****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**студентов группы № 6412 Б 300   
А.Г. Грязнова   
А.А. Епифанцева  
А.А. Рабкиной  
И.С. Рыкова

Тема проекта: **«Система моделирования работы автозаправочной станции»**

**Исходные данные к проекту:**

1. **Характеристика объекта автоматизации:**

## объект автоматизации: автозаправочная станция (АЗС);

## виды автоматизируемой деятельности:

* + процесс конструирования топологии АЗС;
  + процесс моделирования работы АЗС;
  + процесс ведения базы данных;
  + процесс визуализации работы системы;
  + процесс сбора данных о работе АЗС;

## количество шаблонов топологии – 7;

## минимальный размер топологии АЗС по горизонтали – 2;

## максимальный размер топологии АЗС по горизонтали – 10;

## минимальный размер топологии АЗС по вертикали – 2;

## максимальный размер топологии АЗС по вертикали – 10;

## минимальное количество топливно-раздаточных колонок (ТРК) – 1;

## максимальное количество топливно-раздаточных колонок (ТРК) – 5;

## количество касс – 1;

## количество въездов – 1;

## количество выездов – 1;

## минимальное количество резервуаров – 1;

## максимальное количество резервуаров – 5;

## объем резервуара – 20 м3;

## в каждой ТРК может находиться один вид топлива;

## минимальное количество видов топлива – 1;

## максимальное количество видов топлива – 1;

## скорость подачи топлива ТРК – 40 л/ мин;

## скорость подачи топлива дозаправщиком – 500 л/ мин;

## минимальное количество топлива, при котором приезжает дозаправщик, – 30% от полного объёма резервуара;

## вероятность заезда автомобиля на заправку – более 0.5;

## лимит средств в кассе, при котором приезжает инкассация, – 700 тысяч рублей;

## вместимость кассы – 1 миллион рублей;

## количество типов потоков автомобилей – 2;

## количество законов распределения случайной величины – 3;

## минимальное количество автомобилей – 3;

## максимальное количество автомобилей – 10.

1. **Требования к информационному обеспечению:**
2. информационное обеспечение разрабатывается на основе следующих документов:
   * свод правил станции автомобильные заправочные, требования пожарной безопасности [Электронный ресурс]. – mchs.gov.ru/document/3744769 (дата обращения 11.09.2016 г.);
   * СНиП 2.11.03-93 «Склады нефти и нефтепродуктов. Противопожарные нормы» [Электронный ресурс]. – docs.cntd.ru/document/871001020 (дата обращения 14.09.2016 г.);
3. тип АЗС – модульная;
4. АЗС расположена в пределах населённого пункта;
5. для резервуаров зарезервировано отдельное пространство на топологии с въездом и выездом для заправщика;
6. структура базы данных (БД) разрабатывается на основе следующих сведений:
   * информации о типах топлива;
   * информации о типах автомобилей;
   * информации о ТРК;
7. обеспечить целостность БД;
8. в системе должна храниться справочная информация о видах топлива и автомобилях;
9. топология АЗС хранится в файле, структура которого определяется в ходе проектирования;
10. шаблоны АЗС хранятся в графических файлах формата png.
11. **Требования к техническому обеспечению:**
12. тип ЭВМ – IBM PC совместимый;
13. монитор с разрешающей способностью не ниже 800 х 600;
14. манипулятор – мышь;
15. технические характеристики определяются в процессе выполнения курсовой работы.
16. **Требования к программному обеспечению:**
17. тип операционной системы – Windows 98 и выше, Windows 7, Windows Vista;
18. язык программирования – Java 8;
19. среда программирования – Intellij IDEA 16;
20. среда проектирования – StarUML 2.7.0.
21. **Общие требования к проектируемой системе:**

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

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;
18. проверка целостности данных в БД;
19. выдача справочной информации о системе.

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

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

Руководитель   
курсового проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Л.С. Зеленко / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Задание принял  
 к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ А.Г. Грязнов / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ А.А. Епифанцев / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ А.А. Рабкина / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ И.С. Рыков / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**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 12](#_Toc462323453)

[1 Описание и анализ предметной области 14](#_Toc462323454)

[1.1 Описание предметной области 14](#_Toc462323455)

[1.2 Описание систем-аналогов 21](#_Toc462323456)

[1.2.1 Система-аналог 1 **Ошибка! Закладка не определена.**](#_Toc462323457)

[1.2.1 Система-аналог 2 **Ошибка! Закладка не определена.**](#_Toc462323458)

[1.3 Диаграмма объектов предметной области 21](#_Toc462323459)

[1.4 Постановка задачи 24](#_Toc462323460)

[2 Проектирование системы 28](#_Toc462323461)

[2.1 Структурная схема системы 28](#_Toc462323462)

[2.2 Спецификация системы 28](#_Toc462323463)

[2.2.1 Функциональная спецификация 30](#_Toc462323464)

[2.2.2 Спецификация качества 35](#_Toc462323465)

[2.2.3 Перечень исключительных ситуаций 31](#_Toc462323466)

[2.3 Разработка прототипа интерфейса пользователя системы 36](#_Toc462323467)

[2.4 Разработка структур данных и классов 38](#_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 44](#_Toc462323470)

[2.7 Выбор и обоснование комплекса программных средств 44](#_Toc462323471)

[2.7.1 Выбор языка программирования и среды разработки 44](#_Toc462323472)

[2.7.2 Выбор операционной системы 44](#_Toc462323473)

[2.7.3 Выбор среды программирования 44](#_Toc462323474)

[2.7.4 Выбор системы управления базами данных 45](#_Toc462323475)

[3 Реализация системы 46](#_Toc462323476)

[3.1 Разработка и описание интерфейса пользователя 46](#_Toc462323477)

[3.1.1 Разработка и описание пользовательского меню 46](#_Toc462323478)

[3.1.2 Описание тестового примера 46](#_Toc462323479)

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

[3.3 Физическая модель данных (при необходимости) 46](#_Toc462323481)

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

[3.5 Выбор и обоснование комплекса технических средств 47](#_Toc462323483)

[3.5.1 Расчет объема занимаемой памяти 48](#_Toc462323484)

[3.5.2 Минимальные требования, предъявляемые к системе 49](#_Toc462323485)

[Заключение 51](#_Toc462323486)

[Список использованных источников 52](#_Toc462323487)

[Приложение А Руководство пользователя 54](#_Toc462323488)

[Приложение Б Листинг модулей программы 56](#_Toc462323489)

[Приложение В Стили для оформления отчета и записки 57](#_Toc462323490)

[Раздел - 1 Уровень 57](#_Toc462323491)

[3.6 Подраздел (2 уровень) 58](#_Toc462323492)

[3.6.1 Параграф (3 уровень) 58](#_Toc462323493)

[Подпараграф (4 уровень) 59](#_Toc462323494)

Введение

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 [].

В процессе выполнения курсовой работы 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 [].

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 [].

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 [].

* 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 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 по следующим признакам (рисунок 1) [, ]:

1. по месту размещения – городские, дорожные, сельские и гаражные;
2. по конструкции:

* контейнерная автозаправочная станция – АЗС, технологическая система которой предназначена для заправки транспортных средств только жидким моторным топливом и характеризуется надземным расположением резервуаров и размещением топливно-раздаточных колонок (ТРК) в контейнере хранения топлива, выполненном как единое заводское изделие (рисунок 2);

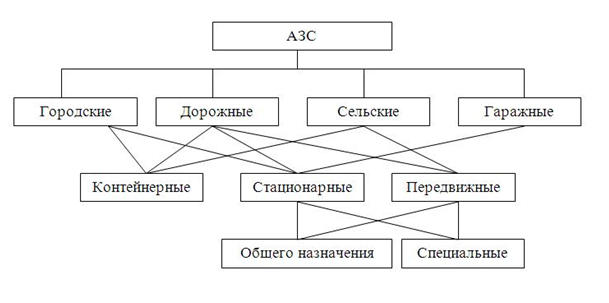
  
Рисунок 1 – Классификация АЗС

  
Рисунок 2 – Контейнерная АЗС

* стационарная (традиционная) автозаправочная станция – АЗС, технологическая система которой предназначена для заправки транспортных средств только жидким моторным топливом и характеризуется подземным расположением резервуаров и их разнесением с ТРК (рисунок 3);
* передвижная автозаправочная станция жидкого моторного топлива (ПАЗС) – АЗС, предназначенная, в том числе, для розничной продажи только жидкого моторного топлива, технологическая система которой установлена на автомобильном шасси, прицепе или полуприцепе и выполнена как единое заводское изделие (рисунок 4);

  
Рисунок 3 – Стационарная АЗС  
Рисунок 4 – Передвижная АЗС

* блочная автозаправочная станция – АЗС, технологическая система которой предназначена для заправки транспортных средств только жидким моторным топливом и характеризуется подземным расположением резервуаров и размещением ТРК над блоком хранения топлива, выполненным как единое заводское изделие (рисунок 5);
* модульная автозаправочная станция – АЗС, технологическая система которой предназначена для заправки транспортных средств только жидким моторным топливом и характеризуется надземным расположением резервуаров и разнесением ТРК и контейнера хранения топлива, выполненного как единое заводское изделие (рисунок 6);

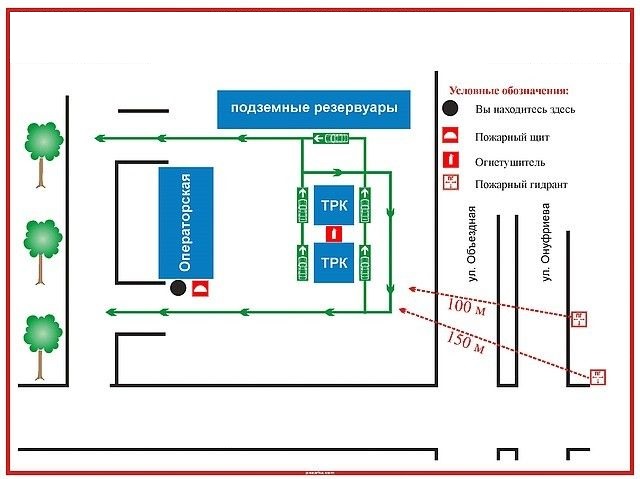
  
Рисунок 5 – Блочная АЗС

  
Рисунок 6 – Модульная АЗС

1. по функциональному назначению – для заправки государственного и общественного автотранспорта, для заправки личных автомобилей и частных фирм.

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 [].

Рассмотрим основные элементы АЗС, особенности работы которых необходимо учесть при проектировании автоматизированной системы. На рисунке 7 приведен пример топологии АЗС.

  
Рисунок 7 – Пример топологии АЗС

Топливораздаточная колонка (ТРК) – единица оборудования автозаправочной станции, предназначенная, согласно технической документации, для отпуска одного или нескольких видов моторного топлива и его залива в топливные баки транспортных средств, оснащенная любым количеством топливораздаточных пистолетов. Отпускаемое топливо измеряется мерными сосудами или объёмными счётчиками и регистрируется контрольным устройством. Наиболее распространены ТРК производительностью 5 – 40 л/мин с минимальной дозой отпуска топлива 2 л [].

Для хранения топлива и масел на АЗС и АЗК применяются подземные и наземные металлические резервуары вертикального и горизонтального типов, одно- и многокамерные. Выбор вместимости резервуаров и их количества зависит от мощности АЗС, которая характеризуется количеством заправок в сутки (250, 500, 750, 1000), видом топлива (бензин, дизельное топливо), емкостью топливного бака. Стационарные АЗС, кроме количества заправок в сутки, характеризуются способностью производить число заправок в час в период максимального скопления автомобилей (часы «пик»). Для 250 – это 57 заправленных автомобилей, для 500 – 100, для 750 – 135, для 1000 – 170. Для стационарных АЗС характерно подземное расположение резервуаров. Как правило, имеется несколько резервуаров для различных марок топлива и автомобильных масел, располагаемых в одной или нескольких группах с общими или отдельными линиями наполнения, выдачи топлива и системами контроля []. В зависимости от объемов резервуары должны соответствовать типам, указанным в таблице 1. Однако, по требованию заказчика допускается изготовлять резервуары типов, указанных в таблице 2 [].

Таблица 1 – Типы резервуаров

|  |  |  |
| --- | --- | --- |
| Обозначение типов | Номинальный объем, куб.м | Область применения |
| P-5 | 5 | Для хранения нефтепродуктов |
| P-10 | 10 |
| P-25 | 25 |
| P-50 | 50 |
| P-75 | 75 |
| P-100 | 100 |

Рассмотрим принцип работы АЗС. Привезенное топливо из автомобильной цистерны через гибкий шланг поступает на входной фильтр топлива с огнепреградителем. Входной фильтр очищает топливо от окалины и грязи. Это первая ступень грубой очистки топлива. Огнепреградитель исключает возможное попадание искры и пламени в резервуар. Шланг присоединяется к фильтру с помощью быстросъемной муфты, которая обеспечивает незамедлительное разъединение в случае необходимости.

Так как большинство аварийных ситуаций может возникнуть во время слива топлива из автомобильной цистерны в резервуар, то для быстрого автоматического прекращения слива применяют электромагнитный клапан (ЭМК).

Таблица 2 – Допустимые типы резервуаров

|  |  |  |
| --- | --- | --- |
| Обозначение типов | Номинальный объем, куб. м | Область применения |
| P-4 | 4 | Для хранения и транспортирования нефтепродуктов |
| P-8 | 8 |
| P-20 | 20 | Для хранения нефтепродуктов |
| P-60 | 60 |

ЭМК независимо от оператора АЗС перекрывает трубопровод линии наполнения и не дает развиться аварийной ситуации. Второй конец сливного трубопровода опускается в резервуар на 100-120 мм ниже всасывающего обратного клапана, то есть он постоянно находится в мертвом остатке, чем создается гидравлический затвор в сливном трубопроводе.

Всасывающий трубопровод изготавливается из газовых труб диаметром 11/2". На конце, опущенном в резервуар, монтируется всасывающий отсечной клапан. Расстояние от нижней части клапана до дна резервуара должно быть 120 - 200 мм. Это позволяет забирать из резервуара чистый нефтепродукт. Всасывающий клапан обеспечивает невозможность перетекания топлива из всасывающего трубопровода в резервуар после прекращения работы насоса [].

* 1. Моделирование транспортных потоков

Транспортный поток – это упорядоченное транспортной сетью движение транспортных средств [13].

Поведение транспортного потока можно описать с помощью математического понятия «поток событий». На рисунке 8 приведен пример потока событий, где Ti – расстояние между событиями.

 T0 T1 T2 T3

t0 = 0 t1 t2  t3 t4

Рисунок 8 – Пример потока событий

* + 1. Детерминированный и случайный поток

Детерминированный поток событий – это последовательность событий со строго фиксированными неслучайными моментами времени между любыми соседними заявками [14]. В этом случае Ti = const.

Случайный поток событий – это последовательность событий, в которой промежутки времени Ti между появлениями события являются случайными величинами [14].

* + 1. Законы распределения случайных величин

*Случайной величиной (СВ)* называется величина, которая в результате опыта может принять то или иное значение, причем неизвестно заранее, какое именно [2].

Непрерывная случайная величина **** имеет *показательное распределение* (показательный, экспоненциальный закон распределения), если множество ее возможных значений , аплотность вероятности (ПВ) имеет вид:

– параметр показательного распределения.

Функция распределения СВ имеет вид:

(1)

Графики ПВ и функции распределения СВ изображены на рисунке 9.

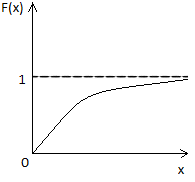
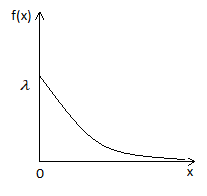


Рисунок 9 – Графики плотности вероятности и функции распределения показательной СВ

Непрерывная случайная величина X имеет равномерное распределение (равномерный ЗР) на отрезке [a, b], если множество ее возможных значений, X лежит внутри этого отрезка, аплотность вероятности постоянна на этом отрезке:

****

Константа *С* при этом однозначно определяется из условия нормировки:

****, то есть .

Таким образом, равномерно распределенная СВ имеет ПВ:

****

Функция распределения равномерно распределеннойСВ:

(2)

Графики ПВ и ФР равномерно распределенной СВ изображены на рисунках 10 и 11 соответственно.

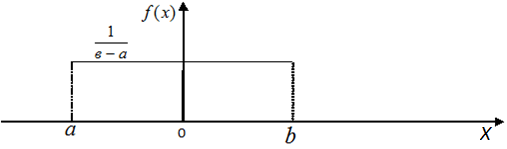


Рисунок 10 – График плотности вероятности  
 равномерно распределенной СВ

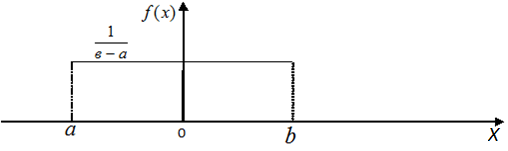


Рисунок 11 – График функции распределения  
 равномерно распределенной СВ

НСВ **** имеет *нормальное распределение* (нормальный, гауссовский ЗР) с параметрами , если множество ее возможных значений   
, аПВ имеет вид:

****.

Сокращенная запись .

Кривая ПВ СВ имеет симметричный вид относительно прямой и имеет максимум в точке .

Если положить и учесть, что Ф(3) = 0,9987 , то получаем:

.

Полученный результат носит название «Правило трех сигма». Он означает, что «практически все» значения СВ находятся внутри интервала в том смысле, что вероятность СВ принять значение, не принадлежащее этому интервалу, пренебрежимо мала

().

На рисунке 12 изображены графики функции распределения и плотности вероятности.

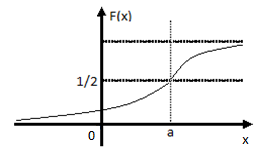
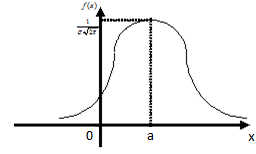


Рисунок 12 – Графики функции распределения и плотности вероятности нормально распределенной СВ

* + 1. Метод генерации нормально распределенных чисел, использующий центральную предельную теорему

Общая идея метода, использующего центральную предельную теорему (ЦПТ), следующая: требуется сложить случайные числа с любым законом распределения, нормализовать их и перевести в нужный диапазон нормального распределения [15].

Допустим, что в целях имитации необходимо получить ряд случайных чисел *x*, распределенных по нормальному закону с заданными математическим ожиданием *mx* и среднеквадратичным отклонением *σx*. Рассмотрим алгоритм решения данной задачи.

1. Сложим *n* случайных чисел, используя стандартный генератор случайных чисел:

.

Согласно ЦПТ, числа *V* образуют ряд значений, распределенный по нормальному закону. Эти числа тем лучше описывают нормальный закон, чем больше параметр *n*. Заметим, что закон распределения чисел *V* имеет математическое ожидание *mV* = *n*/2,*σV* = sqrt(*n*/12). Поэтому он является смещенным относительно заданного произвольного.

1. С помощью формулы *z* = (*V* – *mV*)/*σV* нормализуем этот ряд. Получим нормализованный закон нормального распределения чисел *Z*. То есть *mz* = 0, *σz* = 1.
2. Формулой (сдвиг на *mx* и масштабирование на *σx*) преобразуем ряд *Z* в ряд *x*: *x* = *z* · *σx* + *mx*.
   * 1. Метод обратной (инверсной) функции

Допустим, что нам задан интегральный закон распределения вероятности  , где  – функция плотности вероятности.

Тогда достаточно разыграть случайное число, равномерно распределенное в интервале от 0 до 1. Поскольку функция  тоже изменяется в данном интервале, то случайное событие  можно определить взятием обратной функции по графику или аналитически:

 ,

где  – число, генерируемое эталонным ГСЧ в интервале от 0 до 1,  
 – сгенерированная в итоге случайная величина.

Данным методом особенно удобно пользоваться в случае, когда интегральный закон распределения вероятности задан аналитически и возможно аналитическое взятие обратной функции от него. Графически суть метода изображена на рисунке 13**.**

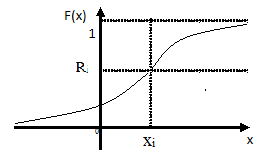


Рисунок 13 – Иллюстрация метода обратной функции

Для показательного закона распределения (см. формулу (1)) обратная функция имеет вид .

Для равномерного ЗР (см. формулу (2)) обратная функция принимает вид .

* 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 14 приведена экранная форма с примером моделирования в среде AnyLogic.

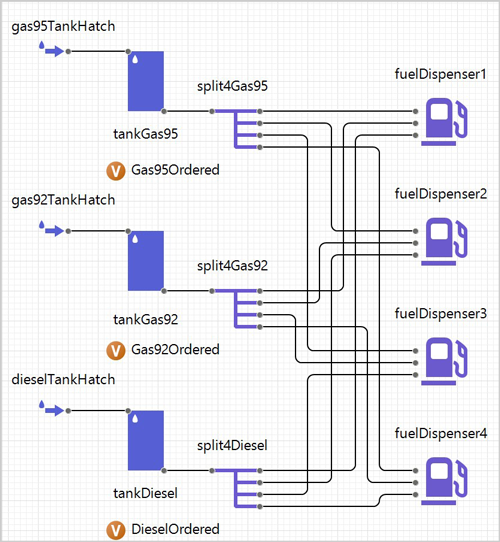
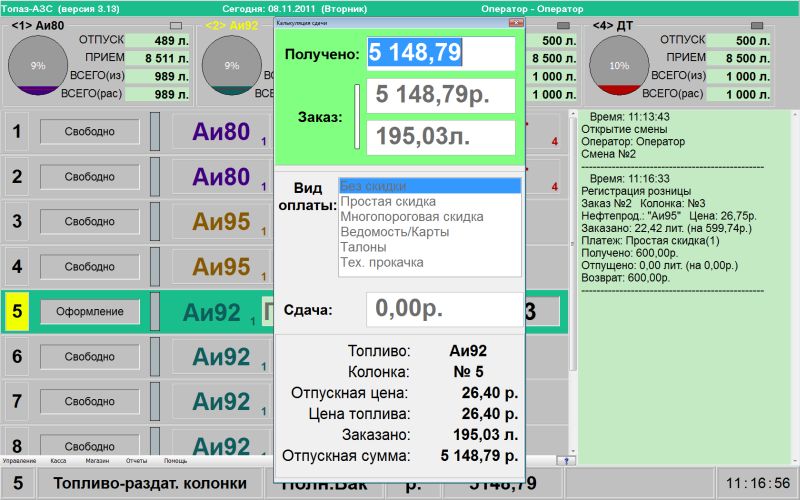


Рисунок 14 – Модель взаимосвязи резервуаров и заправочных колонок  
 в среде «AnyLogic»

* + 1. Топаз-АЗС

Топаз-АЗС – автоматизированная система управления топливораздаточным оборудованием, учета движения горюче-смазочных материалов, 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 15 приведена экранная форма работы Топаз-АЗС.

  
Рисунок 15 – 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 разделены на части [].

На рисунке 16 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).

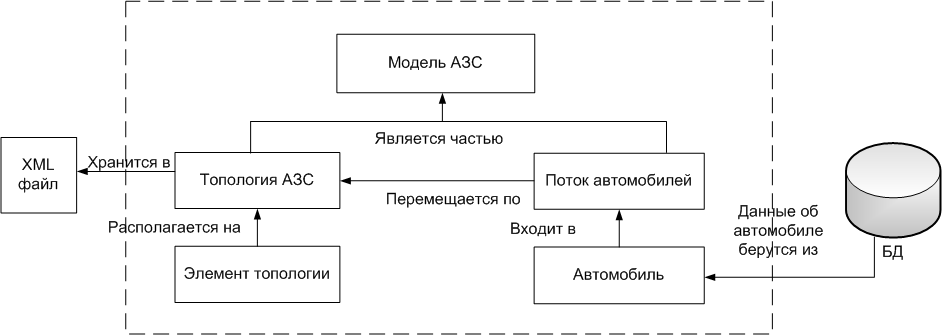


Рисунок 16 – Диаграмма объектов предметной области

* 1. Постановка задачи

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

Система должна функционировать в следующих режимах:

1. *Режим конструирования топологии*. При создании топологии пользователь должен задать размеры поля (от 2х2 до 10х10 клеток). После этого система должна отобразить шаблон (пустое поле заданного размера). Пользователю должна быть предоставлена возможность размещения объектов топологии, представленных в виде шаблонов (изображения хранятся в файлах формата png):

* ТРК (от 1 до 5 шт.);
* касса (1 шт.);
* въезд (1 шт.);
* выезд(1 шт.), который не дожжен совпадать с въездом;
* резервуар (от 1 до 5 шт.);

Дополнительно пользователь должен определены параметры ТРК (тип топлива, скорость заправки) и резервуара (тип топлива, объем (не более 20 м3)). При необходимости топологию можно сохранить в файл.

1. *Режим моделирования.* Моделирование работы данной АЗС может быть запущено только после того, как пользователь создал или загрузил из файла ранее созданную топологию. Для этого пользователь должен выбрать один из двух видов потоков:

* детерминированный (необходимо задать интервал появления автомобилей на АЗС);
* случайный (пользователь должен выбрать закон распределения временных интервалов и указать его параметры (см. п. 1.2.2)).

Система должна рассчитывать оптимальный маршрут для каждого заезжающего на АЗС автомобиля и обеспечить организацию очереди автомобилей.

Также пользователь должен указать, при каком уровне топлива в резервуаре должен вызываться дозаправщик (не менее 30% от полного объема резервуара).

В системе при превышении лимита кассы (700 тыс.руб.) должна приезжать машина инкассации.

Все данные, необходимые для работы системы, должны храниться в базе данных (БД), а именно: сведения об автомобилях (марка, модель, тип топлива, объем бака), о типах топлива (наименование, цена). В системе должна быть обеспечена целостность данных.

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

Во время работы система должна собирать статистику работы АЗС и предоставлять эту информацию в виде графиков.

Кроме того, система должна предоставлять справочную информацию о возможностях системы.

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;
18. проверка целостности данных в БД;
19. выдача справочной информации о системе.
20. Проектирование системы
    1. Структурная схема системы

На основании проведенного выше анализа предметной области формируются общие требования к функциям, которые должна выполнять система. Проектируемую систему разобьем на подсистемы и определим взаимосвязи между ними.

*Система* – это совокупность объектов, компонентов или элементов произвольной природы, образующих некоторую целостность. *Структура системы* – это устойчивая во времени совокупность взаимосвязей между ее элементами или компонентами. Структура системы предполагает вложенность элементов одной системы в другую [].

Структурная схема системы строится на основе функционально-модульного подхода, основанного на принципе алгоритмической декомпозиции с выделением функциональных элементов и установлением строгого порядка выполняемых действий, т.е. в основе лежит иерархический подход с выделением вначале функ­циональных действий, затем независимых компонентов с дальнейшей их детализацией [].

Структурная системы моделирования работы автозаправочной станции приведена на рисунке 17.

В проектируемую систему входит семь подсистем:

1. *Подсистема управления, которая отвечает за взаимодействие всех подсистем.*
2. *Справочная подсистема*, которая обеспечивает выдачу справочной информации об авторах и о системе.
3. *Подсистема конструирования топологии,* в которую входят:

*подсистема настройки топологии*, которая предоставляет пользователю настроить некоторые элементы топологии;

*подсистема конструирования топологии*, которая позволяет пользователю создавать топологию АЗС.

  
Рисунок 17 – Структурная схема автоматизированной системы

1. Подсистема моделирования, в которую входят:

* *Подсистема настройки параметров моделирования,* которая позволяет указать пользователю такие параметры моделирования, тип потока (детерминированный или случайный), уровень топлива в резервуаре, при котором вызывается дозаправщик и, т.д;
* *Подсистема генерации потока автомобилей*, которая по заданным пользователем параметрам генерирует поток автомобилей.

1. *Подсистема визуализации*, которая обеспечивает удобное представление структуры АЗС при конструировании и наглядность работы АЗС при моделировании.
2. *Файловая подсистема*, которая позволяет пользователю сохранять в файл сконструированную им топологию. Также файловая подсистема используется во время моделирования для загрузки изображений автомобилей.
3. *Подсистема работы с БД* обеспечивает работу с базой данных, в которой хранится вся информация об автомобилях.
   1. Спецификация системы

Разработка программной системы (ПС) начинается с формулировки требований, в которой, исходя из пожеланий к ней, должен быть получен документ, в котором должно быть достаточно точно определены требования к разработчикам []. Этот документ называется спецификацией требований. Он играет роль точной постановки задачи и включает две составные части:

спецификация качества на ПС;

функциональная спецификация.

Разработка спецификации качества предшествует функциональной спецификации. В нее включаются требования к технологическим процессам, перечисляющиеся основными показателями качества ПС (надежность, функциональность, эффективность).

* + 1. Функциональная спецификация

*Функциональная спецификация* включает в себя описание всех функций, которые должна выполнять П/С с указанием обрабатываемых результирующих данных. Она строится на естественном языке, но может включать в себя и математические модели, записанные на формализованном языке и диаграммы, выполненные в том или ином стандарте.

Функциональная спецификация включает в себя [20]:

* описание внешней информационной среды;
* определение функций, определенных на множестве состояний этой информационной среды;
* описание исключительных ситуаций, которые могут возникнуть при выполнении функций ПС и реакции на эти ситуации.

*Функциональная спецификация* служит уточнением *структурной схемы*, где определяются все функции, которые выполняет система (с разбивкой на подсистемы), исключительные ситуации, возникающие в каждой из подсистем, а также и информационная среда (входные и выходные параметры) каждой из функций всех *подсистем*.

Функциональная спецификация системы приведена в таблице 1.

* + 1. Спецификация качества

Основу инженерных методов в программировании составляет повышение качества, для достижения которого сформировались методы определения требований к качеству, подходы к выбору и усовершенствованию моделей метрического анализа показателей качества, методы количественного измерения показателей качества на этапах жизненный цикл (ЖЦ) .

Стандарт ГОСТ 2844-94 дает определение качества ПО как совокупность свойств (показателей качества) ПО, которые обеспечивают его способность удовлетворять потребности заказчика в соответствии с

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 Конструирования топологии | 2.1 Задать размер поля по вертикали | Диапазон клеток по вертикали | Целое  2 .. 10 | Значение размера по вертикали | Целое |
| 2.2 Задать размер поля по горизонтали | Диапазон клеток по горизонтали | Значение размера по горизонтали | Целое |
| 2.3 Выбрать элемент топологии | Список элементов топологии | ТРК, касса, резервуар, въезд, выезд | Объект «Элемент топологии» | Структура определяется в ходе проектирования |
| 2.4 Размещение элемента топологии | Объект «Элемент топологии» | Структура определяется в ходе проектирования | Объект «Топология» | Структура определяется в ходе проектирования |
| Координаты клетки | Пара индексов(i, j)  iϵ[0 .. ширина − 1]  jϵ[0 .. высота − 1] |

Продолжение таблицы 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 3 Настройки элементов топологии | 3.1Задать объем резервуара | Допустимый объем резервуара | Целое  5 .. 20 м3 | Объект «Резервуар» | Структура определяется в ходе проектирования |
| 3.2 Задать тип топлива в резервуаре | Список типов топлива | Таблица БД «Топливо» |
| 3.3 Задать тип топлива ТРК | Объект «ТРК» |
| 4 Настройки параметров моделирования | 4.1 Задать тип потока | Список типов потоков | Детерминированный,  случайный | Объект «Поток» | Структура определяется в ходе проектирования |
| 4.2 Задать закон распределения | Список законов распределения | Равномерный, нормальный,  показательный | Объект «Распределение» | Структура определяется в ходе проектирования |
| 4.3 Задать интервал для равномерного ЗР | Левая граница интервала a | Вещественное число,  a > 0 | Объект «Распределение» | Структура определяется в ходе проектирования |
| Правая граница интервала b | Вещественное  число, b > a | Объект «Распределение» |

Продолжение таблицы 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 4 Настройки параметров моделирования | 4.4 Задать параметр λ для показательного ЗР | Интенсивность | Вещественное число, λ>0 | Объект «Распределение» | Структура определяется в ходе проектирования |
| 4.5 Задать минимальное количество топлива в % | Допустимый диапазон | Целое, 0 .. 100% | Уровень топлива | Целое |
| 4.6 Задать параметры для нормального ЗР:  математическое ожидание μ  дисперсию | Математическое ожидание | Вещественное число, μ>0 | Объект «Распределение» | Структура определяется в ходе проектирования |
| Дисперсия | Вещественное число, >0 | Объект «Распределение» |
| 5 Генерирования потока автомобилей | 5.1 Сгенерировать автомобиль | Номер автомобиля | Таблица БД «Автомобиль», | Объект «Автомобиль» | Структура определяется в ходе проектирования |
| Время | Вещественное |

Продолжение таблицы 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 5 Генерирования потока автомобилей | 5.2 Получить время появления автомобиля | Параметры ЗР | Объект «Распределение» | Время | Вещественное |
| 6 Моделирования | 6.1 Переместить автомобиль | Объект «Топология» | Структура объекта определяется в ходе проектирования | Объект «Топология» | Структура объекта определяется в ходе проектирования |
| 6.2 Проверить лимит кассы | Текущий размер кассы | Вещественное | Признак приезда инкассации | Логическое |
| Лимит кассы | 700000 руб. |
| 6.3 Заправить автомобиль | Объект «Автомобиль» | Структура объекта определяется в ходе проектирования | Стоимость заправки | Вещественное |
| Объект «Резервуар» | Структура объекта определяется в ходе проектирования |
| 6.4 Проверить уровень топлива в резервуаре | Объект «Резервуар» | Структура объекта определяется в ходе проектирования | Признак вызова дозаправщика | Логический |

Продолжение таблицы 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 7 Визуализации | 7.1 Отобразить топологию | Объект «Топология» | Структура объекта определяется в ходе проектирования | Визуальное отображение топологии АЗС | – |
| 7.2 Отобразить табло | Список типов топлива | Таблица БД «Топливо» | Визуальное отображение типов топлива и их цен | – |
| 7.3 Отобразить таблицу параметров | Список параметров объекта «Модель» | Структура объекта определяется в ходе проектирования | Визуальное отображение значений параметров модели | – |
| 7.4 Отобразить графики | Список параметров объекта «Модель» | Структура объекта определяется в ходе проектирования | Визуальное отображение зависимостей параметров в процессе работы модели | – |

Продолжение таблицы 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 7 Визуализации | 7.5 Отобразить автомобиль | Объект «Автомобиль» | Структура объекта определяется в ходе проектирования | Визуальное отображение объекта «Автомобиль» в процессе работы модели | – |
| Координаты клетки | Пара индексов(i, j)  iϵ[0 .. ширина]  jϵ[0 .. высота] |
| 8 Файловая | 8.1 Сохранить топологию в файл | Объект «Топология» | Структура объекта определяется в ходе проектирования | Текстовый файл | xml файл |
| Имя файла | Строка \*.xml |
| 8.2 Загрузить топологию из файла | Имя файла | Строка \*.xml | Объект «Топология» | Структура объекта определяется в ходе проектирования |
| Код ошибки | Целое |
| 9 Работы с БД | 9.1 Добавить автомобиль | Марка | Строка | Запись | Таблица БД «Автомобиль» |
| Объем бака | Вещественное |
| Тип топлива | Строка |

Продолжение таблицы 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 9 Работы с БД | 9.2 Удалить автомобиль | ID автомобиля | Целое | Признак успеха удаления | Таблица БД «Автомобиль» |
| 9.3 Изменить параметры автомобиля | ID автомобиля | Целое | Запись |
| Марка | Строка |
| Объем бака | Вещественное |
| Тип топлива | Строка |
| 9.4 Получить объект «Автомобиль» | ID автомобиля | Целое | Объект «Автомобиль» | Структура объекта определяется в ходе проектирования |
| 9.5 Добавить тип топлива | Тип топлива | Строка | Запись | Таблица БД «Тип топлива» |
| Цена | Вещественное |
| 9.6 Удалить тип топлива | ID типа топлива | Строка | Признак успеха удаления |
| 9.7 Изменить тип топлива | Тип топлива | Строка | Запись |
| Цена | Вещественное |

назначением. Этот стандарт регламентирует базовую модель качества и показатели, главным среди них - надежность. Стандарт ISO/IEC 12207 определил не только основные процессы ЖЦ разработки ПС, но и организационные и дополнительные процессы, которые регламентируют инженерию, планирования и управления качеством ПС.

Согласно стандарту ISO/IEC на этапах ЖЦ должен проводиться контроль качества ПО [21]:

* проверка соответствия требований проектируемому продукту и критериев их достижения;
* верификация и аттестация (валидация) промежуточных результатов ПО на этапах ЖЦ и измерение степени удовлетворения достигаемых отдельных показателей; тестирование готовой ПС, сбор данных об отказах, дефектах и других ошибках, обнаруженных в системе;
* подбор моделей надежности для оценивания надежности по полученным результатам тестирования (дефекты, отказы и др.);
* оценка показателей качества, заданных в требованиях на разработку ПС.

Данная ПС должна отвечать следующим требованиям []:

1. *Надежность -* это способность ПС выполнять возложенные на нее функции при поступлении требований на их выполнение в течение заданного интервала времени. Надежность относится к*динамическим требованиям*, предъявляемым к системе, и включает в себя такие элементы:

* *Отказоустойчивость* – возможность восстановления программы и данных в случае сбоев в работе;
* *Безопасность* – сбои в работе программы не должны приводить к опасным последствиям (авариям);
* *Защищенность* от случайных или преднамеренных внешних воздействий («защита от дурака», вирусов, спама);

1. *Удобство применения* (usability) – ПО должно быть легким в использовании, причем именно тем типом пользователей, на которых рассчитано приложение. Это включает в себя интерфейс пользователя и адекватную документацию. Причем, пользовательский интерфейс должен быть не интуитивно, а профессионально понятным пользователю.
   * 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 | 2 | 3 |
| 1 Справочная | 1.1 Невозможно открыть файл справки | Выдача сообщения «Файл справки поврежден» |
| 1.2 Невозможно найти файл справки | Выдача сообщения «Отсутствует файл справки» |
| 2 Конструирования топологии | 2.1 Невозможно найти файл шаблона | Выдача сообщения «Файл шаблона не существует» |
| 2.2 Невозможно открыть файл шаблона | Выдача сообщения «Файл шаблона поврежден» |
| 3 Настройки элементов топологии | 3.1 Невозможно найти файл БД «Топливо» | Выдача сообщения «База данных не существует» |

Продолжение таблицы 2

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 3 Настройки элементов топологии | 3.2 Невозможно открыть файл БД «Топливо» | Выдача сообщения «База данных повреждена» |
| 4 Файловая | 4.1 Попытка открытия файла с несобственным форматом | Выдача сообщения «Файл поврежден или недопустимого формата» |
| 4.2 Файл с заданным именем не существует | Выдача сообщения «Файла с таким именем не существует» |
| 5 БД | 5.1 БД повреждена | Выдача сообщения «База данных повреждена» |

* 1. Разработка прототипа интерфейса пользователя системы

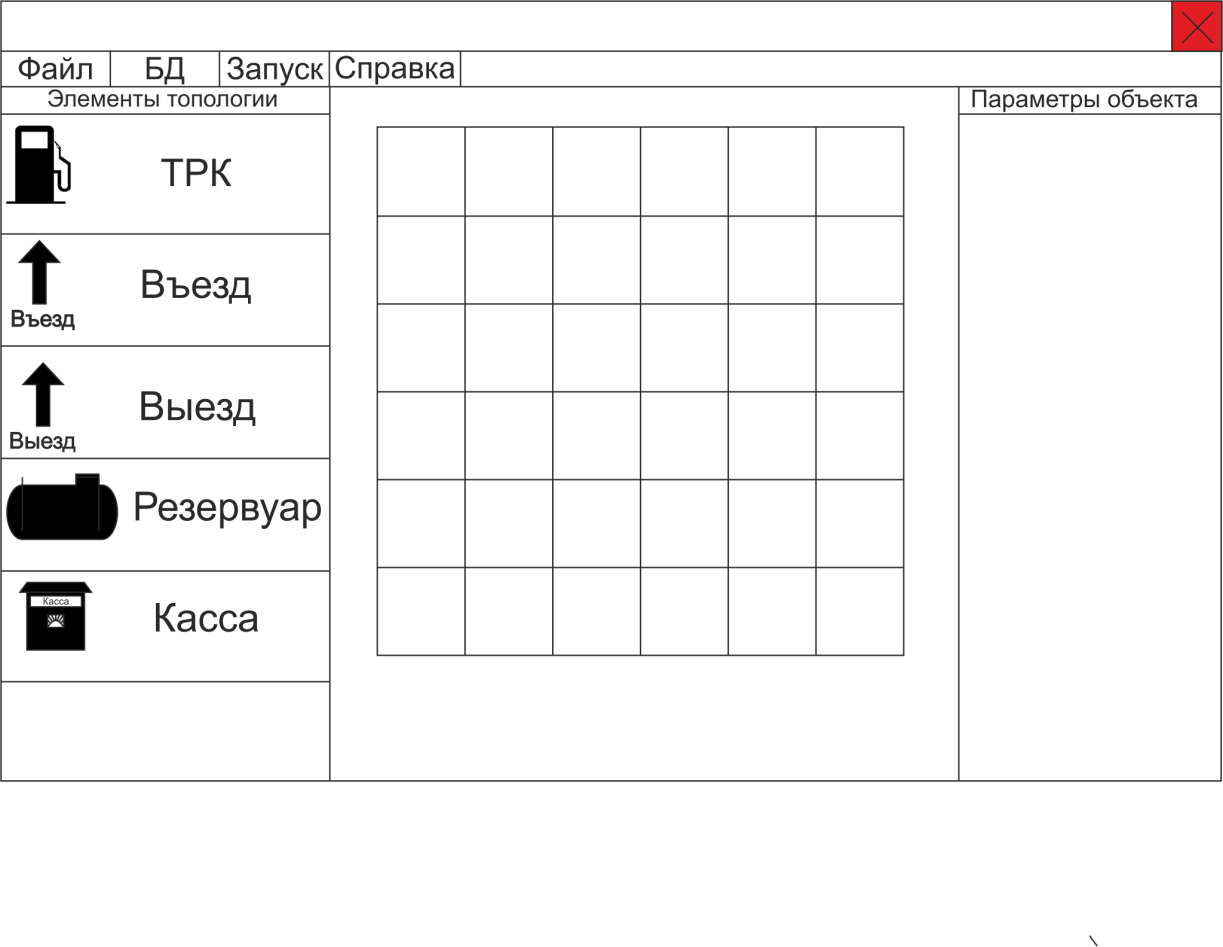
Процесс создания интерфейса начинается с определения целей проекта, а также внутренних и внешних обстоятельств, которые следует принять во внимание. Для того, чтобы правильно расставить приоритеты, необходимо учитывать [20]:

* опыт работы пользователей с компьютером, типовые ситуации использования;
* какая информация необходима и когда, какие результаты должны быть получены;
* технологию разработки и платформа, на которой будут работать пользователи.

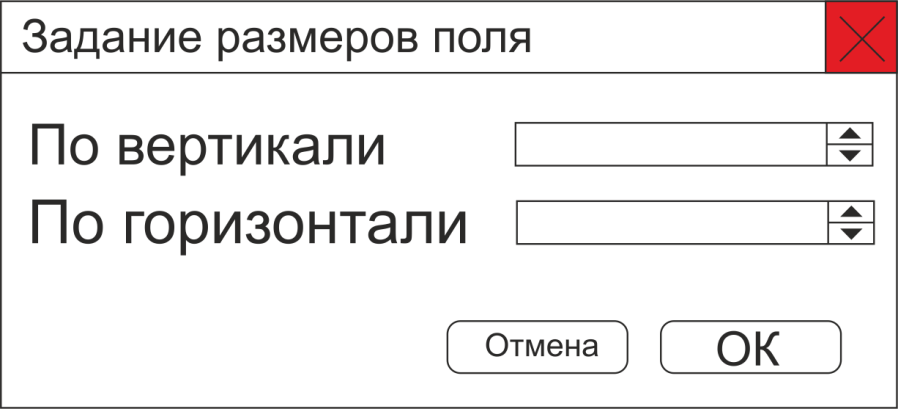
Начальной фазой разработки интерфейса является разработка прототипов экранных форм.

На рисунках 18-25 изображены прототипы экранных форм программы моделирования работы АЗС.

На рисунке 18 приведен прототип экранной формы, с помощью которой пользователь сможет конструировать топологию. Слева располагается список элементов топологии. Справа находятся настройки элементов топологии.

  
Рисунок 18 – Прототип основной экранной формы приложения

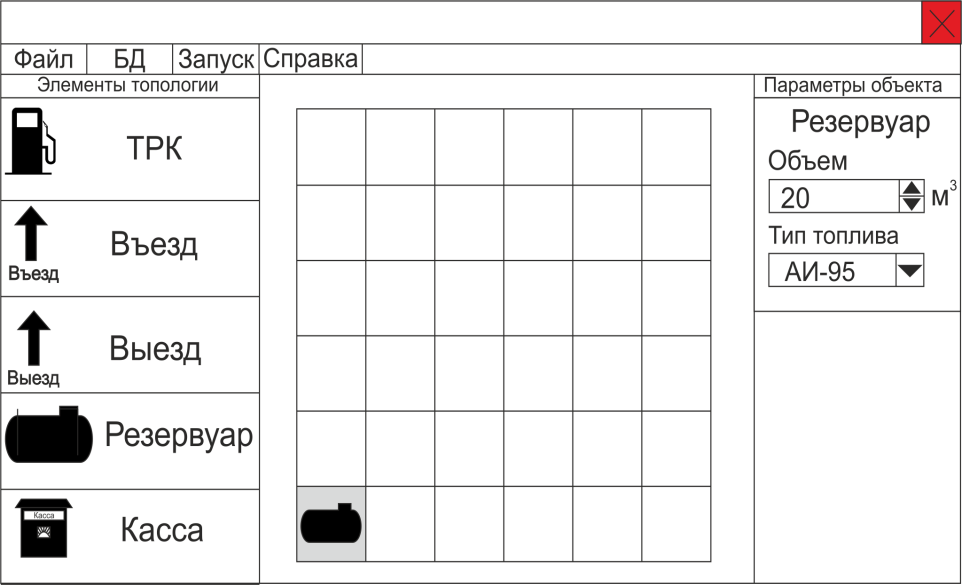
На рисунке 19 изображен прототип экранной формы задания размеров поля. В этом окне пользователь должен задать размеры поля по горизонтали и по вертикали.

  
Рисунок 19 – Прототип экранной формы задания размеров поля

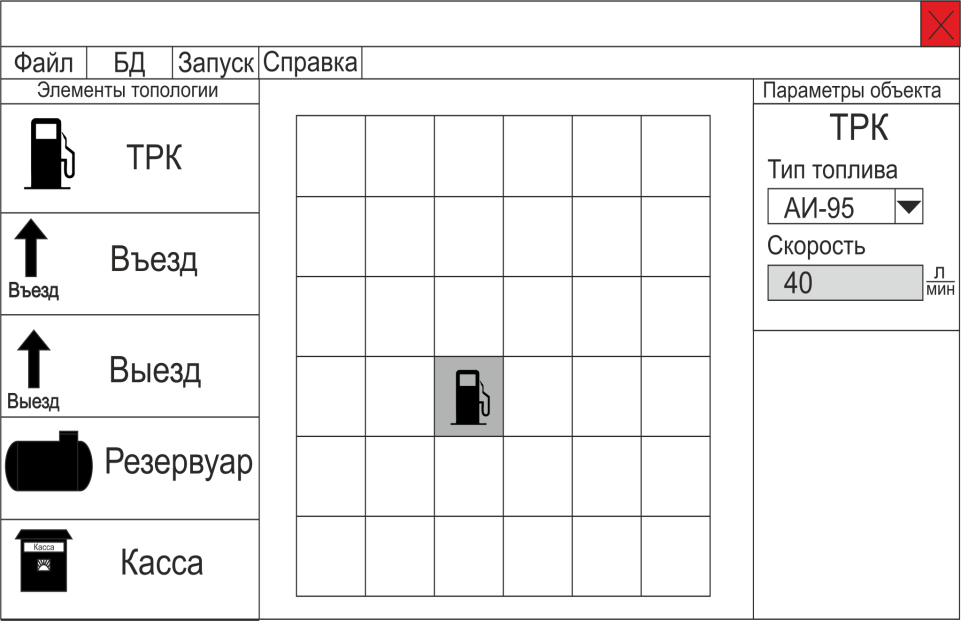
На рисунке 20 представлен прототип экранной формы работы с таблицами базы данных. С помощью этой формы пользователь может добавлять, изменять или удалять записи из БД.

  
Рисунок 20 – Прототип экранной формы управления таблицами БД

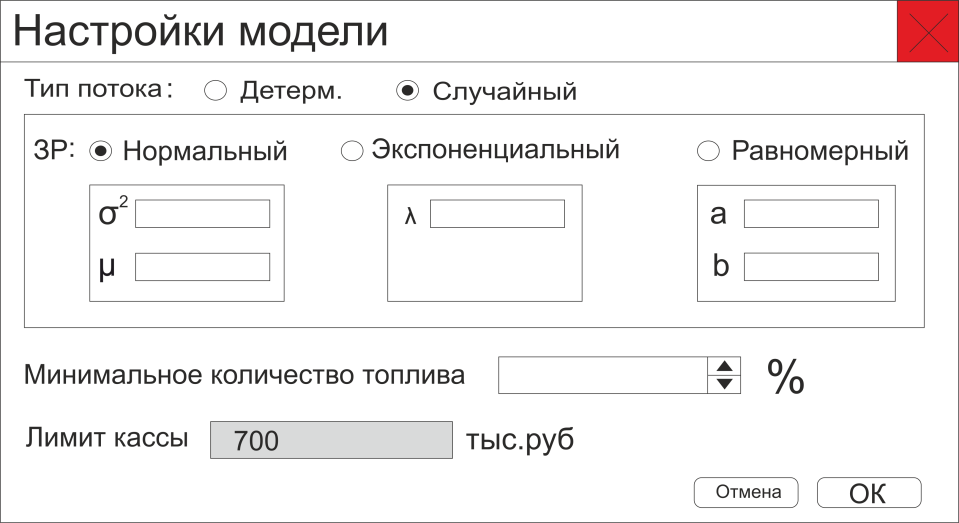
На рисунке 21 изображен прототип экранной формы, иллюстрирующий настройку параметров резервуара. Пользователь должен указать объем резервуара и тип топлива, который хранится в резервуаре.

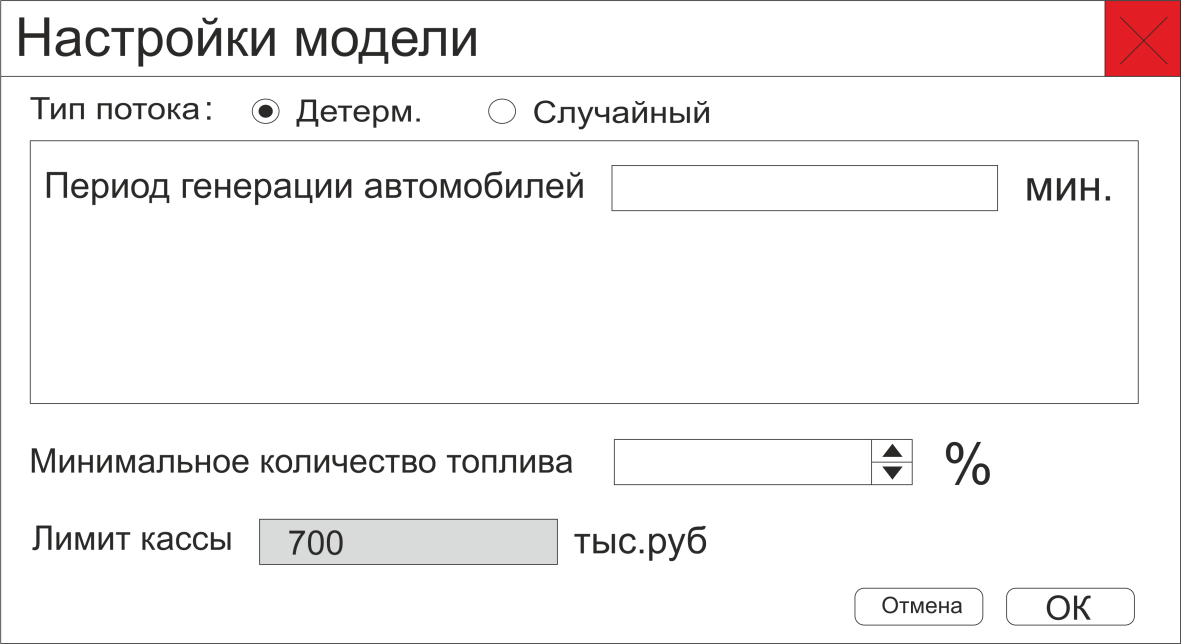
  
Рисунок 21 – Экранная форма настройки резервуара

На рисунке 22 изображен прототип экранной формы, иллюстрирующий настройку параметров ТРК. Пользователь должен указать тип топлива, которым заправляет данная колонка.

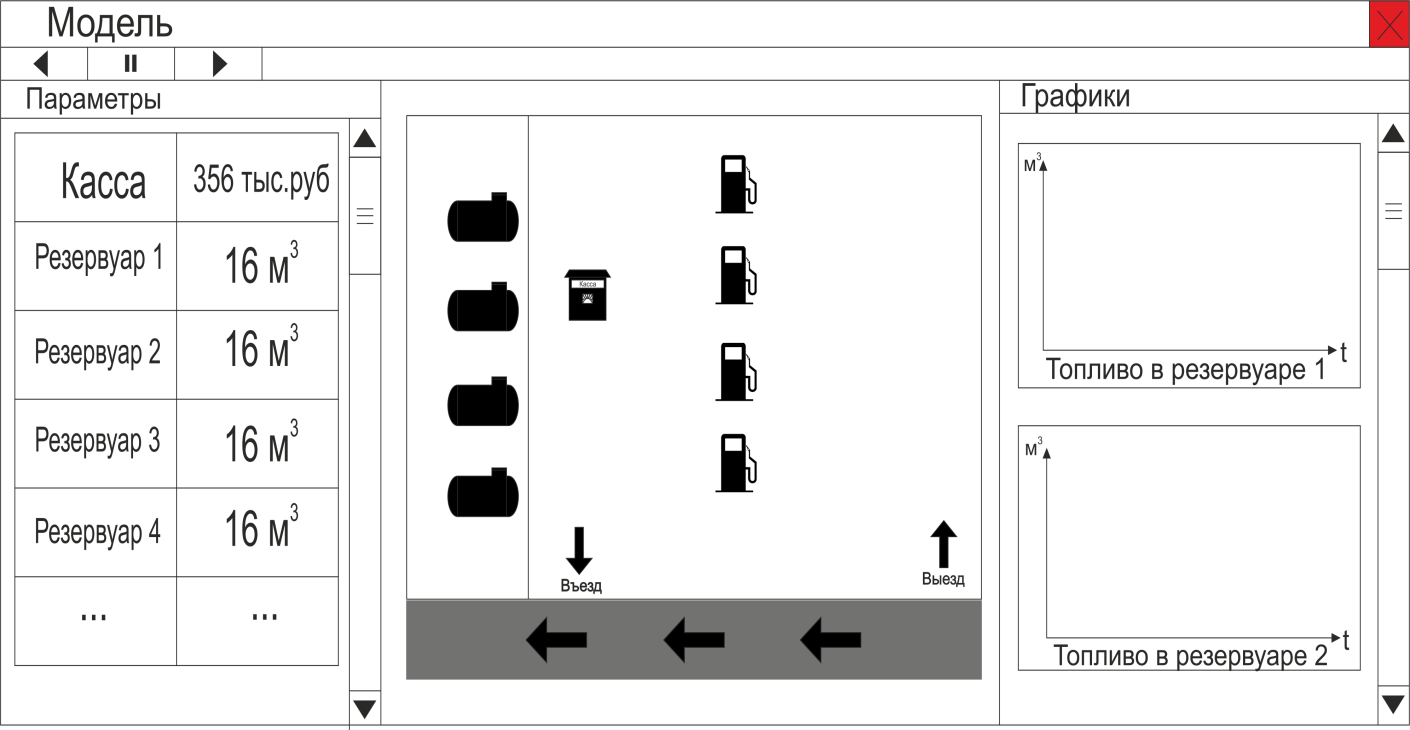
  
Рисунок 22 –Экранная форма настройки ТРК

На рисунках 23 и 24 показаны прототипы экранных форм, позволяющих пользователю выбрать тип потока, указать параметры потока или период генерации автомобилей, а также задать количество топлива (%) в резервуаре, при котором будет вызываться дозаправщик.

  
Рисунок 23 – Экранная форма настройки модели для случайного времени появления автомобилей

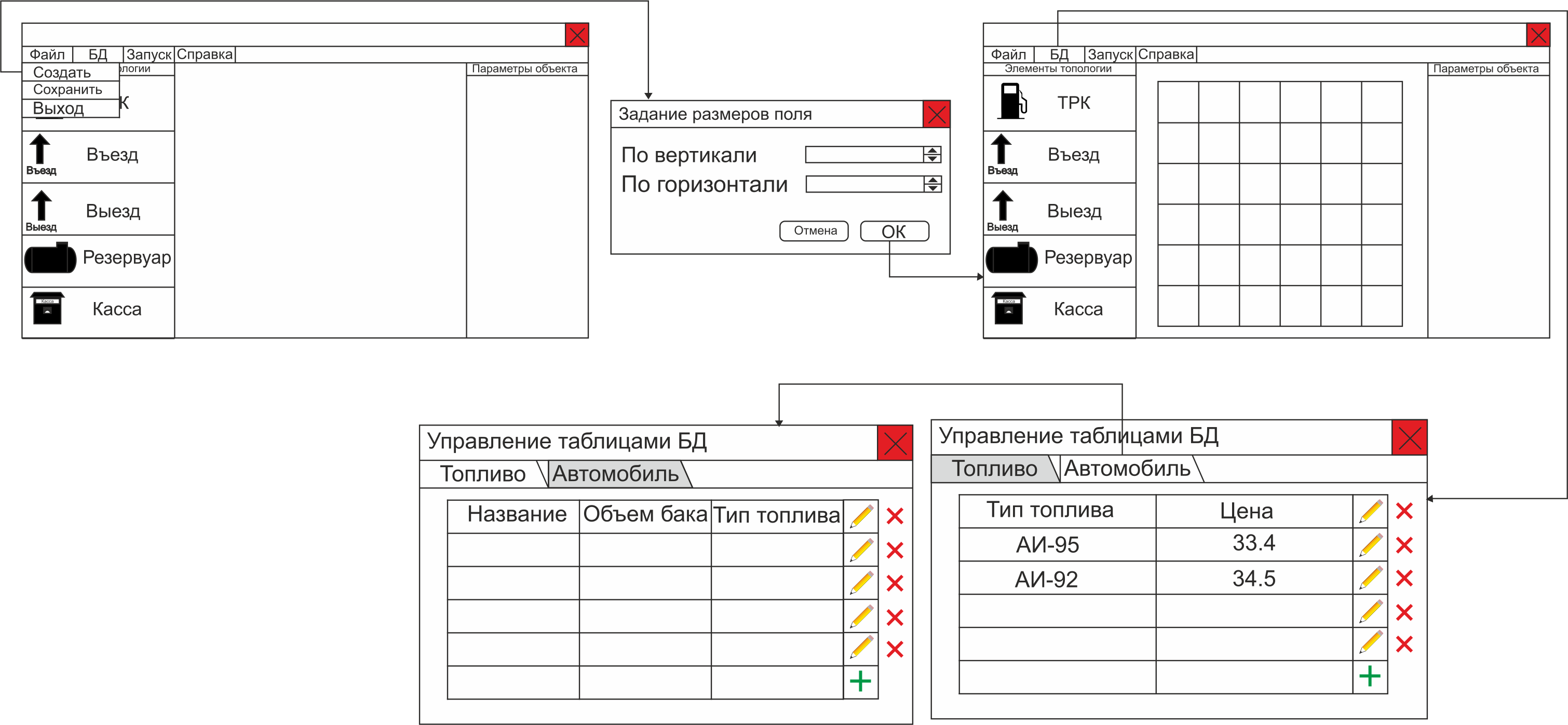
  
Рисунок 24 – Экранная форма настройки модели для детерминированного времени появления автомобилей

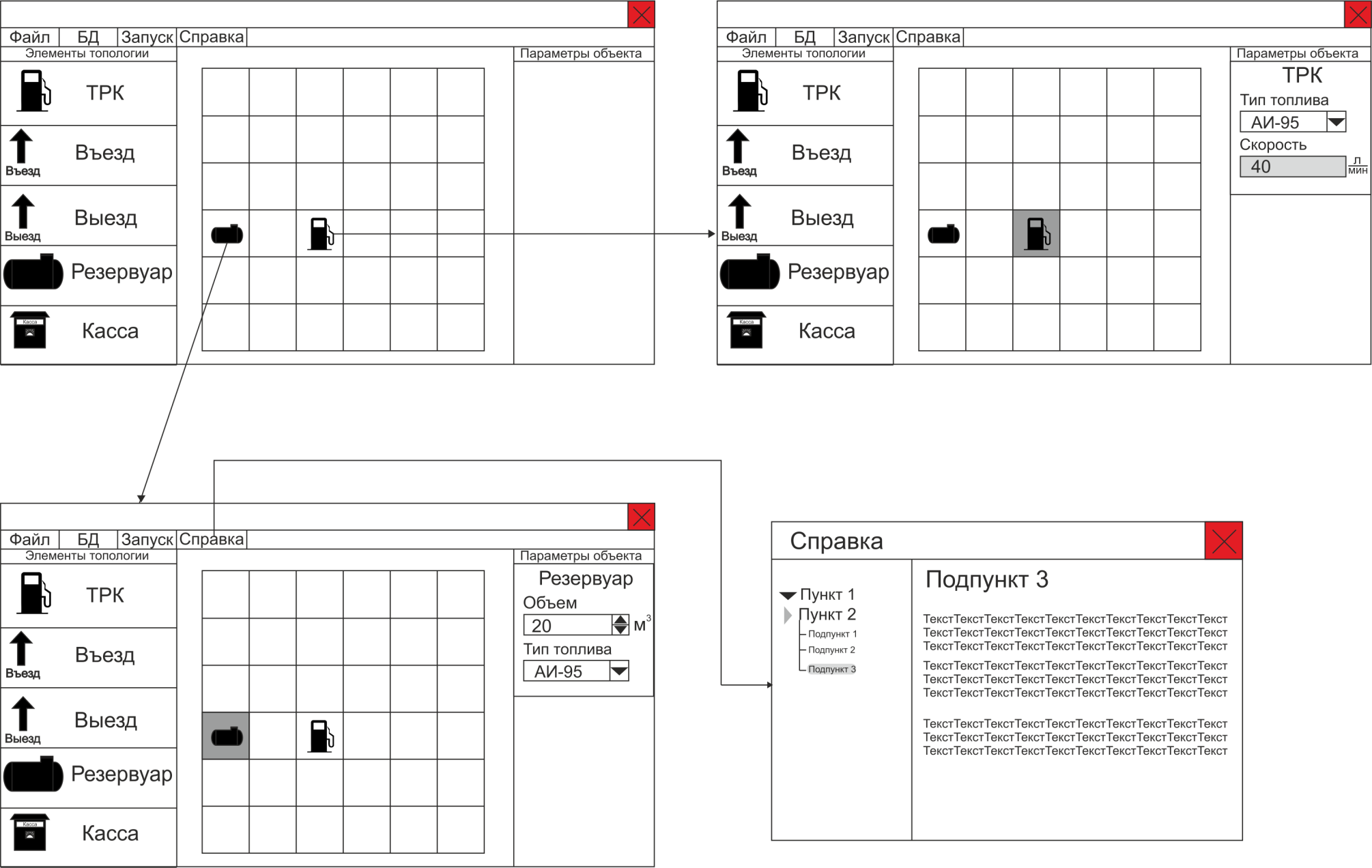
На рисунке 25 изображен прототип экранной формы моделирования. На этой форме будет отображаться процесс работы модели АЗС. В левом фрейме находится таблица с различными параметрами системы (размер кассы, объем резервуаров и т.д.). В правом фрейме находятся графики, отображающее различные зависимости (количество топлива в резервуаре от времени, размер кассы от времени и т.д.). Во время моделирования пользователь может приостановить процесс моделирования, ускорить или замедлить модельное время (ускорить или замедлить работу модели).

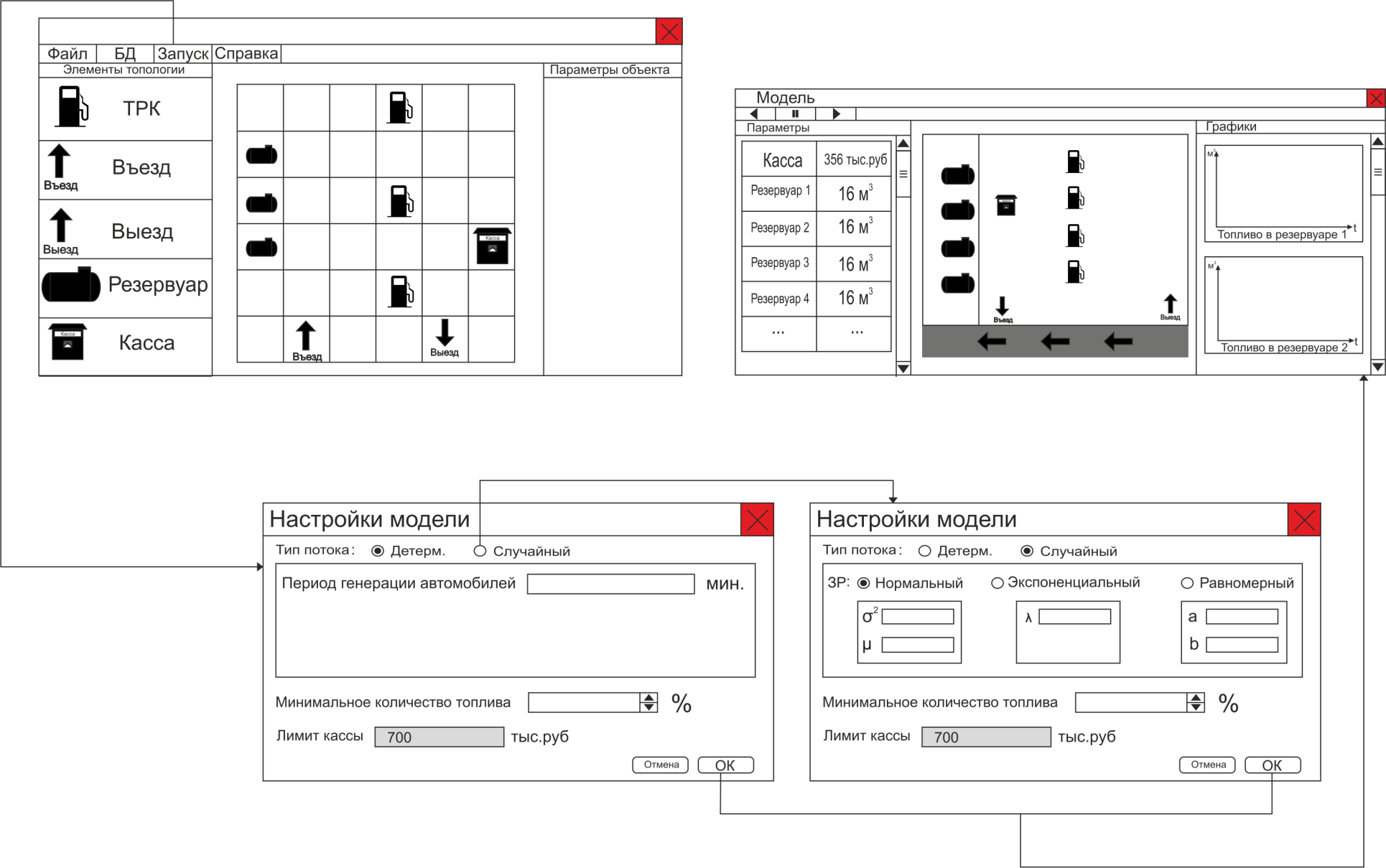
  
Рисунок 25 – Экранная форма процесса моделирования

Интерфейс пользователя является одним из важнейших элементов программы, это та часть программы, которая находится у всех на виду. Недочеты в пользовательском интерфейсе могут серьезно испортить впечатление даже о самых многофункциональных программах.

Навигационная модель показывает, как необходимо распределять функции или задачи между окнами создаваемой программы, а также, как пользователи смогут перемещаться как между различными задачами, так и внутри отдельной задачи. На рисунке 26 приведена навигационная модель разрабатываемого приложения.

  
Рисунок 26‒ Навигационная модель приложения. Режим конструирования (начало)

  
Рисунок 26‒ Навигационная модель приложения. Режим конструирования (продолжение)

  
Рисунок 26‒ Навигационная модель приложения. Режим моделирования (продолжение)

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

Унифицированный язык моделирования (Unified Modeling Language – UML) – это стандартный инструмент для разработки «чертежей» программного обеспечения. Его можно использовать для визуализации, спецификации, конструирования и документирования артефактов программных систем. Язык UML предоставляет стандартный способ написания проектной документации на системы, включая концептуальные аспекты, такие как бизнес-процессы и функции системы, а также конкретные аспекты, такие как выражения языков программирования, схемы баз данных и повторно используемые компоненты ПО.

В нотации языка UML определены следующие виды канонических диаграмм [20]:

1. вариантов использования (use case diagram);
2. классов (class diagram);
3. кооперации (collaboration diagram);
4. последовательности (sequence diagram);
5. состояний (statechart diagram);
6. деятельности (activity diagram);
7. компонентов (component diagram);
8. развертывания (deployment diagram).
   * 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 24.

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. [Автоматизация](http://www.intuit.ru/goto/course/umlbasics/) АЗК [Электронный ресурс] – URL: http://www.technoshop.ru/papers/2003/3/918/918.html (дата обращения: 27.09.2016).
2. Зеленко, Л.С. Методические указания к лабораторному практикуму по дисциплине «Программная инженерия» [Текст]/Л.С.Зеленко.– СГАУ, 2012. – 42 с.
3. Вендров, А.М. CASE-технологии. Современные методы и средств проектирования информационных систем. [Текст] / А. М. Вендров – М.: Финансы и статистика, 1999. – 256 с. : ил.
4. Леоненков, А.В. [Нотация и семантика языка UML](http://www.intuit.ru/goto/course/umlbasics/) [Электронный ресурс] /А.В. Леоненков. – Интернет-университет информационных технологий. – URL: http://www.intuit.ru/department/pl/umlbasics (дата обращения: 27.09.2016).
5. Определение автомобильной заправочной станции [Электронный ресурс]–URL: http://dic.academic.ru/dic.nsf/bse/61276/Автозаправочная (дата обращения: 20.09.2016).
6. Определение автомобильной заправочной станции [Электронный ресурс]–URL: https://ru.wikipedia.org/wiki/Автомобильная\_заправочная\_станция (дата обращения: 20.09.2016).

1. [Классификация](http://www.intuit.ru/goto/course/umlbasics/) АЗС [Электронный ресурс] – URL: http://ros-pipe.ru/clauses/klassifikatsiya\_azs/ (дата обращения: 25.09.2016).
2. СП 156.13130.2014 Свод правил станции автомобильные заправочные требования пожарной безопасности [Электронный ресурс] – URL: http://www.mchs.gov.ru/document/3744769 (дата обращения: 23.09.2016).
3. Топливораздаточная колонка [Электронный ресурс] – URL: http://dic.academic.ru/dic.nsf/bse (дата обращения: 27.09.2016).
4. Вместимость резервуаров АЗС [Электронный ресурс] – URL: http://proofoil.ru/Petrochemical/Petrochemical7.html (дата обращения: 3.10.2016).
5. ГОСТ 17032-71 Резервуары стальные горизонтальные для нефтепродуктов. Типы и основные размеры [Электронный ресурс] – URL: http://neftegaz.ru/standarts/view/971 (дата обращения: 3.10.2016).
6. Принцип работы АЗС [Электронный ресурс] – URL: http://altsi.ru/publ/effective-work-azs.htm (дата обращения: 27.09.2016).
7. Транспортный поток – Википедия [Электронный ресурс] – URL: https://ru.wikipedia.org/wiki/Транспортный\_поток
8. Основные понятия потоков вызова [Электронный ресурс] – URL: http://sernam.ru/t\_11.php
9. Моделирование нормально распределенных случайных величин  
    [Электронный ресурс] – URL: http://stratum.ac.ru/education/textbooks/  
   modelir/lection25.html
10. Формирование модели предметной области [Электронный ресурс]. – URL: <http://5fan.ru/wievjob.php?id=45484> (дата обращения: 27.09.2016).
11. Методология системного анализа и системного моделирования [Электронный ресурс]. – URL: http://www.znannya.org/  
    ?view=summary:Methodology\_analysis\_system\_design (дата обращения: 08.10.2016).
12. Этап концептуализации [Электронный ресурс]. – URL: http://wiki.mvtom.ru/index.php/  
    Этап\_концептуализации (дата обращения: 11.10.2016).
13. Внешнее описание программного средства [Электронный ресурс]. – URL: http://studend.ru/gotovye-raboty/lektsiya-po-teme-vneshnee-opisanie-programmnogo-srdestva-l012.html (дата обращения: 11.10.2016).
14. Зеленко, Л.С. Методические указания к лабораторному практикуму по дисциплине «Программная инженерия» [Текст]/ Л.С. Зеленко. – Самара: СГАУ, 2012. – 67 с
15. Лаврищева, Е.М., Петрухин, В.А. Модели качества и надежности в программной инженерии [Электронный ресурс] / Лаврищева, Е.М., Петрухин, В.А. – Интернет-университет информационных технологий.– URL: http://www.intuit.ru/studies/courses/ (дата обращения: 25.10.2016).
16. Качество ПО (Software Quality) [Электронный ресурс]. –   
    URL: http://fit.distance.ssau.ru/moodle/mod/page/view.php?id=198 (дата обращения: 25.10.2016).
17. 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)