

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий  
Кафедра «Прикладная математика и информатика»

01.03.02 ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА

СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ И КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ

### БАКАЛАВРСКАЯ РАБОТА

на тему: **Разработка автоматизированной обучающей системы «Правила дорожного движения»**

Студент \_\_\_\_\_ Е. В. Абаза \_\_\_\_\_

Руководитель \_\_\_\_\_ А. И. Туищев \_\_\_\_\_

Консультант \_\_\_\_\_ Н. В. Яценко \_\_\_\_\_

**Допустить к защите**  
Заведующий кафедрой А.В. Очеповский \_\_\_\_\_

«\_\_\_\_\_» \_\_\_\_\_ 20\_\_ г.

Тольятти 2017

## АННОТАЦИЯ

Бакалаврская работа представлена на 57 страницах, включает 24 иллюстрации, 3 таблицы, 24 формулы, список используемой литературы, состоящий из 23 источников.

Выпускная квалификационная работа посвящена разработке автоматизированной обучающей системы по «Правилам дорожного движения» в средних общеобразовательных учреждениях.

Цель работы – разработка автоматизированной обучающей системы, которая позволит упростить процесс обучения учеников общеобразовательных учебных заведений правилам дорожного движения.

Объект исследования – процесс обучения учеников общеобразовательных учебных заведений правилам дорожного движения.

Первая часть выпускной работы описывает характеристику предприятия, для которого разрабатывается программа. Приводится классификация обучающих программ. Рассматриваются теоретические аспекты о структуре и алгоритмах функционирования автоматизированных обучающих систем. Обосновывается выбор технологии проектирования. Проводится анализ существующих программ выявляются недостатки.

Вторая глава выпускной бакалаврской работы сосредоточена на разработке математической модели автоматизированной обучающей системы и на применении имитационного моделирование к процессу обучения.

Третья глава описывает выбор и обоснование использования программных средств для разработки программы. Поясняется функциональный код программы.

Данная работа актуальна, так как детский травматизм при несоблюдении правил дорожного движения является важной проблемой в наши дни. Ежегодно на дорогах России совершаются десятки тысяч дорожно-транспортных происшествий с участием детей и подростков. Знание и соблюдение правил дорожного движения поможет сформировать безопасное поведение детей на дорогах.

## **ABSTRACT**

The graduation project consist of an explanatory note on 57 pages, including 24 figures, 3 tables, 24 formulas, the list of 20 references including 23 foreign sources.

This diploma work is devoted to the development of an automated training system on the “Rules of the road” in secondary schools.

The goal of this graduation work is to develop of automated learning system which allows to simplify training activity to traffic regulations in educational institutions.

The object of the graduation work is the process of training activity of pupils educational institutions to traffic regulations.

The first part of the graduation work describes the characteristics of the enterprise which this program is developed for. Classification of learning programs is given. Theoretical aspects about structure and algorithms of functioning of automated learning system learning system are considered. The choice of design technology is justified. The analysis of the existing programs is carried out and deficiencies are revealed.

The second part of the graduation work is concentrated on development of the mathematical model of the automated raining system application imitating modeling to training process.

The third part describes the choice and justification of use of software for development of the program are described. The functional code of the program is explained.

This work is urgent because children's traumatism at non-compliance with traffic regulations is an important problem today. Annually on roads of Russia tens of thousands of the road accidents with participation of children and teenagers are made. Knowledge and observance of traffic regulations will help to create safe behavior of children on roads.

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
Глава 1 АНАЛИЗ ПРОЦЕССА ОБУЧЕНИЯ ШКОЛЬНИКОВ ПРАВИЛАМ ДОРОЖНОГО ДВИЖЕНИЯ.....	4
1.1 Характеристика предприятия.....	4
1.2 Анализ технологий проектирования .....	7
1.3 Анализ существующей технологии обучения.....	9
1.4 Недостатки процесса обучения «КАК ЕСТЬ» и способы усовершенствования с помощью информационных технологий .....	12
1.5 Аналоги обучающих программ.....	15
Глава 2 РАЗРАБОТКА ПРОЕКТНЫХ РЕШЕНИЙ АВТОМАТИЗИРОВАННОЙ ОБУЧАЮЩЕЙ СИСТЕМЫ.....	19
2.1 Классификация обучающих программ .....	19
2.2 Выбор алгоритма функционирования.....	24
2.3 Имитационное моделирование .....	27
2.4 Математическое описание движения агентов.....	34
Глава 3 РЕАЛИЗАЦИЯ КОМПОНЕНТОВ АВТОМАТИЗИРУЕМОЙ ОБУЧАЮЩЕЙ СИСТЕМЫ.....	40
3.1 Выбор языка программирования .....	40
3.2 Конечный автомат. Диаграмма состояний .....	41
3.3 Программная реализация тренажеров.....	44
ЗАКЛЮЧЕНИЕ .....	49
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ .....	50
ПРИЛОЖЕНИЕ А Элементы интерфейса программы .....	53

## **ВВЕДЕНИЕ**

В связи с ежегодными многочисленными дорожно-транспортными происшествиями возрастает количество погибших и раненых. В том числе среди жертв значительный процент составляют дети. Поэтому возникает необходимость обеспечения и организации безопасности дорожного движения с участием несовершеннолетних.

В большинстве случаев это происходит из-за незнания и несоблюдения правил дорожного движения.

Дети – это самые уязвимые участники дорожного движения, поэтому знание ими правил дорожного движения и их соблюдение имеет огромное значение.

Проведение контроля знаний и достижения результатов обучения в данной области с использованием современных средств информационных технологий по сравнению с другими методами контроля имеет ряд преимуществ. При этом удобство выражается в возможности сравнения знаний одних учащихся с другими, и отслеживании динамики усвоения материала в процессе обучения.

Таким образом, актуальность темы выпускной квалификационной работы (ВКР) обусловлена необходимостью разработки автоматизированной обучающей системы по правилам дорожного движения для учеников общеобразовательных учебных заведений.

**Объект исследования** ВКР – процесс обучения учеников общеобразовательных учебных заведений правилам дорожного движения.

**Предмет исследования** ВКР - автоматизированная обучающая система по правилам дорожного движения.

**Цель** работы – разработка автоматизированной обучающей системы, которая позволяет упростить процесс обучения учеников общеобразовательных учебных заведений правилам дорожного движения.

# **Глава 1 АНАЛИЗ ПРОЦЕССА ОБУЧЕНИЯ ШКОЛЬНИКОВ ПРАВИЛАМ ДОРОЖНОГО ДВИЖЕНИЯ**

## **1.1 Характеристика предприятия**

Государственное бюджетное общеобразовательное учреждение Самарской области средняя общеобразовательная школа с. Ягодное муниципального района Ставропольский Самарской области создано в соответствии с постановлением Правительства Самарской области от 12.10.2011 N 576.

Управление Учреждением осуществляется в соответствии с Законом Российской Федерации «Об образовании», Типовым положением об общеобразовательном Учреждении в Российской Федерации и Уставом ГБОУ СОШ с. Ягодное на принципах демократии, гуманизма, общедоступности, приоритета общечеловеческих ценностей, жизни и здоровья человека, гражданственности, свободного развития личности, автономности и светского характера образования, на основе сочетания принципов самоуправления коллектива и единоначалия.

Руководство Учреждением осуществляет директор, прошедший соответствующую аттестацию и назначаемый на должность министерством образования и науки Самарской области. Директор решает все вопросы деятельности Учреждения, не входящие в компетенцию органов самоуправления Учреждения и Учредителей.

Формы самоуправления Учреждением:

- общее собрание трудового коллектива;
- управляющий совет;
- педагогический совет;
- органы ученического самоуправления.

В самом общем виде единое информационное пространство образовательного учреждения представляет собой систему, в которой

задействованы и на информационном уровне связаны между собой все участники учебного процесса.

Участниками информационного пространства, непосредственно вовлекаемыми в процесс создания и функционирования единого информационного пространства образовательного учреждения, являются следующие устойчивые группы:

- администрация;
- учителя;
- ученики;
- родители.

Схема информационных взаимосвязей этих групп изображена на рисунке 1.1.

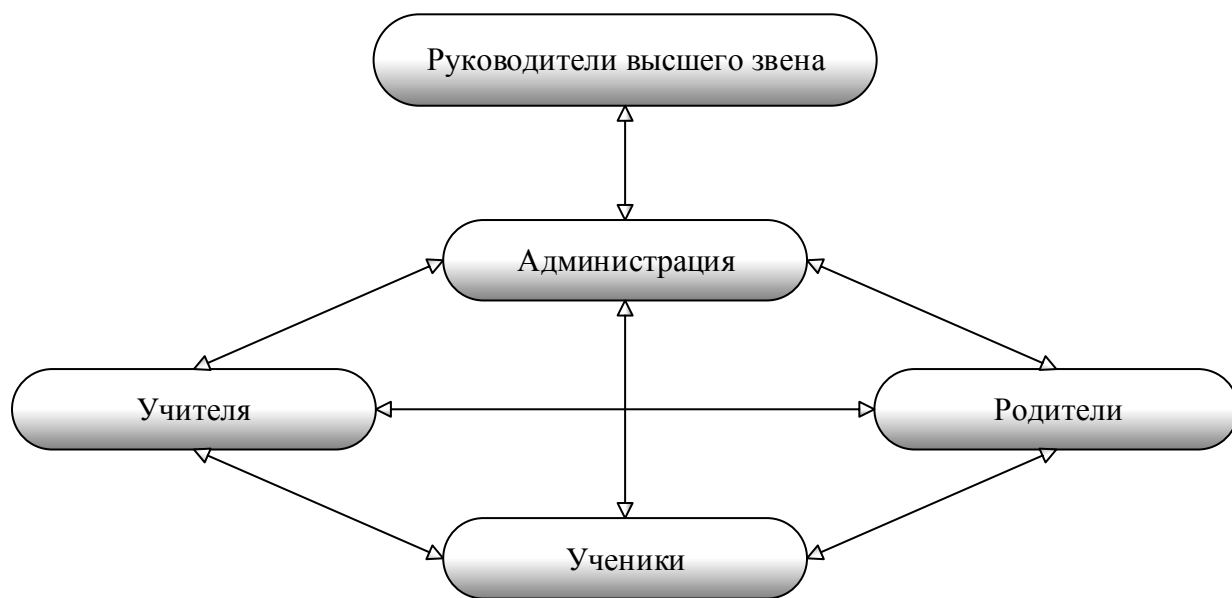


Рисунок 1.1 - Схема информационных связей

С точки зрения функционирующих информационных потоков дополнительным звеном в данной схеме может являться группа с условным названием «Руководители высшего уровня», которая включает в себя все основные внешние контакты образовательного учреждения: органы управления образованием, местные органы власти, общественные организации и т.д.

Анализ основных направлений работы образовательного учреждения и задач, им решаемых, позволяет разделить основные производственные процессы учреждения на три большие группы:

1) планирование, организация и оперативное управление учебным процессом как базовым производственным процессом образовательного учреждения;

2) административное управление функционированием образовательного учреждения и обеспечением образовательного процесса с соблюдением всех необходимых внешних и внутренних форм отчетности;

3) организация и обеспечение содержания образовательного процесса.

Следовательно, можно сделать вывод о том, что информационные связи основных групп участников информационного пространства представляют собой информационные потоки, которые делятся на три основные группы, реально отражающие базовые производственные процессы образовательного учреждения. На рисунке 1.2 представлена структура информационного пространства школы.

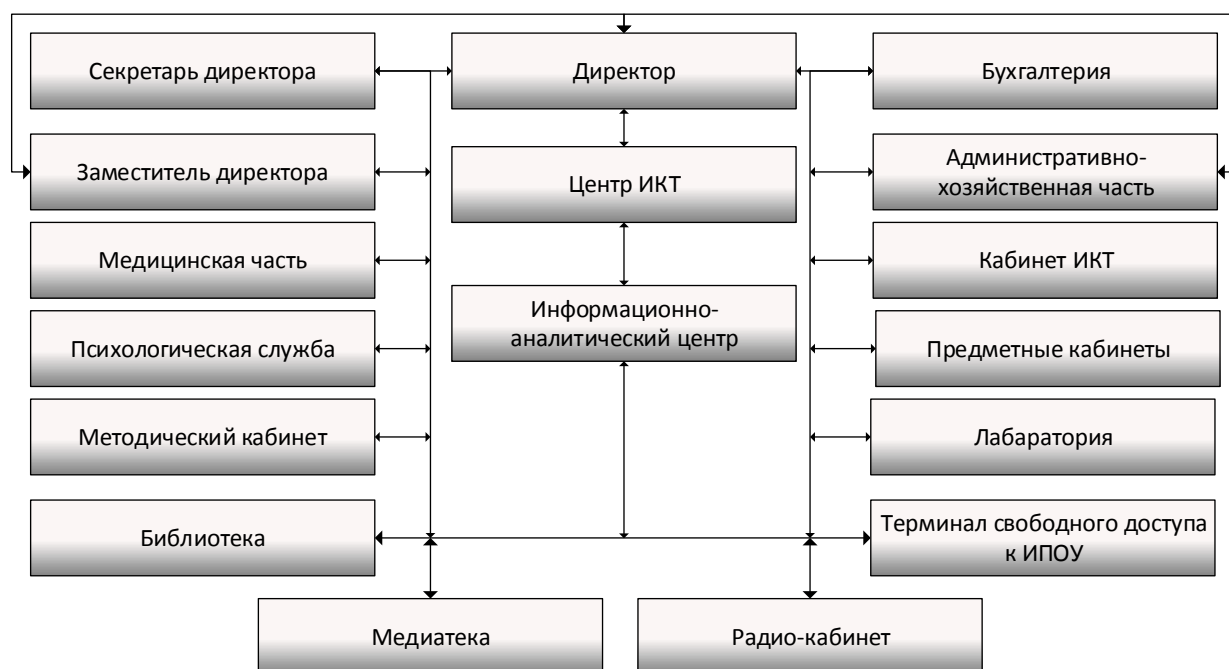


Рисунок 1.2 - Структура информационного пространства школы



Исходя из основных групп информационных потоков, можно предположить, что в состав единого информационного пространства образовательного учреждения входят:

- 1) информационная система организации, планирования, и управления учебным процессом;
- 2) информационная система администрирования деятельности образовательного учреждения;
- 3) информационная система обеспечения содержания учебного процесса цифровыми образовательными ресурсами.

До окончательного формирования структуры единого информационного пространства основные системы должны быть дополнены информационными модулями обеспечения образовательного процесса. Состав информационных модулей в случае конкретных учреждений может различаться.

Министерство образования в Ставропольском районе Самарской области распорядилось о том, чтобы в региональных общеобразовательных учреждениях проводились внеклассные мероприятия по контролю знаний по правилам дорожного движения у 4, 7 и 10 классов, которые ежегодно проходят соответствующее тестирование. В связи с этим необходимо автоматизировать процесс обучения по теме «Правила дорожного движения», с помощью которого будет осуществляться быстрый контроль знаний учащихся в данной области и подготовка к ежегодным тестированиям от ГИБДД.

## **1.2 Анализ технологий проектирования**

Перед началом автоматизации обучающего процесса, необходимо определить методологию проектирования. Для содержательного описания систем можно использовать графический язык и процесс SADT-моделирования. SADT (Structured Analysis and Design Technique) – это методология структурного анализа и проектирования, разработанная для облегчения понимания искусственных систем средней сложности, которая была создана в 1970-х годах. SADT использовали для проектирования телефонных

коммуникаций реального времени, создания программного обеспечения для управляющих и командных систем, автоматизации производства.

Для того, чтобы передать информацию о конкретной системе используют источник естественного языка – люди, и источник графического языка – методология SADT.

Описание систем с помощью данной технологии называется моделью. SADT модель может быть сосредоточена на функциях системы или на её объектах [4].

Существует широкий диапазон методов для схематического изображения процессов, таких как блок-схемы и диаграммы потоков данных, но они не являются строго определённым стандартом.

IDEF0 - стандарт для определения процесса. С конца 1960-х используется многими организациями для простого способа описания процессов.

IDEF0 во многих отношениях достаточно простой метод. Пример этому может служить то, что существует только один тип, включающий эту методологию. Каждое поле представляет собой единственный процесс, как и другие подходы, но IDEF0 отличается в использовании и размещении стрелок, а также стандартными вводами и выводами. Есть два других типа стрелок, которые представляют «средства управления» и «механизмы».

Средства управления - форма ввода, использующаяся для направления действия в процессе. Бывает, что существует степень неопределенности относительно того, является ли элемент вводом или управлением. Простой способ отличить их состоит в том, что вводы преобразованы или изменены в некотором роде для создания выводов, в то время как средства управления изменяются редко. Стандарты, планы, шаблоны и контрольные списки – это все формы управления.

Механизмы – это ресурсы и инструменты, которые требуются для завершения процессов, например, людей с определенными навыками, машины и другие инструменты.



На диаграмме представлены следующие действующие лица (актеры):

- 1) ученик – лицо, обучаемое правилам дорожного движения;
- 2) учитель – лицо, проводящее лекционные занятия, составляет учебные материалы, отслеживает результаты учеников.

В таблице 1.1 подробнее описаны прецеденты диаграммы.

Таблица 1.1 Прецеденты диаграммы вариантов использования

Прецедент	Характеристика
Посещение занятий	Ученик необходим присутствовать на занятиях, проводимых учителем.
Усвоение теоретического материала	Ученик должен понимать материал и входе урока выполнять задания.
Выполнение практических упражнений	Основываясь на теоретические знания выполнять задания.
Подготовка учебного материала	Учитель перед тем, как провести занятие должен подготовить учебный материал, основываясь на последнее издание официальных правил ПДД.
Проведение занятий	Учитель должен обучить учеников, проконтролировать усвоение материала всеми учащимися.
Контроль знаний учеников	Учитель проводит соответствующие тестирование, контрольные для проверки знаний учеников.

Используя вышеописанную диаграмму вариантов можно построить логическую модель данных автоматизированной обучающей системы.

Главная контекстная диаграмма обучающей системы разработана на основе нотации IDEF0. Данная диаграмма показывает входные и выходные ресурсы, правила управления и механизм управления [12].

Контекстная диаграмма процесса обучения правилам дорожного движения представлена на рисунке 1.4.



Рисунок 1.4 – Контекстная диаграмма процесса обучения правилам дорожного движения с применением технологии IDEF0 (0-й уровень)

Входными данными являются цели и задачи обучения: создать условия для формирования у учеников навыков соблюдения правил дорожного движения для сохранения жизни и здоровья.

В качестве механизма используется учитель, читающий теоретический материал для ученика, который является участником процесса обучения.

В управлении задействованы учебный материал и критерий оценивания знаний ученика. Учебный материал включает в себя:

- тематические занятия;
- игровые уроки;
- практические занятия;
- тестирование.

Выходными данными служит оценивание знаний ученика [8].

В декомпозиции контекстной диаграммы (рисунок 1.5) разбиваются процессы на следующие подпроцессы:

- введение в правила дорожного движения;
- проведение лекционных занятий;
- проведение практических занятий;
- проверка знаний ученика.

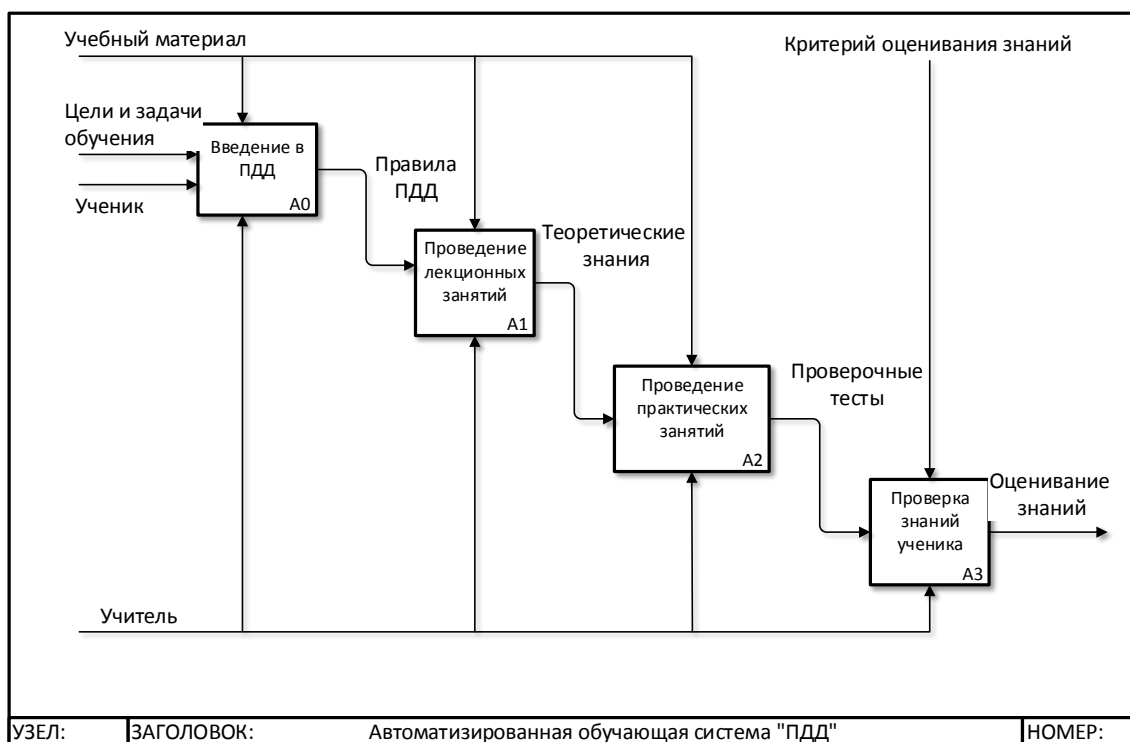


Рисунок 1.5 – Декомпозиция контекстной диаграммы процесса обучения ПДД с применением технологии IDEF0 (1-й уровень)

Таким образом, была разработана модель существующего процесса обучения правилам дорожного движения в среднем общеобразовательном учреждении.

#### 1.4 Недостатки процесса обучения «КАК ЕСТЬ» и способы его усовершенствования с помощью информационных технологий

Исходя из анализа технологии «КАК ЕСТЬ», выявим следующие недостатки существующего процесса обучения:

- недостаточная интерактивность процесса обучения;
- отсутствие наглядного представления дорожной ситуации;
- возможность не усвоения материала.

Опираясь на выявленные недостатки построим контекстную диаграмму усовершенствованного процесса обучения с использованием автоматизированной обучающей системы (рисунок 1.6).

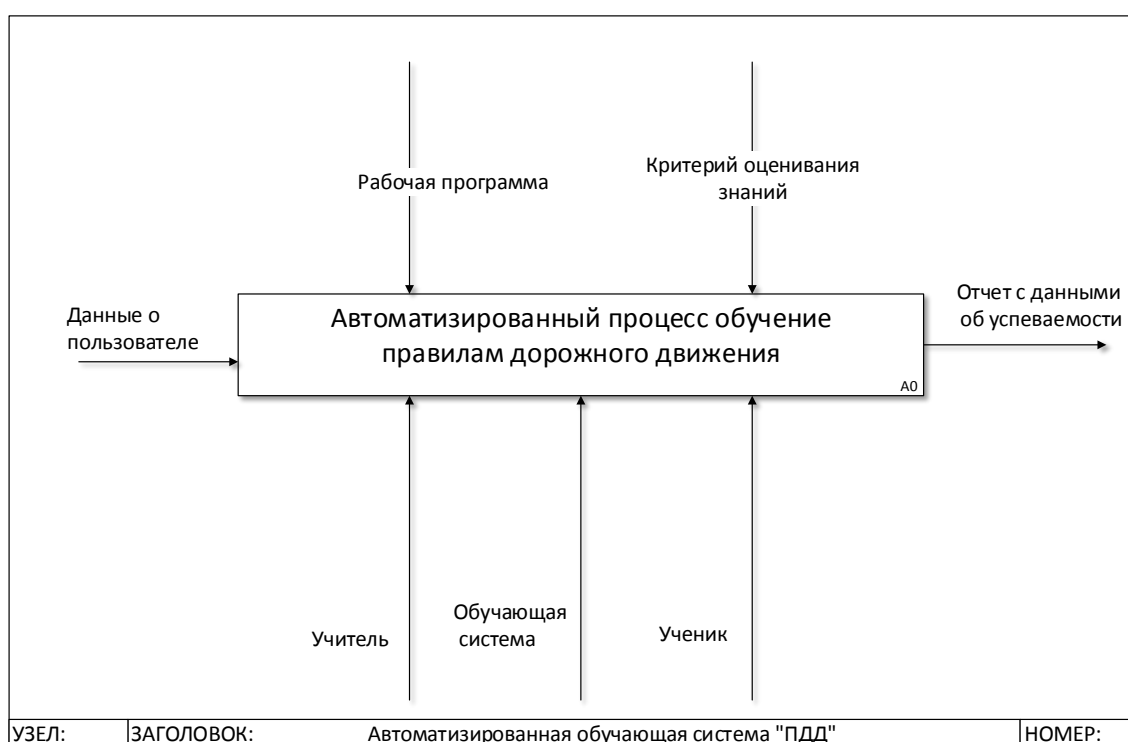


Рисунок 1.6 – Контекстная диаграмма автоматизированного процесса обучения ПДД с применением технологии IDEF0 (0-й уровень)

Сравнив диаграммы на рисунках 1.4 и 1.6, можно увидеть, что механизмами процесса являются учитель и обучающая система.

Исходя из этого, представим декомпозицию процесса «Автоматизация процесса обучения по ПДД» (рисунок 1.7).

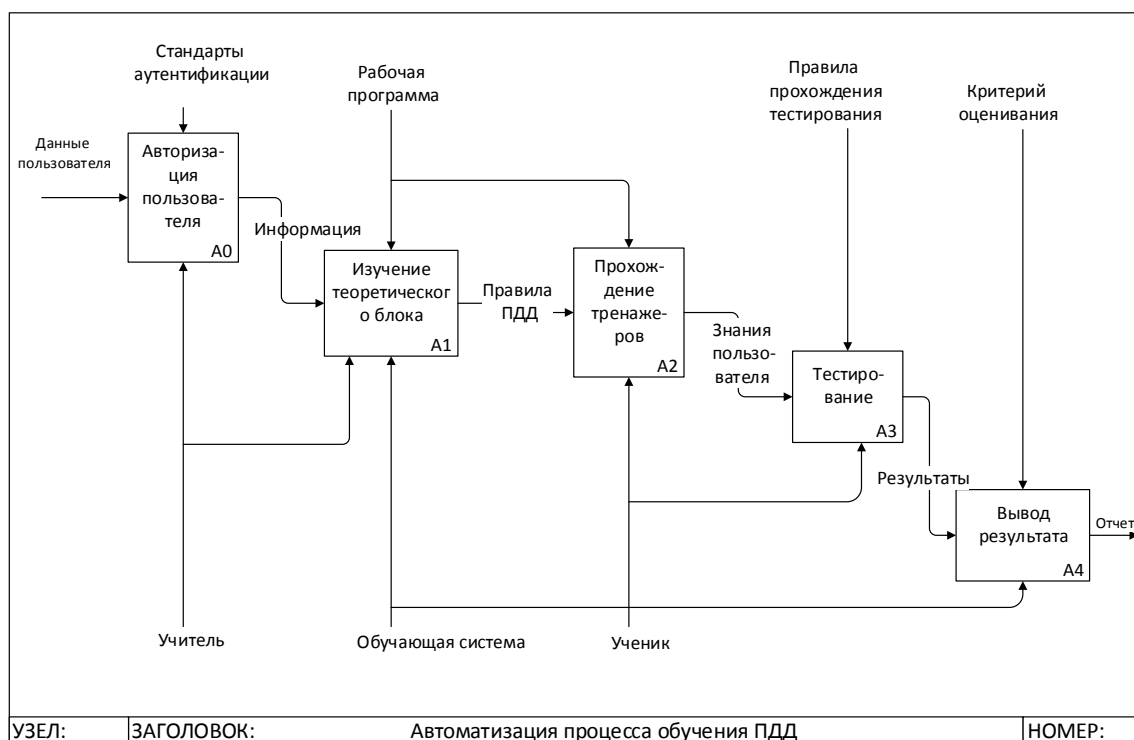


Рисунок 1.7 – Декомпозиция контекстной диаграммы автоматизированного процесса обучения ПДД с применением технологии IDEF0 (1-й уровень)

Опишем подробнее процессы с данной диаграммы:

1. «Авторизация пользователя» - пользователь, используя свой логин и пароль, входит в систему.
2. «Изучение теоретического блока» - пользователь выбирает блок с теоретическим материалом и изучает его. Также учитель может оказывать помощь, объясняя непонятные ученику аспекты.
3. «Прохождение тренажера» - ученик, после изучения теории, отрабатывает навыки в имитационной модели – тренажере.
4. «Тестирование» - ученик проходит тестирование по пройденному материалу.
5. «Вывод результатов» - на экран выводится балл за пройденное тестирование.



Учитель занимается подготовкой теоретического материала, вопросов для тестирования, помогает разобраться в непонятных ситуациях.

Ученик изучает материал, обучается на тренажере и проходит тестирование.

Можно сделать вывод, что применение в процессе обучения информационных технологий значительно эффективнее неавтоматизированного процесса.

Использование обучающей системы позволит изучить сложные ситуации на дороге без риска для жизни учеников.

### **1.5 Аналоги обучающих программ**

Перед тем, как разрабатывать новую технологию, рассмотрим существующие аналоги обучающих систем и выявим их положительные стороны и недостатки. Исходя из этого, сделаем выводы о необходимости разработки нового продукта.

Для оценки программ опишем критерии, по которым будем оценивать эффективность обучения правилам дорожного движения в средних общеобразовательных учреждениях:

- 1) программа должна предоставлять теоретический материал по заданной теме;
- 2) теоретический материал должен быть доступен и понятен для учеников средней школы;
- 3) наличие раздела с имитацией дорожной ситуации;
- 4) тестирование по пройденному материалу;
- 5) доступная стоимость продукта.

Итак, рассмотрим программы для обучения ПДД в школах, найденные в сети Internet:

1. «Мультимедийная обучающая программа по правилам дорожного движения» (рисунок 1.8). Данная программа предназначена для изучения

теоретического обучения мастерству вождения и Правил дорожного движения [15].



Рисунок 1.8 – Скриншоты программы «Мультимедийная обучающая программа по правилам дорожного движения»

Данная программа позволяет:

- использовать компьютерные обучающие и тестирующие методики по контролю и оценок уровня знаний обучаемых;
- применение учебных видеофрагментов в комплексе средств обучения;
- повысить уровень индивидуализации обучения.

Категория обучаемых: лица, достигшие возраста 14 лет.

Доступность: есть платная и бесплатная версия.

2. «1С: Образовательная коллекция. ПДД для школьников. Образовательная программа» (рисунок 1.9) [11].

Данная программа снабжена интуитивно понятным интерфейсом. Дополнительные изображения позволяют наглядно объяснить и напомнить рассматриваемый материал перед началом занятия. Программа генерирует задания, а после ввода ответа дает оценку знаниям.

Доступность: платная.

Возрастные ограничения: 6+.



Рисунок 1.9 – Скриншот программы «1С: Образовательная коллекция. ПДД для школьников»

### 3. «Правила дорожного движения для школьников» (рисунок 1.10).

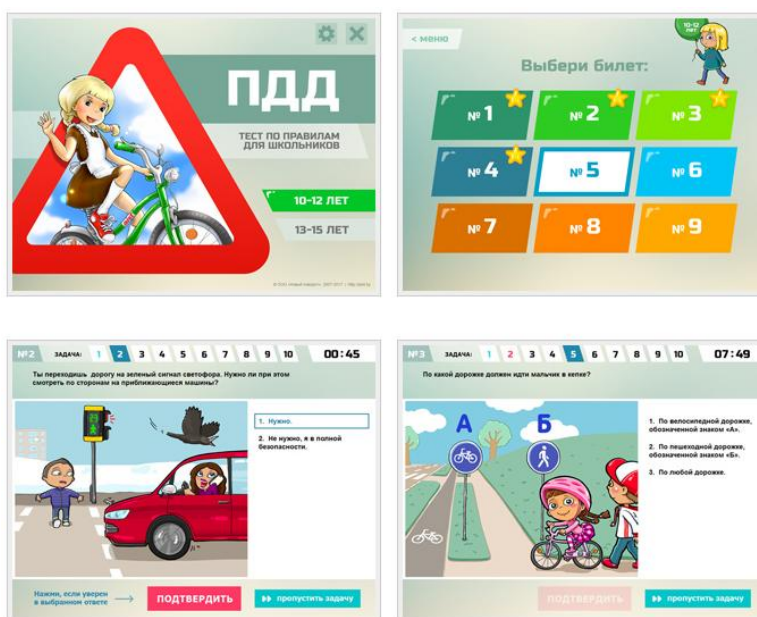


Рисунок 1.10 – Скриншоты программы «Правила дорожного движения для школьников»

Данная программа предназначена для контроля знаний по ПДД [16].

Возрастные ограничения: 10-12 и 13-15 лет.

Доступность: платная.

Произведем оценку аналогов обучающих программ по ПДД (таблица 1.2).

Оценивание будет происходить по следующим критериям:

– теоретический материал;

- доступность материала;
- имитация дорожной ситуации;
- тестирование;
- стоимость.

Программу будем считать эффективной, если по итогам оценивания она достигнет 5 баллов.

Таблица 1.2 Оценка эффективности программных аналогов

Номер программы \ Критерии	Программа 1	Программа 2	Программа 3
Теоретический материал	+	+	+
Доступность материала	-	+	+
Имитация дорожной ситуации	-	-	-
Тестирование	+	+	-
Стоимость	+	-	-
Итог	3	3	2

Исходя из итогов оценивания программ, делаем вывод, что ни один из представленных аналогов не подходит для поставленной задачи. Необходимо разработать новую технологию для обучения правилам дорожного движения в средней школе, удовлетворяющую заданным требованиям.

### **Выводы по первой главе**

Приведена характеристика предприятия, для которого разрабатывается обучающая система. С помощью методологии структурного анализа и проектирования был проведен анализ существующего процесса обучения и его недостатки. Были рассмотрены аналоги программного обеспечения и сделаны выводы о том, что необходима новая системы для поддержки процесса обучения правилам дорожного движения.

# Глава 2 РАЗРАБОТКА ПРОЕКТНЫХ РЕШЕНИЙ АВТОМАТИЗИРОВАННОЙ ОБУЧАЮЩЕЙ СИСТЕМЫ

## 2.1 Классификация обучающих программ

Существует множество подходов к классификации автоматизированных обучающих программ, поэтому рассмотрим две из основных классификаций:

1. Основанная на целях и задачах программ по обучению и режимах использования АОС:

- обучающие;
- операционные;
- тренажеры;
- иллюстрирующие;
- консультирующие.

2. Основанная на обобщении и анализе различных классификаций: имитационные, моделирующие, наставнические, игровые, тренировочные, проблемного обучения.

По форме обучения программы подразделяются:

- электронные учебники (ЭУ);
- тестирующие программы;
- автоматизированные обучающие системы (АОС).

Электронный учебник (ЭУ) – программное средство педагогического назначения для предъявления новой информации, дополняющей печатные издания, которые служат для индивидуализированного обучения и позволяют тестировать полученные навыки и умения обучаемого.

ЭУ позволяет внести изменения в содержание и структуру, которые требуются для конкретного учебного заведения. Поэтому электронный учебник может быть открытой или частично открытой системой. Чтобы не нарушить общую структуру и содержание ЭУ, модификации должны быть разрешены

только опытному преподавателю. Для ограничения доступа применяется пароль.

К элементам ЭУ можно отнести следующее:

- 1) значительно небольшое количество текстовой информации;
- 2) максимум иллюстративного материала;
- 3) мультимедийные файлы;
- 4) гиперссылки по содержанию ЭУ;
- 5) возможность делать закладки в нужном месте;
- 6) список литературы, которая содержится на серверах учебного заведения или в сети Internet.

Электронные учебники с мультимедийными и интерактивными возможностями способствуют лучшему усвоению большого объема информации и сложный материал, возможность по запросу обучаемого предоставить нужный материал, все это приближенно к обучению с преподавателем.

Автоматизированные обучающие системы (АОС) призваны заменить преподавателя, заменяя рутинную деятельность педагогических операций. АОС охватывают различные предметные области и решают задачи обучения на любых этапах человеческой жизни – от начальных классов школы до обучения в высших учебных заведениях.

АОС представляют собой программно-технические комплексы, которые включают учебную, методическую и организационную поддержку процесса обучения, проводящего на базе информационных технологий.

АОС могут решать следующие задачи:

- задачи, связанные со статистическим анализом показателей усвоения нового учебного материала: определить время на решение задач, определить общее число ошибок. К этой задаче относится и задача управления учебной деятельностью;

- задачи, которые связаны с контролем и проверкой уровня знаний и умений учащихся до и после обучения, их индивидуальных способностей;
- задачи, связанные с подготовкой учебного материала и адаптацией его по уровням сложности, подготовкой динамических иллюстраций, лабораторных работ, самостоятельных работ, контрольных заданий учащихся;
- задачи обратной связи с обучаемым.

Отличия АОС от других обучающих программ:

- интерактивный диалог между обучающей системой и пользователем;
- психологические и дидактические аспекты взаимодействия «человек-компьютер»;
- сочетание учебно-методических, программных, технических и организационных обеспечений на базе средств, предназначенных для индивидуализации процесса обучения, и компьютерных технологий.

Можно сделать вывод, что электронные учебники и другие тестирующие программы, проверяющие знания и выдающие информацию, представляются как «ограниченные АОС», которые выполняют «узкую» функцию. Автоматизированные обучающие системы включают в себя функции ЭУ и тестирующих программ, тем самым повышая эффективность обучения.

### 2.1.1 Структура автоматизированной обучающей системы

У каждой АОС есть определенная структура на основе элементов с указанием связей между ними и представлением о системе в целом.

Исходя из этого, структуру системы можно охарактеризовать по имеющимся в ней типам связей.

По структурным признакам обучающей системы с пользователем подразделяются на разомкнутые (без обратной связи) и замкнутые (с обратной

связью) системы, отличающиеся принципиальным подходом к обучению (рисунок 2.1).

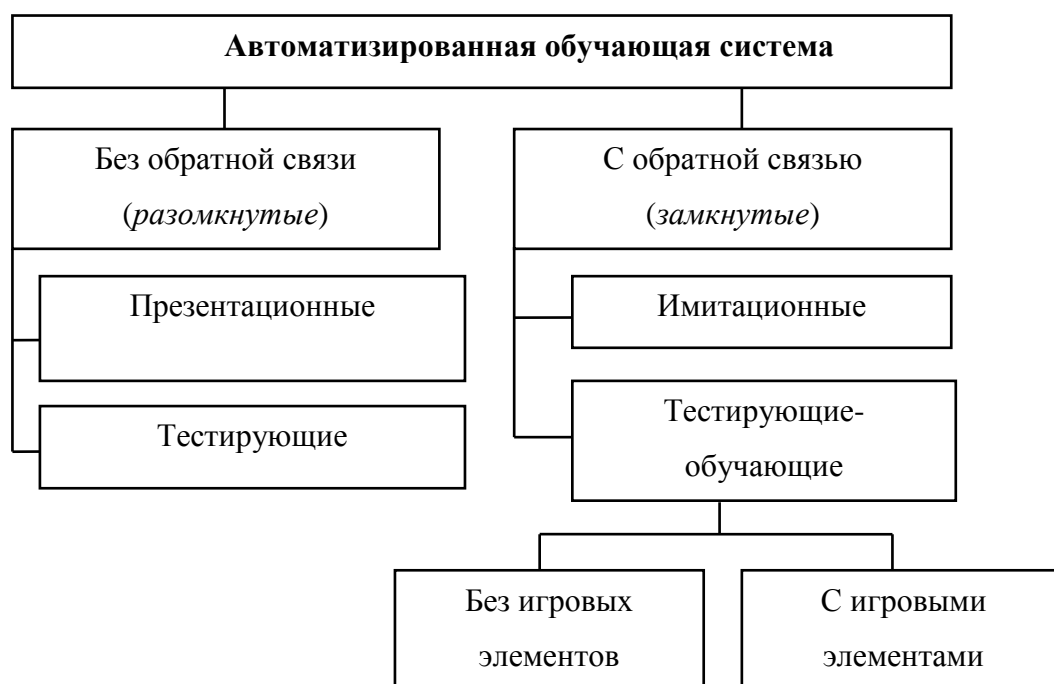


Рисунок 2.1 - Классификация структурного построения АОС

В разомкнутых АОС не ведется учет откликов пользователя на поставленные вопросы и не изменяется последовательность предъявления учебного материала. Такой принцип выполняет лишь определенную последовательность урока, заданную программным путем. Наиболее простые АОС из числа разомкнутых являются системы с презентационной структурой – последовательное включение звеньев «АОС» и «Учащийся».

В автоматизированной обучающей системе данного типа отражается только прямая связь между учащимся и информационной системой. Представляется визуальная информация с монитора. В данном случае учащийся выступает в роли пассивного наблюдателя, от которого не требуются отклики по взаимодействию с АОС.

В тестирующих АОС без обратной связи делается упор на уровень знаний в определенный период обучения. Здесь используются различные методики, такие как предъявление обучаемому закрытый или открытый вариант вопроса.



От пользователя программа ждет отклик в виде ответа на вопрос. Ответ сохраняется в блоке фиксатора ошибок и по результатам тестирования выставляется соответствующая оценка, которая служит критерием по степени усвоения учебного материала.

Замкнутые АОС с обратной связью между учащимся и системой обладают наиболее широкими функциональными возможностями и эффективностью в процессе обучения. Взаимодействие учащегося с АОС можно представить в качестве системы с внешней обратной связью, где задача АОС - это повышение уровня знаний ученика и уменьшение его количества ошибок. Здесь АОС выступает как звено регулирования прямого канала, где объектом выступает «Учащийся».

В зависимости от того, какой характер воздействия несет за собой АОС, ученик принимает определенное решение, которое доказывает факт усвоения им учебного материала, и генерирует его на вход ЭВМ. Если рассматривать систему «АОС – учащийся» в виде некоторой линейной системы, то реакцию учащегося на воздействие АОС можно представить в виде некоторой функции уровня количества ошибок, в зависимости от задания. Задание – совокупность задач, представленные учеником для их решения. Вид такой функции зависит от индивидуальных свойств учащегося и программы. Анализ вида функции помогает принять решение по оценке уровня подготовки ученика, знания учебного материала и готовность к учебному процессу, с одной стороны, а с другой, эффективность автоматизированной обучающей системы.

В первую очередь в АОС необходимо установить допустимое время на принятие решения учащимся на заданный вопрос.

Имитационные АОС являются наиболее распространенными типами среди замкнутых автоматизированных обучающих систем. Здесь функцию ведущего «элемента» выполняет модель или группа моделей реальной ситуации в определённой сфере предметной области. Реакция ученика на предъявляемый учебный материал выступает в роли обратной связи и является

основой непрерывного взаимодействия «АОС – обучаемый». Любое воздействие ученика на систему ведет к ответной реакции со стороны АОС.

Примерами таких систем могут случить различные игровые тренажеры, имитаторы и т.п.

Тестирующие и обучающие АОС используют комплексный подход в обучении. Такая программа помимо того, что обучает учащегося, она проверяет полученные знания на данный момент обучения. В таком случае важным фактором будет являться отклик ученика на то или иное информационное воздействие. В зависимости от отклика, АОС может перестраивать ход урока в том или ином направлении. В такой структуре часто используют игровые элементы [9].

Таким образом, очевидно, что наиболее широкий спектр возможностей имеют замкнутые обучающие системы, которые обеспечивают максимальную «гибкость» в общении с учащимся.

## **2.2 Выбор алгоритма функционирования**

Методика проведения обучающего процесса диктует определенные алгоритмические подходы при реализации любой из ранее рассмотренных структур автоматизированной обучающей системы. Любая обучающая система – это совокупность дидактических материалов (ДМ), которые используются в процессе обучения ученика. Благодаря современной вычислительной технике, обладающей широкими функциональными возможностями, предоставляется возможность использовать в ДМ материале информацию в виде текста, изображений и мультимедийные файлы. При этом можно использовать все возможные представления информации для повышения эффективности процесса обучения.

Классификация алгоритмов функционирования АОС представлена на рисунке 2.2.

Согласно методике, при линейном алгоритме АОС учащемуся предоставляются ДМ, заложенные в системе. Достоинством линейного

алгоритма АОС можно отметить простоту разработки системы. Недостатком может выступать трудоемкость раскрытия некоторых тем и негарантированное закрепления изложенного учебного материала учеником.

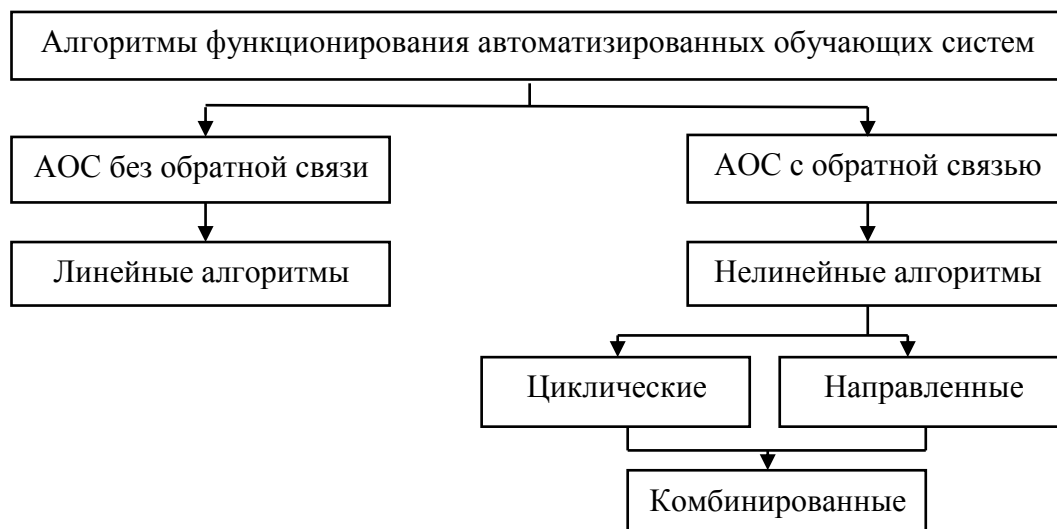


Рисунок 2.2 - Классификация алгоритмов функционирования АОС

Используя нелинейные алгоритмы в АОС появляется возможность менять последовательность предъявления дидактических материалов в зависимости от отклика учащегося на информационное воздействие. В таком случае важнейшую роль играют информационные блоки с вопросами, которые требуют принятия решения учеником. Такие информационные блоки назовем слайдами выбора, в которых используются следующие средства выбора направления обучения:

- закрытые вопросы или вопросы типа «меню»;
- открытые вопросы;
- гиперссылки.

Открытые вопросы состоят только из формулировки вопроса, где ответ должен ввести сам ученик. В качестве ответа может выступать числовое значение, оно может быть однозначным и находится в некотором заданном диапазоне.

Закрытые вопросы состоят из формулировки вопросов и нескольких вариантов ответа. Перед учеником ставится задача выбора одного или нескольких правильных вариантов ответов. Ситуация, когда все вопросы являются неверными недопустима.

Гиперссылки – это один из способов выбора последовательности слайдов, который осуществляет сам ученик. Используется при программированном обучении с использованием сети Internet. Определенная часть слайда сопоставляется с другим. При указании учеником на гиперссылку программа открывает соответствующий ей слайд.

Нелинейные алгоритмы подразделяются на циклические, направленные и комбинированные.

Циклические алгоритмы осуществляют возврат к слайдам с темами, которые недостаточно усвоил обучаемый. Если ученик неверно принимает решение задачи на слайде, то система может повторно предъявить слайды, показанные ранее, для повторного изучения темы. Направленные алгоритмы предполагают наличие слайдов выбора, но в зависимости от ответа учащегося выбирается определенная последовательность без возврата обратно.

Комбинированные алгоритмы используют оба принципа, описанные ранее. В зависимости от принятия решения учеником для ответа на поставленный вопрос, АОС меняет последовательность предъявления слайдов, но на n-ом шаге может осуществляться возврат к предыдущим слайдам.

Таким образом, каждая описанная структура позволяет предъявить учебный материал, обеспечивающий приемлемое представление, согласно требованиям предметной области.

На основе вышеописанного материала делаем вывод, что в качестве разрабатываемой программы, наиболее подходящей будет автоматизированная обучающая система с обратной связью.

### 2.3 Имитационное моделирование

Имитация (от лат. – подражание) – воспроизведение явлений, действий, событий, объектов определенным образом. Термин «имитация» в определенном смысле синоним понятия «модель», что значит мера, образец. Модель определяется как любой нематериальный или материальный образ, например, схема, изображение, описание, представитель и т.п.

Можно сказать, что словосочетание «имитационная модель» некорректно, однако в середине 20 века его ввели в практику математического и физического моделирования.

Имитационные модели – это особый класс математических моделей, и они значительно отличаются от аналитических из-за того, что использование ЭВМ играет важную роль в процессе их реализации. В имитационных моделях нет жестких ограничений в использовании исходных данных, есть возможность использовать всю информацию, собранную в процессе исследования, независимую от формы представления и степени ее формализации.

Имитационное моделирование – метод исследования, который основывается на замене изучаемой системы имитирующей. Далее с имитирующей системой проводят эксперименты и в результате получают информацию об изучаемой системе, не прибегая к экспериментам с реальными объектами. С помощью данного метода возможно имитировать бизнес-процессы так, как бы они проходили в действительности, учитывая графики рабочего времени и занятость ресурсов, а также необходимого количества материальных ресурсов. В итоге получаем возможность оценивать реальное время выполнения одного или заданного множества процессов.

Имитационная модель – описание объекта математической логикой, которое возможно использовать на компьютере в качестве экспериментирования в целях проектирования, анализа и оценки функционирования объекта [2].

### 2.3.1 Основные этапы имитационного моделирования и его виды

Структуру имитационного моделирования можно представить, как последовательно-циклическую, в которой последовательность определяется процессом имитационного моделирования, разбивающийся на ряд этапов, в которых выполнение осуществляется последовательно от предыдущего к последующему. Проявление цикличности отражается в необходимости возвращении к предыдущим этапам и повторении уже пройденного пути с измененными данными и параметрами модели (рисунок 2.3).

Первый этап любого исследования необходим для того, чтобы оценить потребность изучения объекта или проблемы, способы решения задачи, ожидаемые результаты. Данный этап важен для практического применения метода моделирования. Часто к этому этапу осуществляют возврат после окончания исследования модели и обработки результатов, чтобы изменить постановку задачи, либо цели моделирования.

Во втором этапе происходит формализация описания объекта моделирования, основывающаяся на выбранной теоретической базе. На естественном языке описывается состав исследуемого объекта, взаимодействие между элементами объекта и объекта с внешней средой. Выбирается концепция формального определения на основе описания объекта. Следовательно, в завершении этапа словесное описание исследуемой системы преобразуется в абстрактную математическую структуру.

На третьем этапе проводится исследование на разработанной модели, которое реализуется путем запуска ее на ЭВМ. Перед тем как начать исследование нужно составить последовательность запусков модели, которая позволит получить необходимый объем информации при заданных исходных данных. Затем на основе составленного плана эксперимента произвести запуск имитационной модели на ЭВМ. В конце этапа обрабатываются результаты и представляются в удобном виде для анализа.

На заключительном пятом этапе формулируются конечные выводы, происходит разработка рекомендации для применения результатов моделирования для достижения установленных целей. Нередко на основе данных выводов происходит возврат к началу процесса для того, чтобы произвести требуемые изменения в теоретической и практической части модели и повторить исследование с измененной моделью. В результате, после нескольких подобных циклов получается имитационная модель, удовлетворяющая поставленным задачам [5].



Рисунок 2.3 - Этапы имитационного моделирования при исследовании сложной проблемной ситуации

Следовательно, метод имитационной модели предполагает выполнение пяти этапов при исследовании сложной проблемной ситуации.

С помощью имитационных моделей можно проверить, правильно ли понимаются процессы в исследуемом объекте, позволяет выявить в конкретных случаях параметры порядка. Знание этого дает возможность построить простые модели сложных явлений [6].

Имитационное моделирование подразделяется на несколько видов (рисунок 2.4).

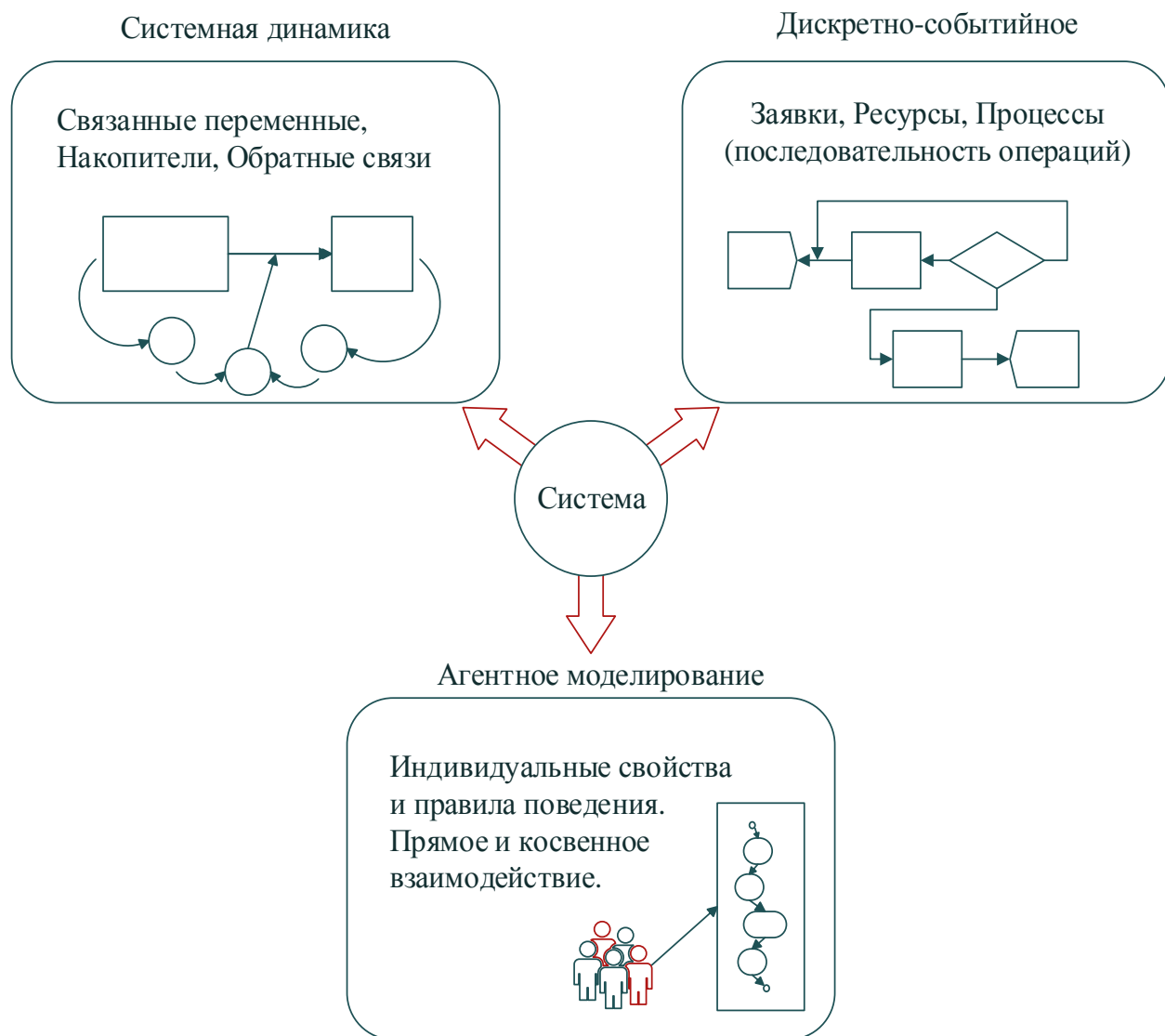


Рисунок 2.4 - Виды имитационного моделирования

Системная динамика – моделирование, основанное на построении графических диаграмм причинных связей и влияний одних параметров на другие для исследуемой системы. На основе этих моделей диаграмма имитируется на компьютере. Системная динамика намного лучше помогает



понять происходящие выявление причинно-следственных связей между явлениями и объектами, чем другие парадигмы моделирования.

Дискретно-событийное моделирование – это такой подход к моделированию, в котором предлагается рассматривать только основные события моделируемой системы и абстрагироваться от постоянной природы событий. Такой вид моделирования наиболее развит и имеет большой контингент приложений: производственные и транспортные системы, логистика, системы массового обслуживания. Дискретно-событийное моделирование в наибольшей степени соответствует моделированию различных производственных процессов. Оно было основано в 1960-х годах Джеффри Гордоном.

Агентное моделирование – одно из новых направлений в имитационном моделировании (1990 – 2000 гг.). Его используют для исследований децентрализованных систем, в которых динамика функционирования определяется не глобальными правилами и законами, а когда последние являются результатом активности членов группы. Целью таких моделей служит получение представления о частном, индивидуальном поведении отдельных активных объектов и взаимодействии их в системе.

В данной работе при разработке тренажера для автоматизированной обучающей системы по ПДД будет использоваться агентное моделирование.

### 2.3.2 Агентное моделирование

Агент – это сущность, которая обладает активностью, автономным поведением и может с помощью некоторого набора правил принимать решения, самостоятельно изменяться и взаимодействовать с окружением.

Агент способен получать информацию о других объектах, находящихся с ним в той же некоторой среде [20].

Выделяют следующие характеристики агентов:

- 1) реактивность – реагирование на изменения среды;

- 2) проактивность – проявление инициативы для достижения цели;
- 3) социальные навыки – взаимодействие с другими агентами для достижения целей.

Агентное моделирование значительно отличается от традиционного.

Основные отличия представлены в виде таблицы (таблица 2.1).

Таблица 2.1 Отличия агентного моделирования от аналитического

Агентное моделирование	Аналитическое моделирование
С помощью аналитических переменных характеризуется состояние агентов	В ходе решения систем уравнений получают значения аналитических переменных.
Агенты оказывают влияние друг на друга, взаимодействуют посредством своего поведения.	Аналитические переменные связываются между собой с помощью уравнений.
Аналитические переменные, характеризующие отдельную часть системы, определяют поведение агентов при моделировании.	Аналитические переменные сфокусированы, как правило, на общем описании системы, а не отдельных ее частей.
Единицей модели является агент с определенным поведением.	Единицей модели является уравнение с переменными.
Лучше подходят для моделирования бизнес-процессов, так как агенты включают модели дискретно-событийных взаимодействий.	Для моделирования физических процессов лучше подходят аналитические процессы.

Множество агентов, взаимодействующих со внешней средой, образуют многоагентную систему.

Многоагентная система (Multi-agent system) для решения сложных задач использует системы, состоящие из множества связанных между собой агентов.

Такая система относится к самоорганизующимся системам. Главным преимуществом многоагентной системы является то, что ее можно дополнить и модифицировать без переписывания частей программы [3].

В данной работе в качестве агентов рассматриваются автомобили, регулировщик и пешеход, а внешней средой агентов будет перекресток и регулируемый пешеходный переход.

Агент распознает текущее состояние  $s_t \in S$  на каждом своем шаге, после этого выбирает действие  $a_t \in A$ . Затем внешняя среда возвращает агенту награду  $r_t = r(s_t, a_t)$  и переводит его в другое состояние  $s_{t+1} = \delta(s_t, a_t)$ . Функции  $\delta$  и  $r$  неизвестны агенту, так как являются частью внешней среды. Получим, что  $S$  – множество состояний агента,  $A$  – действия, доступные агенту,  $t$  – дискретный шаг во времени.

Основная задача агента – найти оптимальную стратегию для выбора дальнейшего действия, относительно его состояния  $\pi: S \rightarrow A$ . Стратегия может считаться оптимальной, если:

$$V^\pi S_t = r_t + \gamma V^\pi S_{t+1} + \gamma^2 V^\pi S_{t+2} + \dots = \sum_{i=0}^{\infty} \gamma^i r_{t+i} \quad (1)$$

Формула (1) описывает максимальную прибыль агента, где  $0 \leq \gamma \leq 1$  – некая константа, которая позволяет учитывать поощрение от внешней среды в настоящий момент времени.

Рассмотрим подробнее работу «регулировщика». Для каждого варианта жестов, регулирующих движение на перекрестке, выполняется оценка организации движения по соответствующим направлениям:

Агент-автомобиль посылает запрос «регулировщику», предполагая направление движения, то есть прямо, налево или направо для соответствующих полос. Затем ожидает положительный или отрицательный ответ.

Каждый агент владеет информацией о количестве машин, который поедут в каждом направлении, благодаря статистике распределения движения.

Допустим, есть вероятности для проезда прямо, налево или направо –  $P_{\text{прямо}}$ ,  $P_{\text{п}}$ ,  $P_{\text{л}}$ . Соответственно они будут вычисляться следующим образом:

$$P_{\text{прямо}} = \frac{c_{\text{прямо}}}{c}, P_{\text{п}} = \frac{c_{\text{п}}}{c}, P_{\text{л}} = \frac{c_{\text{л}}}{c} \quad (2)$$

$c$  – общее количество машин за определенный промежуток времени  $T$ ;

$c_{\text{п}}$  – количество повернувших направо машин из общего количества  $c$ ;

$c_{\text{л}}$  – количество повернувших налево машин из общего количества  $c$ ;

$c_{\text{прямо}}$  – количество проехавших прямо машин из  $c$ .

Среднее время  $t$ , которое автомобили затратят на движение, вычисляется как отношение общего расстояния к средней скорости на всем участке. Если агент-регулировщик при получении запроса на смену позиции, то он будет удерживать ее в интервале  $(t + t_0 - \varepsilon, t + t_0 + \varepsilon)$ , где  $\varepsilon$  – заданная константа доверительного интервала, а  $t_0$  – точка отсчета и текущее время.

Время будет вычисляться как сумма всего времени ожидания на текущем участке и время ожидания на предыдущих шагах.

#### 2.4 Математическое описание движения агентов

Для реализации движения автомобилей по перекрестку будет использоваться геометрическое моделирование. Модель включает в себя алгоритмы реализации систем уравнений. Уравнения, описывающие форму и движение объектов, являются математической основой построения модели. Все разнообразие геометрических объектов является комбинацией различных примитивов – простейших фигур, которые, состоят из графических элементов – линий, точек и поверхностей [15].

Трансформация - это изменение некоторых графиков в нечто другое, применяя определенные правила. Существуют различные типы преобразований, таких как отражение, масштабирование вверх или вниз, поворот, сдвиг и т.д. Когда преобразование происходит на 2D плоскости, оно называется 2D преобразованием.

Трансформации играют важную роль в компьютерной графике, чтобы изменить графику на экране или изменить размер, или ориентацию [6].

Точки на плоскости задаются двумя координатами, поэтому каждая из точек задается значениями координат вектора. Такие координаты можно рассматривать как элементы матрицы  $[x, y]$  – как вектор-столбец или вектор-строка. Управление положением этих точек осуществляется путем преобразования матрицы.

Для переноса точек на плоскости в новые позиции используют добавление к ним константы переноса:

$$x^* \ y^* = x \ y + a \ b = x + a \ y + b \quad (3)$$

Для перемещения точек на плоскости нужно к матрице с координатами прибавить матрицу коэффициентов преобразования.

Рассмотрим умножение двух матриц. Первая определяет точку на плоскости, вторая является матрицей преобразований размерностью  $2 \times 2$ :

$$x \ y \cdot \begin{pmatrix} a & b \\ c & d \end{pmatrix} = ax + cy \quad bx + dy = [x^* \ y^*] \quad (4)$$

Рассмотрим  $x^*$  и  $y^*$  как преобразованные координаты и проведем анализ результатов. Исследуем частные случаи, когда  $a=d=1$  и  $c=b=0$ . Такая матрица преобразований приведет к идентичной исходной:

$$x \ y \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = 1x + 0y \quad 0x + 1y = [x^* \ y^*] \quad (5)$$

Как видно, координаты точки остаются неизменными.

Теперь рассмотрим случай, когда  $d=1, b=c=0, a=const$ :

$$x \ y \cdot \begin{pmatrix} a & 0 \\ 0 & 1 \end{pmatrix} = ax + 0y \quad 0x + 1y = ax \ y = [x^* \ y^*] \quad (6)$$

Видно, что это приводит к изменению масштаба в направлении  $x$ , так как  $x^*=ax$ . Исходя из этого, матричное преобразование эквивалентно перемещению исходной точки в направлении  $x$ .

Если  $b=c=0$ , то:

$$x \ y \cdot \begin{pmatrix} a & 0 \\ 0 & d \end{pmatrix} = ax + 0y \quad 0x + dy = ax \ dy = [x^* \ y^*] \quad (7)$$

Результатом является изменение масштабов в направлении  $x$  и  $y$ . При  $a \neq b$  перемещения неодинаковы. При  $a = d > 1$  увеличивается масштаб исходной точки, а при  $0 < a = d < 1$  уменьшается.

При отрицательных  $a$  и (или)  $b$  отображаются координаты точек. Рассмотрим случай при  $b = c = 0, d = 1$  и  $a = -1$ :

$$\begin{pmatrix} x & y \end{pmatrix} \cdot \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} -1x + 0y & 0x + 1y \end{pmatrix} = \begin{pmatrix} -x & y \end{pmatrix} = [x^* \ y^*] \quad (8)$$

Видно, что отображение точки происходит относительно оси  $y$ . А при  $b = c = 0, d = -1$  и  $a = 1$ , отображение будет относительно оси  $x$ . При отрицательном значении точек  $a$  и  $d$  отражение будет относительно начала координат.

Исходя из этого, можно заметить, что изменение масштаба отображение происходит из-за диагональных элементов матрицы преобразования.

В случае, когда  $a = d = 1$ , а  $c = 0$ , будет следующее:

$$\begin{pmatrix} x & y \end{pmatrix} \cdot \begin{pmatrix} 1 & b \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} x & (bx + y) \end{pmatrix} = [x^* \ y^*] \quad (9)$$

Координата  $x$   $y$  точки остается неизменной, а  $y^*$  линейно зависит от начальных координат. Такой эффект называют сдвигом. Аналогичный эффект при  $a = d = 1$  и  $b = 0$ .

При преобразовании различных геометрических фигур в итоге будет следующее: параллельные прямые преобразуются в параллельные прямые, параллелограмм – в параллелограмм, середина отрезка – в середину отрезка.

Можно получить матрицу  $2 \times 2$ , осуществляющую вращение фигуры относительно начала координат, если рассмотреть вращения единичного квадрата вокруг начала координат (рисунок 2.5).

Точка  $B$  с координатами  $(1,0)$  преобразуется в точку  $B^*$ , для которой  $x^* = (1)\cos\theta$  и  $y^* = (1)\sin\theta$ . Точка  $D$  с координатами  $(0,1)$  преобразуется в точку  $D^*$  с координатами  $x^* = (-1)\sin\theta$  и  $y^* = (1)\cos\theta$ .

Запишем матрицу преобразования общего вида:

$$\begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \quad (10)$$

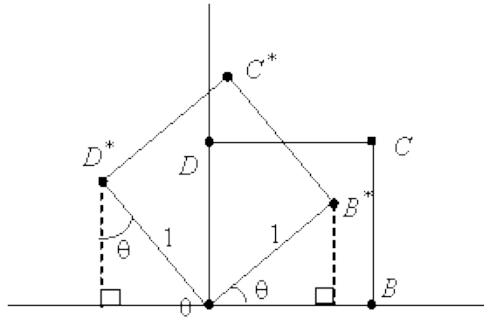


Рисунок 2.5 – Вращение единичного квадрата

Частные случаи:

1) для поворота на  $90^\circ$  используется матрица преобразования:

$$\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \quad (11)$$

2) поворот на  $180^\circ$  можно осуществить с помощью матрицы:

$$\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \quad (12)$$

Если вращение в плоскости осуществляется вокруг перпендикулярной к ней оси, то отображение происходит за счет поворота на  $180^\circ$  вокруг лежащей в плоскости оси.

Вращение вокруг линии  $y = x$  осуществляется матрицей:

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (13)$$

Вращение вокруг линии  $y = 0$  осуществляется матрицей:

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (14)$$

В отличие от поворота и масштабирования, перенос реализуется с помощью сложения, так как ввести константы переноса внутрь матрицы  $2 \times 2$  невозможно. Желательно представить преобразования в единой форме, то есть с помощью умножения матриц. Поэтому вводится третья компонента в вектор  $x$   $y$  и  $[x^* \ y^*]$ , представляя их в виде  $x \ y \ 1$  и  $[x^* \ y^* \ 1]$ . После этого матрица преобразования примет размер  $3 \times 2$ :

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ t & n \end{pmatrix} \quad (15)$$





$$\begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (20)$$

В данном случае, вращение происходит вокруг начала координат. Как говорилось ранее, нужно сначала перенести объект так, чтобы нужный центр оказался в начале координат. Такое действие осуществляется с помощью матрицы переноса:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -4 & -3 & 1 \end{pmatrix} \quad (21)$$

Далее применяется матрица вращения и приводятся результаты к началу координат с помощью обратной матрицы:

$$X \ Y \ Z = x \ y \ 1 \cdot \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ -4 & -3 & 1 & 0 & 0 & 1 & 4 & 3 & 1 \end{pmatrix} \quad (22)$$

Выполним умножения:

$$X \ Y \ Z = x \ y \ 1 \cdot \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 7 & -1 & 1 \end{pmatrix} \quad (23)$$

В результате получается:

$$x^* = \frac{X}{H}, \quad y^* = \frac{Y}{H} \quad (24)$$

Рассмотренные матричные преобразования будут использоваться для реализации движения автомобилей по перекрестку.

### Выводы по второй главе

Во второй главе были рассмотрены классификация обучающих систем.

В качестве основы была выбрана автоматизированная обучающая система с обратной связью на основе имитационного моделирования. Описаны метод агентного моделирования и математическая модель движения автомобилей по перекрестку с регулятором.

## Глава 3 РЕАЛИЗАЦИЯ КОМПОНЕНТОВ АВТОМАТИЗИРУЕМОЙ ОБУЧАЮЩЕЙ СИСТЕМЫ

### 3.1 Выбор языка программирования

JavaFX - это набор графических и мультимедийных пакетов, который позволяет разработчикам проектировать, создавать, тестировать, отлаживать и развертывать многофункциональные клиентские приложения, работающие последовательно на разных платформах.

Библиотека JavaFX написана как Java API, код приложения JavaFX может ссылаться на API из любой библиотеки Java. Например, приложения JavaFX могут использовать библиотеки Java API для доступа к собственным системным возможностям и подключения к серверным приложениям промежуточного уровня.

Внешний вид приложений JavaFX можно настроить. Каскадные таблицы стилей (CSS) отделяют внешний вид и стиль от реализации, чтобы разработчики могли сконцентрироваться на кодировании. Графические дизайнеры имеют возможность легко настраивать внешний вид и стиль приложения с помощью CSS. Если необходимо разделить пользовательский интерфейс (UI) и внутреннюю логику, то предоставляется возможность разработать аспекты представления пользовательского интерфейса на языке сценариев FXML и использовать код Java для логического приложения. Для разработки пользовательского интерфейса без написания кода используется JavaFX Scene Builder. По мере разработки пользовательского интерфейса Scene Builder создает разметку FXML, которую можно портировать в интегрированную среду разработки (IDE), чтобы разработчики могли внедрять бизнес-логику.

Следующие функции включены в JavaFX. Элементы JavaFX соответственно:

1. API Java. JavaFX - это библиотека Java, которая состоит из классов и интерфейсов, написанных в Java-коде.

2. FXML и Scene Builder. FXML - это декларативный язык разметки на основе XML для построения пользовательского интерфейса приложения JavaFX. Scene Builder генерирует разметку FXML, которая может быть перенесена в среду IDE, где разработчик может добавить бизнес-логику.

3. Встроенные элементы пользовательского интерфейса и CSS. JavaFX предоставляет все основные элементы пользовательского интерфейса, необходимые для разработки полнофункционального приложения.

4. Аппаратная ускоренная графика. Графики JavaFX основаны на конвейере рендеринга графики (Prism). JavaFX предлагает гладкую графику, которая быстро визуализируется через Prism, когда используется с поддерживаемой графической картой или графическим процессором (GPU).

5. Модель автономного развертывания приложения. Отдельные пакеты приложений имеют все ресурсы приложения и личную копию среды выполнения Java и JavaFX. Они распространяются как собственные устанавливаемые пакеты и обеспечивают одинаковый опыт установки и запуска в качестве собственных приложений для этой операционной системы.

К основным преимуществам JavaFX можно отнести то, что данная платформа подходит как для создания настольных приложений, запускаемые из-под операционной системы, так и для web-приложений, и для мобильных устройств [10].

### **3.2 Конечный автомат. Диаграмма состояний**

К предмету теории автоматов относят изучение математических моделей преобразователей дискретных данных.

К основным понятиям теории автоматов относятся:

- абстрактный автомат;
- композиция автоматов.

Абстрактный автомат (АА) рассматривает дискретные устройства с точки зрения алгоритмов функционирования, то есть преобразует дискретную информацию.

Абстрактный автомат – это модель, описываемая пятиместным кортежем:

$$A = X, Y, S, f_y, f_s ,$$

в которой первые три компонента являются непустым множеством:

$X$  – множество входных сигналов;

$Y$  – множество выходных сигналов;

$S$  – множество состояний.

Компоненты  $f_y$  и  $f_s$  – это характеристические функции, где  $f_y$  – функция выхода, а  $f_s$  – функция перехода АА из одного состояния в другое.

Если  $X, Y, S$  – конечные множества, то такой абстрактный автомат будет называться конечным автоматом.

Конечный автомат (State machine – машина состояний) используется для описания последовательности состояний, которые переживает объект в течении всего жизненного цикла. Такую последовательность можно рассматривать как ответ на событие или реакцию на это событие. Автоматный подход удобен для формализации динамики отдельных блоков системы.

Общая схема автоматов представляется в виде некоторого «ящика» (рисунок 3.1), который осуществляет преобразование входных сигналов в выходные.

Отсюда следует, что:  $Y = f_y(X, t)$ .



Рисунок 3.1 – Общая схема конечного автомата

При введении вектора состояния  $S$  в уравнение учитывается фактор времени, запоминающий прошлое состояние. Получается, что на один и тот же

набор входных данных вектора  $X$  автомат будет выдавать другие выходные сигналы вектора  $Y$ . В свою очередь, текущее состояние будет определять алгоритм функционирования автомата.

Для отображения конечного автомата используют диаграмму конечных автоматов, которая выбирает поток управления, следующий от состояния к состоянию.

Диаграмма состояний показывает:

- 1) набор состояний системы;
- 2) события, вызываемые переходом от состояния к состоянию;
- 3) действия, происходящие в результате изменения состояния.

Руководствуясь этим, составим диаграмму состояний двух агентов – агент-пешеход (рисунок 3.2) и агент-автомобиль (рисунок 3.3).

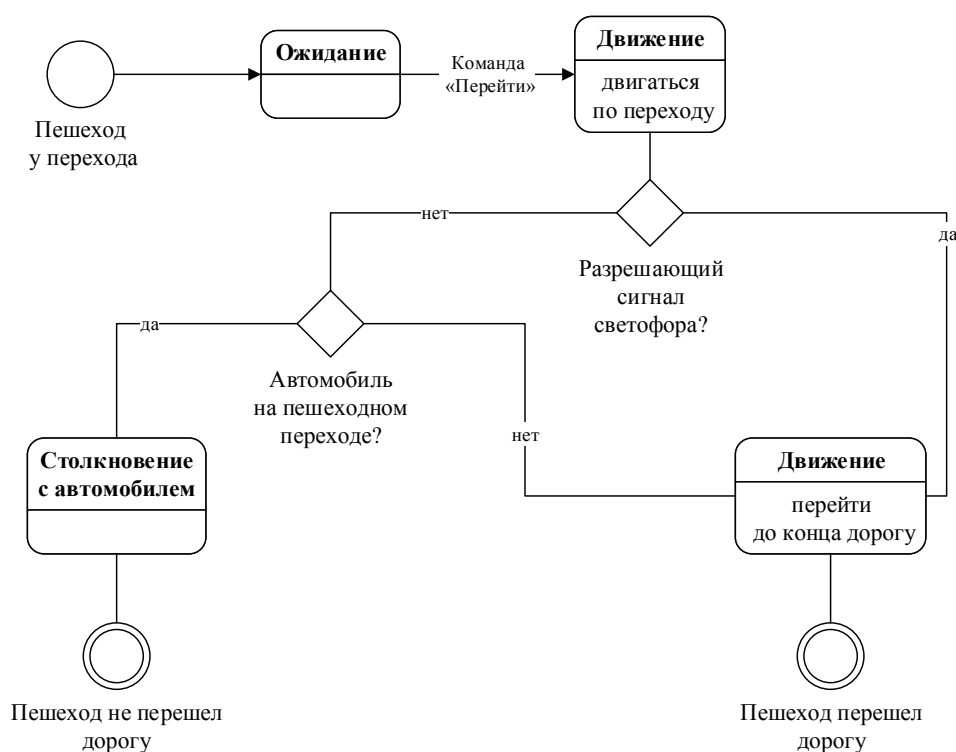


Рисунок 3.2 – Диаграмма состояний агента-пешехода

На диаграмме состояний агента-пешехода основным его состоянием является «Ожидание», оно остается активным, пока пешеход не получит команду «Перейти» от пользователя. Когда это произойдет, состояние изменится на «Движение». То есть пешеход пойдет прямо по переходу. Теперь

это состояние будет активным и сменится только тогда, когда он дойдет до конца перехода, либо произойдет столкновение с автомобилем, при котором пользователь получит уведомление о неправильности своего решения.

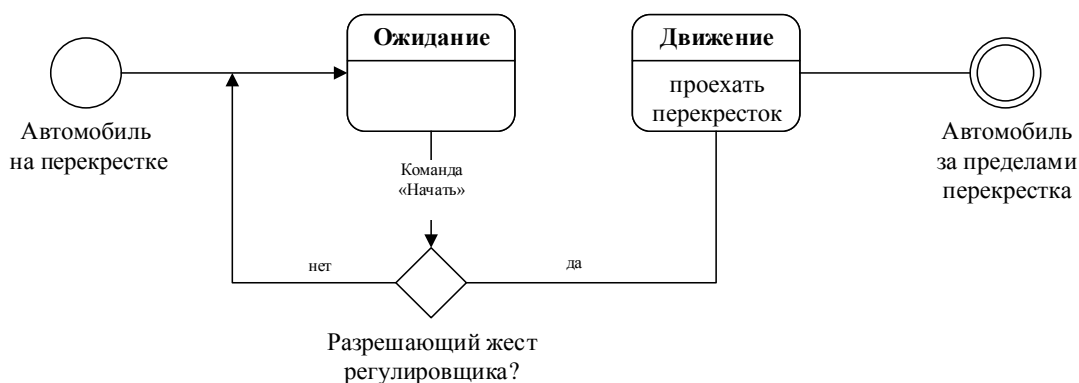


Рисунок 3.3 – Диаграмма состояний агента-автомобиля

На следующей диаграмме основным состоянием автомобиля является состояние «Ожидание». После команды пользователя «Начать» автомобиль принимает решение – оставаться в этом состоянии или сменить его на состояние «Движение». Определяющим фактором является то, какой сигнал на данный момент показывает регулировщик. Если сигнал разрешает движение, то автомобиль двигается в направлении, изначально установленном для него. Если сигнал не позволяет проехать перекресток, то состояние остается прежним – «Ожидание».

### 3.3 Программная реализация тренажеров

Для разработки программы будут использоваться правила, описанные ранее. В программном коде метод `public void go()`, реализованный в классе `Car` из пакета `GameReg`, контролирует движение автомобилей по перекрестку в зависимости от того, куда осуществляется движение регулировщиком.

Рассмотрим программную реализацию алгоритмов движения автомобилей по перекрестку с регулировщиком.

Первый алгоритм проверяет положение автомобиля (Рисунок 3.4).

```

if(!ready){
    posY = Math.max(posY - dist, sizeY - grassRectSize + diff);
    if(posY == sizeY - grassRectSize + diff){
        ready = true;
        inAction = false;
    }
}

```

Рисунок 3.4 – Фрагмент кода с алгоритмом проверки положения автомобиля

Если автомобиль еще не приблизился к перекрестку, то он продолжает движение прямо до него. Затем осуществляется движения по перекрестку прямо, пока он не выедет за пределы поля (рисунок 3.5).

```

if(goUp){
    posY = Math.max(posY - dist, -carSizeY);
    if(posY == -carSizeY){
        finished = true;
        inAction = false;
        goUp = false;
    }
}

```

Рисунок 3.5 – Движение прямо

На рисунке 3.6 представлен алгоритм поворота автомобиля. В первом шаге реализуется поворот направо, затем описывается движение автомобиля прямо до конца дороги.

```

if(goRight){
    if(angle < 90) {
        angle = Math.min(90, angle + dist / 1.25);
        posX = (sizeX - grassRectSize + carSizeY) + 2 * carSizeY * Math.cos(Math.toRadians(180 - angle));
        posY = sizeY - grassRectSize + diff - (diff + carSizeY) * Math.sin(Math.toRadians(180 - angle));
    }
    else{
        stopBlink = true;
        posX = Math.min(sizeY + carSizeY, posX + dist);
        if(posX == sizeY + carSizeY){
            finished = true;
            inAction = false;
            goRight = false;
        }
    }
}

```

Рисунок 3.6 – Фрагмент кода с алгоритмом поворота автомобиля направо

С помощью следующего алгоритма осуществляется поворот налево (рисунок 3.7). На первом шаге происходит выезд автомобиля на перекресток, затем во втором шаге реализуется поворот налево, далее по прямой траектории.

```

if(goLeft){
    if(step1){
        posY = Math.max(190, posY - dist);
        if(posY == 190){
            step1 = false;
            step2 = true;
        }
    }
    else if(step2){
        angle = Math.max(-90, angle - dist);
        posX = 175 + 40 * Math.cos(Math.toRadians(-angle));
        posY = 190 - 25 * Math.sin(Math.toRadians(-angle));
        if(angle == -90){
            step2 = false;
            step3 = true;
        }
    }
    else if(step3){
        posX = Math.max(150, posX - dist);
        if(posX == 150){
            step3 = false;
            step4 = true;
        }
    }
}

```

Рисунок 3.7 – Фрагмент кода с алгоритмом поворота налево (1, 2, 3 шаг)

Следующие шаги реализованы для объезда регулировщика, стоящего в центре перекрестка (рисунок 3.8). То есть автомобиль делает небольшой поворот налево, затем поворотом направо выравнивает свою траекторию и выезжает за пределы дороги.

Итак, были описаны основные алгоритмы движения автомобилей по перекрестку, реализованные на языке программирования JavaFX.

Теперь рассмотрим реализацию графической части обучающей системы.

<canvas> - это элемент разметки, используемый для рисования графических примитивов средствами языков программирования. Его используют для рисования, например, графов, коллажей или анимации.

Класс <canvas> имеет только два атрибута - width и height, которые задают размер изображения. В нем затем представляются команды рисования [16].





- 1) изучить теорию;
- 2) пройти тестирование;
- 3) пройти тренажер «Регулировщик»;
- 4) пройти тренажер «Пешеходный переход».

Графическое представление программы и тренажеров представлены в Приложении А на рисунках А.2-А.9.

Наиболее наглядно данные действия показывает блок-схема алгоритма работы с программой (Приложение А, рисунок А.10).

### **Выводы по третьей главе**

Разработана автоматизированная обучающая система, основанная на агентном моделировании.

Описаны основные функции языка JavaFX.

На базе аппарата конечных автоматов построена диаграмма состояний агентов, положенная в основу реализации программы.

Обучающий процесс представлен в виде интерактивного обучения, где ученик является участником обучающего процесса, изучающий предоставленный материал.

## ЗАКЛЮЧЕНИЕ

В процессе анализа существующих систем обучения по правилам дорожного движения были выявлены процессы, нуждающиеся в автоматизации. Проблемы понимания учебного материала у учащихся средних общеобразовательных учреждений по данной дисциплине вызваны недостаточной визуализацией ситуаций, происходящих на проезжей части. Также необходимо было облегчить процесс контроля знаний по теме у учащихся. Исследование предметной области были проведены на основе концептуальной модели по методологиям IDEF0 и UML, используя SADT-методологии моделирования, с помощью которых был упрощен процесс проектирования модели.

При анализе функциональной модели «КАК ЕСТЬ» были выявлены недостатки, связанные с недостаточным пониманием учебного материала и необходимостью автоматизировать процесс обучения, используя автоматизированную обучающую систему.

На основе модели структурного анализа «КАК ДОЛЖНО БЫТЬ» сформулированы требования к системе. На их основе был сделан вывод, что автоматизированная обучающая система способствует повышению эффективности обучения учащихся, вовлеченных в образовательный процесс.

Основываясь на диаграмму вариантов использования были выявлены общие требования к функциональной системе.

За основу разработки тренажеров, позволяющих визуализировать дорожную ситуацию, было выбрано агентное моделирование. С помощью диаграммы состояний был составлен алгоритм работы обучающих тренажеров, на основе которого реализовывалась программа.

Была разработана автоматизированная обучающая система, позволяющая не только изучить материал и проверить знания по заданной тематике, но и наглядно продемонстрировать последствия действий учащихся, осуществленных на проезжей части, без угрозы для жизни.

## СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

### *Научная и методическая литература*

1. Бродский Ю. И. Лекции по математическому и имитационному моделированию / Ю. И. Бродский. – М.-Берлин : Directmedia, 2015. – 240 с.
2. Запевалина А.А., Трояновский В.М. Моделирование процесса обучения с учетом лабиринтной структуры знаний // Современные информационные технологии и ИТ-образование/ Сборник научных трудов IX Международной научно-практической конференции / под ред. В.А. Сухомлина. - М.: МГУ, 2014. - С. 24-30.
3. Ивашкин Ю.А. Агентные технологии и мультиагентное моделирование // Учебное пособие, ISBN 978-5-7417-0330-4 / МФТИ, Москва, 2013, 267 с.
4. Информационные системы и технологии. Проектирование систем с использованием SADT-методологии. Методические указания. – М., 2013 г.
5. Леонова Н.Л. Компьютерное моделирование: курс лекций /СПбГТУРП. - СПб., 2015. - 88 с.
6. Майер, Р. В. Имитационная модель процесса обучения / Р. В. Майер // Домашняя лаборатория: интернет-журнал. – 2012. – № 3. – С. 358–366.
7. Орлов С. А. Программная инженерия. Учебник для вузов. 5-е издание обновленное и дополненное. Стандарт третьего поколения / С. А. Орлов. – СПб. : Питер, 2016. – 640 с.
8. Сердюков В.И., Сердюкова Н.А. Совершенствование автоматизированной системы контроля знаний // Педагогическое образование в России. - 2014. - №8. - С. 75-80
9. Трухин А.В. Анализ существующих в РФ тренажерно-обучающих систем // Открытое и дистанционное образование. - 2008. - № 1(29). - С. 32-40.
10. Фрэд Лонг, Дхрув Мохиндра, Роберт С. Сиакорд, Дин Сазерленд, Дэвид Свобода. Руководство для программиста на Java. 75 рекомендаций по написанию надежных и защищенных программ - Вильямс, 2014 – 256 с.

### *Электронные ресурсы*

11. 1С:Образовательная коллекция. ПДД для школьников. Образовательная программа [Электронный ресурс]. – Режим доступа : [https://knigamir.com/books/prochie\\_ID211/1s-obrazovatel'naya-kollektsiya-pdd-dlya-shkolnikov-obrazovatel'naya-programma-cd\\_ID550271/](https://knigamir.com/books/prochie_ID211/1s-obrazovatel'naya-kollektsiya-pdd-dlya-shkolnikov-obrazovatel'naya-programma-cd_ID550271/), свободный. – Загл. с экрана. (дата обращения 04.05.2017).

12. JavaFX: Getting Started with JavaFX [Электронный ресурс]. – Режим доступа: <http://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm#JFXST784>, свободный. – Загл. с экрана. (дата обращения 01.05.2017).

13. Syque - The Psychology of Quality and More [Электронный ресурс]. – Режим доступа : [http://www.syque.com/quality\\_tools/tools/Tools19.htm](http://www.syque.com/quality_tools/tools/Tools19.htm), свободный. – Загл. с экрана. (дата обращения 01.05.2017).

14. Золотов С.Ю. Проектирование информационных систем [Электронный ресурс]: учеб. пособие / С.Ю. Золотов; Томский гос. ун-т систем управления и радиоэлектроники. – Томск: Эль Контент, 2013.- 86 с. (дата обращения 01.04.2017).

15. Мультимедийная обучающая программа по правилам дорожного движения [Электронный ресурс]. – Режим доступа : <http://www.fvn2009.narod.ru/ARMS/ExPDD/PddTutors.htm>, свободный. – Загл. с экрана. (дата обращения 04.05.2017).

16. Правила дорожного движения для школьников [Электронный ресурс]. – Режим доступа : <http://pdd.by/school/>, свободный. – Загл. с экрана. (дата обращения 04.05.2017).

17. Сарсекеева Ж.Е., Скакова А.К. Применение компьютерных обучающих программ в процессе преподавания педагогических дисциплин // Теория и практика инновационного менеджмента: отечественный и зарубежный опыт. [2013]. URL: <http://articlekz.com/article/6368> (дата обращения: 21.04.2017).

18. Шеремет Г.Г. Геометрические преобразования и фрактальная геометрия [Электронный ресурс]: учебник/ Шеремет Г.Г.— Электрон. текстовые данные.— Пермь: Пермский государственный гуманитарно-педагогический университет, 2013.— 188 с.— Режим доступа: <http://www.iprbookshop.ru/32031.html>.— ЭБС «IPRbooks» (дата обращения 30.05. 2017).

*Литература на иностранном языке*

19. Doug, Lowe. JavaFX For Dummies / Lowe. Doug. – New Jersey : John Wiley & Sonc, 2015. – 436 с.

20. Gail Anderson, Paul Anderson. JavaFX Rich Client Programming on the NetBeans Platform - Addison-Wesley Professional, 2014.

21. Kishori Sharan. Beginning Java 8 APIs, Extensions and Libraries: Swing, JavaFX, JavaScript, JDBC and Network Programming APIs / Sharan Kishori. – : Apress, 2014. – 808 с.

22. Mark, Heckler. JavaFX 8: Introduction by Example / Heckler. . Mark. и др. – : Apress, 2014. – 420 с.

23. Uri Wilensky, William Rand. An Introduction to Agent-Based Modeling: Modeling Natural, Social, and Engineered Complex Systems with NetLogo / Uri Wilensky, William Rand. – : MIT Press, 2015. – 482 с

# ПРИЛОЖЕНИЕ А

## Элементы интерфейса программы

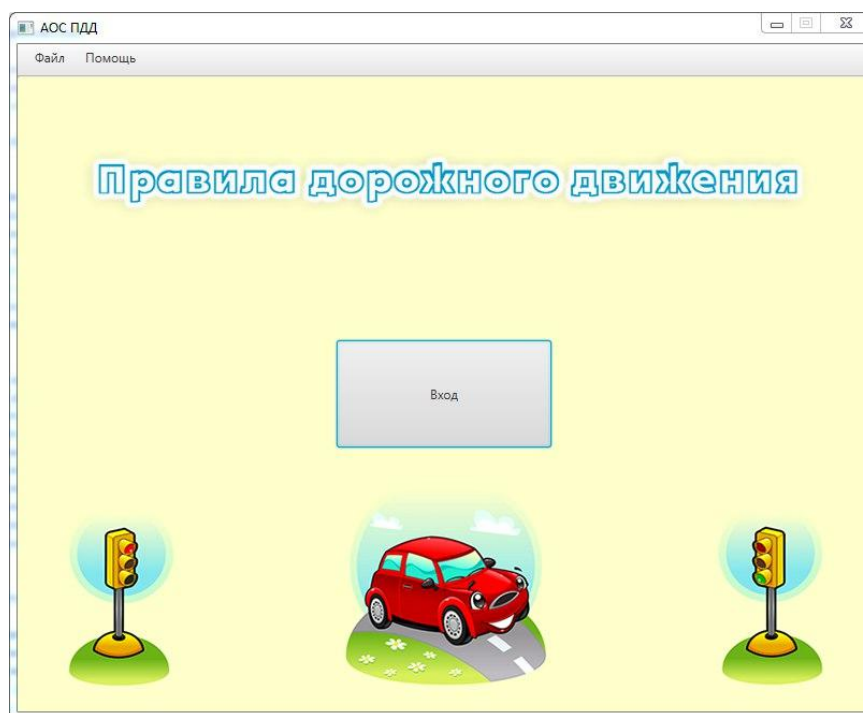


Рисунок А.1 – Стартовое окно программы АОС ПДД

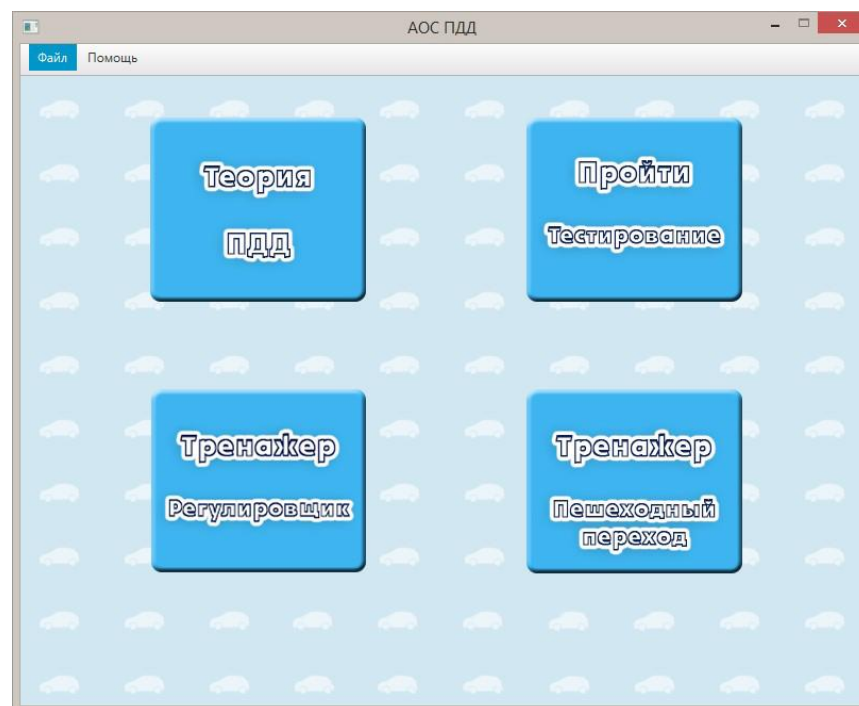


Рисунок А.2 –Окно выбора раздела программы АОС ПДД



Рисунок А.3 – Стартовое окно тренажера «Регулировщик»

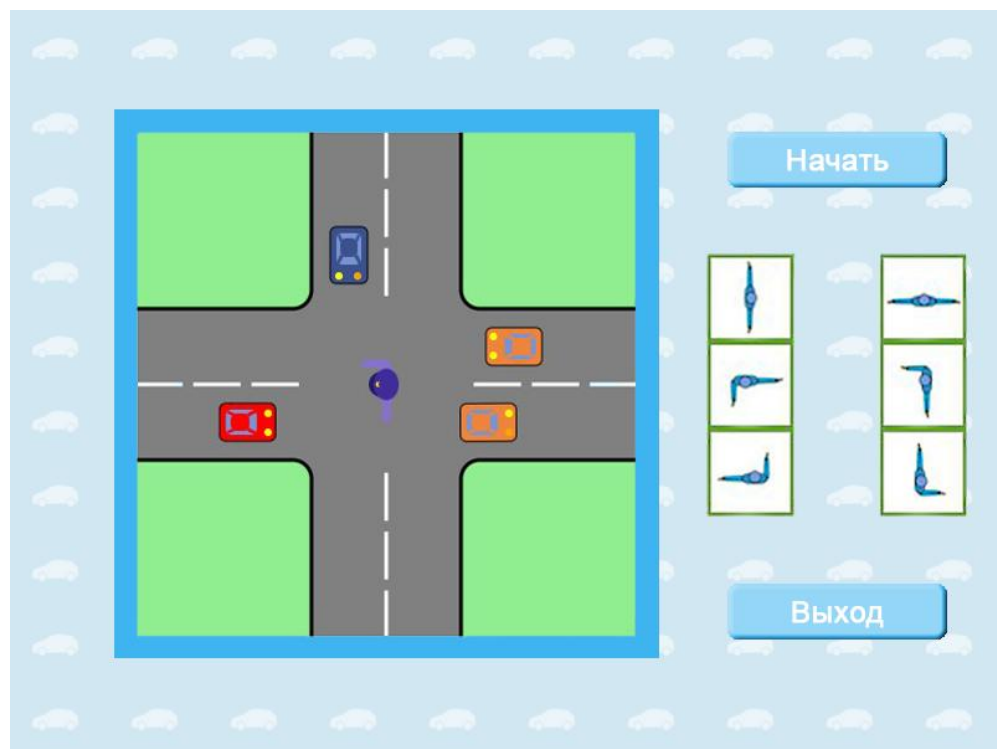


Рисунок А.4 – Запущенное окно тренажера «Регулировщик»



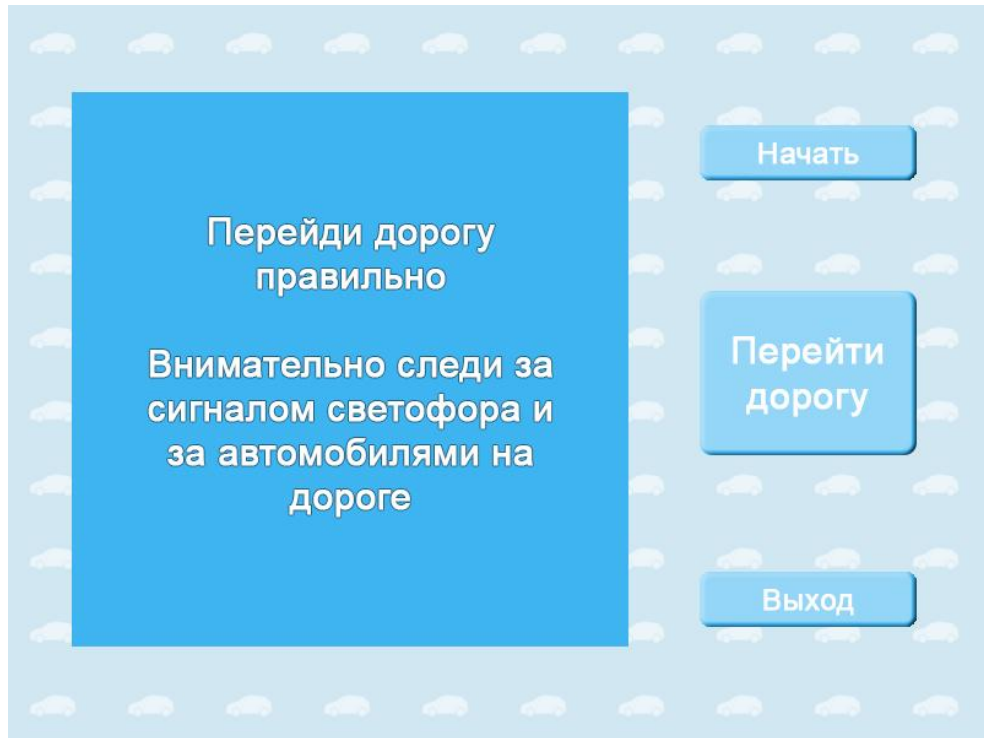


Рисунок А.5 – Стартовое окно тренажера «Пешеходный переход»

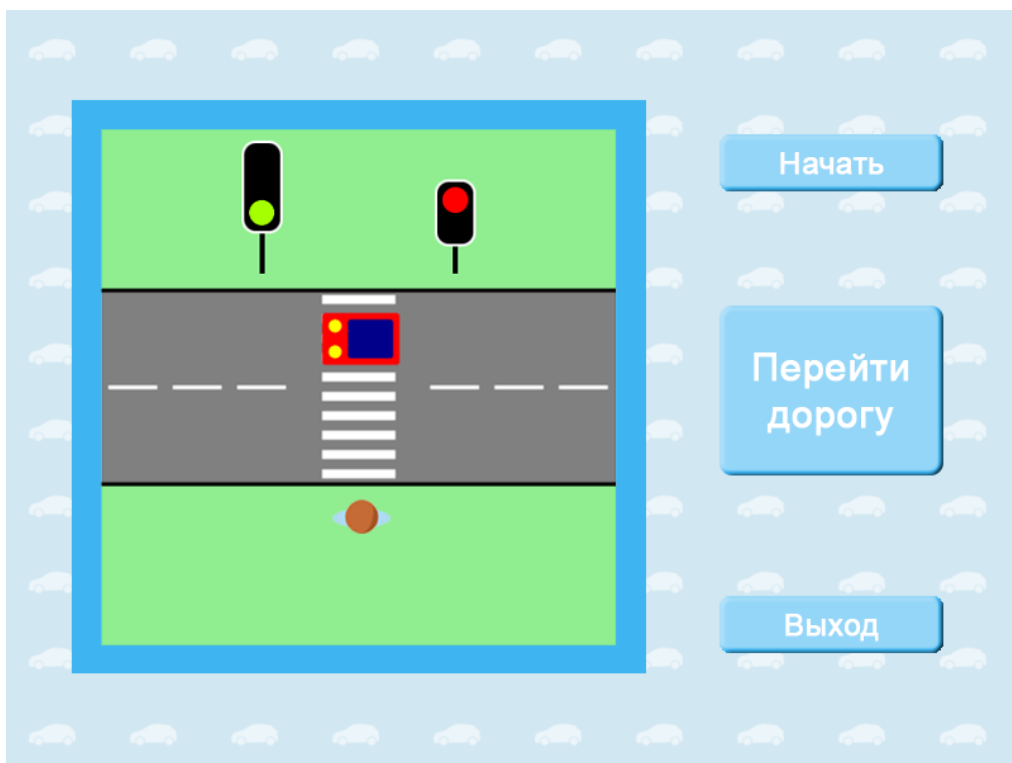


Рисунок А.6 – Запущенное окно тренажера «Пешеходный переход»

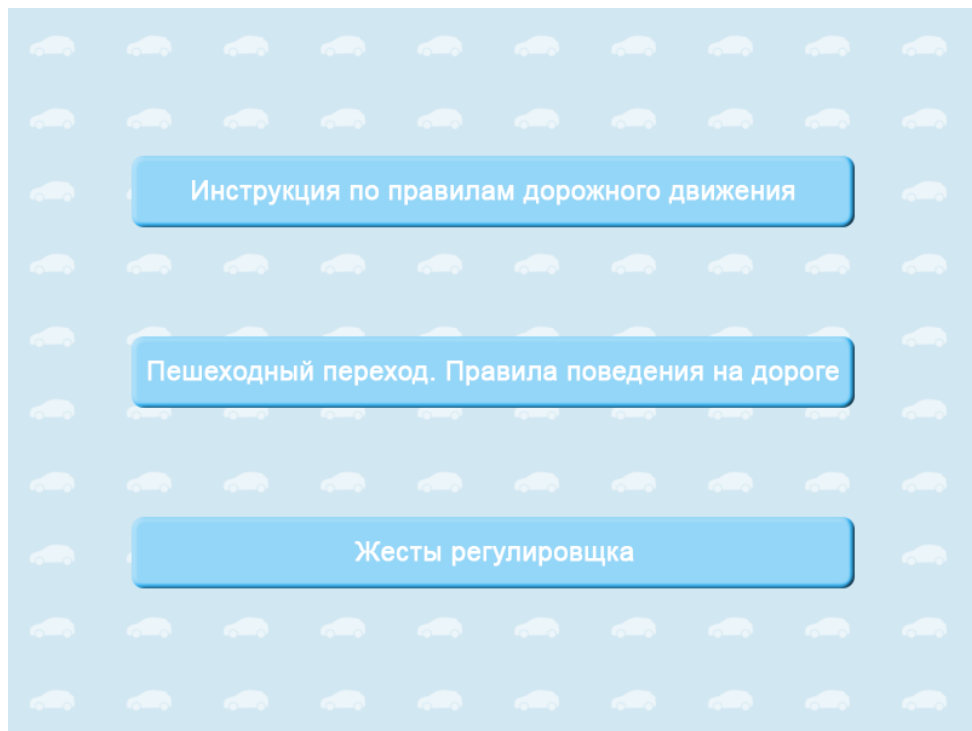


Рисунок А.7 – Запущенное окно «Теория ПДД»



Рисунок А.8 – Стартовое окно «Тест ПДД»

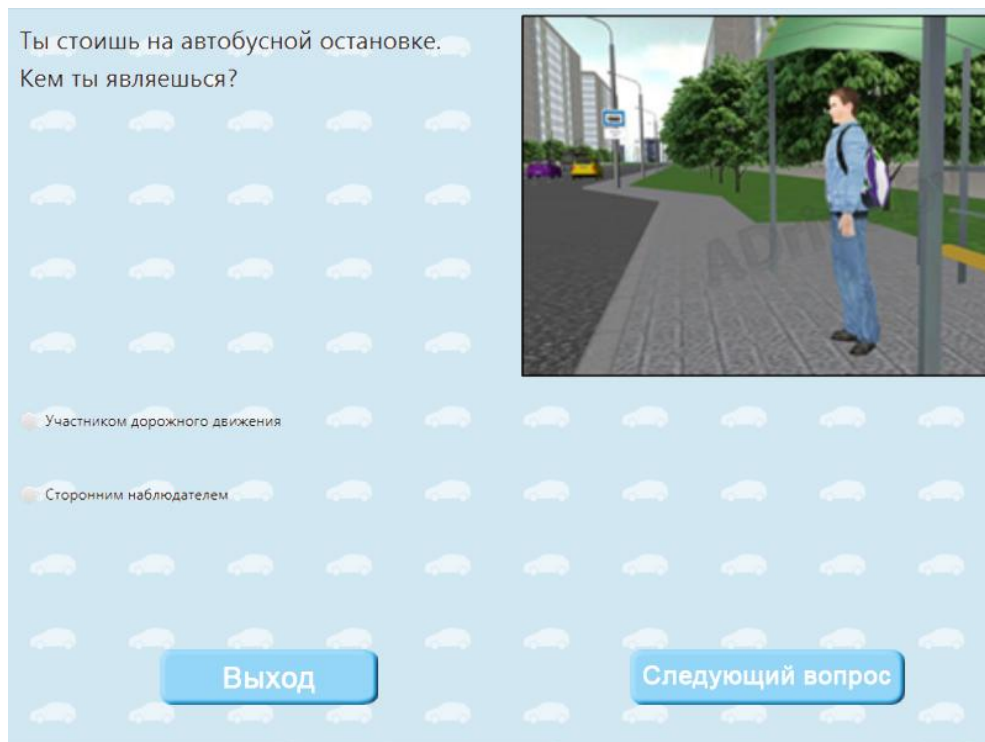


Рисунок А.9 – Запущенное окно «Тест ПДД»

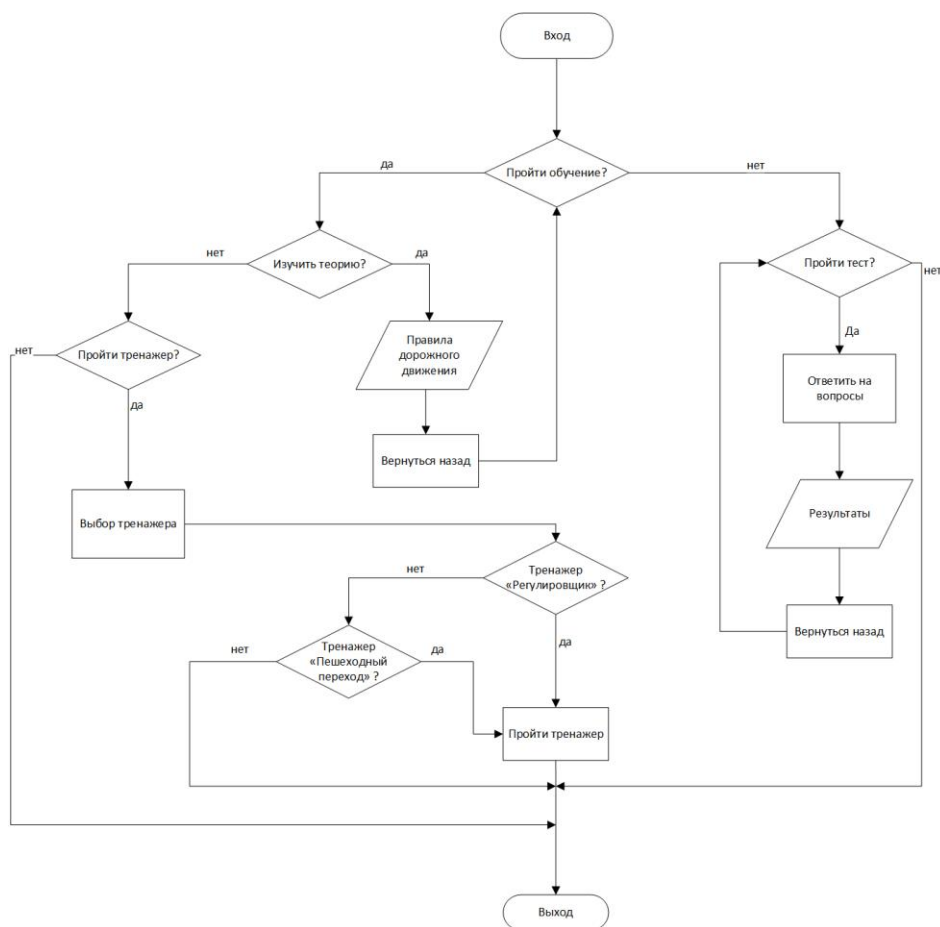


Рисунок А.10 – Блок-схема алгоритма работы в программе «АОС ПДД»