**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ  
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО  
 ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ   
УНИВЕРСИТЕТ имени акад. С.П. КОРОЛЕВА»**

**(национальный исследовательский университет)**

**Кафедра программных систем**

**ОТЧЕТ  
пол лабораторному практикуму по дисциплине  
"Технологии программирования" на тему  
"Автоматизированная система поиска оптимального пути по заданному критерию"**

Выполнили:   
студенты гр.643   
*Чубенко С.А.  
Ермолаев А.Д.  
Назаров Ю.П.*Руководитель проекта:  
доцент каф. ПС *Зеленко Л.С.*Дата сдачи:  
  
Оценка:

Самара 2011 г.

**Кафедра программных систем**

**«УТВЕРЖДАЮ»**Руководитель проекта  
  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Зеленко Л.С./  
  
«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_ 2011 г.

**ЗАДАНИЕ**

на разработку ПС в рамках лабораторного практикума по дисциплине   
«Технологии программирования»

студентам гр. 643  
Чубенко С.А.,   
Ермолаеву А.Д.,  
Назарову Ю.П.

**Тема проекта:**

**«Автоматизированная система для расчета оптимальных маршрутов движения автотранспорта в городских условиях»**

**1 Содержание задания**

1.1 Произвести анализ предметной области по методологии объектной декомпозиции и разработать логический проект системы по технологии RAD.

1.2 Реализовать информационное и программное обеспечения системы в соответствии с проектом, подготовить контрольные примеры и провести автономное и комплексное тестирование и отладку.

1.3 Разработать презентацию на систему.

1.4 Оформить документацию проекта.

**2 Исходные данные**

**2.1 Характеристика объекта автоматизации:**

1. объект автоматизации – участок городской транспортной сети (ГТС);
2. виды автоматизируемой деятельности:
   1. процесс составления (редактирования) участка ГТС;
   2. процесс поиска оптимального пути по заданному критерию;
   3. процесс работы с базой данных (БД);
3. объекты участка ГТС:
   1. узел (перекресток);
   2. дуга (дорога);
   3. светофор;
   4. сотрудник ДПС;
   5. знак «ограничение скорости»;
   6. знак «въезд запрещен»;
   7. знак «одностороннее движение»;
4. характеристики дуги:
   1. длина дуги (положительное число);
   2. коэффициент покрытия (максимальная возможная скорость на данной дуге);
   3. знаки дорожного движения;
   4. наличие сотрудника ДПС;
   5. название;
5. характеристики узла:
   1. наличие светофора;
   2. наличие сотрудника ДПС;
   3. количество дуг, выходящих из данного узла (натуральное число, не более 4);
6. характеристики светофора:
   1. светофоры одинаковые;
   2. количество световых фаз – 2 (красный, зеленый);
   3. начальная фаза (определяется пользователем);
   4. длительность фазы (положительное число);
7. объекты, используемые для нахождения оптимального пути:
   1. водитель;
   2. автомобиль;
8. характеристики водителя:
   1. ФИО;
   2. склонность к нарушениям:

* законопослушный;
* нарушитель;
* может нарушить правила в случае отсутствия сотрудника ДПС;

1. характеристики автомобиля:
   1. максимальная скорость (натуральное число);
   2. марка автомобиля;
   3. тип топлива;
   4. стоимость топлива;
2. критерии нахождения оптимального пути:
   1. расстояние (положительное число);
   2. время (положительное число);
   3. экономичность (положительное число).

**2.2 Требования к информационному обеспечению:**

Информационное обеспечение разрабатывается на основе следующих документов:

1. ГОСТ Р 52290-2004. Технические средства организации дорожного движения. Знаки дорожные. Общие технические требования [Текст] – Введ. 2004-15-12. – М., 2006. - 125 с.;
2. Правила дорожного движения Российской Федерации [Текст]: иллюстрированное издание. – М.: ООО Мир Автокниг, 2011. - 64 с.;
3. Данные и характеристика ГТС хранятся в базе данных;
4. в системе должны вестись следующие справочники:

* тип топлива;
* название улиц;

1. предусмотреть целостность таблиц БД.

**2.3 Требования к техническому обеспечению:**

1. тип ЭВМ - IBM PC совместимый;
2. монитор с разрешающей способностью не ниже 800 х 600;
3. манипулятор – мышь;
4. устройство ввода – клавиатура;
5. конфигурация комплекса определяется в процессе выполнения проекта.

**2.4 Требования к программному обеспечению:**

1. тип операционной системы - Windows XP и выше, Linux;
2. язык программирования – C++;
3. среда программирования – Qt Creator 2.3.0;
4. СУБД – MySQL 5.1 и выше.

**2.5 Общие требования к проектируемой системе.**

2.5.1 Функции, реализуемые системой:

1. настройка параметров системы;
2. составление участка ГТС;
3. сохранение участка ГТС в БД с заданной структурой;
4. загрузка участка ГТС в БД с заданной структурой;
5. работа с редактором участка ГТС:
6. добавление объекта;
7. удаление объекта;
8. настройка характеристик объектов;
9. работа с базой данных:
10. добавление данных объектов;
11. удаление данных объектов;
12. редактирование данных объектов;
13. ведение справочников БД;
14. выдача справочной информации о системе.

2.5.2 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 должны соответствовать ГОСТ 12.1.005, 12.01.007.

**3 Календарный план выполнения работ**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Содержание работы по этапам | | Объем этапа в % к общему объему проекта | Срок  окончания | Фактическое выполнение |
| 1 | Оформление технического задания и его утверждение | | 5 | 21.09.11 |  |
| 2 | Разработка информационно-логического проекта системы и его предъявление руководителю | | 40 | 05.10.11 |  |
| 2.1 | Разработка структурной системы | 10 | 21.09.11 |  |
| 2.2 | Разработка функциональной спецификации системы | 10 | 21.09.11 |  |
| 2.3 | Разработка структур данных и классов | 10 | 05.10.11 |  |
| 2.4 | Разработка алгоритмов обработки данных и функционирования системы | 10 | 05.10.11 |  |
| 3 | Реализация проекта, разработка контрольных примеров. Предъявление реализации руководителю | | 45 | 30.11.11 |  |
| 4 | Корректировка проекта и оформление документации проекта. Защита проекта с представлением презентации. | | 10 | 14.12.11 |  |

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Чубенко С.А. / « » \_\_\_\_\_\_\_\_\_\_\_ 2011 г.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Ермолаев А.Д. / « » \_\_\_\_\_\_\_\_\_\_\_ 2011 г.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Назаров Ю.П. / « » \_\_\_\_\_\_\_\_\_\_\_ 2011 г.

РЕФЕРАТ

Пояснительная записка \_ с, \_ рисунка, \_ таблиц, \_ источников, \_ приложения.

ТРАНСПОРТНАЯ СЕТЬ, ПЕРЕКРЁСТОК, ДОРОГА, КРИТЕРИЙ ОПТИМАЛЬНОСТИ, ПУТЬ, АЛГОРИТМ ПОИСКА, ГРАФ.

Объектом исследования является участок городской транспортной сети

Цель работы – проектирование автоматизированной системы для расчета оптимальных маршрутов движения автотранспорта в городских условиях.

В процессе работы использована теория графов.

В результате работы разработан алгоритм, составлена и отлажена программа конструирования участка транспортной сети. Сохранение производится сразу в базу данных, при любом действии с участком ГТС.

Реализованы функции добавления узла, дуги, сотрудника ДПС, работа со словарями, удаление составляющих ГТС.

Программа реализована на языке C++ в среде Qt Creator 2.3.0 и функционирует под управлением операционных систем Windows.**СОДЕРЖАНИЕ**

[Введение 9](#_Toc307963543)

[1 Систематическая часть 10](#_Toc307963544)

[1.1 Описание и анализ предметной области 10](#_Toc307963545)

[1.1.1 Определение городской транспортной сети 10](#_Toc307963546)

[1.1.2 Элементы городской транспортной сети 10](#_Toc307963547)

[1.1.3 Правила дорожного движения 17](#_Toc307963548)

[1.1.4 Формализация задачи. Теория графов 18](#_Toc307963549)

[1.1.5 Алгоритм Дейкстры 19](#_Toc307963550)

[1.1.6 Описание систем – аналогов 20](#_Toc307963551)

[1.2 Постановка задачи 22](#_Toc307963552)

[1.3 Построение структурной схемы программной системы 24](#_Toc307963553)

[1.4 Функциональная спецификация 25](#_Toc307963554)

[1.4.1 Перечень функций, выполняемых системой 26](#_Toc307963555)

[1.4.2 Перечень исключительных ситуаций и информационных сообщений](#_Toc307963556) 30

# ВВЕДЕНИЕ

С увеличением числа автомобилей на дорогах, цен на бензин, увеличением плотности движения, становится актуальной задача навигации и нахождения оптимального пути в городских условиях.

В настоящее время существуют такие программы, такие как 2Gis, Navitel, Garmin, Яндекс-Карты, Google-Earth, которые предоставляют пользователям спутниковые фотографии городов, схемы городской транспортной сети, актуальные данные о пробках, схемы метро и другие возможности. Для удобной работы с картами сервис предлагает различные инструменты – измерение расстояний, печать карты, прокладка маршрута, хранение точек [1].

Для решения задач оптимизации в геоинформационных системах (ГИС) находит активное применение теория графов. Существующие или вновь проектируемые дома, сооружения, кварталы и т. п. рассматриваются как вершины, а соединяющие их дороги, инженерные сети, линии электропередачи и т. п. — как рёбра. Применение различных вычислений, производимых на таком графе, позволяет, например, найти кратчайший объездной путь или ближайший продуктовый магазин, спланировать оптимальный маршрут.

В рамках лабораторного практикума необходимо разработать систему нахождения пути по заданным критериям времени, расстояния, денежных затрат, которая будет использовать материальные основы теории графов.

Проектирование системы производится по технологии быстрой разработки приложений (RAD), которая поддерживается методологией структурного проектирования и включает элементы объектно-ориентированного проектирования и анализа предметной области.

Для решения задач оптимизации в геоинформационных системах (ГИС) находит активное применение теория графов. Существующие или вновь проектируемые дома, сооружения, кварталы и т. п. рассматриваются как вершины, а соединяющие их дороги, инженерные сети, линии электропередачи и т. п. — как рёбра. Применение различных вычислений, производимых на таком графе, позволяет, например, найти кратчайший объездной путь или ближайший продуктовый магазин, спланировать оптимальный маршрут.

## СИСТЕМОТЕХНИЧЕСКАЯ ЧАСТЬ

### 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 [2].

### Определение городской транспортной сети

*Транспортной сетью (ТС)* называется совокупность транспортных связей, в системе которых осуществляются городские пассажирские и грузовые перевозки [3]. *Городскую транспортную сеть (ГТС)* образует совокупность улиц и транспортных проездов, обслуживаемых различными видами городского транспорта, а также подземные, наземные и надземные транспортные линии, связанные с уличной сетью лишь частично или не связанные с ней вообще (городские железные дороги, эстакадные автомагистрали, метрополитен, монорельсовые дроги).

Транспортная сеть неразрывно связана с обслуживаемым городом, его населением, застройкой, рельефом местности, климатическими условиями. В общем, виде ТС – это некоторая неявная функция планировочных, социально-экономических, демографических, климатических и других характеристик города. В общем виде данная функция может быть предоставлена имитационной моделью ТС. Особенности формирования ТС многих городов связаны с их историческим развитием. ТС городов современной застройки проектируется с учетом комплексного решения градостроительных и транспортных проблем.

### Элементы городской транспортной сети

В состав ГТС входят множество элементов, притом они могут быть одушевлёнными и неодушевлёнными.

**Дороги**

Основным элементом ГТС являются дороги, которые служат в качестве транспортных связей. *Дорога* – путь сообщения для передвижения людей и транспорта, составная часть дорожной инфраструктуры. Согласно Федеральному закону от 10.12.1995 № 196-ФЗ «О безопасности дорожного движения», дорога — обустроенная или приспособленная и используемая для движения транспортных средств полоса земли либо поверхность искусственного сооружения. Дорога включает в себя одну или несколько проезжих частей, а также трамвайные пути, тротуары, обочины и разделительные полосы при их наличии [4].

По направлению движения различают дороги с односторонним движением и дороги с двусторонним движением. Также различаются по ограничению скоростного режима в зависимости от знаков и местонахождения (город, загород, автомагистраль). Существует множество типов покрытия дорог, такие как асфальт, грунтовка и т.д. Пример дороги приведён на рисунке 1.



Рисунок 1 – Пример загородной дороги

**Перекрёстки**

*Перекрёсток* – пересечение дорог или полос движения на одном уровне. В соответствии с Правилами дорожного движения Российской Федерации перекрёсток определяется как «место пересечения, примыкания или разветвления дорог на одном уровне, ограниченное воображаемыми линиями, соединяющими соответственно противоположные, наиболее удаленные от центра перекрёстка начала закруглений проезжих частей». Встречаются равнозначные и неравнозначные перекрёстки, с круговым движением и с примыкающей дорогой, управляемые (регулируемые) и неуправляемые (нерегулируемые) [5]. Виды перекрёстков приведены на рисунке 2. На рисунке 3 приведён пример кругового перекрёстка.



Рисунок 2 – Виды перекрёстков



Рисунок 3 – Пример перекрёстка

**Транспортные средства**

*Транспорт* – совокупность средств, предназначенных для перемещения людей, грузов из одного места в другое. Транспорт делится на три категории: транспорт общего пользования, транспорт специального пользования и личный или индивидуальный транспорт. Транспорт общего пользования обслуживает торговлю (перевозит товары) и население (пассажирские перевозки). Транспорт специального пользования – внутрипроизводственный и внутриведомственный транспорт. Наконец, личный транспорт – это легковые автомобили, велосипеды, яхты, частные самолёты. Персональный автоматический транспорт образует новую категорию, так как соединяет в себе черты городского общественного транспорта и личного автотранспорта.

Автомобильный транспорт сейчас самый распространённый вид транспорта. Автомобильный транспорт моложе железнодорожного и водного, первые автомобили появились в самом конце XIX века. После Второй мировой войны автомобильный транспорт начал составлять конкуренцию железной дороге. Преимущества автомобильного транспорта - маневренность, гибкость, скорость. Грузовые автомобили перевозят ныне практически все виды грузов, но даже на больших расстояниях (до 5 и более тыс. км) автопоезда (грузовик-тягач и прицеп или полуприцеп) успешно конкурируют с железной дорогой при перевозке ценных грузов, для которых критична скорость доставки, например, скоропортящихся продуктов [6].

Абсолютное большинство ныне существующих автомобилей – автомобили индивидуального пользования (легковые). Их используют, как правило, для поездок на расстояния до двухсот километров. На рисунке 4 изображены средства передвижения, различающиеся по типу, массе и назначению.

****

Рисунок 4 – Средства передвижения

**Дорожные знаки**

*Дорожный знак* — техническое средство безопасности дорожного движения, стандартизированный графический рисунок, устанавливаемый у дороги для сообщения определённой информации участникам дорожного движения. С 1 января 2006 года в России вступили в силу новые стандарты на дорожные знаки, светофоры и разметку. На дорогах, в связи с этим, появились 24 новых и 18 изменённых знаков. Все они соответствуют конвенции ООН о дорожных знаках и сигналах, а также последним европейским стандартам [7]. Некоторые дорожные знаки приведены на рисунке 5. В данной работе особое значение будет иметь знак ограничения скорости, используемый для контроля скоростного режима.



Рисунок 5 - Дорожные знаки

**Светофоры**

*Светофор* – устройство оптической сигнализации, предназначенное для регулирования движения людей, велосипедов, автомобилей и иных участников дорожного движения, поездов железной дороги и метрополитена, речных и морских судов [8].

**Виды светофоров**

*Автомобильные светофоры*. Наиболее распространены светофоры с сигналами (обычно круглыми) трёх цветов: красного, жёлтого и зелёного. В некоторых странах в том числе и в России вместо жёлтого используется оранжевый цвет. Сигналы могут быть расположены как вертикально (при этом красный сигнал всегда располагается сверху, а зелёный — снизу), так и горизонтально (при этом красный сигнал всегда располагается слева, а зелёный — справа). При отсутствии других, специальных светофоров они регулируют движение всех видов транспортных средств и пешеходов (но на перекрёстке может быть безсветофорное движение последних). Иногда сигналы светофора дополняют специальным табло обратного отсчёта времени, которое показывает сколько времени ещё будет гореть сигнал. Чаще всего табло обратного отсчёта делают для зелёного сигнала светофора, но в ряде случаев табло отображает и оставшееся время красного сигнала. Практически повсеместно красный сигнал светофора запрещает движение, жёлтый запрещает выезд на охраняемый светофором участок, но допускает завершение его проезда, а зелёный — разрешает движение. Распространено, но не повсеместно использование сочетания красного и жёлтого сигналов, обозначающее предстоящее включение зелёного сигнала. Иногда зелёный сигнал включается сразу после красного без промежуточного жёлтого, но не наоборот. Детали применения сигналов различаются в зависимости от принятых в той или иной стране Правил дорожного движения. Существуют светофоры из двух секций - красной и зелёной. Такие светофоры обычно устанавливаются на пунктах, где пропуск автомобилей производится в индивидуальном порядке, например, на пограничных переходах, при въезде или выезде с автостоянки, охраняемой территории и т. п. Пример светофора на рисунке 6.



Рисунок 6 – Светофор

*Стрелки и стрелочные секции*. На светофорах могут быть дополнительные секции в виде стрелок или контуров стрелок, регулирующие движение в том или ином направлении. Правила (в странах бывшего СССР) таковы:

* Контурные стрелки на красном (жёлтом, зелёном) фоне - это обычный светофор, действующий только в заданном направлении;
* Сплошная зелёная стрелка на чёрном фоне разрешает проезд, но не даёт преимущества при разъезде;
* Светофор с мигающим красным сигналом. Красный мигающий сигнал (как правило, на светофорах с мигающей одной или попеременно мигающими двумя красными секциями) используется для ограждения пересечений с трамвайными линиями при приближении трамвая, мостов при разводке, участков дорог вблизи взлётно-посадочных полос аэропортов при взлёте и посадке самолётов на опасной высоте. Эти светофоры аналогичны тем, что используются на железнодорожных переездах;
* Светофор, устанавливаемый на железнодорожных переездах.

*Реверсивный светофор*. Для регулирования движения по полосам проезжей части (особенно там, где возможно реверсивное движение), применяют специальные светофоры контроля полосы (реверсивные). В соответствии с Венской конвенцией о дорожных знаках и сигналах такие светофоры могут иметь два или три сигнала: красный Х-образный сигнал запрещает движение по полосе; зелёная стрелка, направленная вниз, разрешает движение; дополнительный сигнал в виде диагональной жёлтой стрелки информирует о смене режима работы полосы и указывает направление, в котором её необходимо покинуть. Реверсивный светофор изображён на рисунке 7.

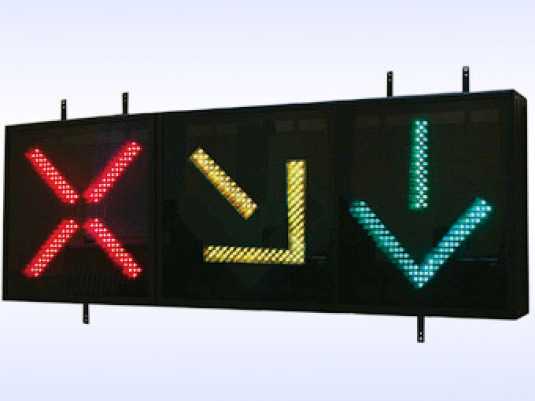


Рисунок 7 – Реверсивный светофор

*Светофоры для маршрутных транспортных средств*. Для регулирования движения маршрутных транспортных средств (трамваев, автобусов, троллейбусов) используют специальные светофоры, вид которых отличается от страны к стране.

* Светофор для пешеходов.
* Светофор для велосипедов.
* Трамвайный светофор.

**Субъекты**

*Водитель* – человек, управляющий своим транспортным средством.

*Пешеход* – участник дорожного движения, находящийся вне транспортного средства. К пешеходам приравниваются лица, передвигающиеся в инвалидных колясках без двигателя, ведущие велосипед, мопед, мотоцикл, везущие санки, тележку, детскую или инвалидную коляску. Правилами дорожного движения определяются права и обязанности пешеходов. В дорожной инфраструктуре для пешеходов предназначены пешеходные дорожки (тротуары), регулируемые и нерегулируемые наземные пешеходные переходы, а также надземные и подземные пешеходные переходы. Пешеходы должны двигаться по тротуарам или пешеходным дорожкам, а при их отсутствии — по обочинам. Пешеходы, перевозящие или переносящие громоздкие предметы, а также лица, передвигающиеся в инвалидных колясках без двигателя, могут двигаться по краю проезжей части, если их движение по тротуарам или обочинам создаёт помехи для других пешеходов. При отсутствии тротуаров, пешеходных дорожек или обочин, а также в случае невозможности двигаться по ним пешеходы могут двигаться по велосипедной дорожке или идти в один ряд по краю проезжей части (на дорогах с разделительной полосой — по внешнему краю проезжей части) [9].

*Сотрудник дорожно-патрульной службы* (ДПС). *Дорожно-патрульная служба* – структурное подразделение Государственной инспекции безопасности дорожного движения Министерства внутренних дел Российской Федерации, входящее в состав милиции общественной безопасности Российской Федерации. Появилось в 1918 году. Основными задачами ДПС являются: сохранение жизни, здоровья и имущества участников дорожного движения, защита их законных прав и интересов, а также интересов общества и государства; обеспечение безопасного и бесперебойного движения транспортных средств; предупреждение и пресечение преступлений и административных правонарушений в области дорожного движения. Выполнение других обязанностей, возложенных на подразделения милиции общественной безопасности [10]. На рисунке 8 изображён контроль дорожного движения сотрудником ДПС.



Рисунок 8 – Сотрудник ДПС

### Правила дорожного движения

*Правила дорожного движения (сокращенно ПДД)* – свод правил, регулирующих обязанности водителей транспортных средств и пешеходов, а также технические требования, предъявляемые к транспортным средствам для обеспечения безопасности дорожного движения.

История современных правил дорожного движения берёт своё начало в Лондоне. 10 декабря 1868 года на площади перед Парламентом был установлен механический железнодорожный семафор с цветным диском.

Прототипами современных дорожных знаков можно считать таблички, на которых указывалось направление движения к населённому пункту и расстояние до него. Решение о создании единых европейских правил дорожного движения было принято в 1909 году на всемирной конференции в Париже, ввиду увеличения числа автомобилей, роста скоростей и интенсивности движения на городских улицах. Следующим важным шагом стало принятие «Конвенции о введении единообразия в сигнализацию на дорогах» в 1931 году в Женеве, на Конференции по дорожному движению, в которой среди прочих стран принял участие и Советский Союз [11].

### Формализация задачи. Теория графов

Для решения задачи можно представить участок ГТС как граф, узлами которого являются перекрёстки, а дугами – улицы (рисунок 10).

## 

Рисунок 10 – Представление участка в виде графа

*Теория графов* – раздел дискретной математики, изучающий свойства графов. В общем смысле граф представляется как множество вершин (узлов), соединённых рёбрами. В строгом определении графом называется такая пара множеств G=(V,E), где V есть подмножество любого счётного множества, а E - подмножество V×V. Теория графов находит применение, например, в геоинформационных системах (ГИС). Существующие или вновь проектируемые дома, сооружения, перекрёстки, кварталы и т. п. рассматриваются как вершины, а соединяющие их дороги, инженерные сети, линии электропередачи и т. п. - как рёбра. Применение различных вычислений, производимых на таком графе, позволяет, например, найти кратчайший объездной путь или ближайший продуктовый магазин, спланировать оптимальный маршрут.

Родоначальником теории графов считается Леонард Эйлер. В 1736 году в одном из своих писем он формулирует и предлагает решение задачи о семи кёнигсбергских мостах, ставшей впоследствии одной из классических задач теории графов [12].

**Основные термины теории графов**

*Граф* – базовое понятие. Включает множество вершин и множество рёбер, являющееся подмножеством декартова квадрата множества вершин (то есть каждое ребро соединяет ровно две вершины). Участок ГТС будет представлен в виде графа.

*Вершина, Узел* – базовое понятие: точка, где могут сходиться/выходить рёбра и/или дуги. Множество вершин графа G обозначается V(G). Перекрёсток – вершина.

*Ребро графа* – базовое понятие. Ребро соединяет две вершины графа. Дорога – ребро.

*Дуга* – это ориентированное ребро.

*Взвешенный граф* – граф, каждому ребру которого поставлено в соответствие некое значение (вес ребра).

*Вес ребра* – значение, поставленное в соответствие данному ребру взвешенного графа. Обычно, вес - вещественное число, в таком случае его можно интерпретировать как «длину» ребра. Длина дороги – вес ребра.

### Алгоритм Дейкстры

*Алгоритм* Дейкстры (Dijkstra’s algorithm) – алгоритм на графах, изобретённый нидерландским ученым Э. Дейкстрой в 1959 году. Находит кратчайшее расстояние от одной из вершин графа до всех остальных. Алгоритм работает только для графов без рёбер отрицательного веса. Алгоритм широко применяется в программировании и технологиях.

Каждой вершине из V сопоставим метку - минимальное известное расстояние от этой вершины до a. Алгоритм работает пошагово - на каждом шаге он «посещает» одну вершину и пытается уменьшать метки. Работа алгоритма завершается, когда все вершины посещены.

Инициализация. Метка самой вершины a полагается равной 0, метки остальных вершин — бесконечности. Это отражает то, что расстояния от a до других вершин пока неизвестны. Все вершины графа помечаются как непосещённые.

Шаг алгоритма. Если все вершины посещены, алгоритм завершается. В противно случае, из ещё не посещённых вершин выбирается вершина u, имеющая минимальную метку. Мы рассматриваем всевозможные маршруты, в которых u является предпоследним пунктом. Вершины, в которые ведут рёбра из u, назовем соседями этой вершины. Для каждого соседа вершины u, кроме отмеченных как посещённые, рассмотрим новую длину пути, равную сумме значений текущей метки u и длины ребра, соединяющего u с этим соседом. Если полученное значение длины меньше значения метки соседа, заменим значение метки полученным значением длины. Рассмотрев всех соседей, пометим вершину u как посещенную и повторим шаг алгоритма [13].

### Описание систем – аналогов

В настоящее время существует немало систем, реализующих процесс нахождения пути и навигации, такие как 2ГИС, Navitel, Gоogle Earth, Яндекс.Карты, Garmin. Интерфейс программы Gоogle Earth приведён на рисунке 10.

Программа имеет множество функций: масштабирование, 3D изображение, метки, пути, отображение погоды, улиц, зданий, поиск объектов, навигация и т.д.

*2ГИС.* Компания «ДубльГИС» является разработчиком электронных справочников, которые помогают пользователям программы с легкостью ориентироваться более чем в 130 городах России и Украины и быстро находить нужную информацию.

Собственные call-центры в каждом городе присутствия 2ГИС постоянно отслеживают информацию и ежемесячно вносят изменения в справочник. Основные сведения о компаниях и предприятиях в справочнике размещаются бесплатно. Карты 2ГИС создаются профессиональным отделом картографического производства по лицензии Федеральной службы геодезии и картографии [14].

Интерфейс программы 2ГИС приведён на рисунке 11.

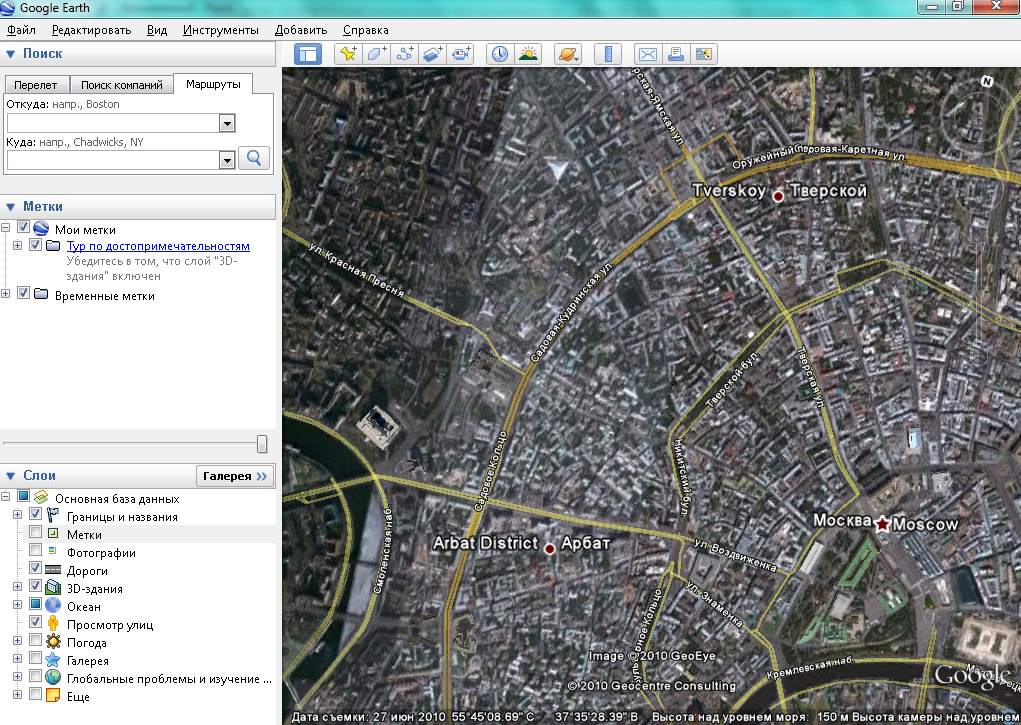


Рисунок 10 - Интерфейс Gоogle Earth

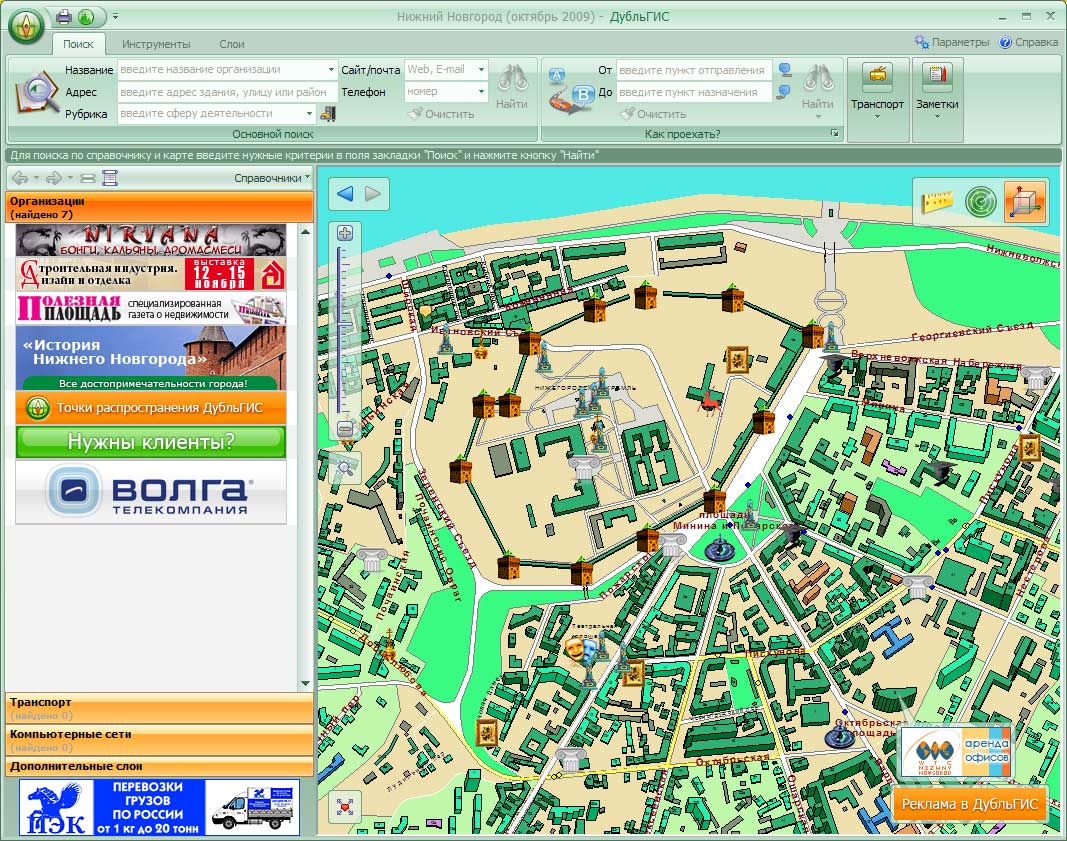


Рисунок 11 - Интерфейс 2ГИС

*Яндекс.Карты.* Предоставляют пользователю доступ к картам городов России, Украины, Казахстана и др. стран с детализацией до дома. Поиск фрагмента карты по городскому адресу или названию населенного пункта. Отображение пробок, панорам улиц и др [15]. Интерфейс программы Яндекс.Карты приведён на рисунке 12.

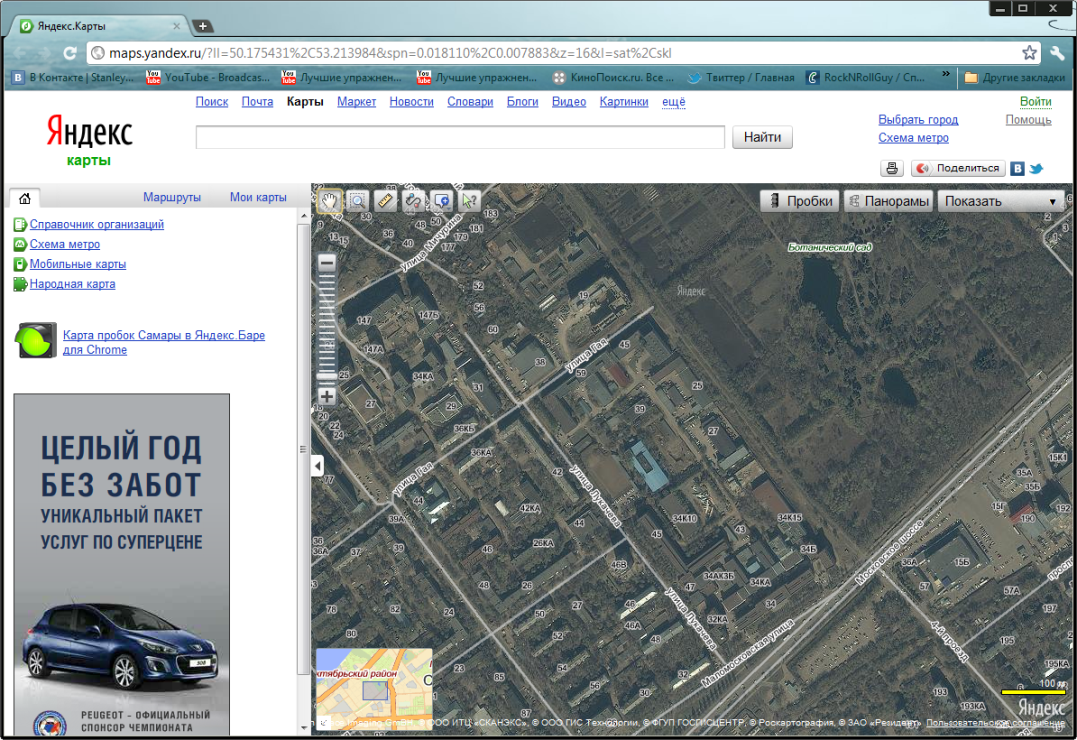


Рисунок 12 - Интерфейс Яндекс.Карты

### Постановка задачи

В рамках поставленной задачи необходимо разработать программную систему, позволяющую моделировать нахождение оптимального пути на участке ГТС.

Пользователь должен иметь возможность загрузить уже имеющийся участок ГТС, составить новый с помощью специального редактора, встроенного в разрабатываемую систему. При составлении участка ГТС пользователь должен использовать такие графические элементы, как узлы (перекрестки), дуги (дороги), знаки запрещения движения (не более одного на дугу), знаки ограничения скорости (в соответствии с правилами дорожного движения ПДД), сотрудников ДПС, светофоры. При этом данным объектам необходимо задавать некоторые характеристики:

* У дуги есть свой тип покрытия, длина (положительное число) и имя (от 1 до 15 символов), также возможно отметить на дуге наличие пробки, наличие сотрудника ДПС, поставить на дугу знак ограничения скорости, знак запрещения движения. Сотруднику ДПС необходимо задать коэффициент строгости (положительное число);
* К узлу можно привязать не более одного светофора и сотрудника ДПС. Также узлу необходимо задать время его прохождения. Пользователь должен иметь возможность задать начальную фазу светофора (красный, зеленый), длительность зеленой и красной фазы (положительное число). Также необходимо отметить, что сотрудник ДПС может быть установлен на узел только при наличии светофора.

После выбора участка ГТС пользователь должен выбрать объекты, участвующие в движении (автомобиль и водитель), а также выбрать критерии нахождения оптимального пути .

К характеристикам автомобиля относятся: максимальная скорость движения (натуральное число), объем бака (натуральное число), расход топлива на 100 км (положительное число), стоимость литра топлива (положительное число). Пользователь задает также марку автомобиля (от 4 до 15 символов).

К характеристикам водителя относятся: имя (от 1 до 15 символов), тип (нарушитель, добропорядочный), критерий нахождения оптимального пути (время — действительное число от 0 до 1, скорость — действительное число от 0 до 1, расстояние — действительное число от 0 до 1).

Все данные и настройки при составлении участка ГТС должны храниться в базе данных (БД). В системе должны вестись следующие справочники: типов топлива, названий улиц, типов сотрудника ДПС. Необходимо предусмотреть целостность таблиц БД.

При возникновении вопросов у пользователя должна иметься возможность обратиться к справочной информации по системе.

Таким образом, система моделирования должна выполнять следующие функции:

1. настройка параметров системы;
2. составление участка ГТС;
3. сохранение участка ГТС в БД с заданной структурой;
4. загрузка участка ГТС в БД с заданной структурой;
5. работа с редактором участка ГТС:
6. добавление объекта;
7. удаление объекта;
8. настройка характеристик объектов;
9. работа с базой данных:
10. добавление данных объектов;
11. удаление данных объектов;
12. редактирование данных объектов;
13. ведение справочников БД;
14. выдача справочной информации о системе.

### Построение структурной схемы программной системы

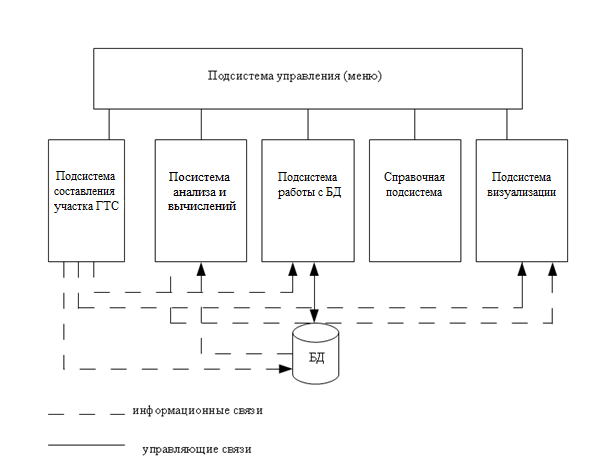
На основании проведенного выше анализа предметной области формируются общие требования к функциям, которые должна выполнять система. Проектируемую систему разобьем на подсистемы и определим взаимосвязи между ними.

*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. Более мелкая система – *это подсистема*. Проектируемую систему разобьем на подсистемы и определим взаимосвязи между ними.

В проектируемую систему входят шесть подсистем:

1. Подсистема управления (меню), которая обеспечивает организацию взаимодействия всех подсистем системы.
2. Подсистема визуализации служит для отображения результатов моделирования в процессе его выполнения и вывод системных сообщений
3. Подсистема анализа и вычислений служит для вычисления оптимального пути по заданному критерию.
4. Подсистема работы с БД служит для сохранения и загрузки данных в БД .
5. Справочная подсистема, которая предназначена для вывода справочной информации о системе, разработчиках, а также руководства пользователя системой.
6. Подсистема составления участка ГТС служит для моделирования дорожно-транспортной сети.

Структурная схема автоматизированной системы поиска оптимального пути по заданному критерию приведена на рисунке 14.

Рисунок 14 - Структурная схема системы

### Функциональная спецификация

Разработка П/С 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:

1. спецификация качества на П/С
2. функциональная спецификация

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

*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:

1. описание внешней информационной среды
2. определение функций, определенных на множестве состояний этой информационной среды.
3. описание исключительных ситуаций, которые могут возникнуть при выполнении функций П/С 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*.

### Перечень функций, выполняемых системой

При детальном проектировании вводятся обозначения для всех функций, специфицируются все входные данные и результаты каждой определенной функции, включая указание их типов и задания всех ограничений, которые должны удовлетворять эти данные и результаты. В таблице 1 приведен перечень функций, выполняемых системой.

### Перечень исключительных ситуаций и информационных сообщений

В рамках функциональной спецификации необходимо указать исключительные ситуации, возникновение которых возможно в процессе работы программы.

Исключительная ситуация - это наиболее существенная с точки зрения пользователя ситуация, при которой система не может нормально выполнить ту или иную функцию.

Таблица 2 – Перечень исключительных ситуаций

|  |  |
| --- | --- |
| Тип исключительной ситуации | Поведение системы |
| Отсутствие файла справки | Предупреждение пользователя о не возможности нормального функционирования справочной подсистемы |
| Ошибка при подключении к БД | Вывод диалога с возможностью выбора: завершить программу или изменить настройки подключения к БД |
| Ошибка при выполнении запроса к БД | Завершение работы программы с возможностью сохранения изменений в файл. |
| БД не прошла контроль целостности | Завершение работы программы, сообщение о необходимости исправления БД. |

### Разработка структур данных и классов объектов

**Структура** - это набор данных, где данные могут быть разного типа. Например, структура может содержать несколько переменных различных типов. Переменные, которые содержатся в структуре, называются членами или полями структуры.

Структура данных «класс», представляющая собой объектный тип данных, внешне похожа на типы данных процедурно-ориентированных языков. При этом есть возможность сокрытия внутреннего устройства объекта и его поведения. Такая возможность называется инкапсуляцией. **Инкапсуляция** — свойство языка программирования, позволяющее пользователю не задумываться о сложности реализации используемого программного компонента (то, что у него внутри), а взаимодействовать с ним посредством предоставляемого интерфейса (публичных методов и членов), а также объединить и защитить жизненно важные для компонента данные. При этом пользователю предоставляется только спецификация объекта.

Наличие инкапсуляции не означает объектной ориентированности языка программирования — для этого требуется наличие наследования. **Наследование** — механизм объектно-ориентированного программирования (наряду с инкапсуляцией, полиморфизмом и абстракцией), позволяющий описать новый класс на основе уже существующего (родительского), при этом свойства и функциональность родительского класса заимствуются новым классом. В нашей системе выделяются классы BaseEntity и PersonEntity как базовые, BaseEntity – базовый класс всех сущностей, PersonEntity – сущности хранящие информацию о людях.

**UML** ([англ.](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA" \o "Английский язык)*Unified Modeling Language*– унифицированный язык моделирования) – [язык](http://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) [графического](http://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%B7%D1%83%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F) описания для [объектного моделирования](http://ru.wikipedia.org/w/index.php?title=%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE%D0%B5_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5&action=edit&redlink=1) в области [разработки программного обеспечения](http://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F). UML является языком широкого профиля, это [открытый стандарт](http://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D0%B9_%D1%81%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82), использующий графические обозначения для создания [абстрактной модели](http://ru.wikipedia.org/wiki/%D0%90%D0%B1%D1%81%D1%82%D1%80%D0%B0%D0%BA%D1%82%D0%BD%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C) [системы](http://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0), называемой *UML-моделью*. UML был создан для определения, визуализации, проектирования и документирования в основном программных систем. UML не является языком программирования, но в средствах выполнения UML-моделей как интерпретируемого кода возможна кодогенерация.

В UML **диаграмма классов** описывает структуру системы, показывая её классы, их атрибуты и операторы, а также взаимосвязи этих классов. На рисунке 15 представлена UML-диаграмма классов.

Таблица 3 – Описание класса «Базовая сущность»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип | Описание |
| идентификатор | Длинное целое | Идентификатор |
| версия | Длинное целое | Номер версии |

Таблица 4 – Описание класса «Карта ГТС»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип | Описание |
| название | Строка |  |
| перекрестки | Набор объектов типа «Узел» |  |
| ширина | Целое |  |
| высота | Целое |  |

Таблица 5 – Описание класса «Узел»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип | Описание |
| координата по вертикали | Целое | Координата узла на карте |
| координата по горизонтали | Целое | Координата узла на карте |
| карта | Объект «Карта ГИС» |  |
| светофор | Объект «Светофор» |  |
| полицейский | Объект «Полицейский» |  |
| начала улиц | Набор объектов "Ребро" |  |
| окончания улиц | Набор объектов "Ребро" |  |

Таблица 6 – Описание класса «Ребро»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип | Описание |
| улица | Объект «Улица» | Улица |
| покрытие | Объект «Покрытие» | Покрытие |
| длина | Действительное |  |
| цвет | Строка |  |
| полицейский | Объект «Полицейский» |  |
| начальный узел | Объект «Узел» |  |
| конечный узел | Объект «Узел» |  |
| список дорожных знаков | Набор объектов типа «Знак» |  |

Таблица 7 – Описание класса «Топливо»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип | Описание |
| марка | Строка |  |
| цена | Денежный |  |

Таблица 8 – Описание класса «Улица»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип | Описание |
| название | Строка |  |

Таблица 9 – Описание класса «Машина»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип | Описание |
| топливо | Строка |  |
| марка | Строка |  |
| модель | Строка |  |
| расход топлива | Действительное |  |
| максимальная скорость | Целое |  |

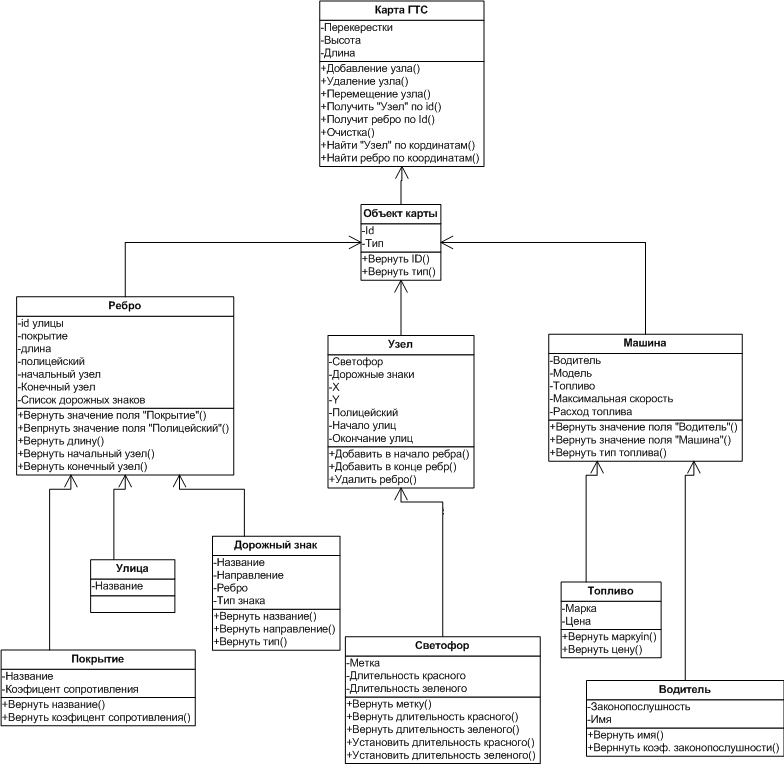
  
Рисунок 14 – UML-диаграмма классов

Таблица 10 – Описание класса «Покрытие»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип | Описание |
| название | Строка |  |
| коэффициент сопротивления | Действительное |  |

Таблица 11 – Описание класса «Дорожный знак»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип | Описание |
| название | Строка |  |
| ребро | Объект «Ребро» |  |
| направление | Целое |  |
| тип знака | Целое |  |
| значение знака | Целое |  |

Таблица 12 – Описание класса «Светофор»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип | Описание |
| метка | Строка |  |
| длительность красного | Целое |  |
| длительность зеленого | Целое |  |

Таблица 13 – Описание класса «Водитель»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип | Описание |
| вероятность нарушения ПДД | Действительное |  |

Таблица 14 – Описание класса «Сущность для описания человека»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип | Описание |
| Фамилия | Строка |  |
| Отчество | Строка |  |
| Имя | Строка |  |
| отображаемое имя | Строка |  |

Таблица 15 – Описание класса «Полицейский»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип | Описание |
| коэффициент жадности | Действительное |  |

1. Выбор и обоснование алгоритмов обработки данных

Алгоритм Дейкстры (Dijkstra) - предназначен для решения задачи Поиск кратчайших путей в графе [8].

Ниже представлен алгоритм нахождения пути минимального веса в графе *G = (V, E)* заданном весовой матрицей *W* у которой элемент *wij* равен весу ребра соединяющего *i*-ую и *j*-ую вершины. При этом предполагается, что все элементы *wij* неотрицательны. Путь ищется из вершины номер *u1* к вершине номер *u2*. Здесь использует алгоритм Дейкстры. Для представления веса, равного бесконечности, используется число *GM*, передаваемое в алгоритм. Это число можно задавать в зависимости от конкретной задачи.

Алгоритм по которому происходит поиск заключается в следующем:

1. всем веpшинам пpиписывается вес - вещественное число, *d(i) := GM* для всех вершин кроме вершины с номером *u1*, а *d(u1 ) := 0*
2. всем веpшинам пpиписывается метка *m(i) := 0*
3. вершина *u1* объявляется текущей: *t := u1*
4. для всех вершин у которых *m(i)* равно *0*, пересчитываем вес по формуле: *d(i) := min{d(i), d(t)+W[t,i]}*
5. среди вершин для которых выполнено *m(i) = 0* ищем ту для которой *d(i)* минимальна, если минимум не найден, т.е. вес всех не "помеченных" вершин равен бесконечности (*GM*), то путь не существует, покидаем алгоритм

иначе найденную вершину c минимальным весом полагаем текущей и помечаем (*m(t) := 1*)

1. если *t = u2*, то найден путь веса *d(t)*, покидаем алгоритм
2. переходим на шаг 4.

На выходе имеем переменную **Length**, которая определяет длину пути (**length** равно null если пути не существует, **length** равно *0*, если *u1* равно *u2*), переменную **Weight** - вес пути и массив **Path** содержащий последовательность номеров вершин определяющих путь.

Схема алгоритма представлена на рисунке 15.

1. Выбор и обоснование комплекса программных средств

### Выбор операционной системы

В качестве операционной системы мы не выбираем конкретную ОС поскольку используем кросплатформенные средства разработки, такие как: Qt 4.7.0, MySQL 5.1. Эти программные инструменты позволяют вести разработку под ряд современнных ОС:

Начало

Инициализация переменных

i=1, N

d(i) := GM

m(i) := 0

t := u1

i := 0

i=1, N

d(i) := min{d(i), d(t)+W[t,i]}

m(i) = 0

i=1, N

m(i) = 0

d(i) = d(i)min

m(i) := 1

t := i

Да

Нет

Нет

Нет

Да

Да

T = u2

Конец

Рисунок 15 – Блок схема алгоритма

семейство Linux 2.6, Unix (FreeBSD), Microsoft Windows. Кроме того, в качестве среды разработки АС «поиска оптимального пути по заданному критерию» была выбрана Qt Creator 2.0 , позволяющая быстро и гибко разрабатывать приложения для бизнеса на платформе C++.

### Выбор языка программирования

C++ — чрезвычайно мощный язык, содержащий средства создания эффективных программ практически любого назначения, от низкоуровневых утилит и драйверов до сложных программных комплексов самого различного назначения. В частности:

* Поддерживаются различные стили и технологии программирования, включая традиционное директивное программирование, ООП, обобщённое программирование, метапрограммирование (шаблоны, макросы).
* Предсказуемое выполнение программ является важным достоинством для построения систем реального времени. Весь код, неявно генерируемый компилятором для реализации языковых возможностей (например, при преобразовании переменной к другому типу), определён в стандарте. Также строго определены места программы, в которых этот код выполняется. Это даёт возможность замерять или рассчитывать время реакции программы на внешнее событие.
* Автоматический вызов деструкторов объектов при их уничтожении, причём в порядке, обратном вызову конструкторов. Это упрощает (достаточно объявить переменную) и делает более надёжным освобождение ресурсов (память, файлы, семафоры и т. п.), а также позволяет гарантированно выполнять переходы состояний программы, не обязательно связанные с освобождением ресурсов (например, запись в журнал).
* Пользовательские функции-операторы позволяют кратко и ёмко записывать выражения над пользовательскими типами в естественной алгебраической форме.
* Язык поддерживает понятия физической (const) и логической (mutable) константности. Это делает программу надёжнее, так как позволяет компилятору, например, диагностировать ошибочные попытки изменения значения переменной. Объявление константности даёт программисту, читающему текст программы дополнительное представление о правильном использовании классов и функций, а также может являться подсказкой для оптимизации. Перегрузка функций-членов по признаку константности позволяет определять изнутри объекта цели вызова метода (константный для чтения, неконстантный для изменения). Объявление mutable позволяет сохранять логическую константность при использовании кэшей и ленивых вычислений.
* Используя шаблоны, возможно создавать обобщённые контейнеры и алгоритмы для разных типов данных, а также специализировать и вычислять на этапе компиляции.
* Возможность имитации расширения языка для поддержки парадигм, которые не поддерживаются компиляторами напрямую. Например, библиотека Boost.Bind позволяет связывать аргументы функций.
* Возможность создания встроенных предметно-ориентированных языков программирования. Такой подход использует, например библиотека Boost.Spirit, позволяющая задавать EBNF-грамматику парсеров прямо в коде C++.
* Используя шаблоны и множественное наследование можно имитировать классы-примеси и комбинаторную параметризацию библиотек. Такой подход применён в библиотеке Loki, класс SmartPtr которой позволяет, управляя всего несколькими параметрами времени компиляции, сгенерировать около 300 видов «умных указателей» для управления ресурсами.
* Кроссплатформенность: стандарт языка накладывает минимальные требования на ЭВМ для запуска скомпилированных программ. Для определения реальных свойств системы выполнения в стандартной библиотеке присутствуют соответствующие возможности (например, std::numeric\_limits <T>). Доступны компиляторы для большого количества платформ, на языке C++ разрабатывают программы для самых различных платформ и систем.
* Эффективность. Язык спроектирован так, чтобы дать программисту максимальный контроль над всеми аспектами структуры и порядка исполнения программы. Ни одна из языковых возможностей, приводящая к дополнительным накладным расходам, не является обязательной для использования — при необходимости язык позволяет обеспечить максимальную эффективность программы.
* Имеется возможность работы на низком уровне с памятью, адресами.
* Высокая совместимость с языком C, позволяющая использовать весь существующий C-код (код на C может быть с минимальными переделками скомпилирован компилятором C++; библиотеки, написанные на C, обычно могут быть вызваны из C++ непосредственно без каких-либо дополнительных затрат, в том числе и на уровне функций обратного вызова, позволяя библиотекам, написанным на C, вызывать код, написанный на С++).

C++ широко используется для разработки программного обеспечения. Область его применения включает создание операционных систем, разнообразных прикладных программ, драйверов устройств, приложений для встраиваемых систем, высокопроизводительных серверов, а также развлекательных приложений (например, видеоигры). Существует несколько реализаций языка C++ — как бесплатных, так и коммерческих. Наиболее популярны проект GNU, Microsoft, Intel и Embarcadero (Borland). C++ оказал огромное влияние на другие языки программирования, в первую очередь на Java и C#.[10]

### Выбор СУБД

В качестве средства проектирования базы данных, была выбрана MySQL 5.1., она является широко распространённой СУБД, используется как на Linux, так и Windows

серверах. Для коммуникации с сервером БД на базе MySQL имеется набор

библиотек для многих языков программирования и программных платформ.

СУБД MySQL имеет ряд преимуществ. Главным из них является то, что она обеспечивает создание информационных систем с архитектурой "клиент-сервер", в которой он играет роль сервера баз данных. Также можно сказать, что он удовлетворяет требованиям, предъявляемым к системам распределенной обработки информации. Эта СУБД поддерживает: тиражирование данных, параллельную обработку, создание и обработку больших баз данных на недорогих аппаратных платформах, отличается простотой управления и использования.

## КОНСТРУКТИВНО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

* 1. Разработка и описание интерфейса пользователя

В процессе дизайна интерфейса можно выделить три основных этапа, а именно первоначальное проектирование, создание прототипа и тестирование/модификация прототипа [9].

На этапе **первоначального проектирования** закладываются основные концепции системы, влияющие абсолютно на все показатели качества её интерфейса. Проектирование состоит из следующих этапов:

1. определение необходимой функциональности системы;
2. создание пользовательских сценариев;
3. проектирование общей структуры;
4. проектирование отдельных блоков;
5. сбор и начальная проверка полной схемы системы.

На этапе **создания прототипа** необходимо учитывать, что первый прототип желательно делать максимально примитивным. Только после того, как тестирование подтверждает его правильность, стоит делать более детализированный прототип.

Все недочеты создаваемого интерфейса должны быть устранены на этапе **тестирования/модификации прототипа**. Технически сеанс тестирования довольно прост. Нужно иметь несколько пользователей, которые ни разу не видели текущего состояния системы. Тестерам дается задание, они его выполняют, после чего результаты анализируются.

После прохождения всех вышеперечисленных этапов программная система будет иметь простой и понятный интерфейс, позволяющий, тем не менее, использовать все ее возможности.

* 1. Разработка и описание пользовательского меню

Все функции программной системы реализуются посредством графического интерфейса пользователя (рисунок 16).

Главное меню имеет три пункта: «Карта», «Правка», «Инструменты» и «Справка». При выборе пункта «Карта» раскрывается список, из которого можно выбрать создание нового участка ГТС, загрузку ГТС, также можно закрыть ГТС и выйти из приложения. С помощью пункта «Правка» можно перейти к редактированию справочников улиц, типов дорожного покрытия, топлива, водителей и автомобилей. С помощью пункта меню «Справка» пользователь может получить информацию о программе, а также прочитать небольшое руководство пользователя к программе. Структура главного меню представлена на рисунке 17.

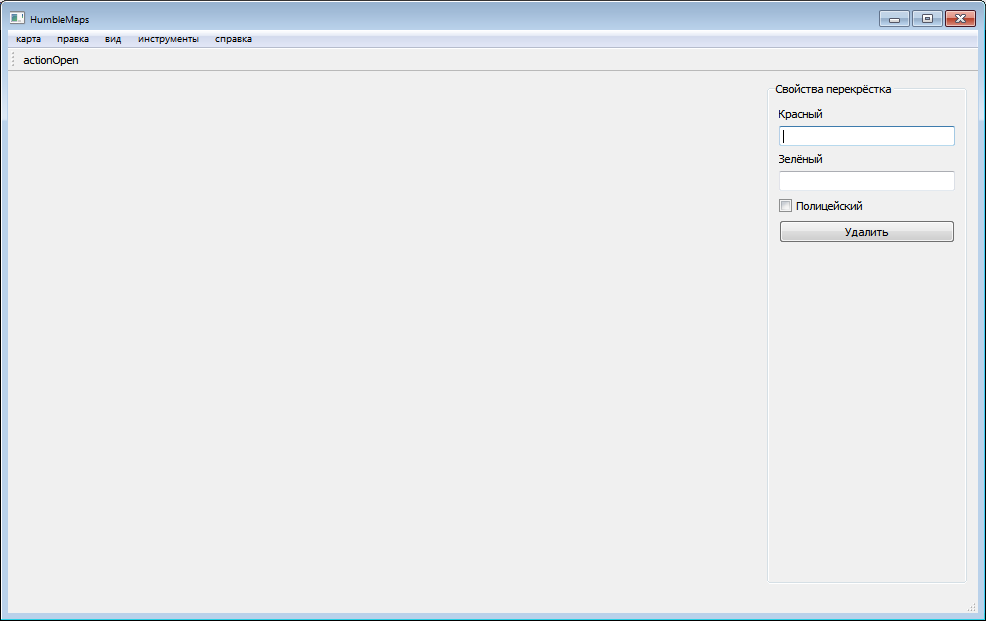


Рисунок 16 – Вид главной формы

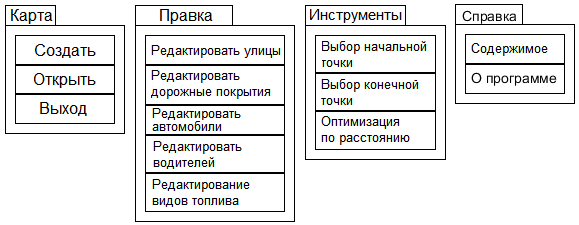


Рисунок 17 – Структура главного меню

* + 1. Разработка панели инструментов

Для удобства пользователя в программе используется панель инструментов. Она позволяет осуществлять быстрый доступ к основным функциям редактирования участка ГТС.

На панели инструментов располагаются инструменты для редактирования участка ГТС: мышь, добавление узла, добавление ребра, начало маршрута, конец маршрута.(Таблица 16).

|  |  |  |
| --- | --- | --- |
| Картинка | Название | Функции |
| mouse | Мышь | Стандартный курсор |
| node | Добавление узла | По клику в области карты добавляем узел ГТС |
| line | Добавление ребра | Кликая по 2м узлам последовательно, добавляет ребро между ними |
| start | Начало маршрута | Отмечает начальный узел маршрута |
| finish | Конец маршрута | Отмечает конечный узел маршрута |

Таблица 16 - Панель инструментов

* + 1. Описание контрольного примера

При входе в программу, создаётся новый участок ГТС и открывается редактор карты (рисунок 18).

Добавление узла: переключиться на инструмент “добавление узла”, затем клик в поле редактора на место, куда хотим поставить узел.

Два узла можно соединить дугой – переключиться на инструмент “добавление дуги”, кликнуть на один затем на второй узел.

В системе представлены справочники улиц, автомобилей, топлива, сотрудников ДПС и водителей. Их вид унифицирован и включает в себя фильтр и таблицу объектов справочника. В справочники можно добавлять объекты, редактировать их и удалять (если на объект не ссылается ни одна карта в БД). На рисунке 19 приведён пример справочника улиц, на рисунке 20 – сотрудников ДПС.

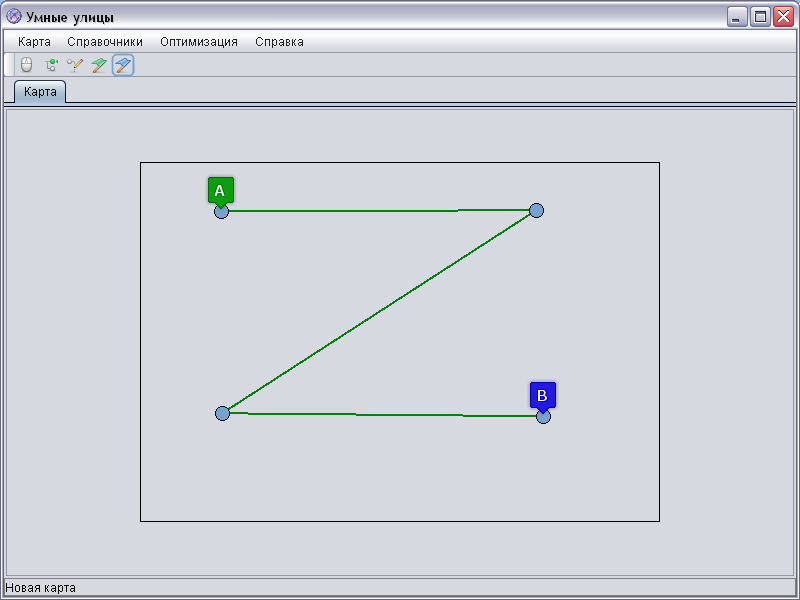


Рисунок 18 – Редактор карты

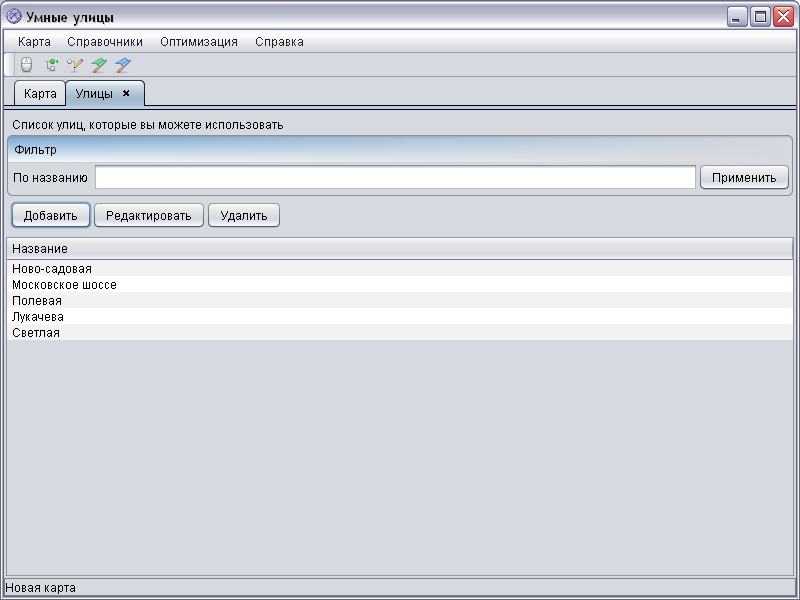


Рисунок 19 – Окно редактирования справочника улиц

Параметры узла и дуги можно редактировать, нажав на соответствующем объекте правой кнопкой мыши и выбрав меню Свойства.

Для удаления компонента, следует выбрать в контекстном меню пункт “Удалить”. При удалении узла, связанные с ним дуги также исчезнут.

### Реализация классов и структур данных

Физическая модель системы следует из логической, когда разработанные на этапе анализа предметной области концептуальные объекты находят свою реализацию в виде

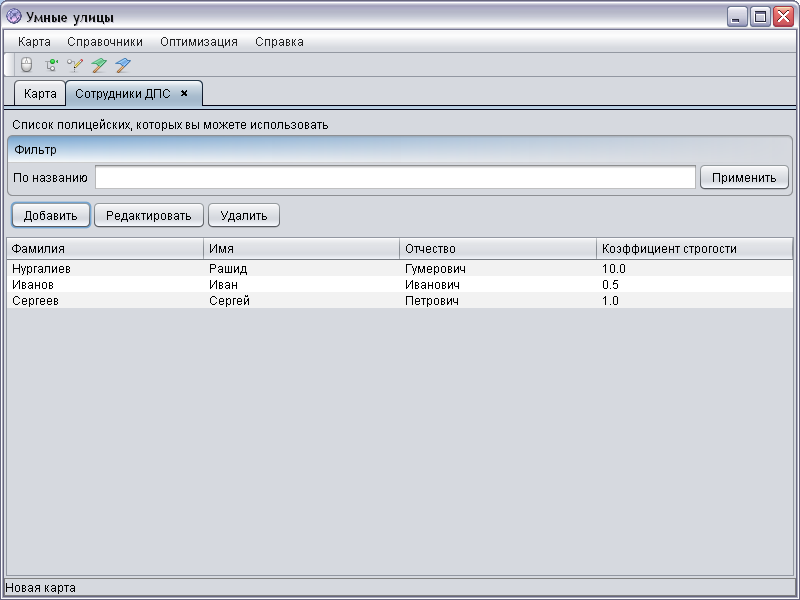


Рисунок 20 – Окно редактирования справочника сотрудников ДПС

классов и других структур данных какого-либо языка программирования*.*

**Модель сущность-связь (ER-модель)** (англ. *entity-relationship model*, ERM) — модель данных, позволяющая описывать концептуальные схемы предметной области. ER-модель используется при высокоуровневом (концептуальном) проектировании баз данных. С её помощью можно выделить ключевые сущности и обозначить связи, которые могут устанавливаться между этими сущностями. В качестве стандартной графической нотации, с помощью которой можно визуализировать ER-модель, была предложена **диаграмма сущность-связь (ER-диаграмма)** [10].

На рисунке 21 представлена диаграмма сущностных классов системы, она полностью идентична ER-модели БД.

Программный код основной формы приложения приведён в приложении А.

* 1. Разработка и описание модулей

Программный модуль – это любой фрагмент описания процесса, оформленный как самостоятельный программный продукт и пригодный для описания этого процесса [11].

В ходе разработки программной системы были разработаны следующие модули, разделенные по принадлежности к различным объектам системы. Они показаны в таблице 17.

|  |  |
| --- | --- |
| Название модуля | Описание модуля |
| global | классы предметной области, маппинг ORM |
| core | сервисы для работы с БД |
| simulation | выполнение оптимизации пути по заданному критерию |
| gui | графический интерфейс пользователя |
| help | справочная информация |

Таблица 17 – Описание модулей системы.

Диаграмма зависимостей модулей и дополнительных библиотек представлена на рисунке 22.

Используемые библиотеки:

* MySQL Connector – драйвер БД для сервера MySQL
* OpenJPA – ORM слой доступа к БД
* Groovy – динамический язык программирования на базе Java
* Log4j – ведение системного журнала
  1. Выбор и обоснование комплексно-технических средств

### Расчет требуемых ресурсов

Расчет требуемых ресурсов включает в себя расчет необходимого объема оперативной памяти, а также памяти жесткого диска.

Необходимый объем **оперативной памяти** рассчитывается как сумма объема оперативной памяти, необходимой для работы непосредственно ПС, минимального объема оперативного запоминающего устройства (ОЗУ), необходимого для нормальной работы операционной системы, а также объема ОЗУ для сервера управления базами данных открытия файла справки, презентации и т.д.

Исходя из этого, рассчитаем требуемый объем ОЗУ, при условии, что программа работает под управлением операционной системы Windows 7. Служба MySQL Server запускается при старте ОС, поэтому она учитывается в памяти, потребляемой ОС.

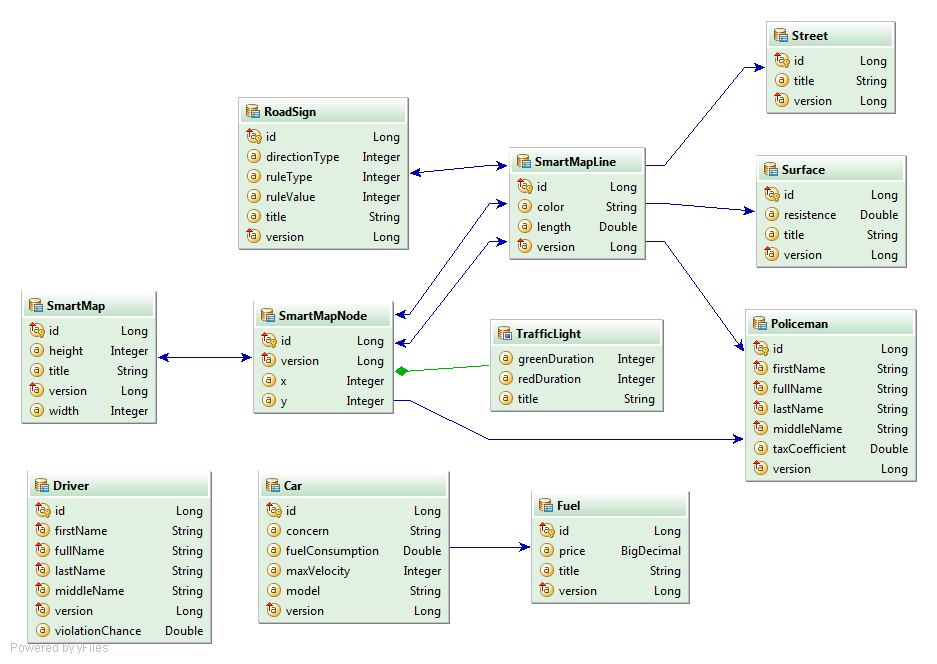


Рисунок 21 – диаграмма сущностных классов системы

Vозу = Vпс + Vос + Vсправки  = 30 Мбайт + 470 Мбайт + 12 Мбайт = 512 Мбайт.

Таким образом, требуемый объем ОЗУ равен 512 Мбайт.

Необходимый объем **памяти жесткого диска** рассчитывается как сумма размера системных файлов операционной системы, размера файла программной системы, размера файла данных, файла справки, презентации.

Суммарный размер файлов системы: 5200 МБ.

* + 1. **Минимальные требования к системе**

Для работы системы необходимо:

1. IBM - совместимый компьютер;
2. Не менее 512 Мбайт оперативной памяти;
3. Не менее 5200 Мбайт на жестком диске;
4. Java Runtime Environment версии 6 или выше;
5. MySQL Server 5.1 или выше;
6. Поддерживаемая Windows XP графическая карта;
7. Поддерживаемая Windows XP мышь и клавиатура;
8. Операционная система Windows XP и выше.

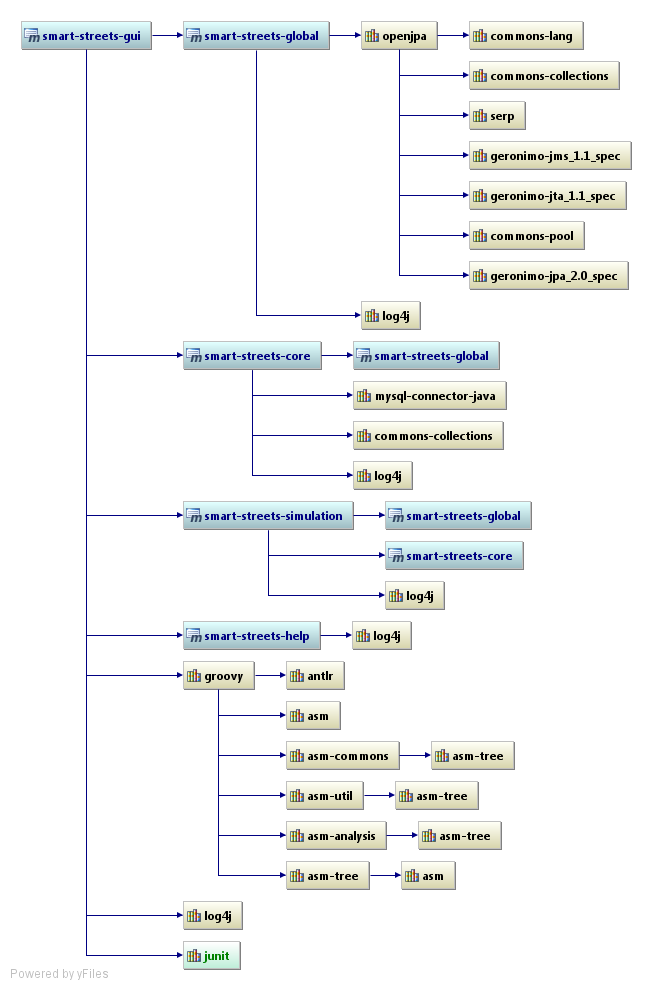


Рисунок 22 – Диаграма модулей системы.

# ЗАКЛЮЧЕНИЕ

В соответствии с утвержденным заданием проведён предпроектный анализ, концептуальное проектирование системы и разработана система. Выполнена физическая реализация проекта в заданной программной среде Intellij Idea. Итогом проделанной работы является автоматизированная система создания и редактирования участка ГТС.

В разработанной автоматизированной системе реализована возможность составления произвольного участка ГТС. Система предоставляет пользователю широкие возможности для описания характеристик составляемого участка ГТС (расстановка знаков дорожного движения, светофоров, сотрудников ДПС и т.д.).

Система позволяет выполнить поиск оптимального пути по критерию расстояния, в дальнейшем планируется расширить набор критериев и добавить оптимизации по критерию времени и стоимости.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Зеленко, Л.С. Методические указания к лабораторному практикуму по дисциплине «Технологии программирования» [Текст]/Л.С. Зеленко. – Самарский аэрокосмический ун-т, 2007г. – 29 с.
2. Проблемы моделирования предметных областей в информационных системах [Электронный ресурс]. – <http://www.codenet.ru/progr/other/modeling-problems/?rss=1>
3. П.И. Поспелов. Дороги [Текст]//Большая Российская энциклопедия: В 30 т. / Председатель Науч.-ред. совета Ю.С. Осипов. Отв. ред. С.Л. Кравец – М.: Большая российская энциклопедия, 2007. – Т. 9. – 767 с.
4. Правила дорожного движения РФ, 2011 [Текст]. – М.: Мир автокниг, 2011. – 48 с.
5. Виды светофоров в мире [Электронный ресурс]. – <http://www.signumplus.ru/index.php?act=pages&id=25>
6. ДПС [Электронный ресурс]. – <http://ru.wikipedia.org/wiki/Дорожно-патрульная_служба>.
7. Тишин В. В. Дискретная математика в примерах и задачах. [Текст] – СПб.: БХВ-Петербург, 2008. — 352 с.
8. Ананий В. Левитин Глава 9. Жадные методы: Алгоритм Дейкстры [Текст]// Алгоритмы: введение в разработку и анализ – М.: «Вильямс», 2006. – С. 189–195.
9. ГОСТ Р 52290-2004. Технические средства организации дорожного движения. Знаки дорожные. Общие технические требования [Электронный ресурс] - <http://www.znakcomplect.ru/gost_r_52290-2004.php>
10. Язык программирования С++ [Электронный ресурс]. – http://ru.wikipedia.org/wiki/С++
11. ER-модель данных [Электронный ресурс]. – <http://ru.wikipedia.org/wiki/ER-модель_данных>
12. СТО СГАУ 02068410-004-2007. Стандарт организации [Текст]. – Самарский аэрокосмический ун-т, 2007г. – 34 с.