****ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»  
(САМАРСКИЙ УНИВЕРСИТЕТ)»

Факультет Информатики   
Кафедра Программных систем

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**  
  
 к курсовой работе по дисциплине «Программная инженерия»

по теме «Полное название темы»

Студент И.И. Иванов

Студент П.П. Петров

Руководитель Л.С. Зеленко

Самара 2016

****ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ   
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА» (САМАРСКИЙ УНИВЕРСИТЕТ)»

Факультет Информатики   
Кафедра Программных систем

**ЗАДАНИЕ**

на курсовую работу по дисциплине

«Программная инженерия»

студентам группы № 6412 Б 300  
И.И. Иванову   
П.П. Петрову

1. **Тема проекта:** «Автоматизированная система составления и разгадывания линейного кроссворда по выбранной теме»
2. **Исходные данные к проекту:** см. приложение к заданию
3. **Перечень вопросов, подлежащих разработке в курсовой работе:**
   1. Произвести анализ предметной области: изучить основные принципы составления кроссвордов, изучить методы и алгоритмы генерации кроссвордов
   2. Выполнить обзор существующих систем-аналогов
   3. Разработать информационно-логический проект по методологии UML
   4. Разработать и реализовать программное и информационное обеспечение, провести его тестирование и отладку .
   5. Оформить документацию курсовой работы
   6. Подготовить презентацию по разработанной системе
4. **Перечень графических разработок**
   1. Структурная схема системы
   2. Диаграммы UML
   3. Схемы основных алгоритмов
5. **Календарный план выполнения работ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Содержание работы по этапам | Объем этапа в % к общему объему проекта | Срок  окончания | Фактическое выполнение |
| 1 | Оформление технического задания и его утверждение | 5 |  |  |
| 2 | Анализ и описание предметной области (1 раздел) | 10 |  |  |
| 3 | Проектирование системы (2 раздел) | 30 |  |  |
| 3.1 | Разработка структурной схемы системы | 5 |  |  |
| 3.2 | Разработка функциональной спецификации системы | 10 |  |  |
| 3.3 | Разработка информационно-логического проекта системы и его предъявление руководителю | 15 |  |  |
| 4 | Реализация проекта, разработка контрольных примеров. Предъявление реализации руководителю (3 раздел). | 45 |  |  |
| 5 | Корректировка проекта и оформление документации проекта. Защита проекта с представлением презентации. | 10 |  |  |

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_ /И.И.Иванов/ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_/П.П.Петров/ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### Роспись Дата (потом удалить)

**ПРИЛОЖЕНИЕ  
к заданию на курсовую работу**студентам группы № 6412 Б 300   
И.И. Иванову

П.П. Петрову

Тема проекта: **«Автоматизированная система составления и разгадывания линейного кроссворда по выбранной теме»**

**Исходные данные к проекту:**

1. **Характеристика объекта автоматизации:**

## объект автоматизации: линейный кроссворд;

## виды автоматизируемой деятельности:

* + процесс составления/генерирования кроссворда;
  + процесс разгадывания кроссворда;
  + процесс работы со словарем понятий;
  + процесс визуализации работы с кроссвордом;

## минимальная длина кроссворда ‑ 50 символов;

## максимальная длина кроссворда ‑ 350 символов;

## минимальная длина одного слова – 3 символа;

## максимальная длина одного слова – 15 символов;

## количество видов отображения кроссворда – 4;

## количество букв в пересечении ‑ от 1 до 3;

## количество видов сортировки словаря понятий – 2;

## минимальное количество подсказок –1;

## максимальное количество подсказок – 10% от количества слов;

## язык записи понятий – русский.

1. **Требования к информационному обеспечению:**
2. информационное обеспечение разрабатывается на основе следующих документов:
   * описание структуры кроссворда [Электронный ресурс]. – ru.wikipedia.org/wiki/Линейный\_кроссворд (дата обращения 15.09.2014 г. );
   * структуры словаря понятий (понятие и его определение располагаются в одной строке, разделены пробелом);
3. словари понятий хранятся в текстовых файлах формата \*.dict;
4. кроссворды хранятся в файлах, структура файла определяется в процессе проектирования.
5. **Требования к техническому обеспечению:**
6. тип ЭВМ – IBM PC совместимый;
7. монитор с разрешающей способностью не ниже 800 х 600;
8. манипулятор – мышь;
9. технические характеристики определяются в процессе выполнения проекта.
10. **Требования к программному обеспечению:**
11. тип операционной системы ‑ Windows 7/8, Windows Vista;
12. язык программирования – С#;
13. среда программирования –Visual Studio 2015;
14. среда проектирования – StarUML 5.0.
15. **Общие требования к проектируемой системе:**

***5.1 Функции, реализуемые системой:***

1. настройка параметров системы:
2. задание размеров кроссворда;
3. выбор вида отображения кроссворда;
4. подключение словаря понятий;
5. автоматическое составление кроссворда по заданным параметрам;
6. создание/редактирование кроссворда:
7. фильтрация списка слов по маске;
8. выбор слова из списка;
9. добавление слова на сетку кроссворда;
10. выбор слова в кроссворде;
11. удаление слова с сетки кроссворда;
12. сохранение кроссворда в файл заданной структуры;
13. загрузка кроссворда из файла;
14. работа со словарями понятий:
15. добавление понятия;
16. удаление понятия;
17. изменение понятия;
18. проверка дублирования понятий;
19. проверка языка записи понятий;
20. сортировка словаря по выбранному критерию;
21. поиск слова по маске;
22. загрузка словаря из файла;
23. сохранение словаря в файл;
24. создание нового словаря понятий;
25. разгадывание кроссворда с организацией системы подсказок:
26. выбор слова;
27. вписывание слова на сетке кроссворда;
28. взятие подсказки;
29. проверка правильности разгадывания слова/кроссворда;
30. визуализация процессов работы с кроссвордом;
31. выдача справочной информации о системе.

***5.2 Технические требования к системе:***

1. режим работы ‑ диалоговый;
2. время автоматической генерации кроссворда – не более 30 с;
3. система должна удовлетворять санитарным правилам и нормам  
    СанПин 2.2.2./2.4.2198-07;
4. условия работы средств вычислительной техники (содержание вредных веществ, пыли и подвижность воздуха) должны соответствовать ГОСТ 12.1.005, 12.01.007;
5. температура окружающего воздуха – 15-25°С;
6. влажность окружающего воздуха ‑ 45-75%.

Руководитель   
курсовой работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Л.С. Зеленко /

Задание принял  
 к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ И.И. Иванов /

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ П.П. Петров /

**РЕФЕРАТ**

Пояснительная записка 35 с, 14 рисунков, 5 таблиц[[1]](#footnote-1), 12 источников,  
2 приложения.

Графическая часть: 15 слайдов презентации PowerPoint.

**ДЕРЕВО ПОИСКА, ГЕНЕРАТОР КРОССВОРДОВ, ГОЛОВОЛОМКА, СЛОВАРЬ ТЕРМИНОВ, ВАРИАНТ ОТОБРАЖЕНИЯ, РАЗГАДЫВАНИЕ**

Во время курсового проектирования разработаны алгоритмы и соответствующая им программа, позволяющая выполнять автоматическую генерацию линейного кроссворда по заданной теме. Задания (понятие и его расшифровка) хранятся в текстовом файле и могут дополняться вручную (с использованием текстового редактора) или внутри программы, при этом ограничений на длину словаря не существует. Тема кроссворда выбирается пользователем в соответствии с содержанием словаря заданий. Программа позволяет сформировать кроссворд, учитывая ограничения на параметры. В системе имеется возможность сохранения кроссвордов в файл с целью последующего их разгадывания.

Программа написана на языке Object Pascal в среде Delphi v.6.0 и функционирует под управлением операционной системы Windows’2003.

**СОДЕРЖАНИЕ**

[Введение 2](#_Toc462323453)

[1 Описание и анализ предметной области 2](#_Toc462323454)

[1.1 Описание предметной области 2](#_Toc462323455)

[1.2 Описание систем-аналогов 2](#_Toc462323456)

[1.2.1 Система-аналог 1 2](#_Toc462323457)

[1.2.1 Система-аналог 2 2](#_Toc462323458)

[1.3 Диаграмма объектов предметной области 2](#_Toc462323459)

[1.4 Постановка задачи 2](#_Toc462323460)

[2 Проектирование системы 2](#_Toc462323461)

[2.1 Структурная схема системы 2](#_Toc462323462)

[2.2 Спецификация системы 2](#_Toc462323463)

[2.2.1 Функциональная спецификация 2](#_Toc462323464)

[2.2.2 Спецификация качества 2](#_Toc462323465)

[2.2.3 Перечень исключительных ситуаций 2](#_Toc462323466)

[2.3 Разработка прототипа интерфейса пользователя системы 2](#_Toc462323467)

[2.4 Разработка структур данных и классов 2](#_Toc462323468)

[2.5 Логическая модель данных (при необходимости) **Ошибка! Закладка не определена.**](#_Toc462323469)

[2.6 Выбор и обоснование алгоритмов обработки данных /Разработка и описание алгоритмов обработки данных 2](#_Toc462323470)

[2.7 Выбор и обоснование комплекса программных средств 2](#_Toc462323471)

[2.7.1 Выбор языка программирования и среды разработки 2](#_Toc462323472)

[2.7.2 Выбор операционной системы 2](#_Toc462323473)

[2.7.3 Выбор среды программирования 2](#_Toc462323474)

[2.7.4 Выбор системы управления базами данных 2](#_Toc462323475)

[3 Реализация системы 2](#_Toc462323476)

[3.1 Разработка и описание интерфейса пользователя 2](#_Toc462323477)

[3.1.1 Разработка и описание пользовательского меню 2](#_Toc462323478)

[3.1.2 Описание тестового примера 2](#_Toc462323479)

[3.2 Реализация классов и структур данных **Ошибка! Закладка не определена.**](#_Toc462323480)

[3.3 Физическая модель данных (при необходимости) 2](#_Toc462323481)

[3.4 Реализация и описание модулей программы **Ошибка! Закладка не определена.**](#_Toc462323482)

[3.5 Выбор и обоснование комплекса технических средств 2](#_Toc462323483)

[3.5.1 Расчет объема занимаемой памяти 2](#_Toc462323484)

[3.5.2 Минимальные требования, предъявляемые к системе 2](#_Toc462323485)

[Заключение 2](#_Toc462323486)

[Список использованных источников 2](#_Toc462323487)

[Приложение А Руководство пользователя 2](#_Toc462323488)

[Приложение Б Листинг модулей программы 2](#_Toc462323489)

[Приложение В Стили для оформления отчета и записки 2](#_Toc462323490)

[Раздел - 1 Уровень 2](#_Toc462323491)

[3.6 Подраздел (2 уровень) 2](#_Toc462323492)

[3.6.1 Параграф (3 уровень) 2](#_Toc462323493)

[Подпараграф (4 уровень) 2](#_Toc462323494)

Введение

Именно оперативное управление в современных, быстроменяющихся условиях (и внешних, и внутренних) становится ключевым фактором, обеспечивающим конкурентные преимущества. Особенно это характерно для предприятий, занимающихся розничной реализацией нефтепродуктов и сопутствующих товаров на одной или нескольких автозаправочных станциях (АЗС) или комплексах (АЗК). Ведь эти компании сочетают в рамках единого бизнеса различные направления деятельности, которыми необходимо одновременно и эффективно управлять. Оперативное управление торговым процессом - настоятельная потребность современного рынка торговых технологий. Поэтому программные системы, которые призваны обеспечить автоматизацию АЗК, должны содержать не только модули, отвечающие за учет хозяйственных операций, но и модули, позволяющие проводить анализ данных, а также инструменты для принятия управленческих решений [Создать ссылку в списке литературы].

В процессе выполнения курсового проекта необходимо разработать систему моделирования работы АЗС, которая позволит создавать различные структуры АЗС и моделировать поток транспортных средств.

Данная система может быть актуальна для предпринимателей, планирующих открыть собственную АЗС. Данное программное обеспечение (ПО) предоставит возможность:

* проверить пропускную способность определенной топологии заправки;
* провести нагрузочное тестирование с целью определения необходимого объема резервуаров, который обеспечит бесперебойную работу АЗС, и т.д.

При проектировании системы будет использована методология UML. UML (*Unified Modeling Language* ) – это стандартный инструмент для разработки «чертежей» программного обеспечения. Его можно использовать для визуализации, спецификации, конструирования и документирования артефактов программных систем. UML подходит для моделирования любых систем – от информационных систем масштаба предприятия до распределенных Web-приложений и даже встроенных систем реального времени [1].

Разработка системы будет производиться по технологии быстрой разработки приложений *RAD (Rapid Application Development),* которая поддерживается методологией структурного проектирования и включает элементы объектно-ориентированного проектирования и анализа предметной области [2].

1. Описание и анализ предметной области

Проектирование любой программной системы начинается с описания и анализа предметной области. Под предметной областью понимают ту часть реального мира, которая имеет существенное значение или непосредственное отношение к процессу функционирования программы. Предметная область включает в себя только те объекты и взаимосвязи между ними, которые необходимы для описания требований и условий решения некоторой задачи [3].

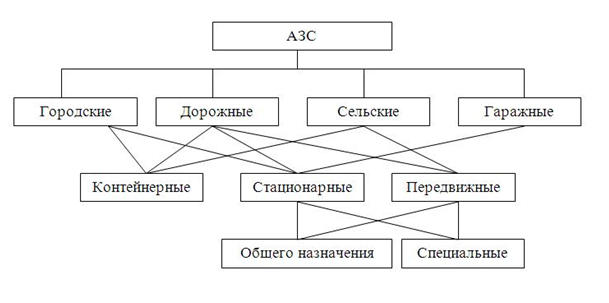
* 1. Описание структуры и работы АЗС

Автозаправочная станция (АЗС) – сооружение для снабжения (заправки) автомобилей, мотоциклов и других самоходных машин жидким топливом, маслом, водой (для омывания стёкол) и воздухом (накачивание шин), а также для продажи фасованных нефтепродуктов, автопринадлежностей и запасных частей. Располагаются на автодорогах и в населённых пунктах, в местах, обеспечивающих удобный заезд и выезд автомобилей. На некоторых загородных АЗС устраиваются площадки для техобслуживания и мойки автомобилей, кафетерий, бытовые помещения [].

Первые АЗС появились в США в начале ХХ века (есть упоминания о 1907). Первые заправки представляли собой одну-две цистерны, стоящие на подпорках, от каждой шли шланги, по которым бензин самотеком поступал в баки автомобилей [].

АЗС можно классифицировать согласно схеме, представленной на рисунке 1:

* по месту размещения – городские, дорожные, сельские и гаражные;
* по конструкции – контейнерные, стационарные, передвижные;
* по функциональному назначению– для заправки государственного и общественного автотранспорта, для заправки личных автомобилей и частных фирм.

  
Рисунок 1 – Классификация АЗС

При проектировании АЗС целесообразно использовать преимущества унификации, распространяя применение одних и тех же конструктивных элементов одновременно на обе группы объектов АЗС – на сооружения и оборудование. Использование типовых АЗС также может дать существенный экономических эффект. Рабочая документация в таких случаях привязывается к участкам строительства АЗС [4].

Рассмотрим принцип работы АЗС…

* 1. Моделирование транспортных потоков
     1. Детерминированный и случайный поток

Транспортный поток — это упорядоченное транспортной сетью движение транспортных средств.

Детерминированный поток событий – это последовательность событий со строго фиксированными неслучайными моментами времени между любыми соседними заявками.

Случайный поток событий – это последовательность событий, в которой промежутки времени между вызовами являются случайными величинами.

T0 T1 T2 T3



t1 t2  t3 t4 t5

Рисунок 2 – Распределение моментов времени t на числовой оси

Поток является детерминированным в том случае, если Ti = const. Если Ti является случайной величиной, то поток является случайным.

* + 1. Методы моделирования случайных величин

Мы говорим, что случайная величина    имеет  показательное  распределение, если

Непрерывная случайная величина X называется распределенной равномерно на отрезке [a,b], если её плотность распределения вероятностей постоянна на данном отрезке:

Пусть непрерывная случайная величина (СВН)  η  задана своим законом распределения:, где – плотность распределения вероятностей, а функция распределения вероятностей. Доказано, что случайная величина распределена равномерно на интервале (0,1).

* 1. Описание систем-аналогов

У разрабатываемой ПС существует мало систем аналогов, которые бы обеспечивали именно моделирование работы АЗС, но существует множество систем и языков, предназначенных для моделирования различных процессов.

* + 1. AnyLogic

Данная система предназначена для имитационного моделирования. Продукт обладает дружественным интерфейсом (для разработчика), но предназначен он не для обычного пользователя, а скорее для разработчика. Так же он не предоставляет готовую модель, нужную для нашей задачи, т.е. сначала нужно разработать модель, а потом проводить эксперименты для получения интересующей пользователя информации. На рисунке 3 приведена экранная форма с примером моделирования в среде AnyLogic.

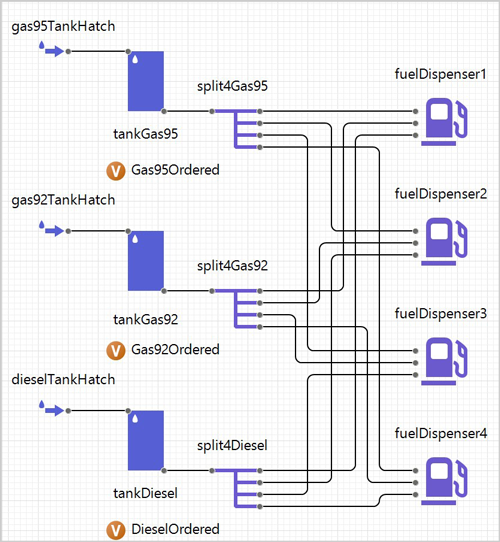
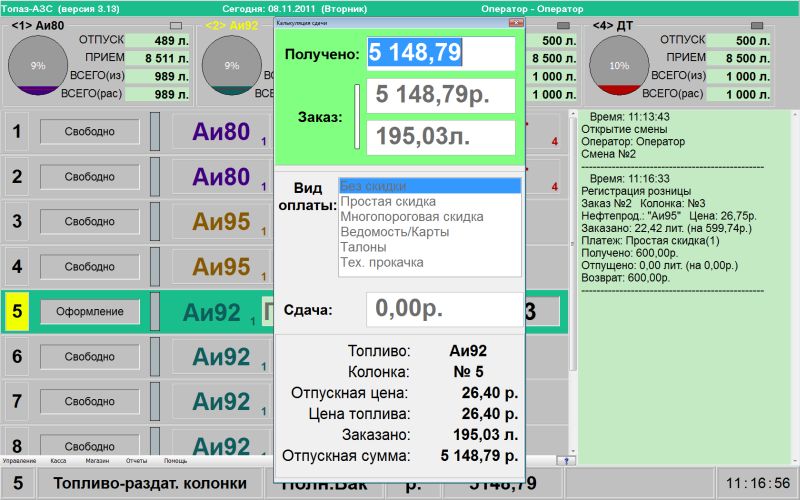


Рисунок 3 – Модель взаимосвязи резервуаров и заправочных колонок в среде «AnyLogic»

* + 1. Топаз-АЗС

Топаз-АЗС – автоматизированная система управления топливораздаточным оборудованием, учета движения ГСМ, реализации товаров, предоставления услуг, получения отчетности. Автоматизирует работу оператора, облегчает администрирование АЗС. Работая с широким спектром основного (ТРК, ГНК) и дополнительного оборудования позволяет успешно решать вопросы комплексной автоматизации.

Топаз-АЗС – дает возможность оператору осуществлять продажу товаров, вести учет дополнительных услуг клиентам. Гибкая система дисконтирования позволяет построить собственную систему обеспечения лояльности как частных клиентов, обслуживаемых за наличный расчет, так и корпоративных клиентов по безналичному расчету. Совместно с ПО "Топаз-Офис" позволяет создать централизованно управляемую сеть автозаправочных комплексов. На рисунке 4 приведена экранная форма работы Топаз-АЗС.

  
Рисунок 4 – Экранная форма Топаз-АЗС

* 1. Диаграмма объектов предметной области

Построение диаграммы объектов предметной области начинается с выделения основных объектов и установления взаимосвязей между ними. Большинство автоматизированных систем являются сложными, что приводит к трудности их описания. По этой причине применяется метод декомпозиции – разделения целого на части. Декомпозиция, как процесс расчленения, позволяет рассматривать любую исследуемую систему как сложную, состоящую из отдельных взаимосвязанных подсистем, которые, в свою очередь, также могут быть расчленены на части [3].

На рисунке 4 приведена диаграмма объектов предметной области. Модель работы АЗС состоит из:

1. Топология АЗС, которая хранится в XML файле, и на которой располагаются элементы топологии, включающие в себя:

* ТРК
* резервуар
* касса
* дорога

1. Поток автомобилей, который перемещается по топологии АЗС и состоит из автомобилей, сведения о которых берутся из базы данных (БД).

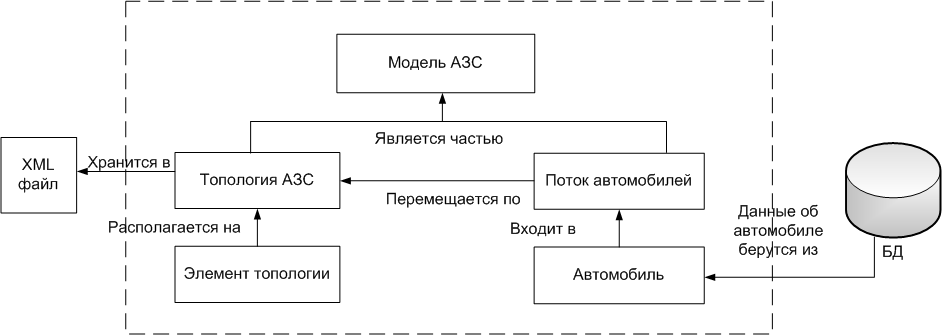


Рисунок 5 – Диаграмма объектов предметной области

* 1. Постановка задачи

Перед авторами поставлена задача – разработать автоматизированную систему, моделирующую работу АЗС.

Система должна функционировать в следующих режимах:

1. Режим конструирования топологии. Пользователь должен задать размеры поля, но на поле наложено следующие ограничения минимальный размер 2х2, а максимальный 10х10. После задания размера поля пользователь должен разместить от одной до пяти топливно-раздаточных колонок (ТРК) и указать тип топлива поступающий в данную колонку, дну кассу, один въезд, один выезд и от одного до пяти резервуаров, указав тип топлива, хранимого в этом резервуаре. После завершения конструирования пользователь может сохранить созданную им топологию в файл.
2. Режим моделирования. После того как пользователь создал или открыл ранее созданную топологию (из файла), он запускает моделирование работы данной АЗС. Во время моделирования пользователь указывает один из двух видов потоков:

* детерминированный;
* случайный;

Если пользователь указывает случайный поток, тогда он должен выбрать один из трёх предложенных законов распределения (ЗР) для этого потока и указать параметры этого ЗР.

Также, пользователь дожжен будет указать, при каком уровне топлива в резервуаре должен вызываться дозаправщик.

Во время разработки системы будет необходимо спроектировать базу данных. В данной БД будут храниться сведения об автомобилях (марка, модель, тип топлива, объем бака), о типах топлива (наименование, цена). На все таблицы базы будут наложены ограничения целостности, и будет исключено полное дублирование записей.

Визуализация работы АЗС необходима для наглядного представления всех процессов проходящих во время моделирования, т.е. представляет пользователю наблюдать за работой сконструированной им АЗС.

Процесс сбора данных о работе АЗС заключается в сборе некой статистики во время работы модели. Данная статистика представляет собой, например, среднее время обслуживания автомобиля, саамы “популярный” тип топлива и т.д. Статистика представляется в удобном для пользователя виде (в виде графиков).

Таким образом, система должна решать следующие задачи:

1. создание/редактирование топологии автозаправочной станции:
2. добавление шаблона на топологию АЗС;
3. удаление шаблона с топологии АЗС;
4. перемещение шаблона по топологии АЗС;
5. задание размера топологии;
6. настройка объектов шаблона:

* резервуар: задание объема и типа топлива;
* ТРК: тип топлива, скорость заправки (литр/секунда);
* касса: вместимость кассы;

1. сохранение в файл и загрузка топологии АЗС из файла с проверкой корректности его структуры;
2. настройка параметров потока автомобилей;
3. выбор типа потока;
4. задание параметров потока;
5. обработка событий автозаправочной станции:
6. работа ТРК;
7. работа кассы;
8. приезд дозаправщика;
9. приезд инкассации;
10. генерация потока автомобилей по заданным параметрам:
11. задание интервала времени между моментами приездов автомобилей (для детерминированного потока);
12. выбор закона распределения и настройка его параметров (для случайного потока);
13. визуализация процессов работы автозаправочной станции;
14. ведение БД:
15. добавление записей в БД;
16. удаление записей из БД;
17. редактирование записей в БД;

проверка целостности данных в БД.

1. Проектирование системы
   1. Структурная схема системы

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

* 1. Спецификация системы

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

* + 1. Функциональная спецификация

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

Функциональная спецификация системы приведена в таблице 1.

Таблица 1 – Перечень функций, выполняемых системой

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название  подсистемы | Название функции | Информационная среда | | | |
| Входные данные | | Выходные данные | |
| Назначение (наименование) | Тип, ограничения | Назначение (наименование) | Тип, ограничения |
|  |  |  |  |  |  |
| 1 Справочная | 1.1 Выдать сведения о разработчиках | Сведения о разработчиках системы (ФИО, номер группы) | Текст (МЕМО) | Визуальное отображение информации | ‑ |
| 1.2 Выдать сведения о системе | Файл справки | Текстовый (\*.HTML) |
| Код ошибки | целое |
| 2 Настройки параметров | 2.1 Задать количество букв в пересечении | Диапазон количества букв: минимальное максимальное | Целое  1  3 | Текущее значение букв в пересечении | Целое |
| 2.2 Подключить словарь понятий | Имя файла | Строка, \*.dict | Список понятий и их определений | Динамический массив строк |
| Код ошибки | Целое |
| 2.3 Задать длину кроссворда | Диапазон длин: минимальное максимальное | Целое  15  250 | Текущее значение длины | Целое |
| Код ошибки | Целое |

Продолжение таблицы 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  | … | … | … | … | … |
| 3 Файловая | 3.1 Загрузить файл с кроссвордом | Имя файла | Строка, \*.kros | Кроссворд | Объект, структура определяется в ходе проектирования |

* + 1. Спецификация качества

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

* + 1. Перечень исключительных ситуаций

*Исключительная ситуация* – это ситуация, при которой система не может выполнить возложенных на нее функций или которая может привести к денормализации работы системы.

В таблице 2 приведен перечень исключительных ситуаций для разрабатываемой системы и описаны реакции системы на их возникновение.

Таблица 2 – Перечень исключительных ситуаций

|  |  |  |
| --- | --- | --- |
| Название  подсистемы | Название исключительной ситуации | Реакция системы |
| 1 Справочная | 1.1 Не возможно открыть файл справки | Выдача сообщения «Файл справки поврежден» |
| 1.2 Не возможно найти файл справки | Выдача сообщения «Отсутствует файл справки» |
| 2 Файловая | 2.1 Попытка открытия файла с несобственным форматом | Выдача сообщения «Файл поврежден или недопустимого формата» |
| 2.2 Файл с заданным именем не существует | Выдача аналогичного сообщения |
| … | … | … |

* 1. Разработка прототипа интерфейса пользователя системы

Вводные слова Вводные слова Вводные слова Вводные слова Вводные слова Вводные слова Вводные слова Вводные слова Вводные слова. Дать определение интерфейса, перечислить основные виды интерфейсов, отметить основные особенности разработки.

Здесь должны быть разработаны прототипы **всех** основных форм приложения с описанием привязанной к ней функциональности, например:

На рисунке 5 приведен прототип экранной формы начальной настройки приложения. Здесь пользователь должен выбрать язык программирования, на котором написан алгоритм, категорию (поиск или сортировка) и нажать кнопку «Далее» для перехода к следующему экрану (форме).

  
Рисунок 1 – Прототип экранной формы начальной настройки приложения

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

На Рисунок ??? приведена навигационная модель разрабатываемого приложения.

****

Рисунок ??? ‒ Навигационная модель приложения

* 1. Информационно-логический проект системы
     1. Язык UML

Унифицированный язык моделирования UML (Unified Modeling Language) – это преемник того поколения методов объектно-ориентированного анализа и проектирования, которые появились в конце   
80-х и начале 90-х годов. Отсутствие единого стандарта и метода построения единой и цельной объектной модели системы мешало широкому распространению объектно-ориентированных методов, поэтому создатели трех наиболее распространенных методологий Г. Буч, Д. Рамбо и А. Якобсон объединили свои усилия под эгидой Rational Software Corporation для создания UML, который и стал стандартом языка объектного моделирования [???].

Текст текст….

* + 1. Диаграмма вариантов использования

Диаграмма вариантов использования является исходным концептуальным представлением или концептуальной моделью системы в процессе ее проектирования и разработки. Она описывает функциональное назначение системы.

Разработка диаграммы вариантов использования преследует цели:

* определить общие границы и контекст моделируемой предметной области на начальных этапах проектирования системы;
* сформулировать общие требования к функциональному поведению проектируемой системы;
* разработать концептуальную модель системы для ее последующей детализации в форме логических и физических моделей;
* подготовить исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями.

В языке UML диаграммы вариантов использования позволяют визуализировать поведение системы, подсистемы или класса, чтобы пользователи могли понять как их использовать, а разработчики – реализовать соответствующий элемент.

Таким образом, суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью так называемых вариантов использования. При этом актером (actor) или действующим лицом называется любая сущность, взаимодействующая с системой извне.

Вариант использования (use case) – внешняя спецификация последовательности действий, которые система или другая сущность могут выполнять в процессе взаимодействия с актерами.

Диаграмма вариантов использования разрабатываемой системы представлена на рисунке ????.

Здесь должно быть краткое описание диаграммы.

Рассмотрим несколько сценариев вариантов использования.

Сценарий «Вход в систему»

*Вариант использования:* вход в систему.

*Краткое описание:* сценарий дает возможность любому пользователю войти в систему.

*Актеры:* пользователь.

*Предусловия:* компьютер пользователя включён, на экране – главное окно операционной системы с набором пиктограмм на рабочем столе.

*Основной поток событий:*

1. Пользователь двойным щелчком левой кнопки мыши по пиктограмме веб-обозревателя запускает данное приложение.
2. Пользователь вводит адрес системы в строку адреса веб-обозревателя.
3. На экране появляется стартовая страница системы.
4. Пользователь переходит по гиперссылке «Вход».
5. На экране появляется страница с формой ввода логина и пароля.
6. Пользователь вводит имя пользователя и пароль и нажимает кнопку «Вход».

А1: пользователь закрыл страницу с формой ввода логина и пароля.  
А2: пользователь перешел по гиперссылке «Справка».

Система проверяет имя пользователя и пароль. Система осуществляет вход с соответствующими правами пользователя. Отображает стартовую страницу приложения в зависимости от прав пользователя, состав пунктов меню настраивается в соответствии с правами пользователя. Вариант использования завершается успешно.

Должно быть описано столько сценариев, сколько членов в команде.

* + 1. Диаграмма классов

Диаграммы классов – это наиболее часто используемый тип диаграмм, которые создаются при моделировании объектно-ориентированных систем, они показывают набор классов, интерфейсов и коопераций, а также их связи. На практике диаграммы классов применяют для моделирования статического представления системы, они служат основой для целой группы взаимосвязанных диаграмм – диаграмм компонентов и диаграмм размещения [???].

На данном этапе для всех информационных объектов, выделенных в системе, разрабатываются классы с указанием полей, методов и свойств, которые регулируют процессы обработки данных (потоки данных заданной структуры) и/или структуры данных.

На рисунке ??? представлена диаграмма классов системы.

Здесь должно быть краткое описание диаграммы.

* + 1. Диаграмма состояний

Каждый объект системы, обладающий определенным поведением, может находиться в определенных состояниях, переходить из состояния в состояние, совершая определенные действия в процессе реализации сценария поведения объекта. Поведение большинства объектов реальных систем можно представить с точки зрения теории конечных автоматов, то есть поведение объекта отражается в его состояниях, и данный тип диаграмм позволяет отразить это графически.

Диаграмма состояний по существу является графом, вершинами которого являются состояние и некоторые другие типы элементов автомата (псевдосостояния), а дугами – переходы из состояния в состояние. Диаграммы состояний могут быть вложены друг в друга, образуя вложенные диаграммы более детального представления отдельных элементов модели [???].

Диаграмма состояний системы в режиме администратора представлена на рисунке ????.

Здесь должно быть краткое описание диаграммы.

* + 1. Диаграмма деятельности

При моделировании поведения проектируемой или анализируемой системы возникает необходимость не только представить процесс изменения ее состояний, но и детализировать особенности алгоритмической и логической реализации выполняемых системой операций.

Для моделирования процесса выполнения операций в языке UML используются диаграммы деятельности. Применяемая в них графическая нотация во многом похожа на нотацию диаграммы состояний, поскольку на этих диаграммах также присутствуют обозначения состояний и переходов. Каждое состояние на диаграмме деятельности соответствует выполнению некоторой элементарной операции, а переход в следующее состояние выполняется только при завершении этой операции [???].

Таким образом, диаграммы деятельности можно считать частным случаем диаграмм состояний. Они позволяют реализовать в языке UML особенности процедурного и синхронного управления, обусловленного завершением внутренних деятельностей и действий. Основным направлением использования диаграмм деятельности является визуализация особенностей реализации операций классов, когда необходимо представить алгоритмы их выполнения. Диаграмма деятельности системы представлена на рисунке ???.

* + 1. Диаграмма последовательности

На данном виде диаграмм показываются взаимодействия объектов, упорядоченные по времени их появления. Таким образом, диаграмма последовательности – временной график работы объектов, связанных между собой для реализации вариантов использования системы.

На диаграмме последовательности приложения показываются взаимодействия между основными классами приложения, описанными ранее. Классы приложения обмениваются сообщениями – элементами модели, предназначенными для представления коммуникации между линиями жизни, вертикальными линиями, отмечающими время взаимодействия между объектами.

Диаграмма последовательности для варианта использования «????» (см. п. 2.4.2) представлена на рисунке ???.

Должно быть столько диаграмм, сколько сценариев было приведено ранее. Описание не нужно, т.к. диаграмма должна точно отражать сценарий.

* + 1. Логическая модель данных (при необходимости)

Проектирование БД является одной из важнейших составных частей процесса создания системы. База данных, рассматриваемая как сложная система, разрабатывается с использованием тех же принципов, что и система в целом.

При проектировании баз данных обычно выделяют три уровня абстракции, на которых происходит последовательное уточнение модели: концептуальный (семантический уровень представления данных в виде абстрактных понятий, учитывающих особенности предметной области), логический (уровень представления в виде структуры данных – сущностей, атрибутов и связей) и физический (уровень реализации базы данных) [???].

Сущность определяется как объект, событие или концепция, информация о котором должна сохраняться. Сущности должны иметь наименование (с четким смысловым значением) в виде существительного в единственном числе. Каждый экземпляр сущности на диаграмме уникален.

Атрибут хранит информацию об определенном свойстве сущности. Атрибут или набор атрибутов, используемый для идентификации экземпляра сущности, называется ключом сущности. В случае если для идентификации экземпляра используется один атрибут, ключ называется простым; в противном случае ключ составной. Каждый экземпляр сущности однозначно определяется ключом.

Объекты могут находиться в определенном отношении, если они по каким-либо свойствам или признакам связаны между собой, правило, по которому экземпляр одной сущности сопоставляется экземпляру другой сущности, известно. Таким образом, множество экземпляров одной сущности отображается по определенному правилу в множество экземпляров другой сущности. Поэтому говорят, что связь – это отображение одной сущности в другую. Связь имеет уникальное имя.

Логическая модель базы данных системы представлена на рисунке ???.

Описание сущностей логической модели базы данных системы представлены в таблицах ???-???.

Таблица 1 – Сущность «Роль пользователя»

|  |  |
| --- | --- |
| Атрибут | Описание |
| Идентификатор | Уникальный идентификатор |
| Название | Название роли пользователя в системе |

* 1. Выбор и обоснование алгоритмов обработки данных /Разработка и описание алгоритмов обработки данных

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

Схемы алгоритмов должны быть оформлены в соответствии с ГОСТ [4].

* 1. Выбор и обоснование комплекса программных средств

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

* + 1. Выбор языка программирования и среды разработки

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

* + 1. Выбор операционной системы

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

* + 1. Выбор среды программирования

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

* + 1. Выбор системы управления базами данных (при необходимости)

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

1. Реализация системы
   1. Разработка и описание интерфейса пользователя
      1. Разработка и описание пользовательского меню
      2. Описание тестового примера
   2. Физическая модель данных (при необходимости)

В процессе проектирования при переходе от концептуальной модели к логической, а затем к физической наблюдается соответствие между основными категориями (см. таблицу ???).

Таблица ??? – Соответствие основных категорий

|  |  |  |  |
| --- | --- | --- | --- |
| Сущность | Отношение | Таблица | Файл |
| Экземпляр | Кортеж | Строка | Запись |
| Атрибут | Домен | Столбец | Поле |

При переходе от логической модели данных (диаграммы сущностных классов (см. рисунок ???)), к физической (схеме таблиц), и наоборот, существует соответствие между элементами модели (таблица ???).

Таблица ??? – Соответствие основных категорий

|  |  |
| --- | --- |
| Логическая модель | Физическая модель |
| Class (Класс) | Table (Таблица) |
| Operation (Операция) | Constraint (Ограничение) |
| Attribute (Атрибут) | Column (Колонка) |
| Package (Пакет) | Scheme (Схема) |
| Component (Компонент) | Database (База данных) |
| Association (Ассоциация) | Relationship (Связь) |
| Нет | Trigger (Тригер) |
| Нет | Index (Индекс) |

* 1. Реализация классов и структур данных
  2. Диаграммы реализации

Диаграммы реализации предназначены для отображения состава компилируемых и выполняемых модулей системы, а так же связей между ними. Диаграммы реализации разделяются на два конкретных вида: диаграммы компонентов (component diagrams) и диаграммы развертывания (deployment diagrams).

* + 1. Диаграмма компонентов

Диаграмма компонентов описывает особенности физической реализации приложения, определяет архитектуру приложения и устанавливает зависимость между компонентами, в роли которых выступает исполняемый код. Диаграмма компонентов отображает общие зависимости между компонентами. Основными графическими элементами диаграммы являются компоненты, интерфейсы и зависимости между ними [????].

Диаграмма компонентов системы изображена на рисунке ????, в нее входят следующие основные компоненты: (перечислить с указанием назначения).

* + 1. Диаграмма развертывания

Диаграмма развертывания применяется для представления общей конфигурации и топологии распределенной системы и содержит распределение компонентов по определенным узлам. Диаграмма показывает наличие физических соединений, маршрутов передачи информации между аппаратными устройствами, задействованными в реализации системы, содержит графические изображения процессоров, устройств, процессов и связей между ними [????].

Диаграмма развертывания системы представлена на рисунке ???.

Должно быть краткое описание диаграммы.

* 1. Выбор и обоснование комплекса технических средств
     1. Расчет объема занимаемой памяти

Расчет объема внешней памяти

Для расчета необходимого объема свободной внешней памяти, необходимой для функционирования системы, воспользуемся следующей формулой:

VЖД = VОС + VПР + [VБД] + [VСПО] + [VФ][[2]](#footnote-2),

где VОС – объем памяти, занимаемый операционной системой (операционная система Windows XP с пакетом обновлений SP3, VОС = 1,5 Гб);

VПР – объем памяти, занимаемый непосредственно файлами приложения (VПР = 80 Мб);

VБД – объем памяти, занимаемый базой данных (всеми таблицами) при ее максимальном заполнении. Пример расчета этой составляющей приведен в таблице 7; исходные данные для расчета взяты из описания таблиц БД.

VСПО – объем памяти, занимаемый всем необходимым сопутствующим программным обеспечением (сюда входят СУБД, фреймворки, MS Office (PowerPoint) и другие средства разработки; дадим оценку сверху VСПО в 2 Гб);

VФ – объем памяти, необходимый для хранения файлов, необходимых для работы программы (дадим ему оценку сверху в 2,5 Мб);

VБД = 4045117680 байт = 3950310 Кб = 3858 Мб = 3,76 Гб.

Таким образом, суммарный объем внешней памяти составит:

VЖД = 1,5 Гб + 80 Мб + 3,76 Гб + 2 Гб + 2,5 Мб ≈ 7,5 Гб.

Таблица 7 – Расчет объема внешней памяти, необходимой для хранения БД (фрагмент)

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица | Размер записи (байт) | Максимум записей | Всего (байт) |
| Пользователь | 396 | 50 | 19800 |
| Роль | 56 | 10 | 560 |
| Назначенная роль | 48 | 500 | 24000 |
| … | … | … | .. |
| Итого | | | 4045117680 |

Расчет объема ОЗУ

Для расчета необходимого объема ОЗУ воспользуемся следующей формулой:

VОЗУ = VОС + VПР + [VСПО] + [VБД],

где VОС – ОЗУ, занимаемое операционной системой (256 Мб);

VПР – ОЗУ, которое займет само приложение (не превысит 8 Мб);

VСПО – ОЗУ, занимаемое СУБД и другим сопутствующим ПО (оценим его сверху значением в 128 Мб);

VБД – объем данных из базы, который может быть одновременно загружен в оперативную память (дадим ему оценку сверху в 10 Мб).

Суммарные объемы ОЗУ составит:

VОЗУ = 256 Мб + 8 Мб + 128 Мб + 20 Мб = 412 Мб.

Таким образом, 512 Мб оперативной памяти можно счесть минимально необходимым для функционирования системы.

* + 1. Минимальные требования, предъявляемые к системе

Для корректного функционирования системы необходимо:

1. тип ЭВМ: x86-64 совместимый;
2. объем ОЗУ – не менее 512 Мб;
3. объем свободного дискового пространства – не менее 10 Гб;
4. клавиатура или иное устройство ввода;
5. мышь или иное манипулирующее устройство.

И другие средства, поддерживающие функционирование системы.

Заключение

Во время лабораторного практикума была разработана автоматизированная система …, позволяющая ….

В заключении должны быть отражены основные результаты работы, желательно сделать это с привязкой к разделам отчета, например:

В первом разделе были изучены основные понятия предметной области, исследованы характеристики систем-аналогов, на основании этого выполнена объектная декомпозиция, отраженная в диаграмме объектов.

Список использованных источников

1. Зеленко, Л.С. Методические указания к лабораторному практикуму по дисциплине «Программная инженерия» [Текст]/Л.С.Зеленко.– СГАУ, 2012. – 42 с.
2. Вендров, А.М. CASE-технологии. Современные методы и средств проектирования информационных систем. [Текст] / А. М. Вендров – М.: Финансы и статистика, 1999. – 256 с. : ил.
3. Формирование модели предметной области [Электронный ресурс]. – URL: <http://5fan.ru/wievjob.php?id=45484> (дата обращения: 27.09.2016).
4. Кнут, Д. Искусство программирования. Том 3. Сортировка и поиск [Текст]/ Д. Кнут. – М.: Издательский дом «Вильямс», 2013. – 824 с.
5. Буч, Г. Язык UML. Руководство пользователя [Текст] / Г. Буч, Д. Рамбо, А. Джекобсон. – М.: ДМК, 2003. – 432 с.
6. ГОСТ 19.701-90. Схемы алгоритмов, программ, данных и систем [Текст]. – М.: Изд-во стандартов, 1991. – 20 с.

Пример оформления электронных ресурсов

1. Диаграмма вариантов использования [Электронный ресурс]. –www.intuit.ru/studies/courses/32/32/lecture/1004 (дата обращения 05.09.2016 г.).

Приложение А  
Руководство пользователя

А.1 Назначение системы

Приводится краткое описание возможностей системы.

А.2 Условия работы системы

***Пример.***

Для корректной работы системы необходимо наличие соответствующих программных и аппаратных средств.

1) Требования к техническому обеспечению:

* ЭВМ типа IBM PC;
* процессор типа x86 или x64 тактовой частоты 1400 МГц и выше;
* …

2) Требования к программному обеспечению:

* операционная система Windows XP SP3 и выше;
* установленная платформа .Net версии 4.0 и выше;
* установленная СУБД ….

А.3 Установка системы

***Пример.***

Система поставляется в виде zip-архива. Данный файл необходимо распаковать в любую директорию на жестком диске. Запускаемым файлом системы является файл ххх.exe.[[3]](#footnote-3)

А.4 Работа с системой

А.4.1 Работа с системой в режиме администратора (если необходимо)

Вход в систему (авторизация)

…

А.4.2 Работа с системой в режиме пользователя

Вход в систему (авторизация)

Вход в систему (регистрация)

Настройка параметров кроссворда

Приложение Б   
Листинг модулей программы

приложение в  
Стили для оформления отчета и записки

раздел - 1 уровень

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

На рисунке 1[[4]](#footnote-4) приведен внешний вид ….

  
Рисунок 1 – Внешний вид компьютера (подрисуночная надпись)

В таблице 1 [[5]](#footnote-5)приведено описание ….

Таблица 1 – Характеристики…

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Шапка таблицы | Шапка таблицы | Шапка таблицы | Шапка таблицы | Шапка таблицы |
| Содержание таблицы |  |  |  |  |
| Содержание таблицы |  |  |  |  |

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

* 1. Подраздел (2 уровень)

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

* + 1. Параграф (3 уровень)

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

1. Элемент Списка нумерованного 2
2. Элемент Списка нумерованного 2
3. Элемент Списка нумерованного 2

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

1. Элемент Списка нумерованного 1
2. Элемент Списка нумерованного 1
3. Элемент Списка нумерованного 1

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

* Элемент Списка маркированного
* Элемент Списка маркированного
* Элемент Списка маркированного

Подпараграф (4 уровень)[[6]](#footnote-6)

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

1. *Количество страниц, рисунков, таблиц указывается с учетом приложений* [↑](#footnote-ref-1)
2. […] – значения, указанные в таких скобках, могут отсутствовать [↑](#footnote-ref-2)
3. Если необходимы дополнительные ресурсы для обеспечения работоспособности системы, то все для них также должны быть перечислены условия установки. *Если установка нестандартная, то она должна быть подробно описана (в объеме, достаточном для понимания пользователя).* [↑](#footnote-ref-3)
4. Ссылка на рисунок должна быть перед рисунком [↑](#footnote-ref-4)
5. Ссылка на таблицу должна быть перед таблицей [↑](#footnote-ref-5)
6. 4-ый уровень заголовков не нумеруется [↑](#footnote-ref-6)