_{\text{м.вр}} + Ц_{\text{п.п}}/C_{\text{сл}},$$

где $P_{\text{э.п}}$ – эксплуатационные расходы потребителя, руб.; $Vp_{\text{п.п}}$ – объем машинного времени в течение года, необходимый для решения данной задачи с использованием программы, ч; $C_{\text{м.вр}}$ – стоимость одного часа машинного времени, руб./ч; $Ц_{\text{п.п}}$ – цена программного продукта, руб.; $C_{\text{сл}}$ – срок службы программного продукта, год. Обычно составляет 1 – 2 года, затем выпускается новая версия программного продукта.

Если на момент внедрения программного продукта у потребителя все работы выполнялись вручную, тогда капитальные затраты рассчитываются по формуле

$$P_{\text{кап}} = \frac{Vp_{\text{п.п}} \cdot K_{\text{ЭВМ}}}{\Phi_{\text{вр}}} + Ц_{\text{п.п}},$$

где $P_{\text{кап}}$ – капитальные расходы потребителя, руб.; $\Phi_{\text{вр}}$ – полезный годовой фонд времени работы вычислительной техники, принимается условно 2000 ч в год; $K_{\text{ЭВМ}}$ – капитальные затраты на вычислительную технику, для которой предназначена программа, руб.

Капитальные затраты на вычислительную технику рассчитываются по формуле

$$K_{\text{ЭВМ}} = Ц_{\text{ЭВМ}} + P_{\text{п.п}},$$

где $Ц_{\text{ЭВМ}}$ – цена вычислительной техники, руб.; $P_{\text{п.п}}$ – прочие расходы потребителя, связанные с помещением (отопление, освещение, уборка и т.д.), принимаются в размере 10 % от стоимости помещения потребителя (или его аренды), руб.

Для расчета годовой экономии эксплуатационных расходов потребителя вычисляются эксплуатационные затраты потребителя при решении задачи вручную:

$$P_{\text{э.руч}} = 1,21 \cdot \text{ФЗП} \cdot 12,$$

где $P_{\text{э.руч}}$ – эксплуатационные расходы потребителя при решении задачи вручную, руб.; ФЗП – фонд заработной платы персонала, обслуживающего решение задачи вручную, руб.; 12 – количество месяцев в году; 1,21 – поправочный коэффициент.

Тогда годовая экономия эксплуатационных расходов у одного потребителя рассчитывается по формуле

$$\text{Э} = P_{\text{э.руч}} - P_{\text{э}}.$$

Срок окупаемости программного продукта рассчитывается по формуле

$$T_{\text{ок}} = \frac{P_{\text{кап}}}{\text{Э}}.$$

Годовой экономический эффект, получаемый одним потребителем, рассчитывается по формуле

$$\text{ЭЭ} = \text{Э} - E_{\text{н}} \cdot P_{\text{кап}},$$

где $E_{\text{н}}$ – нормативный коэффициент эффективности дополнительных капитальных вложений, равный 0,15.

2.8. Анализ мероприятий по безопасности жизнедеятельности и обеспечению экологической чистоты

Задание на эту (третью) часть дипломного проекта (работы) выдается консультантом по безопасности жизнедеятельности, назначаемым выпускающей кафедрой. Бланк задания приведен в приложении 4.

Примерный объем – 10 % от общего объема пояснительной записки. Во время преддипломной практики студент изучает

вопросы промышленной санитарии и техники безопасности на рабочем месте, условия труда на данном производстве и др.

В этой части необходимо изложить следующее:

– анализ мероприятий по безопасности жизнедеятельности (общая характеристика безопасности производственного процесса; анализ вредных производственных факторов и мероприятия по их снижению; анализ опасных производственных факторов и мероприятия по их снижению; оценка электробезопасности помещения и мероприятия по повышению электробезопасности; оценка помещения по пожарной безопасности и мероприятия по повышению пожарной безопасности; анализ неполадок и аварийных ситуаций в рабочем помещении);

- анализ мероприятий по обеспечению экологической чистоты.

При написании данной части студент использует ГОСТы, СНиПы и НПБ, например:

1. ГОСТ 12.0.003-74 – «Классификация опасных и вредных производственных факторов».

2. ГОСТ 12.3.002-75 – «Процессы производственные. Общие требования безопасности».

3. ГОСТ 12.2.003-91 – «Оборудование производственное. Общие требования безопасности».

4. ГОСТ 12.1.005-88 – «Общие санитарно-гигиенические требования к воздуху санитарной зоны».

5. ГОСТ 12.1.012-90 – «Вибрационная безопасность. Общие требования».

6. ГОСТ 12.1.003-83 – «Шум. Общие требования безопасности».

7. ГОСТ 12.1.019-79 изменение 01.86 – «Электробезопасность. Общие требования и номенклатура видов защиты».

8. ГОСТ 12.1.004-91 – «Пожарная безопасность. Общие требования».

9. СНиП 21-01-97 – «Пожарная безопасность зданий и сооружений».

10. СНиП 230595 – «Естественное и искусственное освещение».

11. НПБ 88-01 – «Установки пожаротушения и сигнализации».

2.8.1. Общая характеристика безопасности

Полная безопасность труда человека в производственных условиях определяется тремя факторами: безопасность производственного оборудования, производственного и трудового процессов. Эти составляющие безопасности труда связаны между собой (рис. 1).



Рис.1. Треугольник безопасности

Повышение или понижение пределов отклонений параметров технологического процесса приводит к возникновению опасных и вредных производственных факторов, что оказывает влияние на безопасность труда обслуживающего персонала. Аналогичное воздействие оказывает состояние производственного оборудования. Факторы безопасности производственного оборудования являются составной частью общей системы безопасности труда. Совокупность безопасности производст-

венного процесса и оборудования образует материальный и технический фактор безопасности производства, который на действующем оборудовании и применяемой технологии может совершенствоваться как путем изменения технологии и оборудования, так и путем оснащения технологии и оборудования техническими средствами безопасности и производственной санитарии.

В связи с этим уровень безопасности труда можно рассматривать как произведение уровней безопасности оборудования b_1 , производственного b_2 и трудового b_3 процессов: $V = b_1 \cdot b_2 \cdot b_3$. Предел уровня безопасности для каждой подсистемы равен единице.

Дипломник должен выполнить анализ безопасности труда и определить долю каждого фактора в треугольнике безопасности. При этом дается характеристика роли пользователя в системе, уделяется внимание объективным (производственным) факторам: рабочее место, шумоизоляция, кондиционирование воздуха, освещенность и т.д., а также субъективным (человеческим) факторам: быстрота реакции, достаточно хорошее зрение, внимательность и т.д.

2.8.2. Анализ вредных производственных факторов и мероприятия по их снижению

Вредный производственный фактор – производственный фактор, воздействие которого на человека может привести к заболеванию.

Дипломник определяет основные вредные производственные факторы (запыленность, шум, вибрация, тепловое излучение, электрический ток, электромагнитное излучение и т.д.), их воздействие на жизнедеятельность человека и описывает мероприятия по снижению воздействия этих факторов на пользователя.

2.8.3. Анализ опасных производственных факторов и мероприятия по их снижению

Опасный производственный фактор – производственный фактор, воздействие которого на человека может привести к травме.

Дипломник определяет основные опасные производственные факторы (повышенная температура поверхности оборудования, концентрация горючих газов в воздухе рабочей зоны, взрывоопасность, движущиеся механизмы и машины и т.д.), их воздействие на жизнедеятельность человека и описывает мероприятия по снижению воздействия этих факторов на пользователя.

2.8.4. Оценка электробезопасности помещения и мероприятия по повышению электробезопасности

В данном разделе дается оценка электробезопасности на рабочем месте, определяется класс помещения по опасности поражения электрическим током, характеристики сети и т.д.

По степени опасности поражения человека электрическим током помещения подразделяются на:

1) помещения без повышенной опасности, в которых отсутствуют условия, создающие повышенную или особую опасность;

2) помещения с повышенной опасностью, характеризующиеся наличием в них одного или нескольких условий: сырости, токопроводящей пыли, токопроводящих полов (металлических, земляных, железобетонных, кирпичных), высокой температуры, возможности одновременного прикосновения человека к имеющим соединение с землей металлоконструкциям здания, технологическим аппаратам, с одной стороны, и к металлическому корпусу электрооборудования – с другой;

3) особо опасные помещения, характеризующиеся наличием одного из следующих условий: особой сырости, химически

активной или органической среды, одновременно двух или более условий повышенной опасности.

На основании анализа дипломник определяет конкретные средства защиты (заземление, ограждение, зануление, автоматические отключатели, наличие резиновых ковриков и т.д.).

2.8.5. Оценка помещения по пожарной безопасности и мероприятия по повышению пожарной безопасности

В данном разделе приводится классификация помещения по пожарной опасности, степень огнестойкости всего здания, причины и источники возгорания, классы и виды пожаров.

На основе полученных данных дипломник распределяет оборудование внутри помещения и выбирает необходимые средства пожаротушения (песок, кошма, огнетушители с указанием количества и марки, количество пожарных щитов, количество пожарных гидрантов и т.д.), указывает наличие или отсутствие автоматической противопожарной сигнализации.

2.8.6. Анализ неполадок и аварийных ситуаций

В данном разделе дается прогноз основных возможных неполадок и аварийных ситуаций и определяются мероприятия по защите от неполадок и аварий (источники бесперебойного питания для вычислительной техники; установка контролирующей аппаратуры: датчики, контрольные лампы; установка защитной аппаратуры: автоматы, предохранители; повышенная надежность и быстроедействие системы и т.д.).

2.8.7. Анализ мероприятий по обеспечению экологической чистоты

Контроль за состоянием окружающей среды можно производить при помощи измерения:

- 1) температуры;

- 2) относительной влажности;
- 3) скорости движения воздуха;
- 4) теплоизлучения.

В данном разделе выполняется сравнение приведенных выше параметров с действующими санитарно-техническими требованиями к зданиям и сооружениям. Так, например, температура окружающей среды должна составлять 15 – 25 °С; относительная влажность – 40 – 60 %; скорость движения воздуха: летом – 0,5 – 1,0 м/с, зимой – 0,3 – 0,5 м/с. Приводится перечень мероприятий по снижению вредных воздействий на окружающую среду.

2.9. Заключение

В заключении приводятся теоретические и практические выводы и результаты работы. Они должны излагаться в краткой форме и давать представление о степени выполнения поставленной перед дипломником задачи. В заключении делается акцент на актуальность, степень новизны и практическую значимость разработанного программного продукта. Здесь же приводятся данные об экономической эффективности при внедрении данного программного продукта. Примерный объем заключения – 1 – 2 страницы.

2.10. Приложения

В приложения включают материал, не вошедший в пояснительную записку, но представляющий интерес для более глубокого понимания: 1 – техническое задание (ГОСТ 19.201-78), 2 – схемы и/или диаграммы (ГОСТ 19.701-90), 3 – текст программы (ГОСТ 19.401-78), 4 – спецификация (ГОСТ 19.202-78), 5 – руководство пользователя (ГОСТ 19.503-79, ГОСТ 19.504-79, ГОСТ 19.505-79), 6 – наборы тестовых данных и результатов тестирования (ГОСТ 19.301-79).

2.11. Графический материал

Конкретный графический материал зависит от темы ВКР и согласовывается с руководителем работы.

Плакаты, выносимые на защиту, выполняются на листе ватмана формата А1 (594 × 841) тушью или распечатываются на плоттере. Количество плакатов – 5 – 8.

Примерный перечень плакатов для структурного подхода.

1. Постановка задачи.
2. Структурная и (или) функциональная схемы системы.
3. Диаграммы структур данных (диаграммы классов).
4. Схема взаимодействия программных модулей системы.
5. Алгоритмы основных модулей системы.
6. Результаты испытаний системы.

Примерный перечень плакатов для объектного подхода:

1. Постановка задачи.
2. Результаты испытаний системы.

Раздел 3

ПОРЯДОК ВЫПОЛНЕНИЯ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

3.1. Сроки выполнения ВКР

Дипломный проект (работа) разрабатывается студентом очной пятилетней формы и очной сокращенной формы обучения самостоятельно. Студент составляет календарный план-график выполнения дипломного проекта (работы) с указанием очередности отдельных этапов и согласует его с руководителем. Календарный план-график является основным документом, по которому контролируется процент выполнения дипломного проекта (работы).

Не позднее чем за 10 дней до защиты ВКР студент-дипломник подписывает расчетно-пояснительную записку у

консультантов, отдает ее руководителю для ознакомления и написания отзыва, затем проходит нормоконтроль.

Для получения допуска за 10 дней до защиты студент-дипломник предоставляет на кафедру расчетно-пояснительную записку законченного дипломного проекта (работы), подписанную им самим, консультантами, руководителем и нормоконтролером. Расчетно-пояснительная записка должна содержать отзыв руководителя.

После получения допуска к защите студент-дипломник передает расчетно-пояснительную записку на рецензирование. Рецензентами могут быть высококвалифицированные специалисты соответствующего профиля, работающие на предприятиях, в организациях, научно-исследовательских и проектных институтах. В рецензии дается оценка актуальности темы дипломного проекта (работы), отмечаются достоинства и недостатки разработки, а также указывается практическое значение работы и качество оформления, степень использования выпускником современных источников, технологий, последних достижений науки и техники.

В заключительной части рецензии дается однозначная оценка (отлично, хорошо, удовлетворительно, неудовлетворительно) и указывается, заслуживает или не заслуживает выпускник присвоения квалификации инженер. Подпись рецензента должна быть заверена в кадровой службе по месту его работы. Бланк рецензии приведен в приложении 6 данного учебного пособия.

За 2 дня до защиты ВКР расчетно-пояснительная записка вместе с отзывом руководителя и рецензией сдается на кафедру секретарю Государственной аттестационной комиссии (ГАК).

3.2. Самостоятельная работа студента

Дипломный проект (работа) является личной разработкой студента, руководитель лишь помогает советами и контролирует выполнение плана-графика. При необходимости руководитель делает замечания студенту по дипломной работе. Ответст-

венность за достоверность информации, правильность расчетов и работоспособность программного продукта несет студент.

Студент и руководитель встречаются для обсуждения решений задач дипломного проектирования не реже 1 раза в неделю.

3.3. Допуск дипломного проекта (работы) к защите

Завершенный дипломный проект (работа) подписывается дипломником и консультантами на титульном листе и передается руководителю для написания подробного отзыва. В отзыве руководитель работы оценивает самостоятельность, целеустремленность, трудолюбие студента, а также практическую ценность разработки. Бланк отзыва руководителя приведен в приложении 5 данного учебного пособия. Затем расчетно-пояснительная записка подписывается у нормоконтролера.

Вопрос о допуске студента-дипломника к защите решается на заседании кафедры, как правило, за 10 дней до начала работы ГАК. Все допущенные работы подписываются заведующим кафедрой и директором института.

3.4. Составление текста доклада

При подготовке к защите дипломнику рекомендуется составить тезисы доклада, оформить плакаты, а также продумать ответы на замечания рецензента.

Содержание тезисов доклада и информацию, выносимую на плакаты, дипломник согласовывает с руководителем.

Тезисы доклада обычно состоят из трех частей (вступление, основная часть, выводы). Во вступлении формулируется цель работы, подчеркивается ее актуальность, дается краткая характеристика существующих аналогов и показываются основные пути решения поставленной задачи. В основной части доклада дается описание основных этапов разработки программного обеспечения и демонстрируются преимущества нового варианта по сравнению с уже существующими. В выводах подчеркивается экономическая эффективность внедрения нового про-

граммного продукта, срок его окупаемости, а также дается рекомендация по его использованию.

На плакаты, как правило, выносятся информация, иллюстрирующая основные этапы разработки (структурная и (или) функциональная схемы системы, диаграммы, алгоритмы, таблицы с результатами испытаний). В докладе обязательны ссылки на все представленные плакаты. Примерное время доклада – 10 – 12 минут.

3.5. Защита дипломного проекта (работы)

К защите дипломник предоставляет расчетно-пояснительную записку (со всеми подписями на титульном листе), плакаты, отзыв руководителя и рецензию на дипломный проект (работу).

Защита проводится на открытом заседании ГАК, куда приглашаются преподаватели, сотрудники, аспиранты, студенты, представители предприятий, заинтересованные в результатах дипломных работ.

Перед защитой секретарь объявляет фамилию, имя, отчество выпускника и тему дипломного проекта (работы).

Студент-дипломник в течение 10 – 12 минут излагает основное содержание дипломного проекта (работы). После завершения доклада отвечает на вопросы, которые могут задавать все присутствующие, и демонстрирует с использованием компьютера и мультимедийных средств работу своей программы.

Затем секретарь объявляет оценки руководителя и рецензента, зачитывает рецензию на дипломный проект (работу). Дипломник дает разъяснения на замечания рецензента, после чего защита дипломного проекта (работы) может считаться законченной.

Оценка дипломного проекта (работы) объявляется после обсуждения членами ГАК самой защиты и с учетом оценок руководителя и рецензента.

Все защищенные дипломные проекты (работы) хранятся на выпускающей кафедре в течение пяти лет.

Если защита дипломного проекта (работы) признана неудовлетворительной, то выпускник обязан выполнить новую разработку, определенную выпускающей кафедрой.

Раздел 4

АВТОМАТИЗАЦИЯ ДИПЛОМНОГО ПРОЕКТИРОВАНИЯ С ИСПОЛЬЗОВАНИЕМ CASE-СРЕДСТВ

4.1. VPwin

VPwin имеет достаточно простой и интуитивно понятный интерфейс пользователя, дающий возможность создавать сложные модели. При запуске VPwin по умолчанию появляется основная панель инструментов, палитра инструментов (вид которой зависит от выбранной нотации) и навигатор модели – Model Explorer (рис. 2). Функциональность панели инструментов доступна из основного меню VPwin.

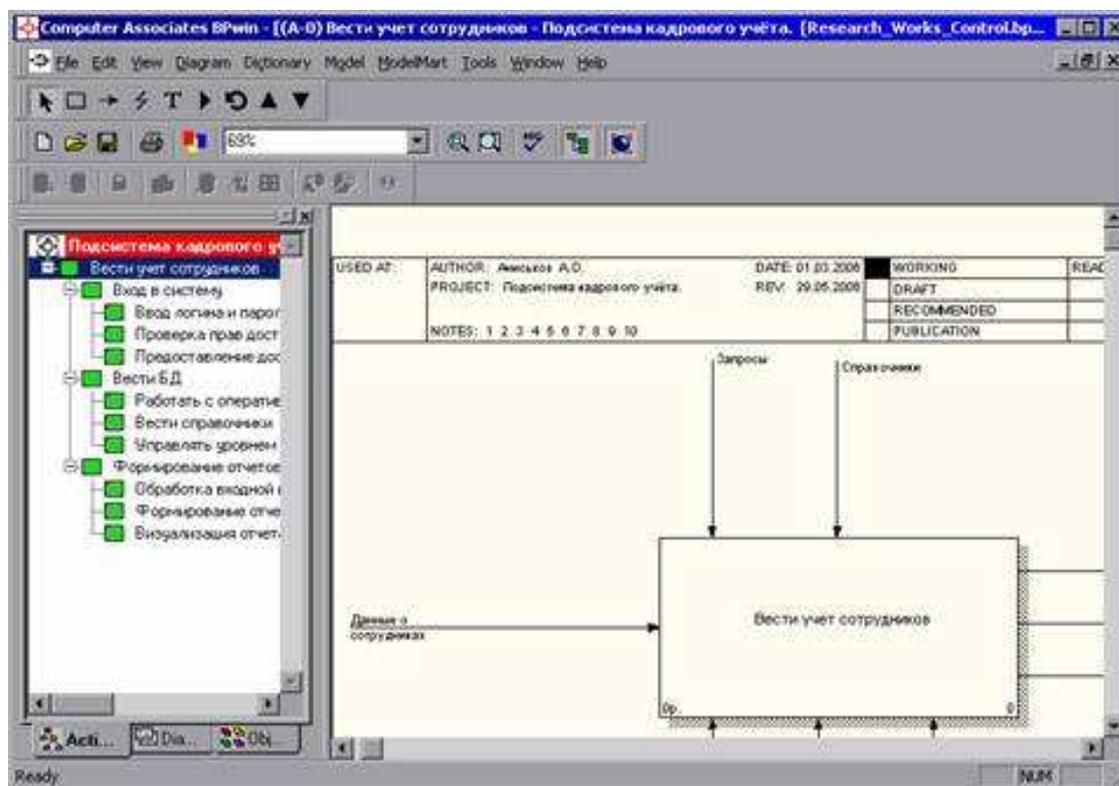


Рис. 2. Интегрированная среда разработки модели VPwin

При создании новой модели возникает диалог, в котором следует указать, будет ли создана модель заново или она будет открыта из файла, внести имя модели и выбрать методологию, в которой будет построена модель (рис. 3).

BPwin поддерживает три методологии – IDEF0, IDEF3 и DFD, каждая из которых решает свои специфические задачи. В BPwin возможно построение смешанных моделей, т.е. модель может содержать одновременно как диаграммы IDEF0, так и диаграммы IDEF3 и DFD. Состав палитры инструментов изменяется автоматически, когда происходит переключение с одной нотации на другую.

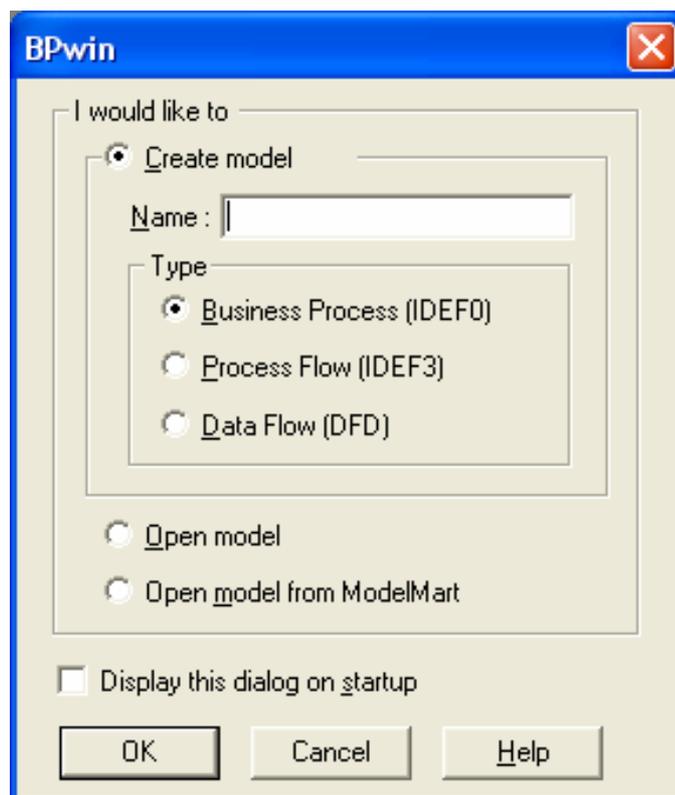


Рис. 3. Диалог создания модели

После щелчка по кнопке ОК появляется диалог Properties for New Models (рис. 4), в котором следует внести свойства модели.

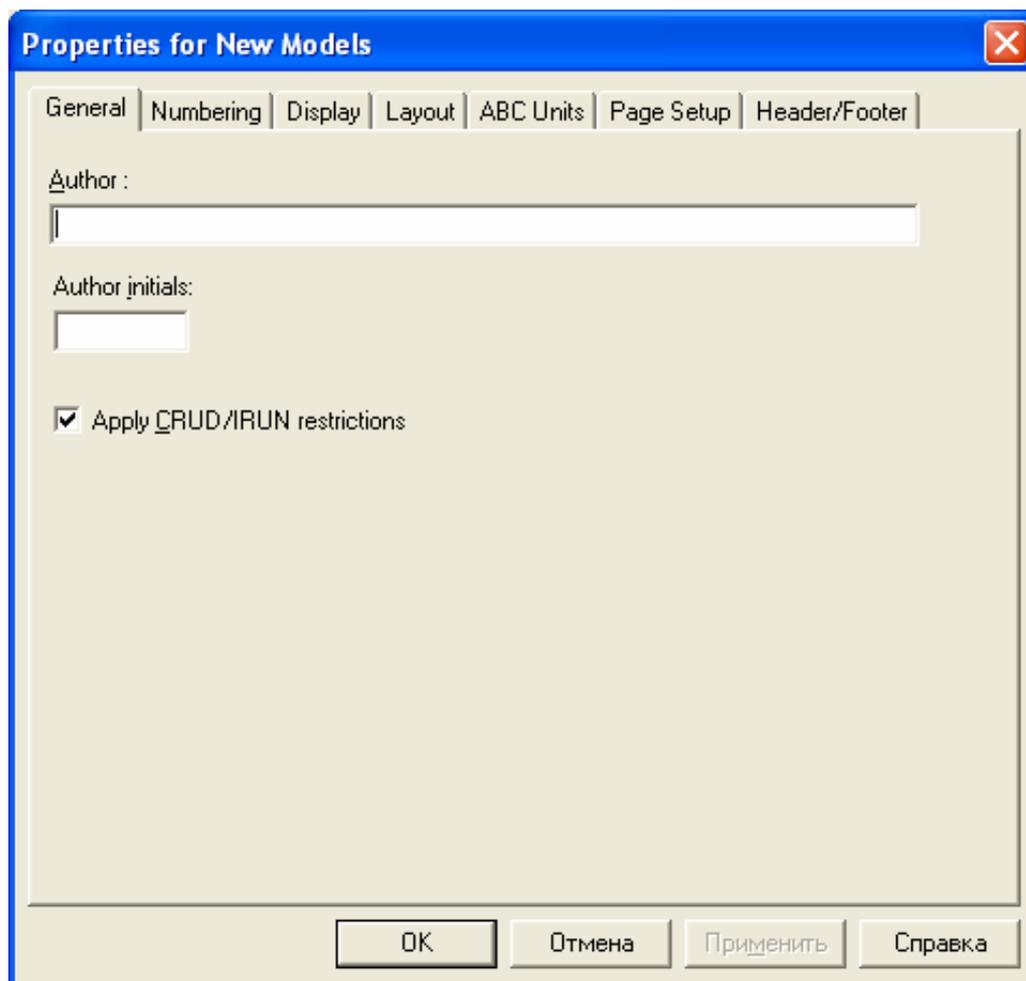


Рис. 4. Диалог Properties for New Models

Модель в VPwin рассматривается как совокупность работ, каждая из которых оперирует некоторым набором данных. Работа изображается в виде прямоугольника, данные – в виде стрелок.

4.1.1. Model Explorer – навигатор модели

Инструмент навигации Model Explorer имеет три вкладки – Activities, Diagrams и Objects. Вкладка Activities показывает в виде раскрывающегося иерархического списка все работы модели. Одновременно могут быть показаны все модели, открытые в VPwin. Работы с диаграмм IDEF0 отображаются зеленым цветом, IDEF3 – желтым, DFD – голубым.

Щелчок по работе во вкладке Activities переключает левое окно Rwin на диаграмму, на которой эта работа размещена. Для редактирования свойств работы следует щелкнуть по ней правой кнопкой мыши, вызвав контекстное меню. Значение пунктов контекстного меню приведено в табл. 4.

Таблица 4

Контекстное меню редактирования свойств работы

Пункт меню	Описание
Insert Before	Создать новую работу на той же диаграмме. В списке работ новая работа будет вставлена перед текущей
Insert After	Создать новую работу на той же диаграмме. В списке работ новая работа будет вставлена после текущей
Decompose	Декомпозировать работу. В результате будет создана новая диаграмма декомпозиции
Name	Вызов редактора имени работы
Definition/Note	Вызов редактора определения и примечания к работе
Font	Изменение шрифта работы
Color	Изменение цвета работы
Costs	Задание стоимости работе
UDP	Задание свойств, определяемых пользователем
UOW	Задание свойств для работ IDEF3

Вкладка Diagrams служит для перехода на любую диаграмму модели (рис. 5).



Рис. 5. Вкладка Diagrams навигатора Model Explorer

После перехода на вкладку Objects на ней отображаются все объекты, соответствующие выбранной на вкладке Diagrams диаграмме.

4.1.2. Создание функциональной модели с помощью VPwin

Создание модели в стандарте IDEF0

В IDEF0 система представляется как совокупность взаимодействующих работ или функций. Такая чисто функциональная ориентация является принципиальной – функции системы анализируются независимо от объектов, которыми они оперируют. Под моделью в IDEF0 понимают описание системы (текстовое и графическое), которое должно дать ответ на некоторые заранее определенные вопросы.

Процесс моделирования какой-либо системы начинается с определения контекста, т.е. наиболее абстрактного уровня описания системы в целом. В контекст входит определение субъекта моделирования, цели и точки зрения.

Итак, на первом этапе проектирования необходимо создать контекстную диаграмму. Для построения следующих диаграмм потребуются предыдущие, поэтому следует сохранять модель, полученную на каждом шаге.

1. После запуска пакета VPwin следует выбрать кнопку , внести имя модели и выбрать тип диаграммы IDEF0 (функциональная диаграмма). При этом шаблон контекстной диаграммы создается автоматически.

2. Во вкладке Activities компонента Model Explorers щелчок правой кнопкой по объекту позволяет редактировать его свойства.

3. В пункте меню Model/Model Properties следует выполнить следующие действия (рис. 6):

- во вкладке General внести имя модели, имя проекта, имя автора и тип модели;
- во вкладке Purpose внести цель и точку зрения;
- во вкладке Definition внести определение и цель.

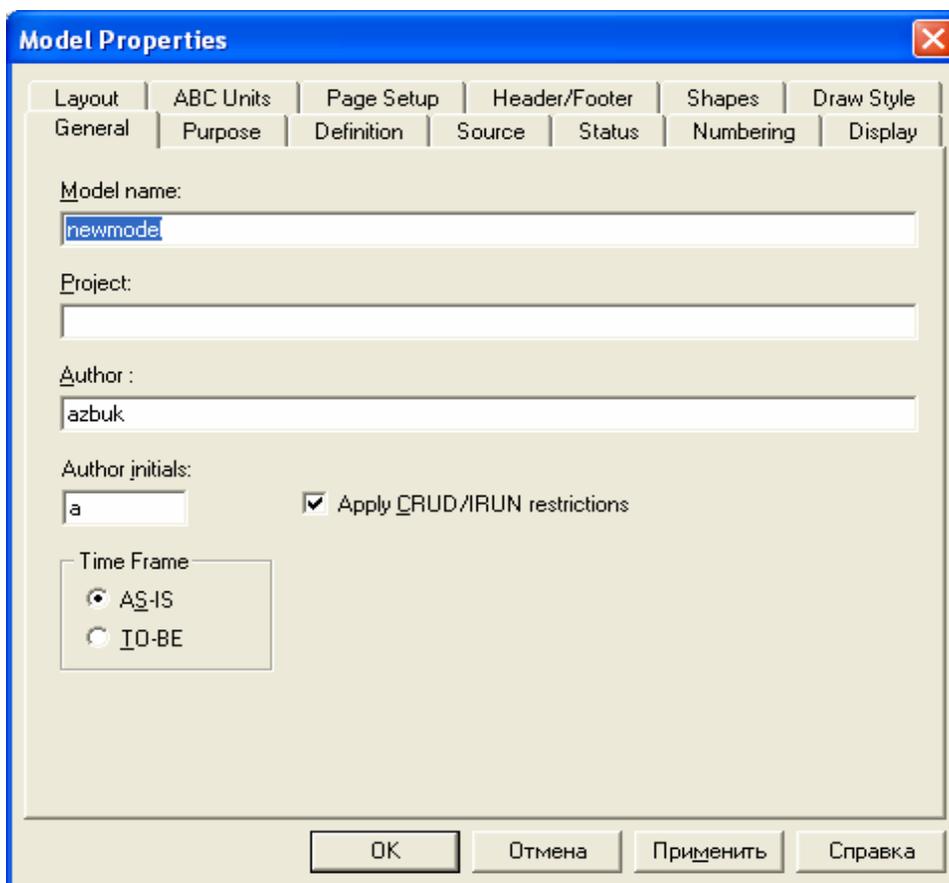


Рис. 6. Диалог задания свойств модели

4. Далее следует перейти на контекстную диаграмму и вызвать контекстное меню. Выбрав пункт Name, следует внести имя функции на контекстной диаграмме, во вкладке Definition – определение.

5. После этого необходимо на контекстной диаграмме создать стрелки, внести для каждой из них имя (Arrow Name), определение (Arrow Definition), тип (Arrow Type) (рис. 7).

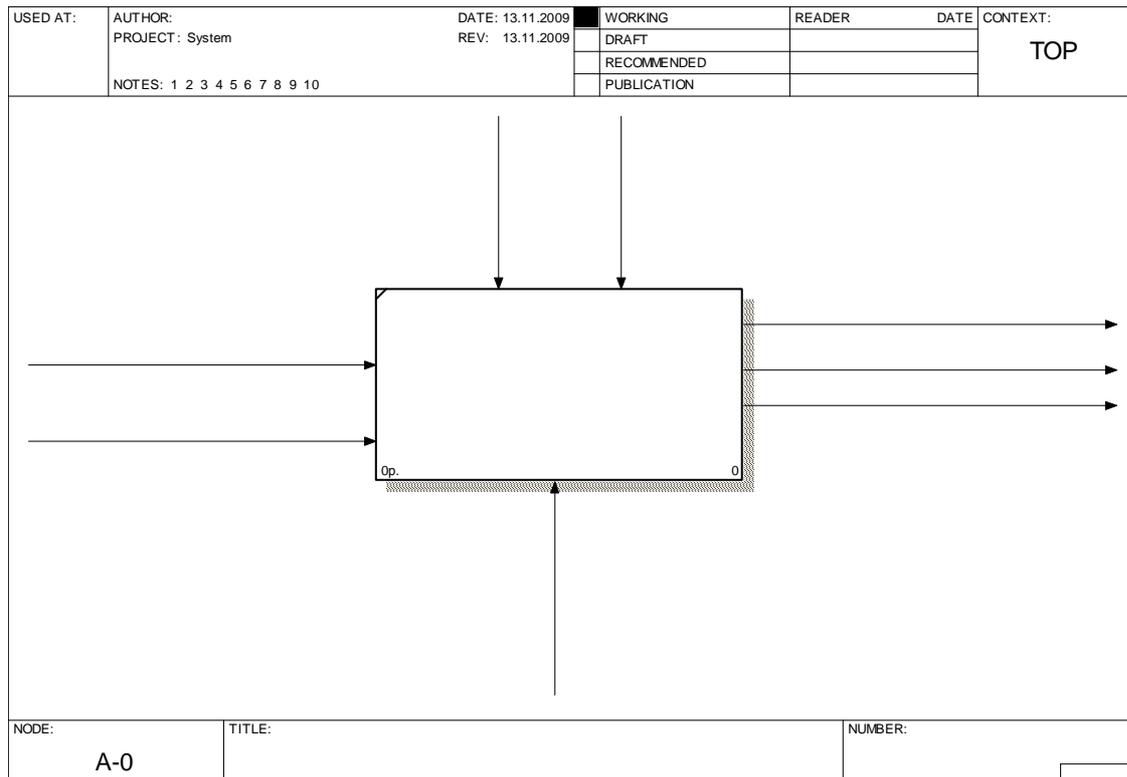


Рис. 7. Заготовка контекстной диаграммы

6. С помощью кнопки  в поле диаграммы можно внести текст, например цель и точку зрения.

7. Результатом предыдущих действий является контекстная диаграмма.

На втором этапе проектирования необходимо создать диаграммы декомпозиции.

1. Для перехода к диаграммам декомпозиции следует выбрать в палитре инструментов кнопку перехода на нижний уро-

вень  и в диалоге Activity Box Count установить количество работ на диаграмме нижнего уровня, а также нотацию IDEF0. После этого автоматически будет создана диаграмма декомпозиции (рис. 8).

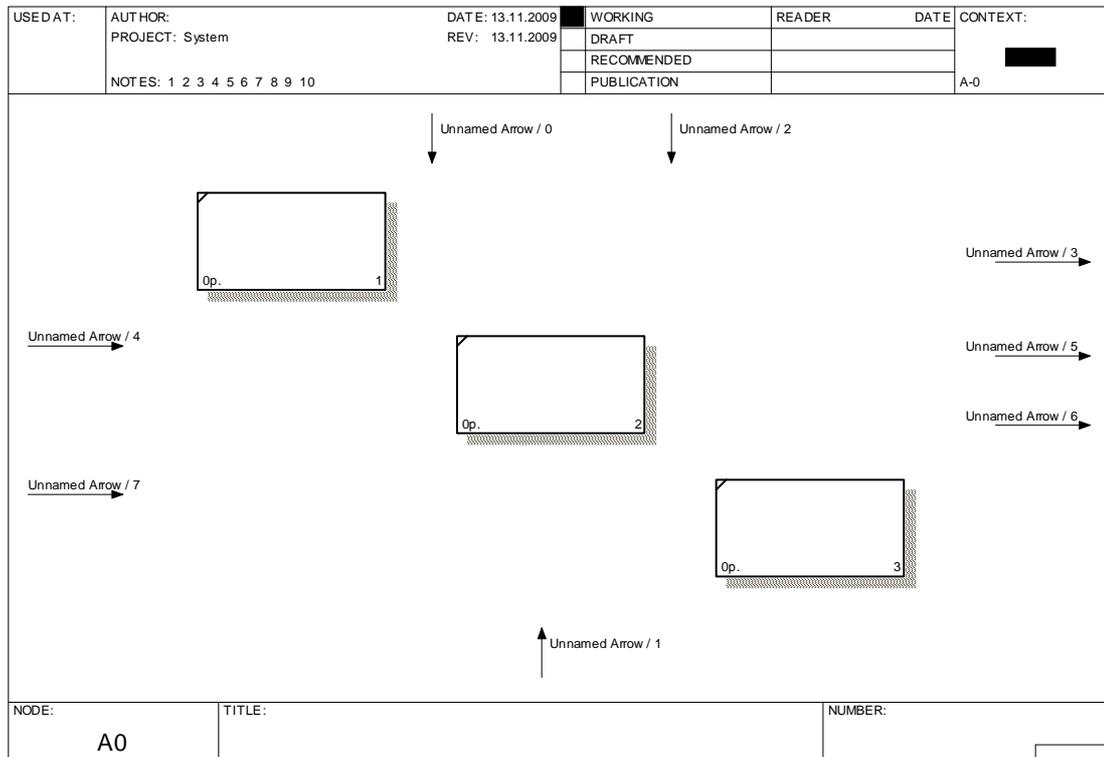


Рис. 8. Заготовка диаграммы декомпозиции

2. Вызвав контекстное меню, следует внести имя (Name) и определение (Definition) для всех работ диаграммы. Для изменения свойств работ после их внесения в диаграмму можно воспользоваться словарем работ. Вызов словаря – меню Dictionary/Activity.

Если описать имя и свойства работы в словаре, то ее можно будет внести в диаграмму позже с помощью кнопки  в палитре инструментов.

Невозможно удалить работу из словаря, если она используется на какой-либо диаграмме. Если работа удаляется с диа-

граммы, из словаря она не удаляется, поэтому ее имя и описание может быть использовано в дальнейшем.

Для добавления работы в словарь необходимо перейти в конец списка и щелкнуть правой кнопкой мыши по последней строке. При этом возникает новая строка, в которой нужно внести данные о работе.

3. Далее в режиме рисования стрелок следует связать граничные стрелки (кнопка  на палитре инструментов) с работами. Имена и свойства стрелок можно внести с использованием словаря стрелок Dictionary/Arrow или щелкнув правой кнопкой мыши по стрелке.

4. Работы следует также соединить внутренними стрелками, при этом могут быть созданы стрелки обратной связи.

5. При создании новой граничной стрелки выхода (если это необходимо) она автоматически не попадает на диаграмму верхнего уровня и имеет квадратные скобки на окончание []. Чтобы удалить квадратные скобки, достаточно щелкнуть по ним правой кнопкой мыши и выбрать пункт меню Arrow Tunnel. В диалоге Border Arrow Editor следует выбрать опцию Resolve it to Border arrow. Для стрелки нужно выбрать опцию Trim из контекстного меню.

Следующий этап проектирования предполагает декомпозицию каждой из созданных на предыдущем этапе работ.

Создание диаграммы IDEF3

1. Контекстную диаграмму нотации IDEF0 можно декомпонировать, выбрав нотацию IDEF3 (диалог Activity Box Count).

2. Вызвав контекстное меню каждой работы, следует внести ее имя, определение и свойства.

3. Можно также внести перекрестки одного из типов (рис. 9), после чего связать их с работами. Если от перекрестка нужно сделать разветвление, следует щелкнуть по нему.

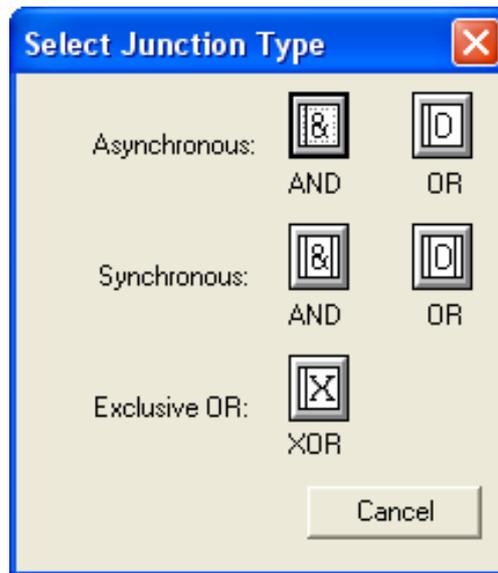


Рис. 9. Типы перекрестков

Связи в IDEF3 показывают взаимоотношения работ. Различают три типа стрелок, изображающих связи, стиль которых устанавливается во вкладке Style диалога Arrow Properties (рис. 10).

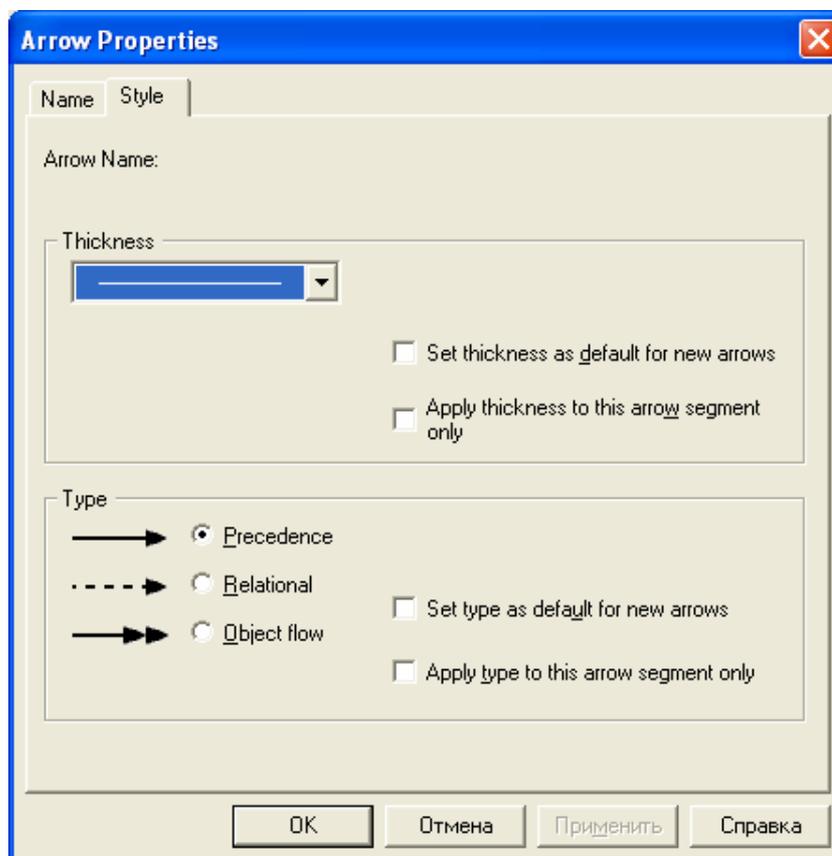


Рис. 10. Вкладка Style диалога Arrow Properties

Создание диаграммы DFD

1. Для декомпозиции диаграммы в нотации DFD следует выбрать соответствующий тип в диалоговом окне Activity Box Count.

2. Имена работ вносятся тем же способом, что и на предыдущих типах диаграмм.

3. Затем нужно удалить граничные стрелки с диаграммы.

4. Если внутренние стрелки должны быть двунаправленными, следует щелкнуть правой кнопкой по стрелке и выбрать в контекстном меню пункт Style, опцию Bidirectional.

Контрольные вопросы

1. Какие методологии поддерживает VPwin?
2. Что понимается под моделью в VPwin?
3. Для чего используется Model Explorer?
4. Кратко опишите этапы создания контекстной диаграммы в методологии IDEF0.
5. Возможно ли удаление работы из словаря, если она используется на какой-либо диаграмме?
6. Каким образом можно декомпонировать контекстную диаграмму нотации IDEF0 в нотацию IDEF3?
7. Для чего служит нотация IDEF3?
8. Какие типы стрелок используются для связей в IDEF3?
9. В чем заключается основное различие между функциональными диаграммами и диаграммами потоков данных?
10. В каких случаях использование диаграмм потоков данных является наиболее целесообразным?

4.2. ERwin

ERwin – средство концептуального моделирования БД, использующее методологию IDEF1X. Методология IDEF1X была разработана с учетом таких требований, как простота изучения и возможность автоматизации. Она базируется на методе IDEF1, разработанном Т. Рэмей и основанном на подходе П. Чена. Метод IDEF1 позволяет построить модель данных, эквивалентную реляционной модели в третьей нормальной форме.

ERwin реализует проектирование схемы БД, генерацию ее описания на языке целевой СУБД (ORACLE, Informix, Ingres, Sybase, DB/2, Microsoft SQL Server, Progress и др.) и реинжиниринг существующей БД. Для ряда средств разработки приложений (PowerBuilder, SQLWindows, Delphi, Visual Basic) выполняется генерация форм и прототипов приложений.

Рабочее окно программы ERwin содержит (рис. 11):

- строку главного меню;
- панель инструментов, где находятся кнопки с командами, которыми можно пользоваться при построении и модифицировании диаграммы ERwin (Новая диаграмма, Открыть диаграмму, Сохранить диаграмму, Печать диаграммы, Уровень сущности, Уровень атрибута, Уровень определения сущности, Уровень физической схемы, Уменьшить, Увеличить, Прямое проектирование (генерация схемы БД), Синхронизация БД, Подсоединиться к базе данных, Выбрать сервер и др.);
- окно инструментов, которые используются при определении модели данных (Инструмент выбора, Создать сущность, Полная связь подтипов, Текстовый блок, Инструмент управления атрибутами, Идентифицирующая связь, Связь «многие ко многим», Неидентифицирующая связь);
- в нижней части окна – статусную строку с информацией о выбранной функции или команде меню;
- окно диаграммы.

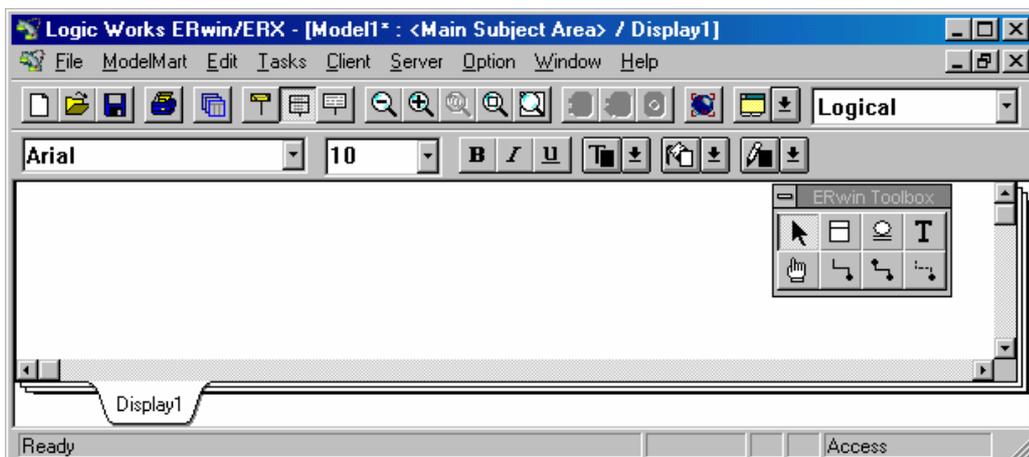


Рис. 11. Рабочее окно программы ERwin

Если работать в нескольких окнах диаграмм, то ERwin запоминает последний выбранный инструмент для каждого окна. При возвращении в окно можно сразу же использовать инструмент, который был выбран до того, как перешли к другому окну.

4.2.1. Пример разработки логической модели в ERwin

Рассмотрим цикл разработки логической модели на примере следующей предметной области. Ведется учет служащих; для каждого служащего хранится информация о детях и о списке должностей, занимаемых этим служащим. Для должностей ведется учет информации об установленных должностных окладах.

Сначала создадим логический уровень модели. Для этого зададим режим отображения сущностей (Display/Entity Level). При помощи линейки инструментов получим сущности «Служащий», «Дети», «История работы», «История зарплаты». Будем именовать сущности на русском языке. Каждую сущность подробно опишем на русском языке в редакторе Entity Definition. Это описание появится в отчетах ERwin и может быть отображено на диаграмме. Укажем связи между сущностями. Например, «Служащий» связан идентифицирующей связью «является родителем» с сущностью «Дети». Описание связи вводится в редакторе Editor/Relationship.

Результат работы отображен на рис. 12.

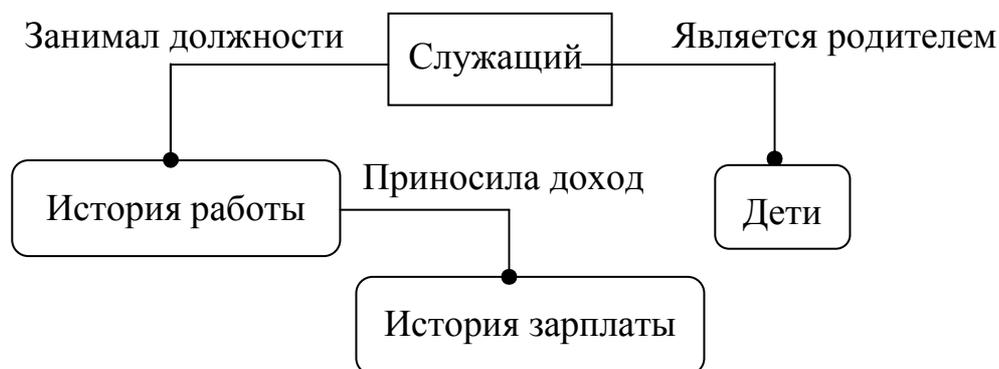


Рис. 12. Диаграмма уровня сущности

Теперь перейдем в режим задания атрибутов (Display/Attribute Level). В редакторе Entity/Attribute на русском языке зададим имена ключевых и неключевых атрибутов. Заметим, что для дочерней сущности «Дети» ключевой атрибут «Номер служащего» не указывается вручную. ERwin обеспечивает его миграцию из родительской сущности. То же происходит с другими дочерними сущностями. Для атрибута «Имя» сущности «Служащий» укажем, что этот атрибут является альтернативным ключом (будем считать, что у всех служащих уникальные имена/фамилии). Для этого после имени атрибута поместим указатель АК1 в скобках. Результат работы отображен на диаграмме ERwin в нотации IDEF1X (рис. 13).

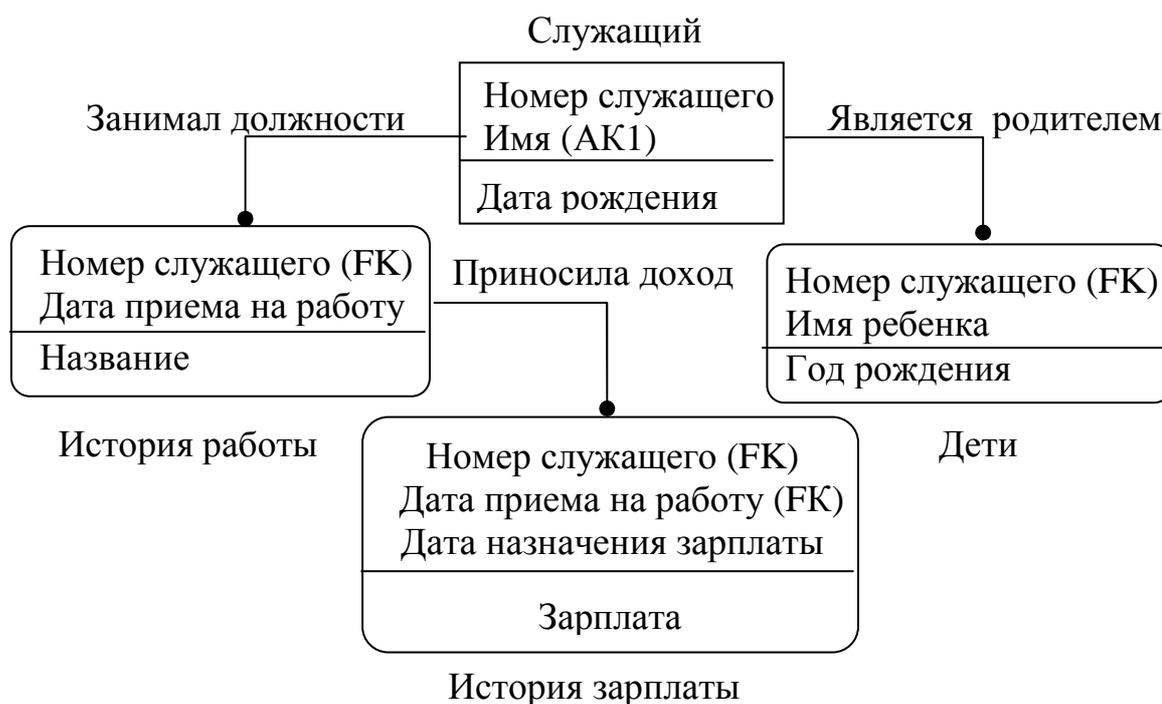


Рис. 13. Диаграмма уровня атрибутов в нотации IDEF1X

Вид той же диаграммы в нотации ИЕ (Information Engineering) показан на рис. 14.

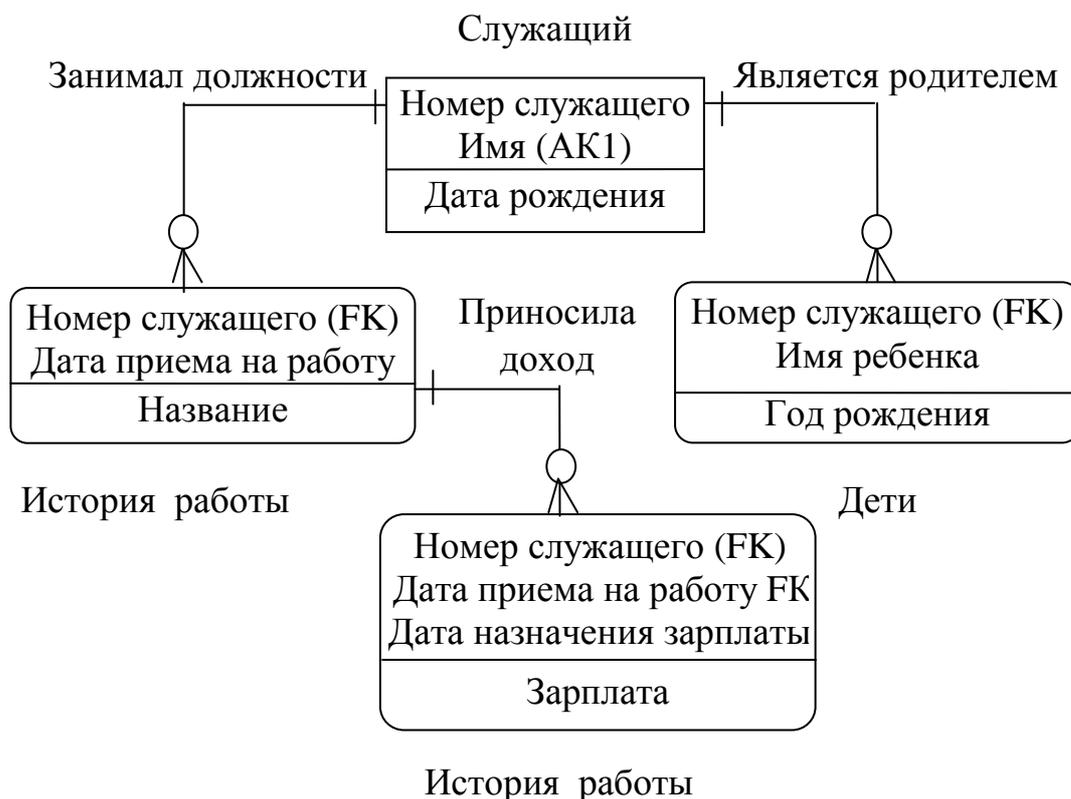


Рис. 14. Диаграмма уровня атрибутов в нотации IE

4.2.2. Физическое моделирование в ERwin

В ERwin существует два уровня представления и моделирования – логический и физический. Логический уровень означает прямое отображение фактов из реальной жизни. Например, люди, столы, отделы, собаки и компьютеры являются реальными объектами. Они именуется на естественном языке, с любыми разделителями слов (пробелы, запятые и т.д.). На логическом уровне не рассматривается использование конкретной СУБД, не определяются типы данных (например, целое или вещественное число) и не определяются индексы для таблиц.

Целевая СУБД, имена объектов, типы данных и индексы составляют второй (физический) уровень модели ERwin. ERwin дает возможность создавать и управлять двумя различными уровнями представления одной диаграммы (модели), а также позволяет иметь много вариантов отображения на каждом уровне.

4.2.3. Пример разработки физической модели в ERwin

Так как имена атрибутов и сущностей заданы на русском языке, для перехода к физическому уровню модели им следует поставить в соответствие идентификаторы таблиц, колонок и ограничений, удовлетворяющие правилам целевой СУБД (обычно это означает использование латинских букв, цифр и некоторых специальных символов). В редакторе Database Schema указываем для каждой сущности соответствующее имя таблицы. Затем в редакторе Attribute Definition задаем имена колонок таблиц, соответствующие атрибутам сущностей. ERwin и здесь обеспечивает миграцию имен колонок в подчиненные таблицы.

На этом этапе можно воспользоваться редактором Extended Attributes для определения расширенных атрибутов PowerBuilder (формата отображения, маски редактирования, правил контроля, выравнивания, заголовков и комментариев). В редакторе Relationship Definitions указывается физическое имя связи, которое соответствует имени ограничения (constraint), создаваемого ERwin в базе данных. Теперь все готово к созданию БД и нужно выбрать целевую СУБД (если этого не было сделано раньше). Выберем, например, СУБД Sybase System 10. В редакторе SYBASE Database Schema зададим типы данных для колонок таблиц. Диалог, в котором происходит выбор типа данных, приведен на рис. 15.

Теперь можно перейти к созданию базы данных. Для этого выполняется команда «Sybase schema generation». ERwin строит пакет SQL-предложений генерации базы данных. В диалоге выбора параметров для генерации БД может быть задан фильтр (генерация не всех таблиц), пакет SQL-предложений можно просмотреть (Preview), распечатать, сохранить в файле (Report), выполнить его генерацию (Generate).

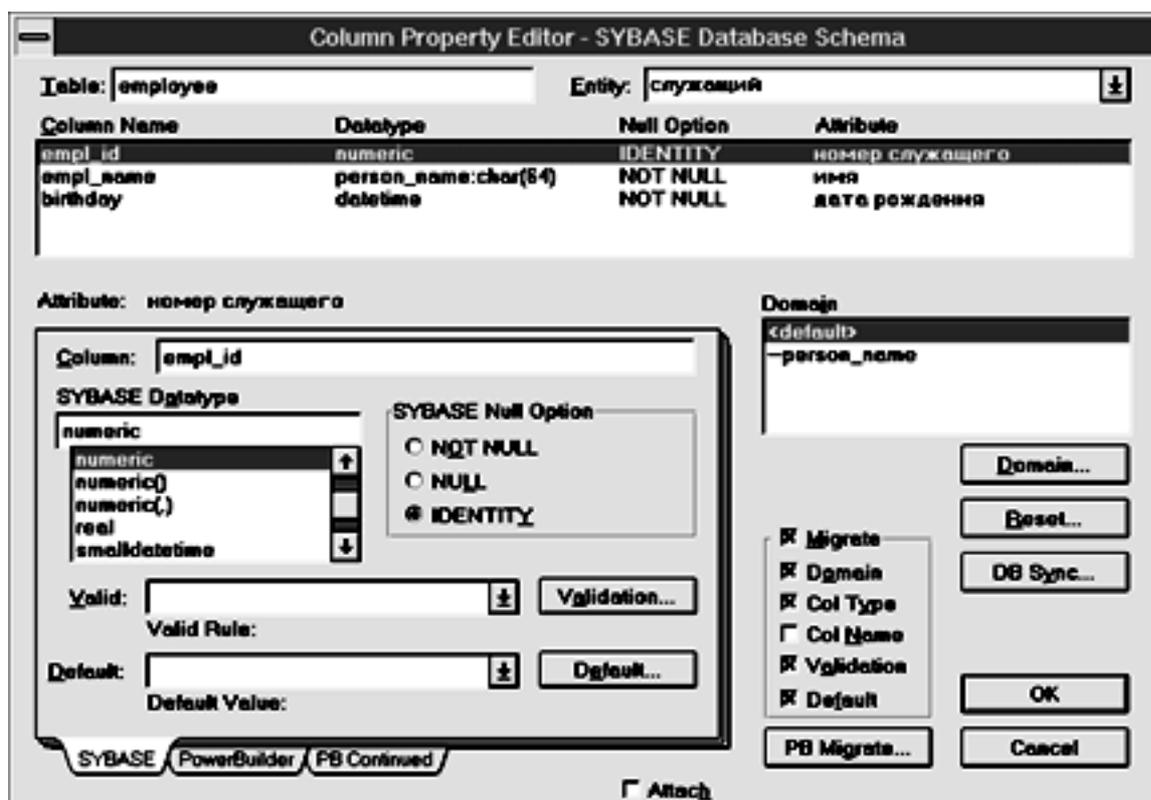


Рис. 15. Определение физической модели

Проектирование на физическом уровне выполняется с использованием терминов той базы данных, которую предполагается использовать в системе. Важно, что ERwin известны соответствия между СУБД различных производителей, вследствие чего возможно преобразование физической схемы, спроектированной для одной СУБД, в другую.

Для создания физической структуры БД может быть запрошена генерация DDL-скрипта (Data Definition Language). При этом используется диалект SQL для выбранного типа и версии сервера. Хотя сгенерированный код не нуждается в модификации, его можно сохранить в файле или распечатать.

4.2.4. Обратное проектирование

Процесс генерации физической схемы базы данных из логической модели данных называется *прямым проектированием* (*Forward Engineering*). Когда генерируется физическая схема,

ERwin позволяет включать триггеры ссылочной целостности, хранимые процедуры, индексы, ограничения. ERwin также имеет средства определения таблиц СУБД.

Процесс генерации логической модели из физической базы данных называется *обратным проектированием (Reverse Engineering)*. ERwin позволяет быстро создавать модель данных путем обратного проектирования имеющейся базы данных. После того как создана модель ERwin, можно произвести обратное проектирование структуры базы данных, а затем легко перенести его в другой формат базы данных.

Объемлющий механизм генерации схемы в ERwin, удобные возможности синхронизации с базой данных, способность подключаться ко всем ведущим источникам данных (SQL- и PC-ориентированным) облегчают проектирование, управление и перенос физических баз данных.

Обратное проектирование, то есть восстановление информационной модели по существующей базе данных, используется при выборе оптимальной платформы для настольной (desktop) базы данных или базы данных на mainframe, а также при расширении (или модификации) структуры, которая была построена без необходимой сопроводительной документации. После завершения процесса восстановления модели ERwin автоматически раскладывает таблицы на диаграмме. Теперь можно выполнять модификации уже с использованием логической схемы – добавлять сущности, атрибуты, комментарии, связи и т.д. По завершении изменений одна команда (синхронизировать модель с базой данных) актуализирует все проведенные изменения.

Построение модели может быть выполнено как на основании данных каталога базы данных, так и на основании пакета операторов SQL, с помощью которого была создана база данных.

В процессе разработки информационной системы может возникнуть ситуация, при которой структура базы данных и информационная модель не соответствуют друг другу. ERwin

дает возможность привести их в соответствие. Для этого предусмотрена функция синхронизации с базой данных. После подключения к СУБД предлагается список несоответствий между существующей структурой данных и моделью. Например, если в базе данных создана новая таблица, то ERwin предложит провести включение ее в модель. Если в модель добавлена новая таблица, ERwin предложит создать ее в реальной базе данных. При добавлении колонок в базе данных или в модели ERwin предлагает провести соответствующие операции по синхронизации.

ERwin знает о таких особенностях хранения данных в отдельных СУБД, как сегменты (в Sybase) и табличное пространство (в Oracle). Информация о физическом размещении может быть включена в модель и использована при прямом и обратном проектировании.

4.2.5. Создание отчетов

ERwin позволяет создавать собственные отчеты, обобщающие графическую информацию, которая содержится в диаграмме ERwin. Для создания отчета можно выбрать один из распространенных форматов, например формат колонок (Columnar), когда информация появляется под соответствующим заголовком, или формат меток (Labeled), когда информация появляется под соответствующей меткой (например, Entity Name: Customer).

ERwin определяет, как будет форматироваться отчет, исходя из режимов форматирования, которые задаются в редакторе Report.

ERwin позволяет создавать отчеты четырех разных типов: Entity Report, Attribute Report, Relationship Report и Constraint Report. В табл. 5 включены все доступные режимы, а также расширения файлов для каждого типа отчета, создаваемого в ERwin.

Типы отчетов и их характеристики

Тип отчета	Расширение файла	Доступные режимы
Entity	.ERE	<p>Детали сущности: имя, определение, notes и т.п.</p> <p>Определения связанных с сущностью атрибутов и связей.</p> <p>Параметры таблицы физической модели.</p> <p>Критерии валидации.</p> <p>Триггер Default или Entity Overide</p>
Attribute	.ERA	<p>Детали атрибута: имя, базовое имя, имя роли и знак ключа.</p> <p>Оглавление и определения колонок таблицы физической модели.</p> <p>Значения связанных с атрибутом ограничений</p>
Relationship	.ER	<p>Детали связи: глагольная фраза, тип связи, имена родительской и дочерней сущностей.</p> <p>Значения режимов нулевых значений, ссылочной целостности и кардинальность.</p> <p>Характеристики физической модели: физические имена родительской, дочерней сущностей и связи</p>
Constraint	.ERC	<p>Тип отчета: выберите тип Domain, Default или Validatio.</p> <p>Имя домена, тип данных и режим нулевых значений.</p> <p>Имя или значение по умолчанию.</p> <p>Имя атрибута логической модели.</p> <p>Имя колонки, тип данных колонки и имя таблицы физической модели.</p> <p>Критерии валидации</p>

ERwin поддерживает несколько распространенных форматов, в соответствии с которыми он выводит на экран данные отчета. В табл. 6 кратко описаны все режимы.

Характеристика режимов форматирования отчета

Режим форматирования отчета	Что делает этот режим
1	2
Report Format	
Labeled	Обозначает каждое значение в отчете при помощи метки (например, Attribute Name: customer_name)
Fixed	Связывает каждое значение в отчете с фиксированным заголовком колонки
Tab Delimited	Использует метки табуляции для разделения значений в отчете
Comma Delimited	Использует запятую для разделения значений в отчете
DDE Table	Пересылает данные отчета в табличной форме в другое приложение
Multi-Valued Format	
Repeating Group	Помещает связанные между собой значения в одну ячейку таблицы, после каждого значения ставит знак «+»
Filled	Повторяет значение родителя для каждого экземпляра повторяющегося дочернего значения
Header	Выводит родительское значение на экран только один раз для всех связанных с ним дочерних значений
Merge	Определяет, будет ли первая строка значения сливаться с родительским значением. Чтобы слить значения, поставьте метку «X» в окне. Чтобы оставить родительское значение в своей строке, уберите метку «X» из окна

1	2
Remove Newline	Убирает символ перехода на новую строку из конца каждой строки отчета. Используйте этот режим для того, чтобы связанные между собой значения поместились на одну строку при экспорте отчета с использованием DDE

Когда ERwin впервые открывает редактор Report, он автоматически загружает определение отчета, устанавливаемое по умолчанию для выбранного типа отчета. ERwin присваивает имя по умолчанию (например, Attribute Report), задает режимы содержания (например, Attribute Name) и форматирования (например, labeled и merge – для заголовка). Можно изменить режимы и определения, задаваемые по умолчанию, включая свои собственные. При нажатии одной из управляющих кнопок внизу редактора ERwin сразу же генерирует отчет, основываясь на заданных режимах содержания и форматирования. Отчет состоит из текстовых данных, взятых из диаграммы.

Порядок, в котором задаются режимы, определяет порядок появления заголовков в отчете. Например, если заданы режимы в следующей последовательности: 1) Attribute Name, 2) Base name, 3) Role Name, то заголовки или метки в отчете будут расположены в том же порядке.

Для получения отчета только один раз необходимо поставить метки в окнах нужных режимов и нажать кнопку Print... Для сохранения выбранных режимов в качестве спецификации, которую потом можно использовать для повторной генерации этого же отчета, следует ввести новое имя определения отчета в текстовое окно Report в верхней части окна-диалога и нажать кнопку New для сохранения определения отчета на диске как части текущей диаграммы.

Для изменения определения отчета нужно выбрать его из списка Report, расположенного в верхней части окна-диалога. Можно изменить имя отчета и (или) задать новые или другие режимы, затем следует нажать кнопку Update для сохранения изменений.

Чтобы удалить определение отчета, необходимо выбрать его из списка Report и нажать кнопку Delete. Файлы отчетов, созданные с использованием удаленного определения отчета, сохраняются.

Можно использовать кнопку Preview... в нижней части редактора Report, чтобы просмотреть содержимое отчета. Для распечатывания готового отчета следует нажать кнопку Print..., а для сохранения выходного файла отчета на диске – кнопку Report...

При нажатии кнопки Close..., расположенной внизу в редакторе Report, ERwin автоматически сохраняет определение отчета и его можно использовать повторно, не задавая снова те же самые режимы каждый раз.

По умолчанию ERwin генерирует отчет по всем сущностям текущей области. Для включения в отчет только каких-то отдельных сущностей нужно нажать кнопку Filter... ERwin открывает редактор Report Filter, содержащий список всех объектов текущей области. Список находится в окне Report Filter. Исключение объектов, не входящих в отчет, возможно с помощью кнопок Remove и Remove All. Объекты, исключенные по неосторожности, можно включить с помощью кнопок Add и Add All.

Прежде чем распечатать отчет, можно просмотреть его в окне Preview редактора Report. Можно ввести текст непосредственно в окно Preview, чтобы снабдить отчет полезной информацией и комментариями.

Чтобы открыть окно Preview, следует нажать кнопку Preview... ERwin откроет окно Preview и покажет в нем текущий отчет. Можно использовать управляющие функции окна для просмотра всего текста отчетов и изменения размера окна.

Если для распечатки или сохранения отчета используются управляющие функции окна Preview, то ERwin включает всю вводимую информацию в выходной текст отчета. Однако при закрытии окна Preview эта информация пропадает.

Отчет в ERwin состоит из двух элементов: определение отчета, в котором заданы режимы содержания и форматирования отчета, и выходной файл отчета, содержащий реальные данные, сгенерированные определением отчета. Определение отчета можно сохранить как часть текущей диаграммы (подобно области), вводя имя отчета и нажимая кнопку New или Update в редакторе Report. Выходной файл можно сохранить в отдельном текстовом файле в формате ASCII, для этого нужно нажать кнопку Report... в редакторе Report, а затем задать имя файла и путь в окне-диалоге Save As.

Можно записать выходной файл отчета на диск, используя для этого кнопку Report... в редакторе Report или в окне Preview. При записи отчета на диск выходной файл отчета сохраняется в формате ASCII. ERwin использует расширение файла отчета, задаваемое по умолчанию (Entity = .ERE, Attribute = .ERA, Relationship = .ER, Constraint = .ERC), и предлагает ввести имя отчета.

Для записи отчета на диск следует нажать кнопку Report... ERwin выводит на экран диалог Generate <Report> Type с расширением файла, задаваемым по умолчанию. Нужно ввести уникальное имя отчета и нажать кнопку ОК.

ERwin сохраняет данные отчета в отдельном файле. Можно открыть отчет ERwin из Microsoft Word, WordPerfect, Excel или любого другого приложения, обрабатывающего тексты или таблицы и читающего файлы ASCII.

4.2.6. Порядок создания отчета

1. Выбрать одну из команд меню Report (Entity Report..., Attribute Report..., Relationship Report... или Constraint Report...) для запуска соответствующего редактора.

2. Для выбора каких-то отдельных сущностей из текущей области и включения в отчет только их следует нажать кнопку Filter... в нижней части окна-диалога. ERwin открывает редактор Report Filter, в котором есть список всех сущностей области – Report Filter. Исключение из списка сущностей возможно с помощью кнопок Remove и Remove All. Для возвращения в редактор Report следует нажать кнопку Close.

3. Задать режимы, определяющие содержание и формат отчета.

Чтобы включить нужный режим, необходимо ввести «X» в окне рядом с названием режима.

Чтобы выключить включенный режим, нужно удалить «X» из окна режима.

4. Чтобы просмотреть содержание и формат отчета в ERwin, следует нажать кнопку Preview, расположенную внизу диалога. ERwin открывает окно Report Preview. Для возвращения в редактор Report нужно нажать кнопку Close.

5. Для изменения режимов содержания и форматирования можно повторить пункты 3 – 4, а затем снова просмотреть отчет.

6. Закончив формирование содержания и форматирование отчета, следует нажать кнопку Print для вывода отчета на печать, а для записи отчета на диск – кнопку Report.

7. Для сохранения определения текущего отчета с новым именем следует щелкнуть по текстовому окну Report, расположенному в верхней части диалогового окна, и, используя стандартные клавиши редактирования текста, ввести новое имя отчета и нажать кнопку New для сохранения нового определения отчета.

8. Закончив работу в редакторе Report, следует нажать кнопку Close для возвращения в окно диаграммы.

4.2.7. Определение отчета

Порядок действий при определении отчета следующий:

1. Выбрать одну из команд меню Report (Entity Report..., Attribute Report..., Relationship Report... или Constraint Report...) для входа в редактор Report.

2. Щелкнуть на стрелке «вниз» рядом со списком Report, расположенным наверху диалога, и выбрать отчет, который нужно изменить.

3. Задать режимы, определяющие содержание и формат отчета.

Чтобы изменить порядок расположения колонок, необходимо щелкнуть по тем колонкам, которые следует переставить местами. Затем снова выбрать их в нужном порядке.

Например, заголовки меток или колонок для отчета расположены в следующем порядке: 1) Attribute Name; 2) Base Name; 3) Definition. Для изменения мест Base Name и Definition следует щелкнуть по этим колонкам и убрать «X» из окон, расположенных рядом с данными заголовками. Затем необходимо щелкнуть, поставив метку сначала в окне Definition, а потом в окне Base Name. ERwin переставит местами колонки или метки: 1) Attribute Name; 2) Attribute Definition; 3) Attribute Base Name.

4. Чтобы просмотреть содержание и формат отчета в ERwin, следует нажать кнопку Preview, расположенную внизу диалога. ERwin открывает окно Report Preview. Для возвращения в редактор Report следует нажать кнопку Close.

5. Для изменения режимов содержания и форматирования можно повторить пункты 3 – 4, а затем снова просмотреть отчет.

6. Закончив формирование содержания и отчета, следует нажать кнопку Update для сохранения определения отчета. Для вывода отчета на печать необходимо нажать кнопку Print, для записи отчета на диск – кнопку Report, для возвращения в окно диаграммы – кнопку Close.

Порядок расположения информации в отчете задается последовательностью выбора режимов для отчета.

Если задается режим Fixed Column или DDE Table в групповом окне Report Format, то кнопки Print и Preview на экране становятся серыми. Чтобы просмотреть отчет в формате Fixed

Column, необходимо экспортировать отчет в другое приложение с помощью режима DDE Table.

4.2.8. Запись выходного файла отчета на диск

1. Нажать кнопку Report..., расположенную в нижней части редактора Report или Report Preview. ERwin открывает окно-диалог Generate <Report> Type с расширением, присваиваемым по умолчанию (например, .ERE, .ERA, .ER, .ERC).

2. Ввести уникальное имя отчета и нажать ОК. ERwin записывает файл отчета на диск и выходит в редактор Report.

4.2.9. Печать отчета

1. Нажать кнопку Print..., расположенную в нижней части редактора Report или Report Preview. ERwin сразу же отправит отчет на печать.

4.2.10. Удаление определения отчета

1. Выбрать одну из команд меню Report (Entity Report..., Attribute Report..., Relationship Report... или Constraint Report...) для входа в соответствующий редактор.

2. Щелкнуть на стрелке «вниз» рядом со списком Report, расположенным вверху диалога, и выбрать отчет для удаления.

3. Нажать кнопку Delete, а затем – кнопку Close для возвращения в окно диаграммы.

При удалении определения отчета в редакторе Report сохраненные файлы отчетов, созданные с использованием этого определения, остаются неизменными.

4.2.11. Использование отчетов ERwin с другими приложениями

ERwin дает возможность работать с данными отчетов ERwin в других приложениях.

Задавая режим DDE Table в редакторе Report, можно экспортировать выходной отчет ERwin в любое приложение обработки текстов или таблиц (например, в Word, Excel и др.).

При сохранении отчета ERwin в любом формате (например, columnar, labeled, tab delimited, comma delimited) как текстового файла (например, .ERE, .ERA, .ER, .ERC) возможно открытие этого файла из любого приложения, которое читает файл ASCII. Чтобы изменить внешний вид отчета Erwin, при открытии этого отчета из приложения можно использовать режимы форматирования.

Контрольные вопросы

1. Какую методологию использует ERwin?
2. Для чего применяется ERwin?
3. Какие компоненты содержит рабочее окно ERwin?
4. Кратко опишите этапы разработки логической модели.
5. Назовите составные части физической модели.
6. Что такое прямое проектирование?
7. Возможно ли с помощью ERwin привести в соответствие информационную модель и структуру базы данных?
8. Назовите типы отчетов, создаваемые с помощью ERwin.
9. Из каких частей состоит отчет в ERwin?
10. Кратко опишите процесс определения отчета.

4.3. Rational Rose

Это визуальный редактор, позволяющий моделировать программные системы любой сложности на основе графических диаграмм языка UML (Unified Modeling Language).

После запуска открывается главное окно программы, показанное на рис. 16.

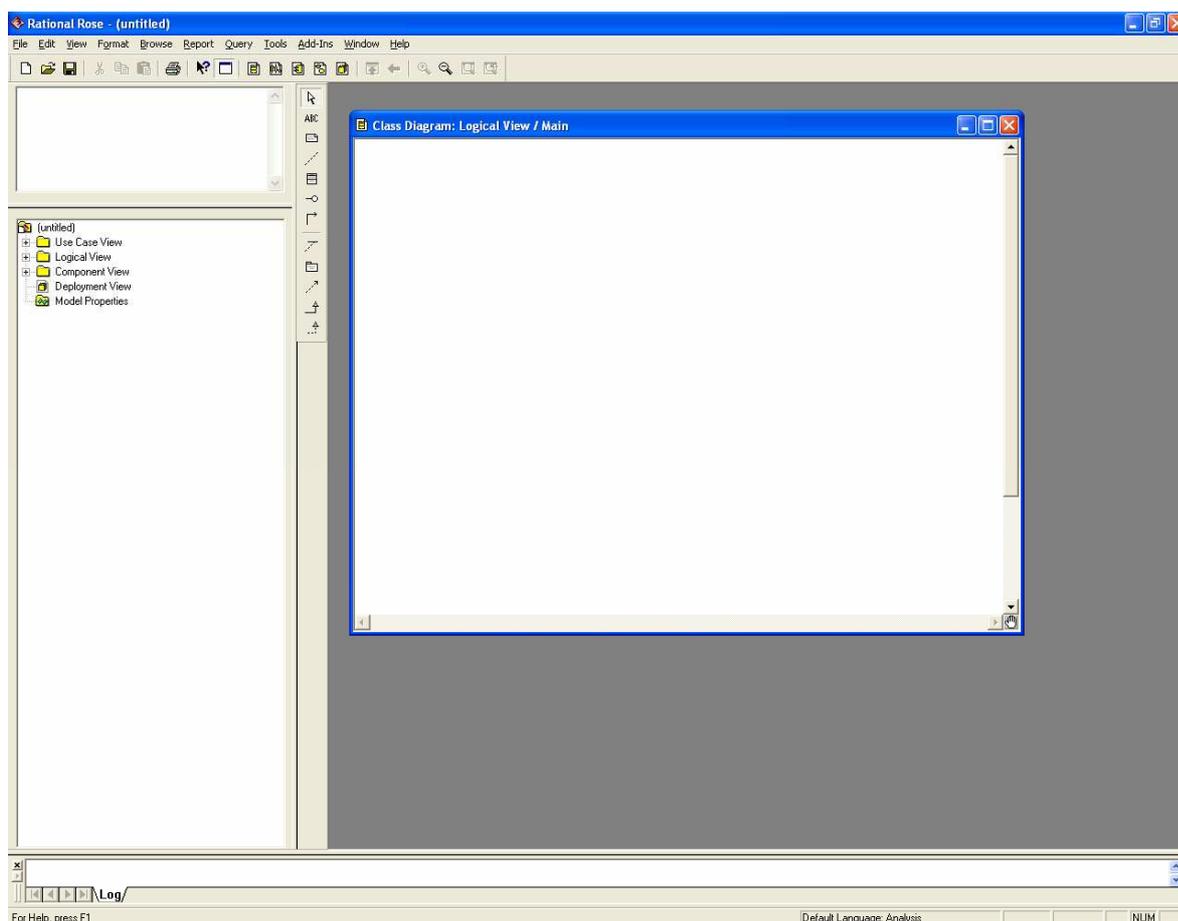


Рис. 16. Главное окно программы

В верхней части экрана находится меню и строка инструментов (Tool Bar), слева находится окно Browser для быстрого доступа к диаграммам. Под окном Browser находится окно Documentation (документация), в котором появляется описание, введенное разработчиком для выделенного в текущий момент элемента. Каждый объект в Rational Rose имеет свое контекстное меню, через которое можно изменить свойства объекта.

В распоряжение проектировщика система Rational Rose предоставляет следующие типы диаграмм, последовательное создание которых позволяет получить полное представление о всей проектируемой системе и об отдельных ее компонентах:

- Use case diagram (диаграммы сценариев);
- Deployment diagram (диаграммы топологии);
- Statechart diagram (диаграммы состояний);

Activity diagram (диаграммы активности);
Interaction diagram (диаграммы взаимодействия);
Sequence diagram (диаграммы последовательностей действий);
Collaboration diagram (диаграммы сотрудничества);
Class diagram (диаграммы классов);
Component diagram (диаграммы компонентов).

Состав и назначение пунктов меню

File (файл) предназначен для сохранения, загрузки, обновления проекта, печати диаграмм и дополнительных настроек.

Edit (редактирование) предназначен для копирования и восстановления данных в буфер обмена Windows, а также для редактирования свойств и стилей объектов.

View (вид) предназначен для настройки представления окон меню и строк инструментов.

Format (форматирование) предназначен для настройки формата текущего значка, цветовой гаммы, линий и т.д.

Browse (просмотр) предназначен для навигации между диаграммами и спецификациями диаграмм, представленных в модели.

Report (отчет) предназначен для получения различного вида справок и отчетов.

Query (запрос) предоставляет возможности контролировать, какие элементы модели будут показаны на текущей диаграмме.

Tools (инструменты) предоставляет доступ к различным дополнительным инструментам и подключаемым модулям.

Add-Ins (добавить) предоставляет доступ к менеджеру подключаемых модулей.

Window (окно) позволяет управлять окнами на рабочем столе.

Help (помощь) позволяет получать справочную информацию.

Для создания новой рабочей модели, в которой будут отражены все нюансы системы, необходимо перейти на диаграмму Use Case, как показано на рис. 17.

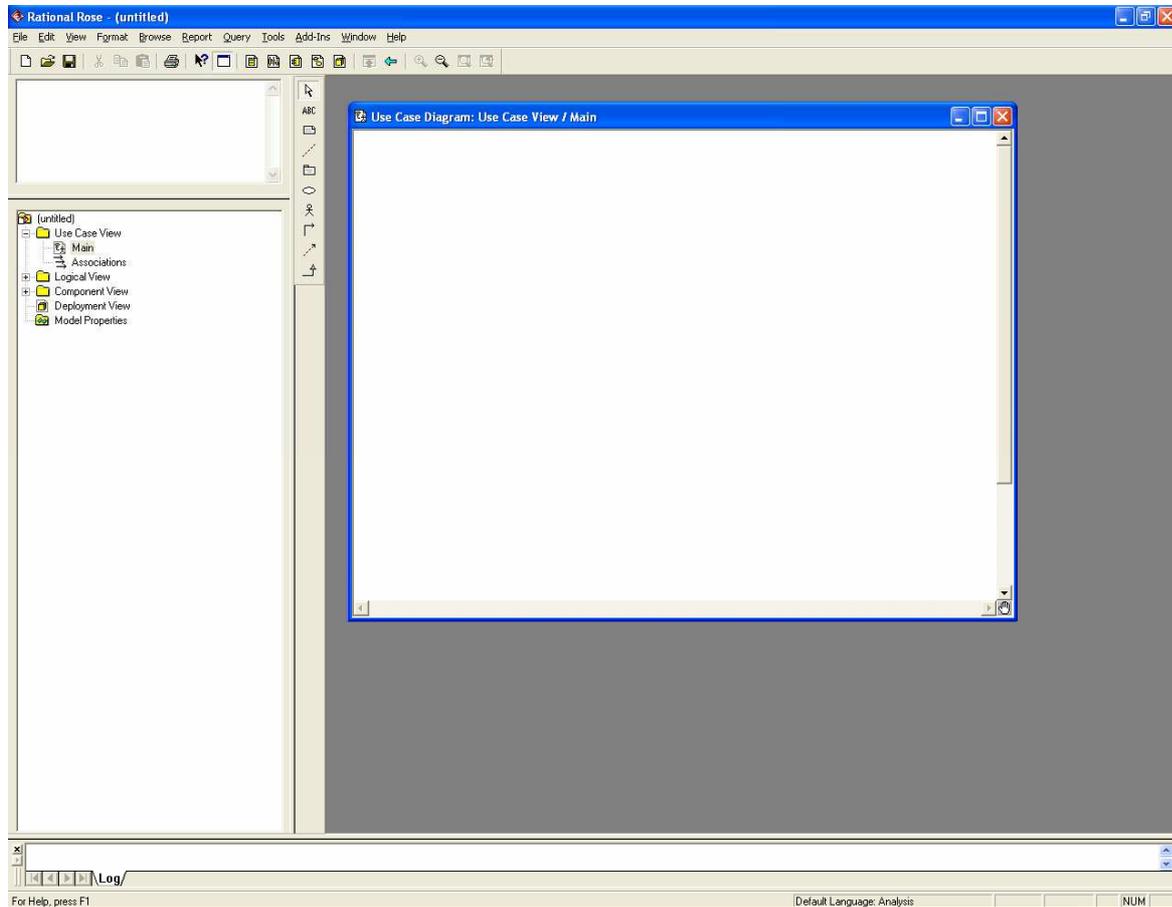


Рис. 17. Активизация Use Case диаграммы

Создание новых элементов

Rational Rose предоставляет несколько способов создания новых элементов в модели: пользуясь контекстным меню, при помощи Menu:Tools=>Create или при помощи строки инструментов.

После активизации диаграммы Use Case строка инструментов диаграммы приобретает вид, представленный на рис. 18.



Рис. 18. Строка инструментов диаграммы Use Case

Для того чтобы увеличить размер значков, можно выбрать **Toolbox=>RClick=>Use Large Buttons**.

По умолчанию строка инструментов состоит из десяти значков. Некоторые доступны только на данной диаграмме, но есть и такие, которые могут присутствовать на разных диаграммах, выполняя при этом одинаковые функции.

Selection Tool (инструмент выбора)



Основной инструмент, который позволяет выбирать элементы диаграммы, для того чтобы производить с ними дальнейшие действия. При создании нового элемента диаграммы необходимо выбрать нужный инструмент в строке инструментов, кнопка «залипает», а после создания необходимо опять перейти в режим Selection Tool.

Text Box (текст)



Данный инструмент позволяет создать произвольную надпись на диаграмме, не привязанную ни к какому элементу. Эта надпись не является полноценным элементом модели и не отображается в окне Browse, а используется как комментарий в конкретной диаграмме.

Для создания надписи необходимо нажать кнопку Text Box, при этом курсор примет вид вертикальной стрелки, и щелкнуть на том месте диаграммы, где необходимо создать надпись. В обозначенном квадратными точками в углах окне можно ввести надпись.

Для того чтобы изменить введенную надпись, нужно активировать редактирование двойным щелчком мыши.

Note (замечание) 

Данный инструмент создает элемент замечания, позволяющий вписать в него принятые во время анализа решения. Заметки могут содержать простой текст, фрагменты кода или ссылки на другие документы. Обычно окно Note соединяют с другими элементами диаграммы при помощи инструмента Anchor Note, для того чтобы показать, к какому элементу диаграммы относится замечание. В этом отличие от элемента Text Box, который располагается на диаграмме без присоединения к другим элементам.

Данный элемент не имеет ограничения на количество вводимых символов, и окно Note может быть растянуто, для того чтобы вместить необходимый текст. При активизации этого инструмента курсор принимает форму креста. Контекстное меню для значка Note позволяет кроме установки шрифта устанавливать цвет линий и заливки. Курсор принимает форму креста при создании элементов диаграммы, являющихся объектами, и форму стрелки при создании связей.

Note Anchor (якорь для замечания) 

Данный инструмент позволяет соединить элемент Note с любым элементом на диаграмме, в том числе и с другим элементом Note. Нельзя соединить между собой два элемента Note Anchor. Для того чтобы присоединить замечание к элементу диаграммы, необходимо выбрать инструмент Note Anchor, при этом курсор приобретает форму вертикальной стрелки, щелкнуть по значку Note и, не отпуская кнопки мыши, «тянуть» линию до нужного значка, по достижении которого кнопку мыши отпустить.

Аналогично происходит соединение при помощи других инструментов для установки связей.

Package (пакет)



Данный инструмент позволяет создавать пакеты, которые могут включать в себя группы элементов Use Case и в данной диаграмме может использоваться для определения более крупных сценариев поведения объектов с дальнейшей детализацией. Причем пакеты могут включать в себя другие пакеты, что позволяет создавать значительный уровень вложенности детализации.

Use Case (сценарии поведения)



Данный инструмент позволяет создавать простые формы сценариев поведения объектов системы. Это представление работы системы с точки зрения исполнителей (actors), то есть объектов, выполняющих в системе определенные функции.

Use Case могут отображать:

- образцы поведения для отдельных объектов системы;
- последовательность связанных транзакций, представляемых объектами или системой;
- получение некоторой информации объектами.

Создание Use Case необходимо для того, чтобы:

- формализовать требования к системе;
- организовать взаимодействие с будущими пользователями системы и экспертами предметной области;
- тестировать систему.

Astor (актер)



Данный инструмент используется для создания действующих лиц в системе. На диаграмме Use Case значком actor часто обозначают пользователей системы, для того чтобы определить задачи, выполняемые пользователями, и их взаимодействие.

Обычно значком Actor обозначают объект, который:

- взаимодействует с системой или использует систему;
- передает или принимает информацию в/из системы;

– является внешним по отношению к системе.

Actor позволяет узнать:

– кто пользуется системой;

– кто отвечает за сопровождение системы;

– внешнее аппаратное обеспечение, которое используется системой;

– другие системы, которые должны взаимодействовать с данной системой.

Unidirectional Association (однаправленная связь)



Данный инструмент позволяет обозначать связи между элементами. На диаграмме Use Case эти связи могут быть определены между use case и actor.

Кроме сценария поведения каждого объекта системы необходимо точно представлять взаимодействие этих объектов между собой, определение клиентов и серверов и порядка обмена сообщениями между ними. Обмен сообщениями происходит в определенной последовательности, и Sequence diagram позволяет получить отражение этого обмена во времени.

При переходе в эту диаграмму (рис. 19) на панели инструментов доступны значки Text Box, Note, Anchor to Item, Object, Message, Message to Self.

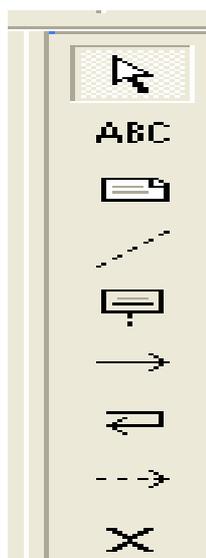


Рис. 19. Строка инструментов Sequence diagram

Object (объект) 

Позволяет включить новый объект в диаграмму. Каждый объект является реализацией класса, поэтому в нем можно указать класс, на основе которого он создан.

Message (сообщение) 

Позволяет создать сообщение, передаваемое от одного объекта к другому. Так как все взаимодействие в объектно-ориентированных системах осуществляется при помощи сообщений между объектами, то классы должны позволять отправку или прием сообщений.

Message to Self (сообщение самому себе) 

Позволяет показать, что отправитель сообщения является одновременно и его получателем.

Return Message (возврат сообщения) 

Позволяет показать, что происходит возврат управления из вызванной подпрограммы на сервере клиенту.

Destruction Marker (маркер уничтожения) 

Показывает, что происходит уничтожение программного объекта.

Диаграмма Statechart (диаграмма состояний) предназначена для описания состояний объекта и условий перехода между ними. Описание состояний позволяет точно описать модель поведения объекта при получении различных сообщений и взаимодействии с другими объектами.

После активизации диаграммы становятся доступны следующие инструменты (рис. 20).

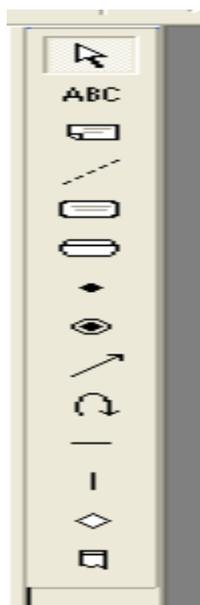


Рис. 20. Строка инструментов диаграммы Statechart

Start State (начало)



Инструмент Start State позволяет создать значок начала работы. Для диаграммы Statechart он обозначает событие, которое переводит объект в первое состояние на диаграмме. Все диаграммы состояний начинаются со значка Start State и должны заканчиваться значком End State. При этом значок начала работы может быть только один, а значков окончания может быть сколько угодно. За этим Rational Rose следит самостоятельно.

End State (завершение)



Инструмент End State позволяет создать значок окончания работы. Направление перехода может быть установлено только в данный значок, однако никаких ограничений на количество переходов в End State, а также на количество таких элементов на диаграмме не налагается.

State Transition (состояние перехода)



Инструмент State Transition позволяет создать значок состояния перехода, который означает, что объект переходит из одного состояния в другое в случае наступления определенного

события или по изменению определенных условий. Пользователь может указать несколько переходов из одного состояния в другое в случае, если каждый такой переход осуществляется при наступлении разных событий или при соблюдении разных условий.

Transition To Self (переход на себя)



Инструмент Transition To Self позволяет создать значок перехода в то же состояние, из которого осуществляется переход. Данный переход похож на State Transition, однако он не осуществляет переход в другое состояние при наступлении некоторого события. Таким образом, при наступлении события оно обрабатывается, и после обработки объект возвращается в то состояние, в котором он находился до наступления события.

Class diagram (диаграмма классов) – основная диаграмма для создания кода приложения. При помощи диаграммы классов создается внутренняя структура системы, описывается наследование и взаимное положение классов друг относительно друга.

При активизации диаграммы строка инструментов приобретает следующий вид (рис. 21).



Рис. 21. Строка инструментов для диаграммы классов

Class (класс)

Данный инструмент позволяет создать новый класс в диаграмме и модели. Для детализации модели поведения классов создаются диаграммы состояний и действий.

Класс в UML нотации изображается как прямоугольник, разделенный на три части (рис. 22). В верхней части записывается название класса, в середине – атрибуты, в нижней части – операции.

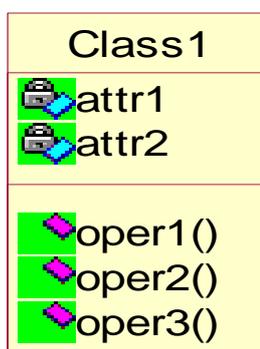


Рис. 22. Изображение класса

Interface (интерфейс)

Позволяет создать объект Interface, который указывает на видимые извне операции класса или компонента. Обычно интерфейс создается только для некоторых строго определенных классов или компонентов и предназначен скорее для логического отображения системы, но может присутствовать как на диаграмме классов, так и на диаграмме компонентов.

В диаграмме классов Interface обычно отображается как значок класса со стереотипом «interface».

Unidirectional Association (однонаправленная связь)

Значок Unidirectional Association позволяет создать однонаправленную связь класса с классом или класса с интерфейсом. Это общий и самый слабый вид связи.

Association Class (ассоциация класса)

Значок Association Class позволяет связать классы ассоциативной связью. Это свойство сохраняется в классе, и для того чтобы его установить, необходимо создать класс и связать его реляцией с другим при помощи этого значка.

Package (пакет)

Позволяет создать элемент Package, который используется для физической или логической группировки элементов.

Dependency of instantiates (зависимость реализации)

Позволяет создать связь Dependency of instantiates, при этом генератор кода C++ Rational Rose создает код класса, включающий определения зависимого класса путем генерации директивы #include. Установка этого типа связей показывает, что класс использует другой класс как параметр в одном из методов.

Generalization (обобщение)

Позволяет создать связь Generalization, для которой Rational Rose создает код наследования, то есть создается подкласс для соединенного этой связью класса, наследуемого из родительского класса.

Realize (выполнять)

Позволяет создать связь Realize между классом и интерфейсом или между компонентом и интерфейсом. Этот тип связи используется для того, чтобы показать, что класс выполняет операции, предоставляемые интерфейсом.

Для создания нового класса и помещения его на диаграмму классов можно воспользоваться соответствующим значком из строки инструментов или меню Menu:Tools==>Create=>Class. Для того чтобы поместить уже созданный класс на диаграмму классов, есть несколько путей:

- перетащить нужный класс мышкой из окна Browse;
- воспользоваться Menu:Query=>Add Classes и в диалоговом окне выбрать необходимые классы для включения в диаграмму.

Контрольные вопросы

1. Назовите область применения Rational Rose.
2. Какие компоненты содержит главное окно Rational Rose?
3. Какие диаграммы можно создавать в Rational Rose?
4. Какими способами можно создать новые элементы в модели?
5. Перечислите базовые отношения между классами.
6. Для чего создается сценарий поведения?
7. С помощью какого значка обозначают пользователя системы?
8. Какая диаграмма используется для создания кода приложения?
9. На какие три части разбивается класс в нотации UML?
10. В каком случае дополнительный и основной варианты использования связаны типом «расширение»?

Раздел 5

ОФОРМЛЕНИЕ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

5.1. Общие требования

При оформлении текстовых и графических материалов, входящих в дипломный проект (работу), следует придерживаться действующих стандартов (ГОСТ 19.ХХХ).

Текстовые документы оформляются на белых листах формата А4, графический материал допускается представлять на листах формата А3. В соответствии с общими требованиями поля листа определяются следующим образом: левое – 30 мм, правое – 10 мм, верхнее – 20 мм, нижнее – 20 мм. Формат текста: Word for Windows, через полтора интервала (около тридцати строк на листе), шрифт – Times New Roman Cyr, размер шрифта – 14, отступ первой строки абзаца – 0,75 см. Количество знаков в строке, считая пробелы, – 60. Текст программы может быть распо-

ложен в две колонки, шрифт – Times New Roman Cyr, размер шрифта – 8.

Нумерация всех страниц (в том числе и приложений) сквозная. Номер проставляется в середине верхнего поля страницы арабской цифрой. Первая страница – титульный лист, вторая страница – аннотация, с третьей страницы начинается оглавление. Номера страниц на титульном листе, аннотации и оглавлении не проставляются.

Наименование разделов, подразделов, пунктов должно быть кратким и соответствовать содержанию. Каждая новая глава (основная часть, технико-экономическое обоснование разработки, анализ мероприятий по безопасности жизнедеятельности и обеспечению экологической чистоты) печатается с новой страницы. Это же правило относится и к другим основным структурным частям работы: аннотации, оглавлению, введению, заключению, списку литературы, приложениям.

Наименование разделов (основных частей) пишется прописными буквами по центру строки. Расстояние между заголовками и текстом, а также между заголовками разделов и подразделов должно быть равно двум интервалам.

Наименования подразделов и пунктов размещаются с абзацного отступа (0,75 см) и печатаются с прописной буквы, без подчеркивания и без точки в конце. Расстояние между последней строкой текста предыдущего раздела и последующим заголовком при расположении их на одной странице должно быть равно трем интервалам.

Разделы и подразделы нумеруются арабскими цифрами с точкой. Разделы имеют порядковые номера 1, 2 и т.д. Номер подраздела состоит из номера раздела и порядкового номера подраздела, входящего в данный раздел, разделенных точкой (например: 1.1, 2.5). При использовании ссылок на пункты, разделы и подразделы указывается порядковый номер раздела или пункта (например: «в разд. 2», «в п. 2.3.1»).

Перечисления нумеруются арабскими цифрами со скобкой (например: 2), 3) и т. д.) с абзацного отступа. Для этого используются нумерованные списки. Допускается также применение маркированных списков одного вида.

Например:

1. ОСНОВНАЯ ЧАСТЬ

.....
.....

1.3. Анализ и уточнение требований к программному обеспечению

1.3.1. Анализ процесса обработки информации и выбор структур данных для ее хранения

.....
.....

1) текст текст текст текст текст ;

2) текст текст текст текст текст ;

3) текст текст текст текст текст .

.....
.....

5.2. Оформление графического материала

В соответствии с ГОСТом 2.105-79 «Общие требования к текстовым документам» иллюстрации (графики, схемы, диаграммы) могут приводиться как в основном тексте, так и в приложении. Все иллюстрации именуется рисунками.

Рисунки, таблицы и формулы нумеруются арабскими цифрами последовательно, так называемая сквозная нумерация, или в пределах раздела (относительная нумерация). В приложении – в пределах приложения.

Каждый рисунок имеет порядковый номер и название, помещаемые под рисунком по центру. Например:

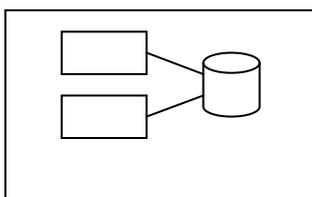
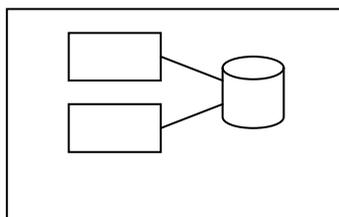


Рис. 5. Название (или)

Рис. 1.5. Название

Если рисунок занимает более одной страницы, то на всех страницах (кроме первой) проставляется порядковый номер рисунка и слово «Продолжение». Например:

1.5. Продолжение



Рисунки размещаются так, чтобы их можно было рассматривать без поворота страницы. В крайнем случае, рисунок располагается так, чтобы для просмотра нужно было повернуть страницу по часовой стрелке. Расположение и размеры полей сохраняются.

Таблицы в основном применяются для оформления цифрового материала. Шрифт – Times New Roman Cyr, размер шрифта – 14. Иногда возможен 10-й размер шрифта. Обозначения единиц физических величин необходимо применять в системе СИ. Номер таблицы размещается в правом верхнем углу над заголовком. Например:

Таблица 1.2

Сравнительная характеристика звуковых плат

Название	Название	Название	Название	Название
значения	значения	значения	значения	значения

При переносе таблицы на следующую страницу необходимо пронумеровать графы и повторить их нумерацию на следующей странице.

Сравнительная характеристика звуковых плат

Название	Название	Название	Название	Название
1	2	3	4	5
значения	значения	значения	значения	значения

Эту страницу начинают с надписи «Продолжение табл.» с указанием номера таблицы. Например:

Продолжение табл. 1.2

1	2	3	4	5
значения	значения	значения	значения	значения

Рисунки и таблицы размещаются сразу после абзаца, в котором они упоминаются в первый раз, или как можно ближе к этому абзацу на следующей странице.

При использовании ссылок на рисунки, таблицы и формулы они оформляются в следующем виде: «...(рис. 2)», или «...в табл. 5», или «...по формуле (9)».

Схемы алгоритмов должны быть выполнены в соответствии с ГОСТом. Толщина сплошной линии составляет 0,6 – 1,5 мм. Надписи на схемах выполняются чертежным шрифтом, высота букв и цифр – не менее 3,5 мм.

Приводимые формулы даются без вывода, если их автором не является сам дипломник. Все формулы должны быть пронумерованы арабскими цифрами. Номер формулы ставится с правой стороны страницы в круглых скобках на уровне формулы. Значения символов и числовых коэффициентов, входящих в формулу, должны быть приведены непосредственно под формулой, начиная со слова «где» без двоеточия. Выше и ниже каждой из формул необходимо оставлять по одной свободной

строчке. Формулы должны быть набраны с помощью редактора формул. Прописные и строчные буквы, надстрочные и подстрочные индексы в формулах должны быть обозначены четко. Рекомендуются следующие размеры знаков для формул: прописные буквы и цифры – 7 – 8 мм, строчные – 4 мм, показатели степеней и индексы не менее 2 мм. Например:

$$V_{p_{в.т}} = K_{д.р} \cdot V_{p_c}, \quad (2.10)$$

где $V_{p_{в.т}}$ – время использования вычислительной техники, ч;
 $K_{д.р}$ – количество дней разработки ПО; V_{p_c} – время работы компьютера в течение суток, ч.

5.3. Оформление приложений

Каждое приложение начинается с новой страницы с указанием в правом верхнем углу слова «ПРИЛОЖЕНИЕ» прописными буквами и имеет тематический заголовок, который располагается ниже по центру. При наличии более одного приложения все они нумеруются арабскими цифрами: ПРИЛОЖЕНИЕ 1, ПРИЛОЖЕНИЕ 2 и т. д.

Рисунки и таблицы в приложении нумеруются арабскими цифрами в пределах каждого приложения с добавлением буквы «П». Например: Рис. П1.3. Название рисунка – третий рисунок первого приложения, Таблица П1.2. Название таблицы – вторая таблица первого приложения.

Если в приложении приводится текст программы, то каждый файл оформляют как рисунок с наименованием файла и его назначением, например: Рис. П2.4. Файл `poisk.pas` – программа поиска.

5.4. Оформление списка литературы

Список литературы включает все использованные источники. В сведениях о книгах (монографиях, учебниках, пособиях,

справочниках и т.д.) содержатся: фамилия и инициалы автора, название книги, место издания, издательство, год издания. При наличии трех и более авторов указывается фамилия и инициалы только первого из них со словами «и др.». Место издания приводится полностью в именительном падеже, кроме двух городов: Москва (М.) и Санкт-Петербург (СПб.).

В сведения о статье из периодического издания включаются: фамилия и инициалы автора, наименование статьи, издания (журнала), серии (если она есть), год выпуска, том (если есть), номер издания (журнала) и диапазон страниц, на которых помещена статья.

При ссылке в тексте расчетно-пояснительной записки на источник из списка литературы указывается порядковый номер по списку литературы, заключенный в квадратные скобки, например [5]. В необходимых случаях (обычно при использовании цифровых данных или цитаты) указываются и страницы, на которых помещен используемый материал, например: [5, с. 6 – 8].

Формирование списка литературы производится в порядке ссылок или в алфавитном порядке.

Например:

1. **Перроун, П. Д.** Создание корпоративных систем на базе Java 2 Enterprise Edition [Текст] : рук. разработчика : [пер. с англ.] / Поль Дж. Перроун, Венката С. Р. «Кришна», Р. Чаганти. – М. [и др.] : Вильямс, 2001. – 1179 с.

2. **ГОСТ Р 517721–2001.** Аппаратура радиоэлектронная бытовая. Входные и выходные параметры и типы соединений. Технические требования [Текст]. – Введ. 2002–01–01. – М.: Издво стандартов, 2001. – IV, 27 с.: ил.; 29 см.

3. **Хисамутдинов, С.Н.** Аппроксимация рядов экспериментальных данных фракталом Мандельброта / С.Н. Хисамутдинов // Известия высших учебных заведений. Черная металлургия. – 2002. - № 11. – С. 63 – 64.

5.5. Употребление сокращений

Правила сокращения слов и словосочетаний устанавливаются государственными стандартами (например, ГОСТ 7.12-77). В качестве приложения к этому ГОСТу приведен список особых случаев сокращения слов и словосочетаний, которые часто встречаются в библиографическом описании, и указаны условия их применения.

При сокращении слов используются три основных способа:

- 1) оставляется только начальная буква слова (год – г.);
- 2) оставляется часть слова (отлично – отл.);
- 3) пропускается несколько букв в середине слова, вместо которых ставится дефис (университет – ун-т).

Сокращение должно оканчиваться на согласную букву и не должно оканчиваться на гласную, на букву «й», на мягкий и твердый знак.

В дипломном проекте (работе) встречаются следующие виды сокращений:

- 1) буквенные аббревиатуры;
- 2) сложносокращенные слова;
- 3) условные графические сокращения по начальным буквам слова.

Первое упоминание вводимых аббревиатур указывается в круглых скобках после полного наименования, в дальнейшем они употребляются без расшифровки. Например, программное обеспечение автоматизированной системы управления (ПО АСУ).

СПИСОК ЛИТЕРАТУРЫ

1. Антипов С.Т., Валуйский В.Я., Панфилов В.А. Алгоритм дипломного проектирования: Учебник для вузов. – М: КолосС, 2006. – 136 с.
2. Лачин В.И. Дипломное проектирование: Учеб. пособие. – М: Феникс, 2003. – 352 с.
3. Вендров А.М. CASE-технологии. Современные методы и средства проектирования информационных систем. – М.: Финансы и статистика, 1998.
4. Иванова Г.С. Технология программирования. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. – 320 с.
5. Кватрани Т. Rational Rose 2000 и UML. Визуальное моделирование. – М.: ДМК Пресс, 2001.
6. Кудрявцев Е.М. Оформление дипломного проекта на ПК. – М.: ДМК Пресс, 2004. – 224 с.
7. Леоненков А.В. Самоучитель UML. – СПб.: БХВ-Петербург, 2006. – 432 с.
8. Маклаков С.В. Моделирование бизнес-процессов с BPWin 4.0. – М.: Диалог-МИФИ, 2002.
9. Маклаков С.В. Создание информационных систем с AllFusion Modeling Suite. – М.: Диалог-МИФИ, 2003.
10. Орлов С.А. Технология разработки программного обеспечения: Разработка сложных программных систем: Учебник для вузов. – СПб.: Питер, 2002. – 463 с.
11. Трофимов С. Технология разработки программного обеспечения. – М.: Финансы и статистика, 2002.
12. Черемных С.В., Семенов И.О., Ручкин В.С. Моделирование и анализ систем. IDEF-технологии: Практикум. – М.: Финансы и статистика, 2002.
13. Единая система программной документации. – М., 1980.
14. Chen P.P. The Entity-Relationship Model: Toward a Unified View of Data // ACM Transactions on Database Systems. – 1976. – Vol. 1.

ПРИЛОЖЕНИЯ

Форма титульного листа

Приложение 1

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
ГОУ ВПО «ЧЕРЕПОВЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт _____

Специальность _____

Выпускающая кафедра _____

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Название работы _____

Студента _____

Ф.И.О.

ДОПУСТИТЬ К ЗАЩИТЕ

Директор института _____ (_____)

Зав. выпуск. кафедрой _____ (_____)

Руководитель выпуск. работы _____ (_____)

Консультант по _____ (_____)

Консультант по _____ (_____)

Нормоконтролер _____ (_____)

Выпускник _____ (_____)

Череповец

_____г.

Форма задания на выпускную квалификационную работу

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
 ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
 ГОУ ВПО «ЧЕРЕПОВЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ЗАДАНИЕ

на выпускную квалификационную работу студента _____

шифр

специальность

фамилия, имя, отчество

1. Тема работы: _____

(утверждена приказом по университету от «___» _____ № _____)

2. Срок сдачи студентом законченной работы: _____

3. Исходные данные: _____

4. Содержание расчетно-пояснительной записки (перечень вопросов, подлежащих разработке):

№	Наименование разделов и подразделов	Срок выполнения	
		по плану	фактически
1	2	3	4
	Введение		
1.	Основная часть		
1.2.	Выбор технологии, среды и языка программирования		
1.3.	Анализ и уточнение требований к программному обеспечению		
1.4.	Разработка спецификаций программного обеспечения		
1.5.	Проектирование программного обеспечения		

1	2	3	4
1.6.	Проектирование интерфейса пользователя		
1.7.	Выбор стратегии тестирования, разработка тестов, программа и методика испытаний		
2.	Технико-экономическое обоснование разработки		
3.	Анализ мероприятий по безопасности жизнедеятельности и обеспечению экологической чистоты		
	Заключение		

5. Содержание графической части:

№	Наименование плаката	Формат

6. Руководитель работы: _____

место работы

должность

фамилия, имя, отчество

подпись

7. Консультанты (по разделам содержания):

Технико-экономическое обоснование разработки

должность, фамилия, имя, отчество

подпись

Анализ мероприятий по безопасности жизнедеятельности и обеспечению экологической чистоты

должность, фамилия, имя, отчество

подпись

8. Дата выдачи задания: _____

Руководитель _____

подпись

9. Задание принял к исполнению: _____

дата

подпись студента

Форма задания по разделу «Технико-экономическое обоснование разработки»

ЗАДАНИЕ

по разделу «Технико-экономическое обоснование разработки»

студента _____ специальности 230105 «Программное обеспечение вычислительной техники и автоматизированных систем»

1. Содержание задания: рассчитать количество разработчиков, фонд заработной платы, стоимость 1 машино-часа, себестоимость и цену разработки, а также годовую экономию эксплуатационных расходов потребителя, срок окупаемости и годовой экономический эффект.

2. Перечень материалов и срок выполнения.

№	Наименование подраздела	Срок выполнения	
		по плану	фактически
2.1.	Расчет себестоимости разработки		
2.2.	Расчет цены разработки		
2.3.	Расчет экономической эффективности		

3. Рекомендуемая литература:

а) Белов А.М., Добрин Г.Н., Карлик А.Е. Экономика организации (предприятия): практикум: Учеб. пособие для вузов. – М.: ИНФРА-М, 2003. – 272 с.

б) Киселева А.В., Кудряшова Ю.В. Экономика предприятия: Учеб.-метод. пособие. – Череповец: ЧГУ, 2001.

4. Дата выдачи задания: «__» _____ 20__ г.

Консультант _____

должность, Ф.И.О.

подпись

Задание принял к исполнению «__» _____ 20__ г. _____

подпись студента

Форма задания по разделу «Анализ мероприятий по безопасности жизнедеятельности и обеспечению экологической чистоты»

ЗАДАНИЕ

по разделу «Анализ мероприятий по безопасности жизнедеятельности и обеспечению экологической чистоты»

студента _____ специальности 230105 «Программное обеспечение вычислительной техники и автоматизированных систем»

1. Содержание задания: провести анализ опасных и вредных производственных факторов, разработать мероприятия по их снижению, оценить электробезопасность и пожаробезопасность помещения, а также проанализировать все возможные неполадки и аварийные ситуации при работе с данным программным обеспечением. Обосновать наличие или отсутствие мероприятий по обеспечению экологической чистоты.

2. Перечень материалов и срок выполнения.

№	Наименование подразделов и пунктов	Срок выполнения	
		по плану	фактически
3.1.	Анализ мероприятий по безопасности жизнедеятельности		
3.1.1.	Общая характеристика безопасности		
3.1.2.	Анализ вредных производственных факторов и мероприятия по их снижению		
3.1.3.	Анализ опасных производственных факторов и мероприятия по их снижению		
3.1.4.	Оценка электробезопасности помещения и мероприятия по повышению электробезопасности		
3.1.5.	Оценка помещения по пожарной безопасности и мероприятия по повышению пожаробезопасности		
3.1.6.	Анализ неполадок и аварийных ситуаций		
3.2.	Анализ мероприятий по обеспечению экологической чистоты		

3. Рекомендуемая литература:

а) Микрюков В.Ю. Безопасность жизнедеятельности: Учебник. – М.: ФОРУМ, 2008. – 464 с.

б) Гуляев Б.В. Дипломное проектирование: раздел «Безопасность жизнедеятельности»: Учеб.-метод. пособие. – Череповец: ЧГУ, 2002.

4. Дата выдачи задания: «__» _____ 20__ г.

Консультант _____

должность, Ф.И.О.

подпись

Задание принял к исполнению «__» _____ 20__ г. _____

подпись студента

Приложение 5

Форма отзыва руководителя выпускной квалификационной работы

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

ГОУ ВПО «ЧЕРЕПОВЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт _____

Специальность _____

Выпускающая кафедра _____

**ОТЗЫВ РУКОВОДИТЕЛЯ
ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ**

Студент

фамилия, имя, отчество

группа

Тема

Форма рецензии на выпускную квалификационную работу

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
ГОУ ВПО «ЧЕРЕПОВЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт _____

Специальность _____

Выпускающая кафедра _____

**РЕЦЕНЗИЯ НА ВЫПУСКНУЮ
КВАЛИФИКАЦИОННУЮ РАБОТУ**

Студент _____

фамилия, имя, отчество

_____ группа

Тема

В рецензии следует дать общую характеристику выпускной квалификационной работы (ее достоинства, недостатки, ошибки в расчетах, определениях, формулировках и др.); качество изложения и оформления; степень использования выпускником современных источников, технологий, последних достижений науки и техники.

Пример оглавления (при использовании структурного подхода)

ОГЛАВЛЕНИЕ

Введение	4
1. Основная часть	5
1.1. Сравнительный анализ отечественных и зарубежных аналогов проектируемого объекта	5
1.2. Выбор технологии, среды и языка программирования	7
1.3. Анализ и уточнение требований к программному обеспечению	9
1.3.1. Анализ процесса обработки информации и выбор структур данных для ее хранения	9
1.3.2. Выбор методов и разработка основных алгоритмов решения задачи	11
1.4. Разработка спецификаций программного обеспечения	13
1.4.1. Построение функциональных диаграмм	13
1.4.2. Построение диаграмм описания процессов	17
1.4.3. Построение диаграмм потоков данных	20
1.4.4. Проектирование структур данных и построение диаграмм отношений компонентов данных	25
1.4.5. Построение диаграмм переходов состояний	30
1.5. Проектирование программного обеспечения	32
1.5.1. Разработка структуры программного обеспечения	32
1.5.2. Разработка функциональной схемы	34
1.5.3. Разработка модульной структуры программного обеспечения	35
1.5.3.1. Модульное описание программного обеспечения	35
1.5.3.2. Описание взаимосвязей и взаимодействий модулей ..	36
1.5.3.3. Описание интерфейсов модулей	37
1.5.3.4. Спецификации программных модулей	38
1.5.4. Разработка программных компонентов распределенной обработки данных	47
1.6. Проектирование интерфейса пользователя	52
1.6.1. Построение графа диалога	52
1.6.2. Разработка форм ввода-вывода информации	53
1.7. Выбор стратегии тестирования, разработка тестов, программа и методика испытаний	57

1.7.1. Объект испытаний	57
1.7.2. Цель испытаний	57
1.7.3. Требования к программному обеспечению	57
1.7.4. Требования к программной документации	59
1.7.5. Состав и порядок испытаний	61
1.7.6. Методы испытаний	63
1.7.7. Результаты проведения испытаний	65
2. Техничко-экономическое обоснование разработки	67
2.1. Расчет себестоимости разработки	67
2.2. Расчет цены разработки	71
2.3. Расчет экономической эффективности	73
3. Анализ мероприятий по безопасности жизнедеятельности и обеспечению экологической чистоты	78
3.1. Анализ мероприятий по безопасности жизнедеятельности ..	78
3.1.1. Общая характеристика безопасности	78
3.1.2. Анализ вредных производственных факторов и меро- приятия по их снижению	80
3.1.3. Анализ опасных производственных факторов и меро- приятия по их снижению	82
3.1.4. Оценка электробезопасности помещения и мероприятия по повышению электробезопасности	85
3.1.5. Оценка помещения по пожарной безопасности и меро- приятия по повышению пожаробезопасности	88
3.1.6. Анализ неполадок и аварийных ситуаций	91
3.2. Анализ мероприятий по обеспечению экологической чистоты	93
Заключение	94
Список литературы	95
Приложения	96
<i>Приложение 1. Техническое задание</i>	<i>97</i>
<i>Приложение 2. Схемы и/или диаграммы</i>	<i>104</i>
<i>Приложение 3. Текст программы</i>	<i>107</i>
<i>Приложение 4. Спецификация</i>	<i>112</i>
<i>Приложение 5. Руководство пользователя</i>	<i>114</i>
<i>Приложение 6. Наборы тестовых данных</i>	<i>120</i>

Пример оглавления (при использовании объектного подхода)

ОГЛАВЛЕНИЕ

Введение	4
1. Основная часть	5
1.1. Сравнительный анализ отечественных и зарубежных аналогов проектируемого объекта	5
1.2. Выбор технологии, среды и языка программирования	7
1.3. Анализ и уточнение требований к программному обеспечению	9
1.3.1. Анализ процесса обработки информации и выбор структур данных для ее хранения	9
1.3.2. Выбор методов и разработка основных алгоритмов решения задачи	11
1.4. Разработка спецификаций программного обеспечения	13
1.4.1. Построение диаграмм вариантов использования	13
1.4.2. Построение контекстной диаграммы классов	18
1.4.3. Построение диаграмм последовательностей системы для каждого варианта использования	20
1.4.4. Построение диаграмм деятельности сценариев вариантов использования	23
1.4.5. Проектирование структур данных и построение диаграмм отношений компонентов данных	25
1.4.6. Построение диаграмм переходов состояний	28
1.5. Проектирование программного обеспечения	30
1.5.1. Проектирование структуры программного обеспечения и построение диаграмм пакетов	30
1.5.2. Проектирование классов	33
1.5.2.1. Построение исходных диаграмм классов для каждого пакета	33
1.5.2.2. Построение диаграмм последовательностей действий	35
1.5.2.3. Построение диаграмм кооперации	37
1.5.2.4. Построение уточненных диаграмм классов для каждого пакета	38
1.5.2.5. Детальное проектирование классов	40
1.5.2.5.1. Разработка детальной структуры класса	40
1.5.2.5.2. Построение диаграмм состояний объектов	43
1.5.2.5.3. Построение диаграмм деятельности методов класса	45
1.5.3. Построение диаграмм компонентов	47

1.5.4. Построение диаграмм размещения	49
1.5.5. Разработка программных компонентов распределенной обработки данных	51
1.6. Проектирование интерфейса пользователя	54
1.6.1. Построение графа диалога	54
1.6.2. Разработка форм ввода-вывода информации	56
1.7. Выбор стратегии тестирования, разработка тестов, про- грамма и методика испытаний	58
1.7.1. Объект испытаний	58
1.7.2. Цель испытаний	58
1.7.3. Требования к программному обеспечению	58
1.7.4. Требования к программной документации	60
1.7.5. Состав и порядок испытаний	61
1.7.6. Методы испытаний	63
1.7.7. Результаты проведения испытаний	65
2. Техничко-экономическое обоснование разработки	67
2.1. Расчет себестоимости разработки	67
2.2. Расчет цены разработки	71
2.3. Расчет экономической эффективности	73
3. Анализ мероприятий по безопасности жизнедеятельности и обеспечению экологической чистоты	78
3.1. Анализ мероприятий по безопасности жизнедеятельности .	78
3.1.1. Общая характеристика безопасности	78
3.1.2. Анализ вредных производственных факторов и меро- приятия по их снижению	80
3.1.3. Анализ опасных производственных факторов и меро- приятия по их снижению	82
3.1.4. Оценка электробезопасности помещения и мероприятия по повышению электробезопасности	85
3.1.5. Оценка помещения по пожарной безопасности и меро- приятия по повышению пожаробезопасности	88
3.1.6. Анализ неполадок и аварийных ситуаций	91
3.2. Анализ мероприятий по обеспечению экологической чис- тоты	93
Заключение	94
Список литературы	95
Приложения	96
<i>Приложение 1.</i> Техническое задание	97
<i>Приложение 2.</i> Схемы и/или диаграммы	104
<i>Приложение 3.</i> Текст программы	107
<i>Приложение 4.</i> Спецификация	112
<i>Приложение 5.</i> Руководство пользователя	114
<i>Приложение 6.</i> Наборы тестовых данных	120

Пример заполнения спецификации на программное изделие

СПЕЦИФИКАЦИЯ*

Обозначение	Наименование	Примечание
Документация		
Расчетно-пояснительная записка	«Разработка программного обеспечения	
Приложение 2	Схемы и/или диаграммы	
Приложение 3	Текст программы	
Приложение 5	Руководство пользователя	
Приложение 6	Наборы тестовых данных	
...	...	
Комплексы**		
Компоненты		
Main.exe	Центральный модуль программы	
MathRoutines.dll	Модуль математических функций	
EngineF.doc	Модуль настройки параметров двигателя	
...	...	

* Правила оформления см. ГОСТ 19.202-78 (Переиздание (июль 1982 г.) с изменением № 1, утвержденным в сентябре 1981 г. (ИУС 11-81)

** Раздел «Комплексы» присутствует в Спецификации, когда несколько законченных программных продуктов объединяются в один глобальный программный продукт.

ОГЛАВЛЕНИЕ

Предисловие	3
<i>Раздел 1. Назначение и тематика выпускной квалификационной работы</i>	4
<i>Раздел 2. Структура выпускной квалификационной работы</i>	5
<i>Раздел 3. Порядок выполнения выпускной квалификационной работы</i>	22
<i>Раздел 4. Автоматизация дипломного проектирования с использованием CASE-средств</i>	26
<i>Раздел 5. Оформление выпускной квалификационной работы</i> ...	66
Список литературы	74
Приложения	75

**Е.В. Ершов, В.В. Селивановских,
О.Л. Селяничев, Л.Н. Виноградова**

**ТЕХНОЛОГИЯ РАЗРАБОТКИ ДИПЛОМНОГО ПРОЕКТА
С ИСПОЛЬЗОВАНИЕМ CASE-СРЕДСТВ**

Учебное пособие

Редактор Г.В. Иванова
Компьютерная верстка: Т.С. Камыгина
Лицензия А № 165724 от 11.04.06 г.

Подписано к печати 4.04.10 г. Тир. 300.
Уч.-изд. л. 6,54. Формат 60 × 84 ¹/₁₆. Усл. п. л. 5,6.
Гарнитура Таймс. Зак.

ГОУ ВПО «Череповецкий государственный университет»
162600 г. Череповец, пр. Луначарского, 5.