



МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«МИНСКИЙ ГОСУДАРСТВЕННЫЙ ВЫСШИЙ
РАДИОТЕХНИЧЕСКИЙ КОЛЛЕДЖ»

ПОДЛЕЖИТ ВОЗВРАТУ

УТВЕРЖДАЮ
Проректор по учебной работе
_____Ф. С. Шумчик
«15» мая 2013 г.

ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

**УЧЕБНАЯ ПРОГРАММА, МЕТОДИЧЕСКИЕ УКАЗАНИЯ
И КОНТРОЛЬНЫЕ ЗАДАНИЯ**

ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Учебная программа, методические указания
и контрольные задания
для учащихся заочной формы получения образования специальности
2-40 01 01 «Программное обеспечение информационных технологий»

МИНСК 2013

Минск
МГВРК
2013

УДК 32.973.2-018-02я7
ББК 681.3.068(075)
Т38

Рекомендовано к изданию кафедрой информатики (протокол № 6 от 31 января 2013 г.) и Научно-методическим советом учреждения образования «Минский государственный высший радиотехнический колледж» (протокол № 8 от 21 марта 2013 г.)

С о с т а в и т е л ь

Т. М. Тарасова, ассистент кафедры информатики МГВРК

Р е ц е н з е н т

М. А. Бельчик, старший преподаватель
кафедры информатики МГВРК

Технология разработки программного обеспечения :
Т38 учеб. программа, метод. указания и контрол. задания для
учащихся заочной формы получения образования специаль-
ности 2-40 01 01 «Программное обеспечение информацион-
ных технологий» / сост. Т. М. Тарасова. – Минск : МГВРК,
2013. – 64 с.
ISBN 978-985-526-184-2

Приведены учебная программа дисциплины, требования, предъявляемые к выполнению и оформлению контрольной работы и ее варианты, вопросы для самоконтроля, требования, предъявляемые к составлению технического задания, а также примеры его оформления и построения диаграмм.

Предназначено для учащихся и преподавателей колледжа.

УДК 32.973.2-018-02я7
ББК 681.3.068(075)

© Тарасова Т. М., составление, 2013

ISBN 978-985-526-184-2 © Учреждение образования «Минский
государственный высший радиотех-
нический колледж», 2013

Предисловие

Главной целью дисциплины «Технология разработки програм-
много обеспечения» является формирование у учащихся профес-
сиональных навыков анализа, проектирования и программирова-
ния задач различного уровня и типа с использованием стандарт-
ных наборов инструментальных средств, включая интеграцию с
CASE-системами. Особое внимание уделяется методам и про-
мышленным технологиям проектирования программных систем,
оценке качества программного обеспечения (ПО), проектирова-
нию интерфейса пользователя.

В результате изучения дисциплины учащиеся должны:

- 1) получить представление об организации ПО и технологии
его проектирования;
- 2) овладеть методом структурного подхода к проектированию:
- научиться строить функциональные модели системы и диа-
граммы потоков данных;
- 3) овладеть методом объектного подхода к проектированию
ПО:
- уметь разрабатывать объектную модель системы;
- представлять статические и динамические аспекты системы
с использованием диаграмм языка UML;
- 4) программировать интерфейсную оболочку приложения, ис-
пользуя методологию RAD;
- 5) ознакомиться с современными технологиями создания ПО;
- 6) уметь оценивать затраты на разработку ПО;
- 7) выполнять тестирование ПО;
- 8) оформлять комплект документации на ПО.

Учащиеся должны уметь разрабатывать программы в соот-
ветствии с промышленными требованиями, обеспечивая высокий
уровень качества, документированности и экономической эффек-
тивности.

Изучение дисциплины «Технология разработки программного
обеспечения» опирается на знания и умения учащихся, получен-
ные при изучении таких дисциплин, как «Основы алгоритмизации
и программирования», «Операционные системы», «Конструиро-
вание программ и языки программирования». В свою очередь
знания, умения и навыки, полученные при изучении дисциплины
«Технология разработки программного обеспечения», найдут при-
менение в курсовом и дипломном проектировании, а также в прак-
тической деятельности по специальности.

Учитывая большое практическое значение дисциплины, предусматривается выполнение значительного количества лабораторных занятий. При этом, в качестве формы организации обучения рекомендуется использовать коллективную работу учащихся как прообраз группы (бригады) программистов, моделируя обстановку максимально приближенную к реальным производственным условиям. Основными задачами лабораторных работ являются:

1) систематизация знаний, полученных при изучении теоретического курса «Технология разработки программного обеспечения»;

- 2) применение полученных знаний на практике;
- 3) получение навыков самостоятельной работы;
- 4) приобретение навыков исследовательской работы.

При коллективном проектировании несколько учащихся объединяются в творческую группу, которая разрабатывает сложную программную систему. При коллективной разработке требуется четкая организация и распределение обязанностей.

Содержание лабораторных работ имитирует реальную разработку ПО, проходя при этом все этапы его жизненного цикла (анализ, моделирование, проектирование, разработка и внедрение).

Структуру учебной дисциплины определяет логическая взаимосвязь излагаемого материала. Во введении рассматриваются основные понятия технологии программирования, индустрия создания программных систем, организация ПО. В первом разделе – понятие жизненного цикла программного обеспечения (ЖЦ ПО) и его модели, требования, предъявляемые к программным продуктам (ПП). Во втором разделе – разработка пользовательского интерфейса, методы проектирования и программирования программных комплексов. В третьем разделе – промышленные технологии проектирования ПО. В четвертом разделе – тестирование, документирование, сопровождение ПП, экономические и юридические аспекты создания и использования ПО, критерии оценки его качества.

В соответствии с действующим учебным планом на полный курс изучения дисциплины «Технология разработки программного обеспечения» по специальности 2-40 01 01 «Программное обеспечение информационных технологий» для учащихся заочной формы получения образования отводится 36 обязательных аудиторных занятий (16 – лекционных и 20 – лабораторных).

1 УЧЕБНАЯ ПРОГРАММА ДИСЦИПЛИНЫ

1.1 Примерный тематический план

Наименование раздела и темы	Количество часов			
	Дневная форма		Заочная форма	
	Лекция	Лаб. работа	Лекция	Лаб. работа
Введение	2	-	2	-
Раздел 1 Жизненный цикл ПО	6	8	2	-
Тема 1.1 Понятие и основные этапы ЖЦ ПО	2	-	1	-
Тема 1.2 Модели ЖЦ ПО	2	-	-	-
Тема 1.3 Управление требованиями к системе	2	8	1	-
Раздел 2 Методы проектирования и программирования ПО	50	42	10	18
Тема 2.1 Принципы проектирования пользовательского интерфейса	2	2	2	2
Тема 2.2 Структурный подход к разработке ПО	4	-	2	-
Тема 2.3 Функциональное моделирование. Проектирование структуры базы данных	4	8	-	8
Тема 2.4 Объектно-ориентированный подход	14	18	2	-
Тема 2.5 Методологические основы CASE-технологии	4	-	2	-
Тема 2.6 Унифицированный язык моделирования UML	16	14	-	8
Тема 2.7 XP-экстремальное программирование	4	-	2	-
Тема 2.8 Концепция шаблонов проектирования	2	-	-	-
Раздел 3 Промышленные технологии проектирования ПО	8	-	2	-
Тема 3.1 Методология RAD	2	-	1	-
Тема 3.2 Технология DATARUN	2	-	-	-
Тема 3.3 Технология RUP	2	-	1	-
Тема 3.4 Метод Oracle	2	-	-	-

Наименование раздела и темы	Количество часов			
	Дневная форма		Заочная форма	
	Лекция	Лаб. работа	Лекция	Лаб. работа
Раздел 4 Вспомогательные средства поддержки ЖЦ ПО	10	18	-	2
Тема 4.1 Управление конфигурацией ПО. Средства документирования и тестирования	6	14	-	2
Тема 4.2 Качество ПО и критерии его оценки	2	-	-	-
Тема 4.3 Оценка затрат на разработку ПО	2	4	-	-
Итого	76	68	16	20
Всего	144		36	

1.2 Содержание дисциплины

Введение

Основные понятия методологии и технологии проектирования ПО как продукции промышленного назначения. Сложность ПО. Состав и структура ПО. Становление и развитие программной «инженерии», CASE-технология.

РАЗДЕЛ 1 ЖИЗНЕННЫЙ ЦИКЛ ПО

ТЕМА 1.1 Понятие и основные этапы ЖЦ ПО

Программы с большим и малым временем жизни. ЖЦ ПО. Этапы ЖЦ ПО.

Литература [5, с. 15–32].

ТЕМА 1.2 Модели ЖЦ ПО

Модели ЖЦ ПО: спиральная, каскадная, с промежуточным контролем. Их достоинства и недостатки. ЖЦ ПО в соответствии со стандартом ISO/IEC 12207:1995.

Литература [5, с. 15–48].

ТЕМА 1.3 Управление требованиями к системе

Функциональные и нефункциональные требования. Специализированные системы управления требованиями (Requisite Pro, DOORS). Документирование требований.

Литература [5, с. 287–293].

РАЗДЕЛ 2 МЕТОДЫ ПРОЕКТИРОВАНИЯ И ПРОГРАММИРОВАНИЯ ПО

ТЕМА 2.1 Принципы проектирования пользовательского интерфейса

Принципы проектирования пользовательского интерфейса. Стратегия разработки интерфейса человек–компьютер. Составные части интерфейса: процессы ввода–вывода, процесс диалога, сообщения, проверка входных данных, подсказки.

Литература [6, с. 60–63], [8, с. 25–43].

ТЕМА 2.2 Структурный подход к разработке ПО

Принципы структурного подхода. Алгоритмическая декомпозиция системы. Иерархическое построение программной системы. Восходящее и нисходящее проектирование. Базовые конструкции. Типовая структура программного комплекса. Модульное программирование. Структура и основные принципы формирования модулей. Области видимости и локализации данных.

Литература [7, с. 17–22], [5, с. 60–63].

ТЕМА 2.3 Функциональное моделирование.

Проектирование структуры базы данных

Метод функционального моделирования SADT. Принципы построения модели IDEF0. Модели представления данных: реляционная, древовидная, сетевая. Моделирование потоков данных. Концептуальное моделирование структуры данных. Модель «сущность–связь» (ERD). Диаграмма сущностей и диаграмма «сущность–связь» в нотации Ричарда Баркера. Создание физической модели данных.

Литература [7, с. 55–63], [5, с. 63–115], [9, с. 43–187].

ТЕМА 2.4 Объектно-ориентированный подход

Объектный подход. Структурная и типовая иерархия.

Объектная декомпозиция. Объектно-ориентированный анализ, объектно-ориентированное проектирование, объектно-ориентированное программирование, прикладной анализ. Принципы абстрагирования, ограничения доступа, модульности, иерархии, типизации, параллелизма и устойчивости. Понятие объекта. Отношения между объектами. Сущность «класс». Отношения между классами. Взаимосвязь классов и объектов.

Литература [7, с. 22–30], [3, с. 23–54, с. 236–262], [5, с. 115–118].

ТЕМА 2.5 Методологические основы CASE-технологии

CASE-технология – индустриальная разработка систем обработки информации. CASE-технология: ключевые характеристики. Задачи, решаемые при помощи CASE-средств. Технология освоения и внедрения CASE-средств: определение потребностей, оценка и выбор CASE-средств, выполнение пилотного проекта, практическое внедрение CASE-средств. Классификация CASE-средств.
Литература [5, с. 185–263].

ТЕМА 2.6 Унифицированный язык моделирования UML

Концептуальная модель UML. Типичные приемы при моделировании словаря системы. Механизмы расширения.

Диаграммы вариантов использования. Диаграммы классов. Диаграммы взаимодействия: последовательности, кооперативная. Диаграммы состояний. Диаграммы деятельности. Диаграммы реализации.

Литература [7, с. 70–288], [5, с. 119–184].

ТЕМА 2.7 XP-экстремальное программирование

Понятие XP. Игра в планирование, тестирование до начала разработки, парное программирование, коллективное владение кодом, заказчик на рабочей площадке, быстрый выпуск версий, стандарты кодирования.

Литература [1, с. 13–220].

ТЕМА 2.8 Концепция шаблонов проектирования

Описание шаблонов проектирования. Понятия абстрактной фабрики, строителя, фабрики метода, прототипа, синглтона, адаптера, моста, композита, декоратора, фасада, флайвейта, прокси, интерпретатора, итератора, медиатора, наблюдателя. Каталог шаблонов проектирования. Классификации шаблонов проектирования: креационные шаблоны, структурные шаблоны, шаблоны поведения.

Литература [5].

РАЗДЕЛ 3 ПРОМЫШЛЕННЫЕ ТЕХНОЛОГИИ ПРОЕКТИРОВАНИЯ ПО

ТЕМА 3.1 Методология RAD

Подход RAD (Rapid Application Development). Особенности RAD-методологии. Среда разработки Delphi и C++ Builder. Ограничения RAD-методологии.

Литература [5, с. 48–53].

ТЕМА 3.2 Технология DATARUN

Модели, создаваемые по технологии DATARUN. CASE-средство Silverrun. Функции инструментального средства SE.

Литература [5, с. 263–271].

ТЕМА 3.3 Технология RUP

Общий вид процесса RUP. Динамические аспекты процессов: циклы, фазы, итерации, контрольные точки. Статические аспекты процессов: действия, исполнители, рабочие процессы.

Литература [5, с. 271–278].

ТЕМА 3.4 Метод Oracle

Метод разработки прикладного ПО (CDM). Метод управления проектом (PJM). Метод внедрения прикладного ПО (AIM). Реинжиниринг бизнес-процессов (BPR). Метод создания хранилищ данных (DWM).

Литература [5, с. 278–286].

РАЗДЕЛ 4 ВСПОМОГАТЕЛЬНЫЕ СРЕДСТВА ПОДДЕРЖКИ ЖЦ ПО

ТЕМА 4.1 Управление конфигурацией ПО.

Средства документирования и тестирования

Средство Rational ClearCase. Требования, предъявляемые к документации. Средство управления конфигурацией ПО (PVCS). Средства документирования. Система Soda. Средство тестирования (Rational Team Test).

Литература [5, с. 298–315].

ТЕМА 4.2 Качество ПО и критерии его оценки

Качество ПО и критерии его оценки. Надежность как один из основных показателей качества ПО. Факторы, определяющие надежность системы. Технологические методы и средства разработки качественного ПО.

Литература [5, с. 55–63], [11, с. 115–143].

ТЕМА 4.3 Оценка затрат на разработку ПО

Оценка размера разрабатываемого продукта. Оценка трудоемкости. Оценка продолжительности проекта. Оценка стоимости проекта.

Литература [5, с. 294–298].

1.3 Перечень лабораторных работ

Учитывая то, что содержание лабораторных работ имитирует реальную разработку ПО, проходя при этом все этапы жизненного цикла, необходимо соблюдать порядок выполнения работ.

Примерный перечень лабораторных работ:

- 1 Выдача и обсуждение заданий.
- 2 Написание и обсуждение технического задания (ТЗ).
- 3 Построение функциональной модели системы*.
- 4 Разработка диаграмм потоков данных*.
- 5 Построение диаграммы «сущность-связь»*.
- 6 Построение и обсуждение UML-диаграмм*.
- 7 Разработка и обсуждение технического проекта (ТП). Выбор средств реализации задачи.
- 8 Разработка интерфейса пользователя*.
- 9 Программирование задачи.
- 10 Формирование пакета данных отладки прототипа.
- 11 Тестирование и отладка.
- 12 Документирование*.
- 13 Оценка качества и затрат на создание ПП.

1.4 Примерный перечень тем для выполнения практической части контрольной работы

1 Программа создания универсальной обучающе-тестирующей оболочки, позволяющая сформировать обучающий блок и по десятибалльной системе оценить знания учащегося.

2 Программа создания универсальной обучающей оболочки, позволяющая формировать и модифицировать обучающий блок в виде гипертекстового документа.

3 Программа создания универсальной тестирующей оболочки, позволяющая по десятибалльной системе оценить знания учащегося, сохранить результаты в архив и сформировать отчетные документы.

- 4 Обработка информации приемной комиссии.
- 5 Калькулятор.
- 6 Электронные игры.
- 7 Обработка экзаменационных ведомостей.

* Работы выполняются учащимися заочной формы получения образования

- 8 АРМ сотрудника ЖЭСа.
- 9 Обменный пункт валют.
- 10 Текстовый редактор.
- 11 Графический редактор.
- 12 Эмулятор команд ОС.
- 13 АРМ библиотекаря.
- 14 АРМ сотрудника отдела кадров.
- 15 АРМ сотрудника учебного отдела (составление расписания).
- 16 Интернет-магазин.
- 17 Адресная книга (с отображением найденной информации на карте города).
- 18 Информаторий железнодорожного вокзала (с отображением информации на электронном табло).
- 19 Информаторий аэропорта (с оформлением бронирования и продажи билетов).
- 20 Справочная театральных билетных касс.
- 21 АРМ сотрудника лечебного учреждения.
- 22 Работа с графическими объектами на плоскости. Создание композиций из стандартных объектов и их модификация.
- 23 Электронная сваха.
- 24 Фотоальбом.
- 25 Ежедневник.
- 26 Будильник.
- 27 Трансформация трехмерных объектов.
- 28 Головоломки (типа sudoku и японских кроссвордов).
- 29 Генератор кроссвордов.
- 30 Ханойская башня.

2 ТРЕБОВАНИЯ К ВЫПОЛНЕНИЮ И ОФОРМЛЕНИЮ КОНТРОЛЬНОЙ РАБОТЫ

В соответствии с учебным планом дисциплины учащимся должна быть выполнена одна контрольная работа, включающая ответы на два теоретических вопроса и три практических задания. Тема для выполнения практического задания выбирается по номеру учащегося в журнале учебных занятий или предлагается учащимся и согласовывается с преподавателем.

При выполнении работы необходимо изучить теоретический материал по каждому вопросу.

Излагая ответ на теоретический вопрос, необходимо давать точные формулировки всех определений и полностью раскрывать сущность самого вопроса.

В конце каждого теоретического ответа приводится ссылка на использованную литературу или Интернет-источник.

Контрольная работа выполняется машинописным способом и сдается в срок, установленный в учебном графике. В конце работы необходимо указать список использованной литературы и оставить место для рецензии преподавателя. Вариант контрольной работы определяется последней цифрой шифра учащегося.

Проверенная контрольная работа предъявляется на экзамен. На экзамене преподаватель может проверить знания не только по билету, но и по вопросам контрольной работы. Работа должна быть подписана с указанием даты ее выполнения и иметь поля для замечаний рецензента. Страницы необходимо пронумеровать.

Порядок выполнения работы:

1) теоретическая часть: ответить на вопросы 1 и 2 со ссылкой на источники информации;

2) практическая часть (задания 3–5):

- тема для выполнения практической части выбирается из подраздела 1.4 в соответствии с номером учащегося по журналу;

- тема может быть заменена по согласованию с преподавателем;

- разработать ТЗ в соответствии с заданием 3 контрольной работы, приложением А и примером, приведенным в приложении Б;

- построить диаграмму деятельности в соответствии с заданием 4 контрольной работы и информацией, содержащейся в приложении В;

- построить диаграмму вариантов использования в соответствии с заданием 5 контрольной работы и информацией, содержащейся в приложении Г.

3 ВАРИАНТЫ КОНТРОЛЬНОЙ РАБОТЫ

Вариант 0

1 Структурный подход к разработке ПО. Принципы построения иерархии при структурном подходе. Функциональная иерархия данных, иерархия программных компонентов, свойства иерархической системы. Алгоритмическая декомпозиция, восходящее и нисходящее проектирование.

2 Моделирование потоков данных (процессов). Принципы построения модели потоков данных (DFD).

3 Разработать по утвержденной теме ТЗ на создание ПО.

4 Построить диаграмму деятельности в соответствии с требованиями, изложенными в ТЗ.

5 Построить диаграмму вариантов использования в соответствии с требованиями, изложенными в ТЗ.

Вариант 1

1 Объектно-ориентированный подход к разработке. Составляющие объектного подхода: абстрагирование, ограничение доступа, модульность, иерархия, типизация, параллелизм, устойчивость.

2 Принципы построения диаграммы «сущность-связь» в нотации Ричарда Баркера.

3 Разработать по утвержденной теме ТЗ на создание ПО.

4 Построить диаграмму деятельности в соответствии с требованиями, изложенными в ТЗ.

5 Построить диаграмму вариантов использования в соответствии с требованиями, изложенными в ТЗ.

Вариант 2

1 Структура ПО. Общее и специальное ПО. ЖЦ ПО: основные этапы. Модели ЖЦ ПО.

2 Принципы построения диаграмм деятельности.

3 Разработать по утвержденной теме ТЗ на создание ПО.

4 Построить диаграмму деятельности в соответствии с требованиями, изложенными в ТЗ.

5 Построить диаграмму вариантов использования в соответствии с требованиями, изложенными в ТЗ.

Вариант 3

1 Метод функционального моделирования SADT. Принципы построения модели IDEF0.

2 Технология освоения и внедрения CASE-средств. Классификация CASE-средств.

3 Разработать по утвержденной теме ТЗ на создание ПО.

4 Построить диаграмму деятельности в соответствии с требованиями, изложенными в ТЗ.

5 Построить диаграмму вариантов использования в соответствии с требованиями, изложенными в ТЗ.

Вариант 4

- 1 Унифицированный язык моделирования (UML). Принципы построения диаграмм вариантов использования.
- 2 XP-экстремальное программирование. Основные принципы и методики.
- 3 Разработать по утвержденной теме ТЗ на создание ПО.
- 4 Построить диаграмму деятельности в соответствии с требованиями, изложенными в ТЗ.
- 5 Построить диаграмму вариантов использования в соответствии с требованиями, изложенными в ТЗ.

Вариант 5

- 1 Качество ПО и критерии его оценки. Надежность как один из основных показателей качества ПО. Факторы, определяющие надежность системы.
- 2 Унифицированный язык моделирования (UML). Принципы построения диаграмм состояния.
- 3 Разработать по утвержденной теме ТЗ на создание ПО.
- 4 Построить диаграмму деятельности в соответствии с требованиями, изложенными в ТЗ.
- 5 Построить диаграмму вариантов использования в соответствии с требованиями, изложенными в ТЗ.

Вариант 6

- 1 Три аспекта согласованности интерфейса. Стратегия разработки интерфейса «человек–компьютер». Форматирование экрана. Составные части интерфейса: процессы ввода–вывода, процесс диалога, сообщения, проверка входных данных, подсказки. Многооконные WIMP-интерфейсы.
- 2 Типы ошибок при проектировании сложной системы. Методы отладки и тестирования программ. Сопровождение ПО.
- 3 Разработать по утвержденной теме ТЗ на создание ПО.
- 4 Построить диаграмму деятельности в соответствии с требованиями, изложенными в ТЗ.
- 5 Построить диаграмму вариантов использования в соответствии с требованиями, изложенными в ТЗ.

Вариант 7

- 1 Документирование ПП. Структура документации. Единая система программной документации (ЕСПД). Централизованный

фонд алгоритмов и программ (ЦФАП). Передача программных средств в ЦФАП.

- 2 Унифицированный язык моделирования (UML). Принципы построения диаграмм классов.
- 3 Разработать по утвержденной теме ТЗ на создание ПО.
- 4 Построить диаграмму деятельности в соответствии с требованиями, изложенными в ТЗ.
- 5 Построить диаграмму вариантов использования в соответствии с требованиями, изложенными в ТЗ.

Вариант 8

- 1 Понятие объекта. Отношения между объектами. Понятие класса. Отношения между классами. Взаимосвязь классов и объектов.
- 2 Унифицированный язык моделирования (UML). Принципы построения диаграмм последовательности и кооперации.
- 3 Разработать по утвержденной теме ТЗ на создание ПО.
- 4 Построить диаграмму деятельности в соответствии с требованиями, изложенными в ТЗ.
- 5 Построить диаграмму вариантов использования в соответствии с требованиями, изложенными в ТЗ.

Вариант 9

- 1 Объектно-ориентированный анализ, проектирование и программирование. Методы прикладного анализа.
- 2 Унифицированный язык моделирования (UML). Диаграммы реализации.
- 3 Разработать по утвержденной теме ТЗ на создание ПО.
- 4 Построить диаграмму деятельности в соответствии с требованиями, изложенными в ТЗ.
- 5 Построить диаграмму вариантов использования в соответствии с требованиями, изложенными в ТЗ.

4 ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ

- 1 Сложная система. Признаки сложной системы.
- 2 ЖЦ ПО. Модели ЖЦ ПО.
- 3 Алгоритмическая декомпозиция.
- 4 Восходящее и нисходящее проектирование.
- 5 Функциональная иерархия данных.

- 6 Базовые конструкции, используемые при разработке структурированной программы.
- 7 Принципы построения модели IDEF0.
- 8 Диаграмма потоков данных (DFD).
- 9 Модель «сущность-связь» (ERD) в нотации Ричарда Баркера.
- 10 Объектная декомпозиция. Отличие от алгоритмической.
- 11 Структурная и типовая иерархия при объектном подходе.
- 12 Признаки сложной системы при объектном подходе.
- 13 Класс. Отношения между классами.
- 14 Объект. Отношения между объектами.
- 15 Объектный подход. Принципы абстрагирования, ограничения доступа, модульности, иерархии, типизации, параллелизма и устойчивости.
- 16 Объектно-ориентированный анализ, проектирование и программирование.
- 17 Методы прикладного анализа. Отличие от объектно-ориентированного анализа.
- 18 Задачи, решаемые с помощью CASE-систем.
- 19 CASE-технология: ключевые характеристики.
- 20 Классификация CASE-средств.
- 21 Технология освоения и внедрения CASE-средств, выполнение пилотного проекта.
- 22 Унифицированный язык моделирования (UML). Структурные диаграммы. Диаграммы поведения системы.
- 23 Принципы построения UML-диаграмм.
- 24 Методология RAD, ее особенности.
- 25 Методологии RUP, DATARUN и Oracle.
- 26 Принципы и методики экстремального программирования.
- 27 Качество ПО и критерии его оценки.
- 28 Надежность как один из основных показателей качества ПО. Факторы, определяющие надежность системы.
- 29 Стратегия разработки интерфейса «человек–компьютер».
- 30 Составные части интерфейса: процессы ввода–вывода, процесс диалога, сообщения, проверка входных данных, подсказки.
- 31 Три аспекта согласованности интерфейса.
- 32 Методы отладки и тестирования программ.
- 33 Типы ошибок.
- 34 Сопровождение ПО. Задачи этапа сопровождения.
- 35 Документирование ПП.
- 36 Оценка затрат на разработку ПО.

Рекомендуемая литература

- 1 Бек, К. Экстремальное программирование : пер. с англ. / К. Бек. – СПб. : Питер, 2003. – 224 с.
- 2 Бозм, Б. У. Инженерное проектирование программного обеспечения : пер. с англ. / Б. У. Бозм. – М. : Радио и связь, 1985. – 511 с.
- 3 Буч, Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++ : пер. с англ. / Г. Буч. – 2-е изд. – М. : Бином ; СПб. : Невский диалект, 2000. – 560 с. : ил.
- 4 Буч, Г. Язык UML. Руководство пользователя : пер. с англ. / Г. Буч, Д. Рамбо, А. Джекобсон. – М. : ДМК, 2000. – 432 с. : ил.
- 5 Вендров, А. М. Проектирование программного обеспечения экономических информационных сетей : учебник / А. М. Вендров. – М. : Финансы и статистика, 2000. – 352 с. : ил.
- 6 Калянов, Г. Н. CASE. Структурный системный анализ (автоматизация и применение) / Г. Н. Калянов. – М. : Лори, 1996. – 242 с.
- 7 Леоненков, А. UML : самоучитель / А. Леоненков. – 2-е изд. – СПб. : БХВ-Петербург, 2004. – 427 с.
- 8 Липаев, В. В. Системное проектирование сложных программных средств для информационных систем / В. В. Липаев. – М. : СИНТЕГ, 1999. – 224 с.
- 9 Маклаков, С. В. Bpwin и Erwin. CASE-средства разработки информационных систем / С. В. Маклаков. – М. : Диалог-МИФИ, 1999. – 304 с.
- 10 Марка, Д. А. Методология структурного анализа и проектирования / Д. А. Марка, К. МакГоуэн. – М. : МетаТехнология, 1993. – 240 с.
- 11 Петров, В. Н. Информационные системы / В. Н. Петров. – СПб. : Питер, 2002. – 688 с. : ил.
- 12 Программное обеспечение CASE // Computerweek. – М., 1996. – № 19.

Требования, предъявляемые к составлению ТЗ

Министерство образования Республики Беларусь
Учреждение образования
«Минский государственный высший радиотехнический колледж»

Руководитель _____/Фамилия И. О./
(подпись)

Разработчик _____ /Фамилия И. О./
(подпись)

201_

(Год разработки)

- 1 Общие сведения
 - 1.1 Формулировка задания
 - 1.2 Цели, достигаемые разработкой
 - 1.3 Категории пользователей
 - 1.4 Наименование организации заказчика
 - 1.5 Основание для проведения работ
- 2 Описание предметной области
 - 2.1 Описание (схемы) бизнес-процессов
 - 2.2 Состав данных и алгоритмы обработки информации
 - 2.3 Недостатки существующих проектных решений
 - 2.4 Текущий уровень автоматизации
- 3 Требования к разработке
 - 3.1 Информационная модель
 - 3.2 Структура меню
 - 3.3 Функциональные требования
 - 3.4 Требования к информационному обеспечению
 - 3.5 Требования к интерфейсу
 - 3.6 Требования к алгоритмам
 - 3.7 Прочие требования
- 4 Порядок контроля и обеспечение качества
 - 4.1 Экспертиза
 - 4.2 Тестирование
 - 4.3 Опытная эксплуатация
- 5 Требования к документированию
 - 5.1 Требования к справочной подсистеме
 - 5.2 Требования к документации пользователя
- Приложения

Примерное содержание разделов ТЗ

Данный раздел должен содержать информацию, предоставляющую общие сведения о проекте.

Краткая формулировка задания на разработку.

1.2 Цели, достигаемые разработкой

Например, «Автоматизация бизнес-процедур, выполняемых при заключении и исполнении долгосрочных договоров ...»

1.3 Категории пользователей

Категории пользователей, на которых должен быть ориентирован результат разработки (директор, менеджер, бухгалтер и т. д.).

Области применения (универсальное решение, отражение специфики отраслей, регионов, отдельных организаций). Дополнительно указываются ограничения на применение, обусловленные спецификой предметной области, либо проектными решениями, например, «... предназначено только для организаций розничной торговли, ведущих количественно-стоимостной учет».

1.4 Наименование организации заказчика

Наименование организации заказчика (заказчиков), город, фамилия, имя, отчество, телефон и адрес электронной почты ответственных представителей.

1.5 Основание для проведения работ

Основание для проведения работ: приказ, распоряжение, план разработки версии, договор с заказчиком.

2 Описание предметной области

Данный раздел должен содержать информацию, достаточную для понимания разработчиками сути, специфики и объема работ по ТЗ.

В случаях, когда разработке ТЗ предшествовала стадия анализа и имеется документ «Описание предметной области», содержащий необходимую информацию, он включается в состав приложений к ТЗ со ссылкой в данном разделе. При необходимости здесь могут быть даны пояснения и дополнения. Дублирование информации, содержащейся в указанных документах, не производится.

2.1 Описание (схемы) бизнес-процессов

Описываются состав и алгоритмы выполнения бизнес-процессов предметной области. Приводятся источники формирования первичных документов, общие правила обработки и формирова-

ния выходных документов. Формы входных и выходных документов с примерами заполнения приводятся в приложении. Указываются количество и периодичность поступления первичных документов и формирования отчетов, другие характеристики (для конкретных заказчиков либо типичных случаев).

Схемы бизнес-процессов включаются по согласованию с разработчиком ТП или исполнителем, если ТП не разрабатывается.

При необходимости могут включаться описания (схемы) организационной структуры конкретных предприятий, перечни функциональных обязанностей.

Рекомендуется пояснять описания и схемы конкретными примерами (включая числовые), вынося их в приложения.

2.2 Состав данных и алгоритмы обработки информации

По каждому документу или потоку данных дается перечень учитываемых реквизитов (показателей). Приводится описание процедур обработки, взаимосвязь различных документов (текстовое или в виде схем).

2.3 Недостатки существующих проектных решений

Краткое описание недостатков или указание на отсутствие проектных решений, необходимых для реализации описанных процессов в существующих аналогах или существующей версии (номер).

2.4 Текущий уровень автоматизации

Раздел при необходимости включается только для заказных доработок и содержит сведения, поясняющие требования к разработке (например: техническая и программная оснащенность, наличие программ-аналогов и взаимодействующих систем).

3 Требования к разработке

3.1 Информационная модель

Приводятся текстовое описание и схемы реализуемых бизнес-объектов (сущностей), их атрибутов, состояний и взаимосвязей.

Приводится схема, отображающая новые объекты, их взаимосвязи между собой и объектами, существующими в системе (рисунок А.2).

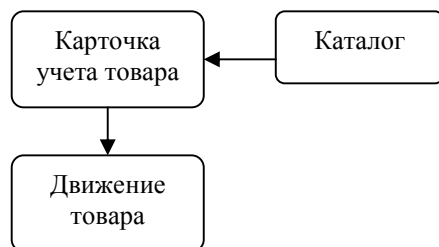


Рисунок А.2

Приводится перечисление наиболее существенных для дальнейшей реализации атрибутов объектов с указанием возможных значений.

П р и м е ч а н и е – Подраздел 3.1 является обязательным и не подлежит исключению.

3.2 Структура меню

В виде схемы приводится развернутая структура меню модуля. Для нового модуля изображается целиком (кроме меню администратора и общесистемного), для дорабатываемого – только те фрагменты меню, в которые вносятся изменения. При этом новые функции выделяются жирным шрифтом или курсивом.

Пример:

Документы

Карточка учета товара

Накладная на внутреннее перемещение

...

Накладная на реализацию

Акт на списание

Накладная на возврат

от покупателя

поставщику

на оптовый склад

3.3 Функциональные требования

Здесь приводится перечень функций (режимов выполнения), с указанием назначения и краткой характеристики.

Описание может быть выполнено в виде таблицы. Примерная форма перечня (допускается расширение путем добавления колонок, например: ограничения, примечания и т. д.) – таблица А.1.

Т а б л и ц а А.1

Раздел меню, наименование	Назначение, ссылка на описание бизнес-процесса	Использует объекты (входные данные)	Оперировать с объектами (выходные данные)
Меню «Документы»			
Приходные накладные	Ввод и редактирование накладных, приход от поставщика непосредственно в розницу (раздел 2.XX, рисунок N, операция M)	Каталог подразделений Каталог МОЛ Каталог контрагентов Каталог КУТ	Создает приходы на заданный разрез хранения
Акт на списание	Ввод и редактирование (раздел 2.XX, рисунок N, операция M)	Каталог подразделений Каталог МОЛ Записи по приходам	Записи по расходам выбранных приходов

Ограничения по работе с системой, предусмотренные данной функциональностью, например, «... необходимо ведение учета МЦ по партиям».

П р и м е ч а н и е – Подразделы 3.2 и 3.3 при описании новых модулей или функций считаются обязательными, исключение не допускается. Если предметом ТЗ является одиночная функция, допускается вместо структуры меню указывать только путь (способ активизации), а вместо табличного описания приводится текстовое.

3.4 Требования к информационному обеспечению

Требования по составу и структуре входных данных, включая:
- данные, вводимые пользователем при работе с функциями, перечисленными в подразделе 3.2 «Функциональные требования»;
- данные, хранящиеся в базе данных (БД) системы.

Требования по составу и структуре выходных данных.

Требования по реализации печатных форм первичных и отчетных документов.

Требования по организации обмена данными между компонентами (модулями) системы.

Требования по срокам хранения данных.

Требования по организации обмена данными с другими программами (включая экспорт/импорт данных).

Требования по использованию общегосударственных (отраслевых) классификаторов, справочников, перечней.

Требования по изменению (дополнению) общесистемных справочников и классификаторов.

Требования по изменению (дополнению) параметров настройки. Требования по изменению (дополнению) других таблиц БД. Требования по использованию конкретных СУБД. Требования по обеспечению конвертации.

Примечание – При необходимости последовательность перечисленных выше требований может быть изменена.

3.5 Требования к интерфейсу

Приводятся требования к пользовательскому интерфейсу, не оговоренные действующим стандартом либо детализирующие его положения.

3.6 Требования к алгоритмам

Краткие описания алгоритмов, которые должны быть изменены или специально разработаны для обеспечения функциональных требований.

3.7 Прочие требования

Требования по изменению (дополнению) общесистемных и сервисных функций.

Требования по обеспечению быстродействия.

Требования по обеспечению надежности.

Требования по защите информации.

Требования по совместимости с имеющимися у заказчика аппаратными и программными средствами (только для заказных доработок).

Другие требования.

Примечание – В подразделе 3.7 не следует приводить требования, которые уже реализованы в системе или описаны в других подразделах.

4 Порядок контроля и обеспечение качества

4.1 Экспертиза

Требуется (не требуется) разработка макета на стадии «Технический проект» на уровне (меню, основных интерфейсов и т. д.).

Требуется (не требуется) проведение экспертизы с участием представителей (наименования служб и подразделений, внешних организаций):

- ТП;
- макета.

4.2 Тестирование

Тестирование проводить в соответствии с требованиями (наименование документа).

Требуется (не требуется) участие в тестировании представителей (наименования служб и подразделений, внешних организаций) на этапе (внутреннего, внешнего) тестирования.

4.3 Опытная эксплуатация

Требуется (не требуется) проведение опытной эксплуатации в перечисленных далее внешних организациях:

- 1 Организация 1
- 2 Организация 2
- 3 ...

Вместо наименований могут указываться также типы организаций.

5 Требования к документированию

5.1 Требования к справочной подсистеме

Приводятся требования к справочной подсистеме, не оговоренные действующим стандартом либо детализирующие его положения.

5.2 Требования к документации пользователя

Требования:

- по разработке отдельной книги документации;
- дополнения существующих книг (новые разделы, либо добавление в существующие разделы).

Дополнительно могут указываться примерный объем и структура документации.

Примечание – Могут включаться требования по разработке или представлению иных видов документации, включая проектную.

Приложения

Перечень документов, прилагаемых к ТЗ, включая копии законодательных (нормативных) актов, форм первичных и отчетных документов (и т. д.) в виде таблицы А2.

Т а б л и ц а А.2

Номер	Наименование (источник, дата публикации)	Кол-во страниц	Файл
1			
2			
3			

Графа «Файл» заполняется, если приложение передается в виде отдельного файла. Допускается также его включение в виде объекта средствами MS Word, использование гиперссылок.

Приложение Б (справочное)

Пример оформления ТЗ

Оформление титульного листа

Министерство образования Республики Беларусь
Учреждение образования
«Минский государственный высший радиотехнический колледж»

«АВТОМАТИЗАЦИЯ СКЛАДСКОГО УЧЕТА АВТОСЕРВИСА»
Версия 1.01

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Руководитель _____ /Н. Н. Нестеров/
(подпись)

Разработчик _____ /А. Б. Николаев/
(подпись)

2013

Оформление содержания

Содержание	
1 Общие сведения	3
1.1 Формулировка задания	3
1.2 Цели, достигаемые разработкой	3
1.3 Категории пользователей	3
1.4 Наименование организации заказчика	3
1.5 Основания для проведения работ	3
2 Описание предметной области	3
2.1 Описание (схемы) бизнес-процессов	3
2.2 Состав данных и алгоритмы обработки информации	4
2.3 Недостатки существующих проектных решений	4
2.4 Текущий уровень автоматизации	4
3 Требования к разработке	5
3.1 Информационная модель	5
3.2 Структура меню	5
3.3 Функциональные требования	6
3.4 Требования к информационному обеспечению	7
3.5 Требования к интерфейсу	7
3.6 Требования к алгоритмам	7
4 Порядок контроля и обеспечение качества	7
4.1 Экспертиза	7
4.2 Тестирование	7
4.3 Опытная эксплуатация	8
5 Требования к документированию	8
5.1 Требования к справочной подсистеме	8
5.2 Требования к документации пользователя	8
Приложение А Пароль Администратора	9
Приложение Б Накладная на получение товара организацией ОАО «Руно-Авто»	10
Приложение В Квитанция, выдаваемая клиенту при покупке запчастей	11
Приложение Г Квитанция, выдаваемая заказчику при сдаче машины в ремонт	12
Приложение Д Квитанция, выдаваемая заказчику при получении машины из ремонта	13
Приложение Е Квитанция, выдаваемая клиенту при покупке автомобиля	14

Оформление разделов ТЗ

1 Общие сведения

1.1 Формулировка задания

Разработать ПП, предназначенный для автоматизации складского учета крупного автосервиса. ПП должен обеспечивать ведение БД, учет поступления товара, внутреннее движения товара по складам, формирование, обработку и ввод документации.

1.2 Цели, достигаемые разработкой

Автоматизация обработки информации и документооборота малого предприятия, занимающегося ремонтом и продажей автомобилей, а также продажей запасных частей по наличному и безналичному расчету.

1.3 Категории пользователей

ПП должен быть рассчитан на работника склада, обслуживающего автосервис.

1.4 Наименование организации заказчика

Заказчиком на разработку данного ПП является ОАО «Руно-Авто» (ул. Калинина, 12/12, 220393, г. Минск, тел./факс 278-65-42).

1.5 Основания для проведения работ

Основанием для разработки данного ПП является договор между Заказчиком (ОАО «Руно-Авто») и Разработчиком (ООО «ЯВХС») от 10.09.2012 г.

2 Описание предметной области

2.1 Описание процессов

Автосервис ОАО «Руно-Авто» выполняет следующие функции: ремонт автомобилей, продажа автомобилей, запчастей и сопутствующих товаров по наличному и безналичному расчету.

Предприятие имеет три склада. ПП должен не только учитывать товар, поступающий на склады головного предприятия, но и осуществлять учет внутреннего перемещения товаров по складам, корректировку БД, создание стандартных форм на списание, заключение договоров. Должны быть автоматизированы бизнес-процедуры, выполняемые при заключении и исполнении договоров.

С помощью разрабатываемого ПП должны осуществляться учет и обработка как первичных документов (договоров с поставщиками, договоров с заказчиками), так и выходных (товарных накладных (ТН) на получение товара, актов отгрузки, актов приема полученных товаров, гарантийных талонов). ПП должен осуществлять учет внутреннего движения товаров по складам посредством корректировки БД.

Доступ к информации имеет администратор после успешного завершения процедуры авторизации.

ПП должен осуществлять ведение БД, обработку запросов на поиск и фильтрацию информации по следующим критериям: дате поступления на склад, дате продажи, стоимости, отметке о выполнении работ, фамилии, имени, отчеству или коду заказчика; формировать следующие документы: гарантийный талон, договор о продаже, чек-квитанцию на сданную в ремонт технику.

По запросу пользователя может быть осуществлена настройка интерфейса каждого из складов.

Должна быть предусмотрена возможность расчета скидки для постоянных клиентов.

2.2 Состав данных и алгоритмы обработки информации

Входными данными для разрабатываемого ПП являются:

- пароль администратора (приложение А);
- параметры настройки, перечисленные в подразделе 3.3

«Функциональные требования»;

- данные, вводимые пользователем в таблицы БД;
- критерии поиска, фильтрации;
- тип формируемого документа.

Выходными данными разрабатываемого ПП являются:

- информация, отображаемая на экран по запросу пользователя;

- документы, создаваемые разработанным ПП (структура документации приведена в приложениях Б–Е).

Постоянными данными являются:

- шаблоны документов;
- БД.

Разрабатываемый ПП должен создавать и модифицировать следующие БД:

- *БД продажи автомобилей, содержащая две таблицы:*

- таблица поступления автомобилей со структурой: марка автомобиля, все реквизиты поставщика, технические характеристики, цена, дата подписания договора (эта таблица содержит ссылку на подфункцию, формирующую договор поставки на продажу);

- таблица продажи автомобилей со структурой: марка автомобиля, цена, дата подписания договора;

- *БД запасных частей, содержащая три таблицы:*

- таблица учета поступления запчастей со структурой: все реквизиты поставщика, наименование и номер документа, подтверждающего поставку, наименование детали, себестоимость, розничная цена, дата поступления (таблица имеет ссылку на подфункцию формирования документов, подтверждающих получение запчастей (приложение В));

- таблица учета продажи запчастей со структурой: реквизиты покупателя, наименование детали, цена, дата продажи;

- таблица учета внутреннего перемещения запчастей со структурой: наименование детали, дата перемещения, назначение перемещения;

- *БД автомобилей, находящихся в ремонте, содержащая две таблицы:*

- таблица поступления автомобилей в ремонт со структурой: реквизиты заказчика, код заказчика, дата поступления в ремонт, марка автомобиля, модель автомобиля, описание неисправностей, назначенная стоимость, назначенная дата окончания ремонта;

- таблица сдачи выполненных работ: код заказчика, перечень выполненных работ, список использованных запчастей, конечная стоимость ремонта, дата окончания работы;

- *БД клиентов, содержащая одну таблицу со структурой:* код заказчика, информация о заказчике, дата первого обращения, периодичность обращения в автосервис. Информация о клиенте, не обращающемся в автосервис более двух лет, удаляется из БД.

2.3 Недостатки существующих проектных решений

На данный момент существует несколько аналогов, таких как Image v1.2, Склад v1.0, Base Editor v1.3. Однако все эти ПП не обеспечивают полноценного учета всей деятельности Заказчика, а лишь автоматизируют учет отдельных ее частей. Image v1.2 может вести БД только по продажам, Склад v1.0 автоматизирует складской учет при ремонте автомобилей, Base Editor v1.3 не учи-

тывает операции по продаже запасных частей по безналичному расчету. Ни один из приведенных выше аналогов не обеспечивает доступ к данным по локальной сети и не осуществляет фильтрацию данных по всем необходимым Заказчику критериям.

2.4 Текущий уровень автоматизации

На данный момент Заказчик не располагает каким-либо ПП, обеспечивающим полноценную автоматизацию всей его деятельности.

Заказчик располагает оборудованием:

- пять компьютеров P-III-500 (256RAM RIMM), Unix Free BSD, Star Office;
- четыре компьютера П-133 (16RAM 4VRAM), Windows 98, Microsoft Office 97.

3 Требования к разработке

3.1 Информационная модель

Информационная модель представлена на рисунке Б.1.

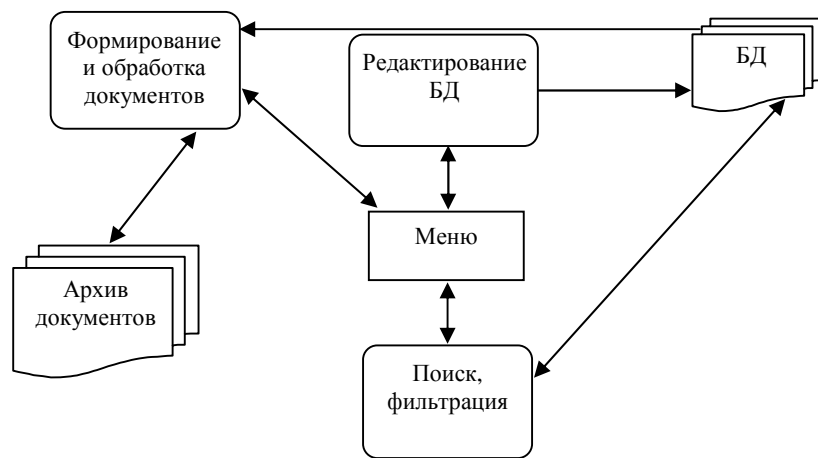


Рисунок Б.1

3.2 Структура меню

Автомобили в продаже

- Вывести список
- Внесение изменений

Автомобили в ремонте

- Вывести список
- Внесение изменений

Запасные части

- Вывести список
- Внутреннее движение
- Внесение изменений

Документы

- Просмотр созданных
- Создать новый

Накладная на внутреннее перемещение

Накладная на реализацию

Акт на списание

Накладная на возврат

Договор

Гарантийный талон

Внесение изменений

Поисковая система

Поиск автомобиля

Поиск автомобиля в продаже

По марке автомобиля

По модели

По цене

Поиск автомобиля в ремонте

По коду заказчика

По сроку выполнения

По марке

Поиск запчастей

По наименованию

Расчет скидки

Сохранить

Help

3.3 Функциональные требования

1 Ведение БД по продаже автомобилей.

Функция предназначена:

- для ведения учета поступления автомобилей, включающего все реквизиты поставщика, марку автомобиля, технические харак-

теристики, цену, дату подписания договора, а также формирование такого договора;

- ведения учета продажи автомобиля, включающего все реквизиты покупателя, марку автомобиля, технические характеристики, цену, дату продажи, а также формирование договора «купли/продажи».

2 Ведение БД по продаже запасных частей.

Функция предназначена:

- для ведения учета поступления запчастей, включающего все реквизиты поставщика, формирование документов, подтверждающих получение запчастей, наименование детали, себестоимость, розничную цену, дату поступления, а также учет документов, подтверждающих поставку запчастей (наличие сертификатов, ТН, под-писанных поставщиком);

- ведения учета продажи запчастей, включающего все реквизиты покупателя, наименование детали, цену, дату продажи, а также формирование и учет сопроводительных документов (квитанция о покупке, гарантийный талон);

- ведения учета внутреннего движения запчастей, включающего наименование детали, дату перемещения, назначение перемещения, а также формирование и учет внутренних ТН.

3 Ведение БД по автомобилям, находящимся в ремонте.

Данная функция должна осуществлять следующее:

- ведение учета поступления в ремонт, включающего все реквизиты заказчика, его код, дату поступления в ремонт, марку и модель автомобиля, описание неисправностей, назначенную стоимость, назначенную дату исправления поломки, а также формирование и учет документов, подтверждающих заказ (квитанция, договор приема заказа).

4 Учет сдачи выполненных работ формирует перечень выполненных работ, список использованных запчастей, рассчитывает конечную стоимость ремонта, дату окончания работы, а также формирует сопроводительные документы (акт «приема/сдачи выполненных работ», квитанция об оплате).

5 Контроль сроков выполнения работ.

6 Ведение БД клиентов.

7 Авторизация. Предназначена для контроля допуска к информации.

8 Поиск, фильтрация. Формирует списки и осуществляет поиск по заданным критериям.

9 Расчет скидки для постоянных клиентов.

Для постоянных клиентов, пользующихся услугами автосервиса более трех лет, устанавливается скидка по оплате 5 % от стоимости всех работ и приобретенных сопутствующих товаров. Для клиентов, пользующихся услугами автосервиса более пяти лет, устанавливается скидка по оплате 10 % от стоимости всех работ и приобретенных сопутствующих товаров.

Постоянными клиентами считаются физические лица или организации, приобретающие автомобиль в данном автосервисе и (или) пользующиеся его услугами для профилактических осмотров, ремонта или приобретения сопутствующих товаров не реже одного раза в полгода в течение указанных выше сроков.

10 Настройка интерфейса.

3.4 Требования к информационному обеспечению

Разрабатываемый ПП должен:

- сохранять данные, вводимые пользователем;
- формировать и сохранять отчетные документы;
- содержать необходимые классификаторы и таблицы;
- обеспечивать обмен данными с другими ПП в реальном времени;
- обеспечивать доступ к данным по локальной сети;
- иметь изменяемые настройки поиска, классификации данных, сопровождаемые кратким описанием;
- иметь изменяемые настройки обрабатываемых объектов (склады, филиалы).

3.5 Требования к интерфейсу

Разрабатываемый ПП должен быть обеспечен упрощенной формой интерфейса:

- глубина вложенности окон должна быть не более трех;
- содержать краткие описания элементов меню;
- должен быть понятным и простым в использовании.

Панель интерфейса должна содержать кнопку, предназначенную для сохранения необходимой информации.

Интерфейс должен содержать в себе все необходимые функции для работы Заказчика. Каждая функция оснащается всплыва-

ющей подсказкой и кратким описанием, что будет способствовать быстрому освоению разрабатываемого ПП.

При необходимости интерфейс изменяется *только* Разработчиком.

3.6 Требования к алгоритмам

Разрабатываемые алгоритмы должны легко модифицироваться. В течение сроков, установленных в договоре между Заказчиком и Разработчиком, в ПП могут быть внесены изменения, не требующие глобальной переработки алгоритмов.

3.7 Прочие требования

Разрабатываемый ПП должен функционировать в ОС Win 9x, NT, 2K. Другие ОС не поддерживают формат создаваемых файлов в процессе работы с ПП.

Хранение данных пользователя должно производиться в файлах с встроенной защитой, не подлежащих удалению без пароля Администратора (приложение А).

ПП должен использовать нижние регистры памяти, что обеспечивает защиту от сбоев на 95 %.

Каждая созданная таблица должна записываться на HDD в виде файла, открываемого любым текстовым редактором без права редактирования, что обеспечивает удобство переноса данных и возможность максимально быстрого просмотра данных.

ПП должен обеспечивать сетевой обмен информацией с другими БД. При создании БД происходит резервное сохранение информации, вводимой пользователем в файлы с расширением «.role». Сетевой обмен происходит за счет передачи/получения файлов с таким же расширением от других систем, для этого необходимо всю нужную информацию другой БД сохранить в файл с расширением «.txt» (текстовый), затем переименовать текстовый файл с расширением «.role». После этого полученный файл поместить в директорию:

C:\Program Files\AutoServAutomatisation\main\bd\info\pole.

4 Порядок контроля и обеспечение качества

4.1 Экспертиза

Не требуется разработка макета на стадии «Технический проект». Требуется проведение экспертизы при участии представителя Заказчика.

4.2 Тестирование

Тестирование должно проводиться Разработчиком в соответствии с установленными правилами. Присутствие представителя Заказчика необязательно.

Тестирование проводится Разработчиком по предложенной схеме Заказчика:

- создается БД небольшого объема по всем таблицам и пунктам меню;

- в течение срока, указанного в договоре, Разработчик проводит тестирование ПП и передает Заказчику его готовый экземпляр, который будет сопровождаться Разработчиком в течение сроков, предусмотренных договором.

4.3 Опытная эксплуатация

Опытная эксплуатация проводится Заказчиком при контроле Разработчика в соответствии с договором.

5 Требования к документированию

5.1 Требования к справочной подсистеме

Справочная система должна содержать описание всех команд, использующихся в ПП. Справочная система должна быть встроенной в интерфейс и располагаться непосредственно в окнах.

5.2 Требования к документации пользователя

Разработанный ПП сопровождается полным пакетом документов, предусмотренным ГОСТ 7685–2000, а именно:

- 1) лицензионное право использования разработанного ПП;
- 2) документация по эксплуатации;
- 3) талон на последующее обслуживание ПП;
- 4) документы, подтверждающие монопольное использование разработанного ПП.

Оформление приложений

Приложение А (справочное)

Пароль Администратора

Данный ПП должен быть установлен только на диск C:\Program Files.

Изменения вступят в силу после ввода пароля Администратора, который будет запрошен при запуске ПП. *Доступ к паролю имеют только лица, купившие лицензионную версию ПП.*

Пароль Администратора: **ZIDBK{JVCEL**

го варианта использования. Стандартным графическим обозначением

зываются количество и периодичность поступления первичных документов

ками и записанные в виде: «первое число .. второе число». Данная запись в языке UML соответствует нотации для множества или интервала целых чисел. Пример такой формы записи кратнопериодичности

- для ведения учета поступления автомобилей, включающего периодичность

Приложение Б (справочное)

Накладная на получение товара организацией ОАО «Руно-Авто»

ОРДЕР №		0402510001	
Число			
От кого	ДЕБЕТ	СУММА	
	сч № 1010		
Наименование товара	кредит/оплата	Общая	
Получатель	ОАО «Руно-Авто»		
Номера тов. единиц	2019	Вид опер.	
		Наз. плат	
Бухгалтер	Зав. Складом		

Приложение В
(справочное)

**Квитанция, выдаваемая клиенту
при покупке запчастей**

КВИТАНЦИЯ №			0402510001															
Число																		
От кого	ОАО «Руно-Авто» Для зачисл.		Сумма покупки															
Наименование тов. единиц	ПЕРВОМАЙСКОЕ ОТДЕЛЕНИЕ		Сумма цифрами															
Получатель																		
Сумма прописью																		
Форма оплаты			2019															
<div style="display: flex; justify-content: space-between;"> М.П. (код отчетности) </div> <div style="display: flex; justify-content: space-between;"> Бухгалтер Деньги принял кассир </div>																		
<p>ГАРАНТИЙНЫЕ ОБЯЗАТЕЛЬСТВА (приложение к квитанции)</p> <p>В течение гарантийного срока бесплатно устраняются все дефекты, возникшие в процессе эксплуатации, при условии соблюдения покупателем правил установки и эксплуатации изделия.</p> <p align="center">Гарантийный талон СПЕЦИФИКАЦИЯ</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Наименование</th> <th>Тип (модель)</th> <th>Серийный номер</th> <th>Кол</th> <th>Гар. срок (мес)</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td></tr> </tbody> </table> <p align="right">Дата изготовления </p>				Наименование	Тип (модель)	Серийный номер	Кол	Гар. срок (мес)										
Наименование	Тип (модель)	Серийный номер	Кол	Гар. срок (мес)														

Приложение Г
(справочное)

**Квитанция, выдаваемая заказчику
при сдаче машины в ремонт**

КВИТАНЦИЯ №			0402510001
Число			
От кого (заказчик)			Сумма услуги
Наименование автомобиля	ОАО «Руно-Авто»		Сумма цифрами
Исполнитель			
Сумма прописью			
Описание неиспр.			
Заказ принял			
Дата выполнения			

Приложение Д
(справочное)

**Квитанция, выдаваемая заказчику
при получении машины из ремонта**

КВИТАНЦИЯ № 		Число	0402510001	
Для кого (заказчик)		Сумма услуги		
Марка автомобиля		<div style="border: 1px solid black; width: 100%; height: 40px;"></div>		
Исполнитель ОАО «Руно-Авто»				
Описание не-испр.				
Сумма прописью				
Работу принял (заказчик)				
Дата получения				
ГАРАНТИЙНЫЕ ОБЯЗАТЕЛЬСТВА (приложение к квитанции)				
<p>В течение гарантийного срока бесплатно устраняются все дефекты, возникшие в процессе эксплуатации, при условии соблюдения покупателем правил эксплуатации изделия.</p>				
Гарантийный талон СПЕЦИФИКАЦИЯ				
Наименование	Тип (модель)	Серийный номер	Кол	Гар. срок (мес)
Дата получения из ремонта		<div style="border: 1px solid black; width: 100%; height: 30px;"></div>		

Приложение Е
(справочное)

**Квитанция, выдаваемая клиенту
при покупке автомобиля**

КВИТАНЦИЯ № 		Число	0402510001		
От кого		Для зачисл.			
ОАО «Руно-Авто»		Сумма покупки			
Марка автомобиля		<div style="border: 1px solid black; width: 100%; height: 40px;"></div>			
Единиц					
Получатель		Сумма цифрами			
ПЕРВОМАЙСКОЕ ОТДЕЛЕНИЕ					
Сумма прописью					
Форма оплаты					
		2019			
М.П.		(код отчетности)			
Бухгалтер		Деньги принял кассир			
ГАРАНТИЙНЫЕ ОБЯЗАТЕЛЬСТВА (приложение к квитанции)					
<p>В течение гарантийного срока бесплатно устраняются все дефекты, возникшие в процессе эксплуатации, при условии соблюдения покупателем правил эксплуатации изделия.</p>					
Гарантийный талон СПЕЦИФИКАЦИЯ					
Марка автомобиля	Номер двигателя	Номер кузова	Серийный номер	Кол	Гар. срок (мес)
Дата покупки		<div style="border: 1px solid black; width: 100%; height: 30px;"></div>			

Приложение В (справочное)

Построение диаграмм деятельности (activity diagram)

При моделировании поведения системы возникает необходимость детализировать особенности алгоритмической и логической реализации выполняемых системой операций. Для моделирования процесса выполнения операций в языке UML используются диаграммы деятельности. Каждое состояние на диаграмме деятельности соответствует выполнению некоторой элементарной операции, а переход в следующее состояние срабатывает только при завершении этой операции в предыдущем состоянии. Графически диаграмма деятельности представляется в форме графа деятельности, вершинами которого являются состояния действия, а дугами – переходы от одного состояния действия к другому. На диаграмме деятельности отображается логика или последовательность перехода от одной деятельности к другой, при этом внимание фиксируется на результате деятельности. Компонентами диаграммы деятельности являются:

- состояния действия;
- переходы;
- дорожки;
- объекты.

Состояние действия. Состояние действия (action state) является специальным случаем состояния с некоторым входным действием и, по крайней мере, одним выходящим из состояния переходом. Этот переход неявно предполагает, что входное действие уже завершилось. Состояние действия не может иметь внутренних переходов, поскольку оно является элементарным. Обычное использование состояния действия заключается в моделировании одного шага выполнения алгоритма (процедуры) или потока управления.

Графически состояние действия изображается как показано на рисунке В.1. Внутри фигуры записывается выражение действия (action-expression), которое должно быть уникальным в пределах одной диаграммы деятельности.

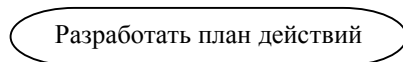


Рисунок В.1 – Графическое изображение состояния действия

Действие может быть записано на естественном языке, некотором псевдокоде или языке программирования. Рекомендуется в качестве имени простого действия использовать глагол с пояснительными словами.

Иногда возникает необходимость представить на диаграмме деятельности некоторое сложное действие, в свою очередь состоящее из нескольких более простых действий. В этом случае можно использовать специальное обозначение состояния поддеятельности (subactivity state). Оно обозначается специальной пиктограммой в правом нижнем углу символа состояния действия (рисунок В.2).

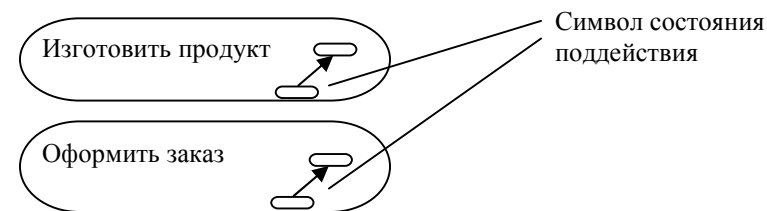


Рисунок В.2 – Графическое изображение состояния поддеятельности

Каждая диаграмма деятельности должна иметь единственное начальное и единственное конечное состояния. Они имеют следующие обозначения (рисунок В.3):



Рисунок В.3 – Графическое изображение начального и конечного состояний на диаграмме состояний

Диаграмму деятельности принято располагать таким образом, чтобы действия следовали сверху вниз. В этом случае начальное состояние будет изображаться в верхней части диаграммы, а конечное – в ее нижней части.

Переходы. При построении диаграммы деятельности используются только нетриггерные переходы, то есть такие, которые срабатывают сразу после завершения деятельности или выполнения соответствующего действия. На диаграмме такой переход изображается сплошной линией со стрелкой.

Если из состояния действия выходит более одного перехода, то сработать может только один из них. Тогда для каждого из таких переходов должно быть явно записано сторожевое условие в форме булевского выражения в прямых скобках. В этот ромб может входить только одна стрелка. Принято входящую стрелку присоединять к верхней или левой вершине символа ветвления. Выходящих стрелок может быть две или более, но для каждой из них явно указывается соответствующее сторожевое условие.

В качестве примера рассмотрим фрагмент алгоритма нахождения корней квадратного уравнения (рисунок В.4). В общем случае после приведения уравнения второй степени к каноническому виду: $a \cdot x \cdot x + b \cdot x + c = 0$ необходимо вычислить его дискриминант. Причем, в случае отрицательного дискриминанта, уравнение не имеет решения на множестве действительных чисел, и дальнейшие вычисления должны быть прекращены. При неотрицательном дискриминанте уравнение имеет решение, корни которого могут быть получены на основе конкретной расчетной формулы.

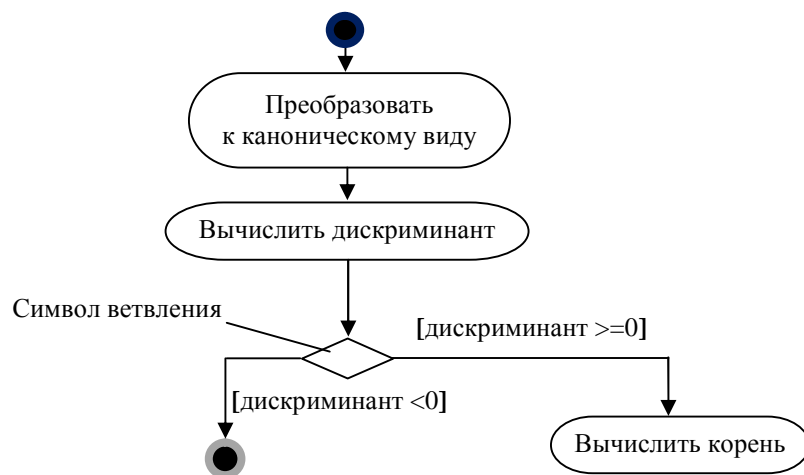


Рисунок В.4 – Фрагмент диаграммы деятельности для алгоритма нахождения корней квадратного уравнения

Первое из состояний рассматриваемого алгоритма можно считать состоянием поддеятельности, так как приведение квадратного уравнения к каноническому виду может потребовать нескольких элементарных действий. Поэтому для данного со-

стояния можно добавить соответствующую пиктограмму (как на рисунке В.2).

В примере, приведенном на рисунке В.5, рассчитывается общая стоимость товаров, покупаемых по кредитной карточке. Если эта стоимость превышает \$50, то выполняется аутентификация личности владельца карточки. В случае положительной проверки (карточка действительная) или если стоимость товаров не превышает \$50, происходит снятие суммы со счета и оплата стоимости товаров. При отрицательном результате (карточка недействительная) оплаты не происходит, и товар остается у продавца. Вместо сторожевого условия допускается использовать слово «иначе» (else).

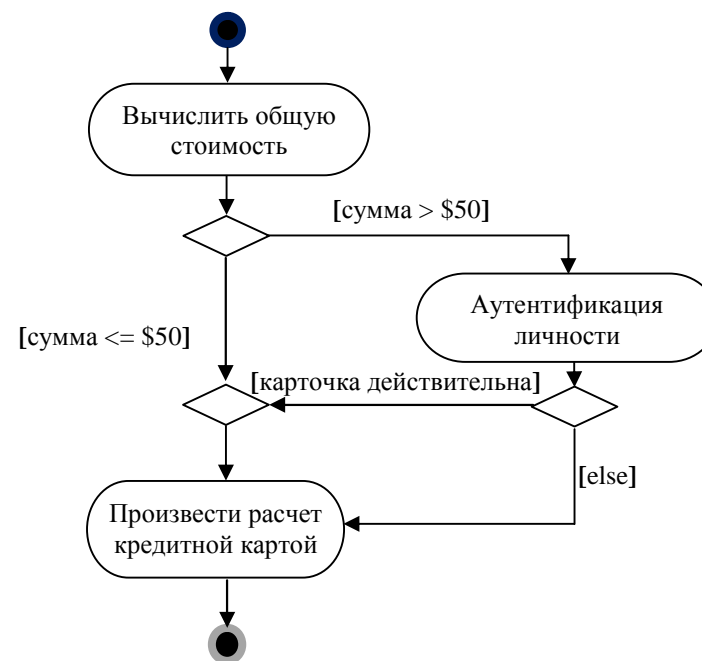


Рисунок В.5 – Диаграмма деятельности расчета общей стоимости товаров, покупаемых по кредитной карточке

В языке UML используется специальный символ для разделения и слияния параллельных вычислений или потоков управления.

Таким символом является прямая горизонтальная линия, толщина которой несколько шире основных сплошных линий на диаграмме. При этом разделение (concurrent fork) имеет один входящий переход и несколько выходящих (рисунок В.6, а), а слияние (concurrent join) – несколько входящих переходов и один выходящий (рисунок В.6, б).



Рисунок В.6 – Графическое изображение разделения и слияния параллельных потоков управления

Для иллюстрации параллельных процессов рассмотрим пример с приготовлением кофе (рисунок В.7).

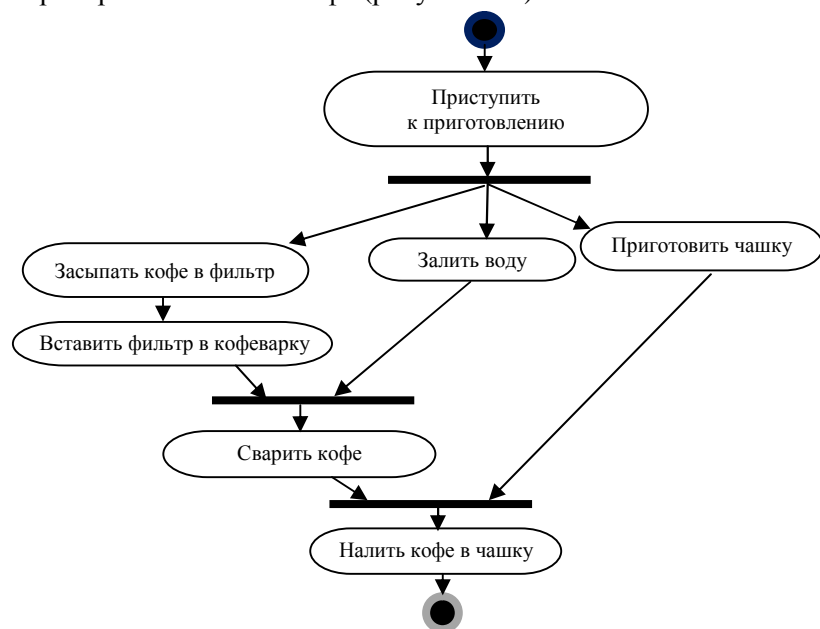


Рисунок В.7 – Диаграмма деятельности, иллюстрирующая параллельные процессы

Дорожки. Диаграммы деятельности могут быть также использованы для моделирования бизнес-процессов. Применитель-

но к бизнес-процессам желательно выполнение каждого действия ассоциировать с конкретным подразделением компании. В этом случае подразделение несет ответственность за реализацию отдельных действий, а сам бизнес-процесс представляется в виде переходов действий из одного подразделения к другому.

Для моделирования этих особенностей в языке UML используется специальная конструкция, получившее название дорожки (swimlanes). При этом все состояния действия на диаграмме деятельности делятся на отдельные группы, которые отделяются друг от друга вертикальными линиями. Две соседние линии и образуют дорожку, а группа состояний между этими линиями выполняется отдельным подразделением.

Названия подразделений явно указываются в верхней части дорожки. Пересекать линию дорожки могут только переходы, в этом случае обозначающие выход или вход потока управления в соответствующее подразделение компании.

В качестве примера рассмотрим фрагмент диаграммы деятельности торговой компании, обслуживающей клиентов по телефону. Подразделениями компании являются отдел приема и оформления заказов, отдел продаж и склад (рисунок В.8).

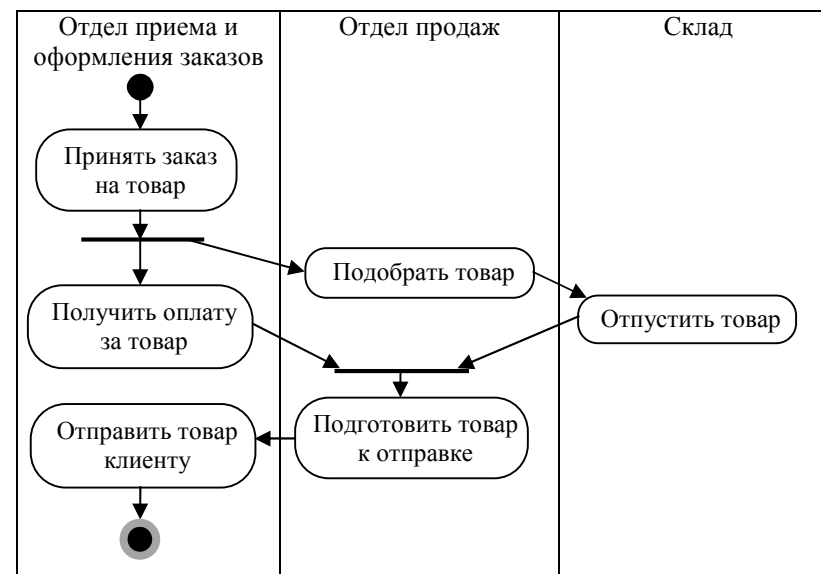


Рисунок В.8 – Фрагмент диаграммы деятельности для торговой компании

Объекты. В общем случае действия на диаграмме деятельности выполняются над теми или иными объектами. Эти объекты либо инициируют выполнение действий, либо определяют некоторый результат этих действий. Для графического представления объектов используется прямоугольник класса с тем отличием, что имя объекта подчеркивается. Далее после имени может указываться характеристика состояния объекта в прямых скобках. Такие прямоугольники объектов присоединяются к состояниям действия отношением зависимости пунктирной линией со стрелкой.

Возвращаясь к предыдущему примеру с торговой компанией, можно заметить, что центральным объектом процесса продажи является заказ или вернее состояние его выполнения. Данная информация может быть представлена графически в виде модифицированного варианта диаграммы деятельности этой же торговой компании (рисунок В.9).

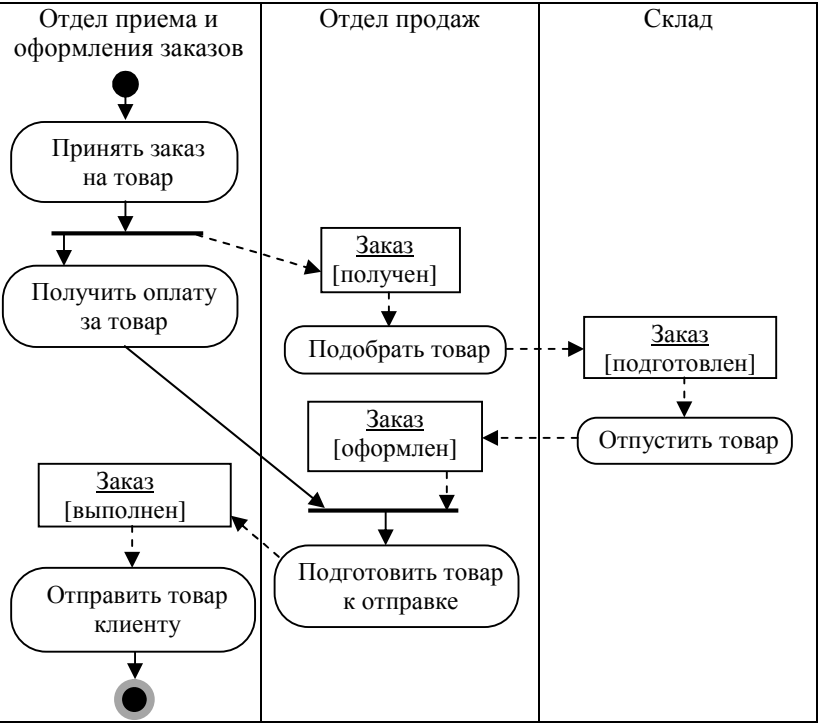


Рисунок В.9 – Фрагмент диаграммы деятельности торговой компании с объектом-заказом

Приложение Г (справочное)

Построение диаграмм вариантов использования (use case diagram)

Визуальное моделирование в UML можно представить как некоторый процесс поуровневого спуска от наиболее общей и абстрактной концептуальной модели исходной системы к логической, а затем и к физической модели соответствующей программной системы. Для достижения этих целей вначале строится модель в форме диаграммы вариантов использования (use case diagram), которая описывает функциональное назначение системы. Диаграмма вариантов использования является исходным концептуальным представлением или концептуальной моделью системы в процессе ее проектирования и разработки.

Диаграммы вариантов использования определяют общие границы и контекст моделируемой предметной области на начальных этапах проектирования системы, формулируют общие требования к функциональному поведению проектируемой системы, подготавливают исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями.

Основными компонентами диаграммы вариантов использования являются:

- варианты использования;
- актеры;
- интерфейсы;
- примечания;
- отношения.

Суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью вариантов использования. При этом актером (actor) или действующим лицом называется любая сущность, взаимодействующая с системой извне. Это может быть человек, техническое устройство, программа или любая другая система, служащая источником воздействия на моделируемую систему так, как определит сам разработчик. В свою очередь, вариант использования (use case) служит для описания сервисов, которые система предоставляет актеру. Другими сло-

вами, каждый вариант использования определяет некоторый набор действий, совершаемый системой при диалоге с актером. При этом ничего не говорится о том, каким образом будет реализовано взаимодействие актеров с системой. Отдельные компоненты диаграммы могут быть заключены в прямоугольник, который обозначает проектируемую систему в целом.

Вариант использования. Конструкция или стандартный элемент языка UML вариант использования применяется для спецификации общих особенностей поведения системы или любой другой сущности предметной области без рассмотрения внутренней структуры этой сущности. Каждый вариант использования определяет последовательность действий, которые должны быть выполнены проектируемой системой при взаимодействии ее с соответствующим актером. Диаграмма вариантов может дополняться пояснительным текстом, который раскрывает смысл или семантику составляющих ее компонентов. Такой пояснительный текст получил название примечания или сценария.

Отдельный вариант использования обозначается на диаграмме эллипсом, внутри которого содержится его краткое название или имя в форме глагола с пояснительными словами (рисунок Г.1).

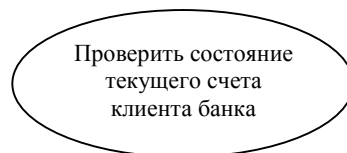


Рисунок Г.1 – Графическое обозначение варианта использования

Цель варианта использования заключается в том, чтобы определить законченный аспект или фрагмент поведения некоторой сущности без раскрытия внутренней структуры этой сущности. В качестве такой сущности может выступать исходная система или любой другой элемент модели, обладающий собственным поведением, подобно подсистеме или классу в модели системы.

Каждый вариант использования соответствует отдельному сервису, который предоставляет моделируемую сущность или систему по запросу пользователя (актера), то есть определяет способ применения этой сущности.

Варианты использования описывают не только взаимодействия между пользователями и сущностью, но также реакции сущности на получение отдельных сообщений от пользователей и восприятие этих сообщений за пределами сущности. Множество вариантов использования в целом должно определять все возможные стороны ожидаемого поведения системы.

Примерами вариантов использования могут являться следующие действия: проверка состояния текущего счета клиента, оформление заказа на покупку товара, получение дополнительной информации о кредитоспособности клиента, отображение графической формы на экране монитора и другие действия.

Актеры. Актер представляет собой любую внешнюю по отношению к моделируемой системе сущность, взаимодействующую с системой и использующую ее функциональные возможности. При этом актеры служат для обозначения согласованного множества ролей, которые могут играть пользователи в процессе взаимодействия с проектируемой системой. Каждый актер может рассматриваться как некая отдельная роль относительно конкретного варианта использования. Стандартным графическим обозначением актера на диаграммах является фигурка «человечка», под которой записывается конкретное имя актера (рисунок Г.2).



Рисунок Г.2 – Графическое обозначение актера

Имя актера должно быть достаточно информативным с точки зрения семантики. Примерами актеров могут быть: клиент банка, банковский служащий, продавец магазина, менеджер отдела продаж, пассажир авиарейса, водитель автомобиля, администратор гостиницы, сотовый телефон и другие сущности, имеющие отношение к концептуальной модели соответствующей предметной области.

В качестве актеров могут выступать другие системы, подсистемы проектируемой системы или отдельные классы.

Исходя из того, что в общем случае актер всегда находится вне системы, его внутренняя структура никак не определяется.

Для актера имеет значение только его внешнее представление, то есть то, как он воспринимается со стороны системы. Актеры взаимодействуют с системой посредством передачи и приема сообщений от вариантов использования. Сообщение представляет собой запрос актером сервиса от системы и получение этого сервиса. Кроме этого, с актерами могут быть связаны интерфейсы, определяющие, каким образом другие элементы модели взаимодействуют с этими.

Интерфейсы. Интерфейс (interface) служит для спецификации параметров модели, которые видимы извне без указания их внутренней структуры. Интерфейсы определяют совокупность операций, обеспечивающих необходимый набор сервисов или функциональности для актеров. Интерфейсы не могут содержать ни атрибутов, ни состояний, ни направленных ассоциаций. Они содержат только операции без указания особенностей их реализации.

На диаграмме вариантов использования интерфейс изображается в виде маленького круга, рядом с которым записывается его имя (рисунок Г.3, а). В качестве имени может быть существительное, характеризующее соответствующую информацию или сервис (например, «датчик», «сирена», «видеокамера»), но чаще строка текста (например, «запрос к базе данных», «форма ввода», «устройство подачи звукового сигнала»). Если имя записывается на английском языке, то оно должно начинаться с заглавной буквы I, например, ISecureInformation, ISensor (рисунок Г.3, б).

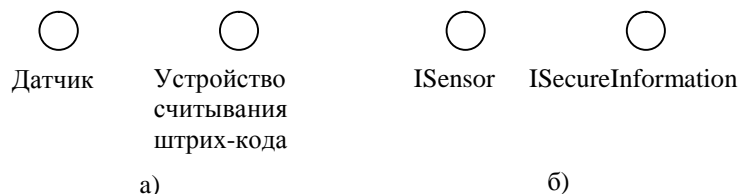


Рисунок Г.3 – Графическое изображение интерфейсов на диаграммах вариантов использования

Графический символ отдельного интерфейса может соединяться на диаграмме сплошной линией с тем вариантом использования, который его поддерживает. Сплошная линия в этом случае указывает на тот факт, что связанный с интерфейсом вариант использования должен реализовывать все операции, необходимые для данного интерфейса, а возможно и больше (рисунок Г.4, а).

Кроме этого, интерфейсы могут соединяться с вариантами использования пунктирной линией со стрелкой (рисунок Г.4, б), означающей, что вариант использования предназначен для спецификации только того сервиса, который необходим для реализации данного интерфейса.

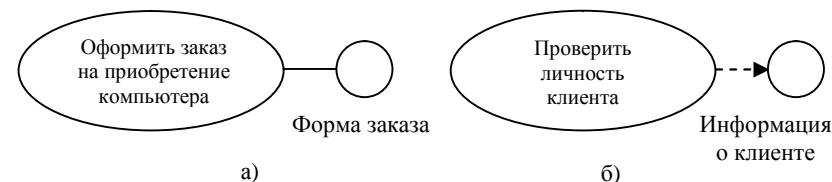


Рисунок Г.4 – Графическое изображение взаимосвязей интерфейсов с вариантами использования

Важность интерфейсов заключается в том, что они определяют стыковочные узлы в проектируемой системе, что совершенно необходимо для организации коллективной работы над проектом.

Примечания. Примечания (notes) в языке UML предназначены для включения в модель произвольной текстовой информации, имеющей непосредственное отношение к контексту разрабатываемого проекта. В качестве такой информации могут быть комментарии разработчика (например, дата и версия разработки диаграммы или ее отдельных компонентов), ограничения (например, назначения отдельных связей или экземпляры сущностей) и помеченные значения. Применительно к диаграммам вариантов использования примечание может носить самую общую информацию, относящуюся к общему контексту системы.

Графически примечания обозначаются прямоугольником с «загнутым» верхним правым углом (рисунок Г.5). Внутри прямоугольника содержится текст примечания. Примечание может относиться к любому элементу диаграммы, в этом случае их соединяет пунктирная линия. Если примечание относится к нескольким элементам, то от него проводятся, соответственно, несколько линий. Разумеется, примечания могут присутствовать не только на диаграмме вариантов использования, но и на других канонических диаграммах.

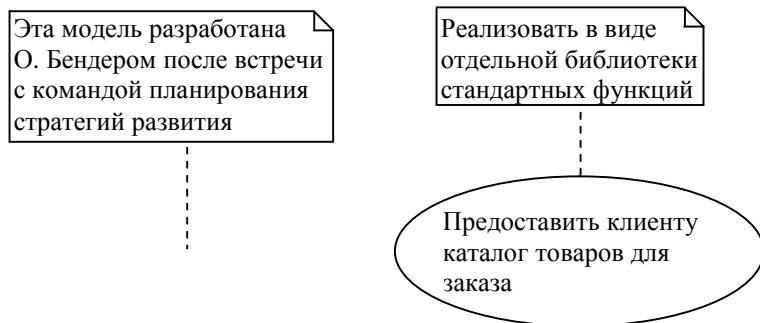
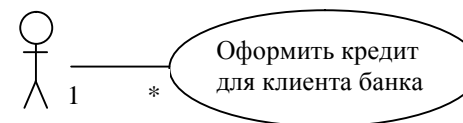


Рисунок Г.5 – Примеры примечаний в языке UML

Отношения. Между компонентами диаграммы вариантов использования могут существовать различные отношения, которые описывают взаимодействие экземпляров одних актеров и вариантов использования с экземплярами других актеров и вариантов использования. Один актер может взаимодействовать с несколькими вариантами использования. В этом случае этот актер обращается к нескольким сервисам данной системы. В свою очередь один вариант использования может взаимодействовать с несколькими актерами, предоставляя для всех них свой сервис. Следует заметить, что два варианта использования, определенные для одной и той же сущности, не могут взаимодействовать друг с другом, поскольку каждый из них самостоятельно описывает законченный вариант использования этой сущности. В языке UML имеется несколько стандартных видов отношений между актерами и вариантами использования:

- отношение ассоциации (association relationship);
- отношение включения (include relationship);
- отношение расширения (extend relationship);
- отношение обобщения (generalization relationship).

Отношение ассоциации. Оно служит для обозначения специфической роли актера в отдельном варианте использования. Это отношение устанавливает, какую конкретную роль играет актер при взаимодействии с экземпляром варианта использования. Оно обозначается сплошной линией между актером и вариантом использования. Эта линия может иметь дополнительные условные обозначения, такие, например, как имя и кратность (рисунок Г.6).



Клиент банка

Рисунок Г.6 – Пример графического представления отношения ассоциации между актером и вариантом использования

Кратность (multiplicity) ассоциации указывается рядом с обозначением компонента диаграммы, являющегося участником данной ассоциации. Кратность характеризует общее количество конкретных экземпляров данного компонента, которые могут выступать в качестве элементов данной ассоциации.

Формы записи кратности:

- целое неотрицательное число (включая цифру 0). Примером этой формы записи кратности ассоциации является указание кратности «1» для актера «Клиент банка» (рисунок Г.6). Эта запись означает, что каждый конкретный кредит оформляется на единственного клиента этого банка;

- два целых неотрицательных числа, разделенные двумя точками и записанные в виде: «первое число .. второе число». Данная запись в языке UML соответствует нотации для множества или интервала целых чисел. Пример такой формы записи кратности ассоциации – «1..5». Эта запись означает, что количество отдельных экземпляров данного компонента, которые могут выступать в качестве элементов данной ассоциации, равно некоторому заранее неизвестному числу из множества целых чисел {1, 2, 3, 4, 5};

- два символа, разделенные двумя точками. При этом первый из них является целым неотрицательным числом или 0, а второй – специальным символом «*». Здесь символ «*» обозначает произвольное конечное целое неотрицательное число, значение которого неизвестно на момент задания соответствующего отношения ассоциации. Пример такой формы записи кратности ассоциации – «2...*»;

- единственный символ «*», который является сокращением записи интервала «0...*». В этом случае количество отдельных экземпляров данного компонента отношения ассоциации может быть любым целым неотрицательным числом. При этом 0 означает, что для некоторых экземпляров соответствующего компонента данное

отношение ассоциации может вовсе не иметь места. В качестве примера этой записи можно привести кратность отношения ассоциации для варианта использования «Оформить кредит для клиента банка» (рисунок Г.6). Здесь кратность «*» означает, что каждый отдельный клиент банка может оформить для себя несколько кредитов, при этом их общее число заранее неизвестно и ничем не ограничивается. При этом некоторые клиенты могут совсем не иметь оформленных на свое имя кредитов (вариант значения 0).

Если кратность отношения ассоциации не указана, то по умолчанию принимается ее значение, равное 1.

Отношение включения. Отношение включения между двумя вариантами использования указывает, что некоторое заданное поведение для одного варианта использования включается в качестве составного компонента в последовательность поведения другого варианта использования. Данное отношение является направленным бинарным отношением в том смысле, что пара экземпляров вариантов использования всегда упорядочена в отношении включения.

Один вариант использования может быть включен в несколько других вариантов, а также включать в себя другие варианты. Отношение включения, направленное от варианта использования А к варианту использования В, указывает, что каждый экземпляр варианта А включает в себя функциональные свойства, заданные для варианта В. Эти свойства специализируют поведение соответствующего варианта А на данной диаграмме. Графически данное отношение обозначается пунктирной линией со стрелкой (вариант отношения зависимости), направленной от базового варианта использования к включаемому. При этом данная линия со стрелкой помечается ключевым словом «include» («включает»), как показано на рисунок Г.7.

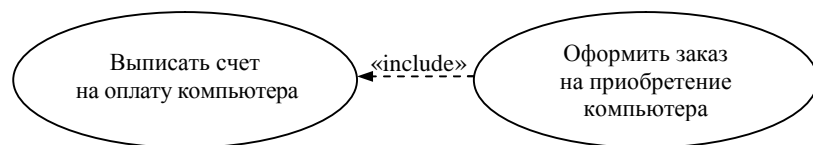


Рисунок Г.7 – Пример графического изображения отношения включения между вариантами использования

Отношение расширения. Отношение расширения определяет взаимосвязь экземпляров отдельного варианта использова-

ния с более общим вариантом. Отношение расширения является направленным. Так, если имеет место отношение расширения от варианта использования А к варианту использования В, то это означает, что свойства экземпляра варианта использования В могут быть дополнены благодаря наличию свойств у расширенного варианта использования А.

Отношение расширения между вариантами использования обозначается пунктирной линией со стрелкой, направленной от того варианта использования, который является расширением для исходного варианта использования. Данная линия со стрелкой помечается ключевым словом «extend» («расширяет»), как показано на рисунок Г.8.

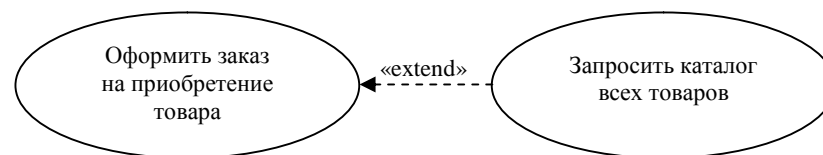


Рисунок Г.8 – Пример графического изображения отношения расширения между вариантами использования

Один из вариантов использования может быть расширением для нескольких базовых вариантов, а также иметь в качестве собственных расширений несколько других вариантов.

В представленном выше примере (рисунок Г.8) при оформлении заказа на приобретение товара только в некоторых случаях может потребоваться предоставление клиенту каталога всех товаров. При этом условием расширения является запрос от клиента на получение каталога товаров (рисунок Г.9).

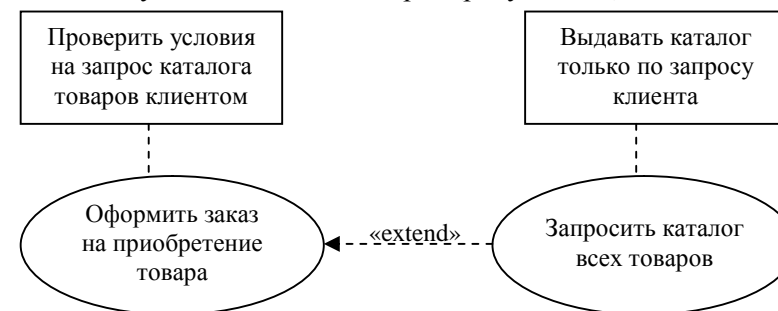


Рисунок Г.9 – Графическое изображение отношения расширения с примечаниями условий выполнения вариантов использования

Отношение обобщения. Отношение обобщения служит для указания того факта, что некоторый вариант использования А может быть обобщен до варианта использования В. В этом случае вариант А будет являться специализацией варианта В. При этом вариант В называется предком или родителем по отношению к варианту А, а вариант А – потомком по отношению к варианту использования В. Следует подчеркнуть, что потомок наследует все свойства и поведение своего родителя, а также может быть дополнен новыми свойствами и особенностями поведения. Графически данное отношение обозначается сплошной линией со стрелкой в форме незакрашенного треугольника, которая указывает на родительский вариант использования (рисунок Г.10). Эта линия со стрелкой имеет специальное название – стрелка «обобщение».

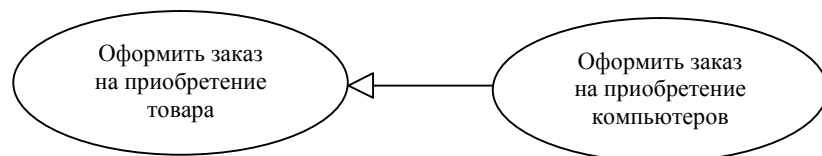


Рисунок Г.10 – Пример графического изображения отношения обобщения между вариантами использования

Отношение обобщения между вариантами использования применяется в том случае, когда необходимо отметить, что дочерние варианты использования обладают всеми атрибутами и особенностями поведения родительских вариантов. Один вариант использования может иметь несколько родительских вариантов. С другой стороны, один вариант использования может быть предком для нескольких дочерних вариантов, что соответствует таксономическому характеру отношения обобщения.

Между отдельными актерами также может существовать отношение обобщения.

Пример построения диаграммы вариантов использования

В качестве примера рассмотрим процесс моделирования системы продажи товаров по каталогу, которая может быть использована при создании соответствующих информационных систем.

В качестве актеров данной системы могут выступать два субъекта, один из которых является продавцом, а другой – покупателем. Они оба обращаются к соответствующему сервису «Оформить заказ на покупку товара» (рисунок Г.11).

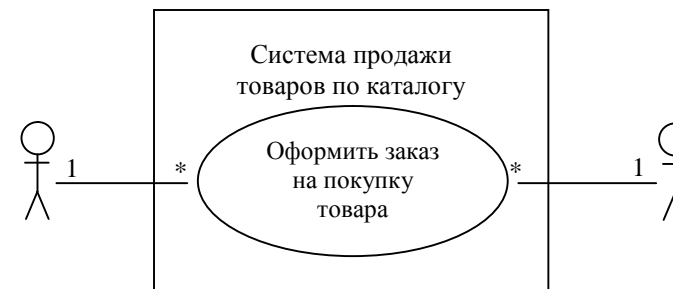


Рисунок Г.11 – Контекстный вид диаграммы вариантов использования

Один продавец может участвовать в оформлении нескольких заказов, в то же время каждый заказ может быть оформлен только одним продавцом. Покупатель может оформлять на себя несколько заказов, но, в то же время, каждый заказ должен быть оформлен на единственного покупателя.

Проведем детализацию диаграммы.

На следующем этапе разработки диаграмма уточняется на основе введения в рассмотрение дополнительных вариантов использования: обеспечить покупателя информацией о товаре, согласовать условия оплаты товара и заказать товар со склада. Указанные действия раскрывают поведение исходного варианта использования в смысле его конкретизации, и поэтому между ними будет иметь место отношение включения.

Продажа товаров по каталогу предполагает наличие самостоятельного информационного объекта – каталога товаров, который не зависит от реализации сервиса по обслуживанию покупателей. Каталог товаров может запрашиваться покупателем или продавцом при необходимости выбора товара и уточнения деталей его продажи. Сервис «Запросить каталог товаров» можно рассматривать в качестве самостоятельного варианта использования.

Полученная в результате детализации уточненная диаграмма показана на рисунке Г.12.

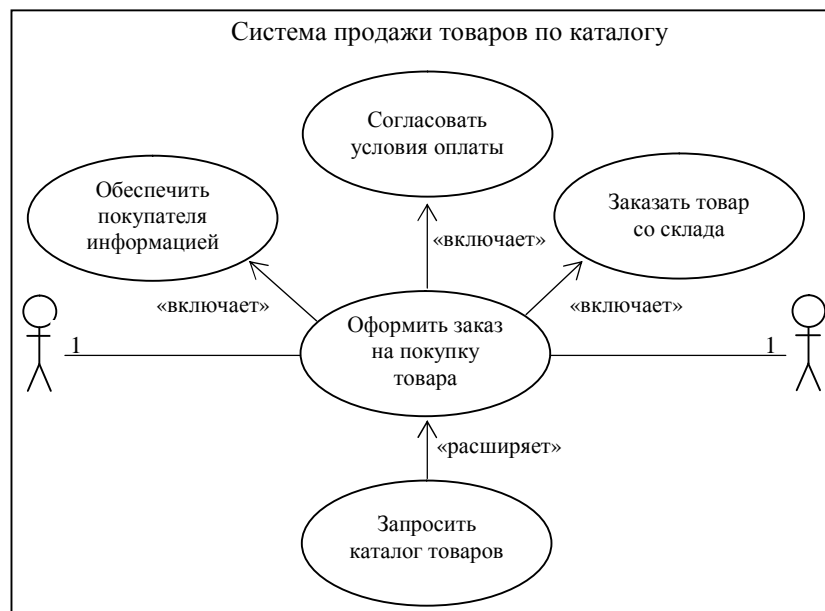


Рисунок Г.12 – Уточненный вариант диаграммы вариантов использования для примера системы продажи товаров по каталогу

Оглавление

Предисловие	3
1 Учебная программа дисциплины	5
1.1 Примерный тематический план	5
1.2 Содержание дисциплины	6
1.3 Перечень лабораторно-практических занятий	10
1.4 Примерный перечень тем для выполнения практической части контрольной работы	10
2 Требования по выполнению и оформлению контрольной работы	11
3 Варианты контрольной работы	12
4 Вопросы для самоконтроля	15
Рекомендуемая литература	17
Приложение А Требования, предъявляемые к составлению ТЗ	18
Приложение Б Пример оформления ТЗ	27
Приложение В Построение диаграмм деятельности (activity diagram)	44
Приложение Г Построение диаграмм вариантов использования (use case diagram)	51

Учебное издание

**ТЕХНОЛОГИЯ РАЗРАБОТКИ
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

Учебная программа, методические указания
и контрольные задания
для учащихся заочной формы получения образования специальности
2-40 01 01 «Программное обеспечение информационных технологий»

Составитель
Тарасова Татьяна Михайловна

Ответственный за выпуск О. П. Козельская
Редактор Ю. С. Гришкевич
Корректор Г. Л. Говор
Компьютерная верстка Ю. С. Гришкевич

Подписано в печать 15.05.2013. Формат 60×84¹/₁₆. Бумага писчая.
Ризография. Усл. печ. л. 3,72. Уч.-изд. л. 3,17. Тираж 40 экз. Заказ 58.

Издатель и полиграфическое исполнение:
учреждение образования «Минский государственный высший
радиотехнический колледж»
ЛИ № 02330/0494033 от 08.01.2009.
Пр. Независимости, 62, 220005, Минск.