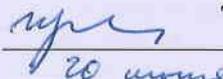


МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК
Кафедра программного обеспечения

ДОПУЩЕНО К ЗАЩИТЕ В ГЭК
И ПРОВЕРЕНО НА ОБЪЕМ
ЗАИМСТВОВАНИЯ

д.п.н., профессор

 И.Г. Захарова
20 июня 2016 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

ПРОГРАММНАЯ ПОДДЕРЖКА ДИСЦИПЛИНЫ «ОСНОВЫ
РОБОТОТЕХНИКИ» ДЛЯ СТУДЕНТОВ НАПРАВЛЕНИЯ
«ПЕДАГОГИЧЕСКОЕ ОБРАЗОВАНИЕ»

02.04.03. Математическое обеспечение и администрирование
информационных систем

Магистерская программа «Высокопроизводительные вычислительные
системы»

Выполнила работу
студентка 2 курса
очной формы обучения


(Подпись)

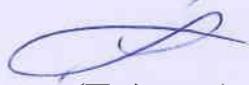
Прудаева
Ирина
Владимировна

Научный руководитель
к.т.н., доцент


(Подпись)

Воробьева
Марина
Сергеевна

Рецензент
к.т.н.


(Подпись)

Григорьев
Михаил
Викторович

ОГЛАВЛЕНИЕ

Введение.....	3
Глава 1. Описание содержания дисциплины «Основы робототехники»	6
1.1. Анализ учебных материалов и программ в области образовательной робототехники	6
1.2. Содержание дисциплины «Основы робототехники».....	7
1.3. Описание заданий лабораторных работ дисциплины «Основы робототехники»	9
Глава 2. Описание проектирования программного приложения	12
2.1. Постановка задачи.....	12
2.2. Описание алгоритма работы web - ресурса.....	12
2.3. Описание проектирования структуры базы данных web-ресурса	14
2.4. Описание проектирования редактора программ.....	23
Глава 3. Описание разработки программного приложения.....	26
3.1. Описание используемого программного обеспечения	26
3.2. Разработка web-ресурса.....	27
3.3 Разработка редактора программ	29
3.4 Алгоритм разбора лабораторной задачи.....	32
Глава 4. Описание работы программного приложения	35
4.1. Описание работы web – ресурса.....	35
4.2. Описание интерфейса редактора программ	38
Заключение	42
Список литературы	44
Приложение 1	46

ВВЕДЕНИЕ

Робототехника – одно из самых передовых направлений науки и техники, а образовательная робототехника – это новое междисциплинарное направление обучения школьников, интегрирующее знания о физике, мехатронике, технологии, математике, кибернетике и ИКТ. Кроме обучающей задачи, образовательная робототехника позволяет вовлечь в процесс инновационного научно-технического творчества и повысить престиж инженерных профессий среди молодежи, развить навыки практического решения актуальных инженерно-технических задач и работы с техникой.

На сегодняшний день образовательная робототехника активно внедряется в школьное образование: ежегодно проводятся робототехнические фестивали, соревнования и олимпиады всех уровней, проводятся летние образовательные площадки и т.п. Однако следует отметить, что знакомство школьников с основами робототехники осуществляется преимущественно в рамках дополнительного образования в виде кружков, клубов, секций, факультативных и элективных курсов.

В связи с этим возникает необходимость в подготовке педагогических кадров новым образовательным программам, включающим изучение методики и технологий применения образовательной робототехники в школе. Реализовать такую программу возможно в форме программы бакалавриата, в которой изучение вопросов образовательной робототехники будет включено как неотъемлемый элемент в общую подготовку педагога в области математического образования.

В качестве ресурса профессиональной подготовки такого рода специалистов на кафедре математики и информатики Института математики и компьютерных наук Тюменского государственного университета в учебный план подготовки бакалавра педагогического образования, включена дисциплина «Основы робототехники», целью которой является формирование профессиональных компетенций будущих педагогов в области применения образовательной робототехники в школе.

Подготовка педагогов в области образовательной робототехники требует наличия определенной материально-технической базы. Рынок аппаратных решений и программного обеспечения для реализации этого инновационного направления достаточно широк и развивается.

Цель дисциплины «Основы робототехники» заключается не в обучении студентов на конкретном оборудовании, а прежде всего, чтобы будущие педагоги имели достаточно информации о возможностях учебных конструкторов робототехники для школьников.

В связи с этим возникает цель разработать инструментальное средство для поддержки дисциплины «Основы робототехники» для обучения студентов направления «Педагогическое образование» основам программирования робототехнических систем.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Изучить учебную и методическую литературу в области образовательной робототехники.
2. Разработать учебно-методический комплекс дисциплины «Основы робототехники».
3. Составить задания лабораторных работ по дисциплине «Основы робототехники».
4. Разработать инструментальное средство для организации и проведения лабораторных работ по дисциплине.
5. Провести тестирование работоспособности разработанного программного приложения.

Разрабатываемое программное приложение предназначено для организации и проведения лабораторных занятий по дисциплине «Основы робототехники» и должно включать в себя редактор для программирования движений виртуального робота и возможность размещения дополнительной справочной поддержки по содержанию дисциплины. В связи с этим, в рамках данной работы, инструментальное средство для поддержки дисциплины

«Основы робототехники» будет состоять из двух частей: web-ресурса для размещения заданий лабораторных работ и редактора для программирования движения виртуального робота.

ГЛАВА 1. ОПИСАНИЕ СОДЕРЖАНИЯ ДИСЦИПЛИНЫ «ОСНОВЫ РОБОТОТЕХНИКИ»

1.1. Анализ учебных материалов и программ в области образовательной робототехники

Одна из серьезных проблем в описываемой области – отсутствие проработанных учебных программ и учебных материалов для учителей. Пока робототехника распространена в основном в области дополнительного образования, и потому слабо методически формализована.

Среди самых популярных среди школьников робототехнических конструкторов на сегодняшний день (Lego Mindstorms EV3, HUNO, ROBOTIS, Arduino, TETRIX) только конструкторы компании Lego имеют в своем составе средства обучения и методические материалы. Компания Lego (подразделение Lego Education) разработала целостную концепцию обучения. Деятельность Lego Education направлена на формирование у детей творческих навыков, создание ими проектных работ, сотрудничество в команде. Помимо самих конструкторов, компания предлагает пособия для учителей, рабочие тетради, справочники и программное обеспечение [15].

В отечественной педагогике накоплен позитивный опыт разработки учебных курсов по робототехнике, как с использованием материалов Lego Education, так и на базе собственных разработок (Л.Г. Белиовская, А.С. Злаказов, Г.А. Горшков, С.Г. Шевальдина, Л.Ю. Федосов, С.А. Филиппов, А.В. Чехлова, С.А. Якушин) [6].

Существующие учебные курсы и пособия по рассматриваемому направлению могут быть разделены на следующие группы.

Первая группа обучающих материалов по основам робототехники использует роботов для обучения основам программирования.

Вторая группа - это курсы, основанные на методиках проектной деятельности [3].

Третья группа - это материалы, ориентированные на выполнение олимпиадных задач по робототехнике.

Таким образом, на сегодняшний день имеется огромный дидактический потенциал робототехники, отвечающий в полной мере на вопрос «чему учить».

1.2. Содержание дисциплины «Основы робототехники»

Цель освоения дисциплины «Основы робототехники» заключается в применении будущими педагогами технологии робототехнического творчества в урочной и внеурочной деятельности в системе общего образования для развития творческих способностей подростков и юношества в процессе конструирования и программирования роботов.

Дисциплина «Основы робототехники» относится к вариативной части учебного плана. Изучается в 7 семестре в форме лекционных и лабораторных занятий, форма итоговой аттестации – зачет. Общая трудоемкость дисциплины составляет 108 часов: из них на лекционные занятия и лабораторные работы отводится по 36 часов.

В результате освоения программы дисциплины у выпускника должны быть сформированы компетенции:

- Общекультурная компетенция (ОК-3) – способность использовать естественнонаучные и математические знания для ориентирования в современном информационном пространстве.
- Профессиональная компетенция (ПК-4) - способность использовать возможности образовательной среды для достижения личностных, метапредметных и предметных результатов обучения и обеспечения качества учебно-воспитательного процесса средствами преподаваемых учебных предметов.

В дисциплине «Основы робототехники» интегрируются составляющие профессиональной подготовки преподавателя информатики и математики – психолого-педагогическая, предметная, методическая и т.д. Используются знания и виды деятельности, сформированные в процессе изучения дисциплин «Педагогика», «Психология», «Технологии программирования», «Объектно-

ориентированное программирование», «Методика обучения предмету (информатика)», «Информационные технологии в образовании».

Содержание дисциплины «Основы робототехники» включает в себя следующие разделы:

- Основы конструирования и механики;
- Программирование робототехнических устройств;
- Формы и методы обучения.

Раздел «Основы конструирования и механики» направлен на изучение базовых механических принципов и элементарных технических решений, лежащих в основе всех современных конструкций и устройств (устойчивость и прочность конструкции, зубчатые и ременные передачи, рычаги, колеса, оси, блоки, преобразование движения). Освоение этих знаний дает студентам возможность разработки и сборки модели для реализации поставленной задачи.

Раздел подготовки «Программирование робототехнических устройств» основывается на требованиях к результатам изучения предметной области «Математика и информатика», включающих в себя «развитие алгоритмического мышления, необходимого для профессиональной деятельности в современном обществе; развитие умений составить и записать алгоритм для конкретного исполнителя; формирование знаний об алгоритмических конструкциях, логических значениях и операциях; знакомство с одним из языков программирования и основными алгоритмическими структурами – линейной, условной и циклической». Любая робототехническая конструкция выступает в роли исполнителя программы (алгоритма).

Раздел «Формы и методы обучения» включает в себя:

- общие подходы к формированию содержания учебного курса по робототехнике на разных ступенях общего образования;
- дидактические принципы отбора содержания учебного курса по робототехнике для интеграции с предметами естественно-научного и

технологического направления (математике, информатике, физике, технологии);

- раскрытие метапредметных связей робототехники и предметов естественно-научного и технологического направления (математики, информатики, физики, технологии);
- практические приемы внедрения робототехники в деятельность образовательного учреждения;
- возможные способы интеграции образовательной робототехники в учебный процесс;
- методы и приемы формирования универсальных учебных действий у учащихся, а также планируемые результаты в соответствии с ФГОС;
- использование сетевых возможностей организации и проведения практических занятий по робототехнике.

1.3. Описание заданий лабораторных работ дисциплины «Основы робототехники»

В учебном плане дисциплины «Основы робототехники» на выполнение лабораторных работ отводится 36 часов. Лабораторный практикум состоит из семи лабораторных работ:

1. Образовательная робототехника в России (2 часа).
2. Детали машин и основы конструирования (8 часов).
3. Программирование моторов (6 часов).
4. Алгоритмические конструкции программирования (2 часа).
5. Управление роботом с использованием датчиков (4 часа).
6. Элементы олимпиадной робототехники (2 часа).
7. Внедрение основ робототехники в систему школьного образования (2 часа).

Рассмотрим содержание лабораторной работы на тему «Программирование моторов».

Робот — это автоматическое устройство для осуществления производственных и других операций по определенной программе (алгоритму).

Главной функцией робота является «Движение». Двигаться может весь робот (движение тележки), двигаться могут части робота: движение манипулятора (клешня, рука), движение сенсора. В движение робота приводит серводвигатель (Рис. 1).



Рис. 1 - Серводвигатель Lego Mindstorms NXT 2.0

Серводвигатель – это сочетание электродвигателя, шестерёнчатого редуктора и датчика вращения, объединённых в одном корпусе (Рис. 2) [12].

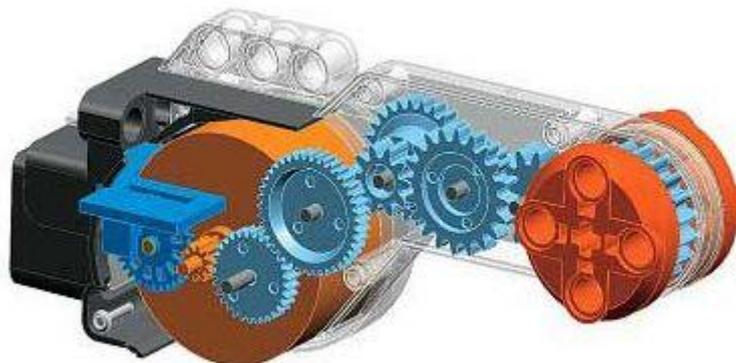


Рис. 2 - Внутреннее устройство серводвигателя Lego Mindstorms NXT 2.0

Серводвигатели подключаются к портам программируемого блока, и могут работать как каждый отдельно, так и в синхронизированном режиме. Этот режим позволяет обеспечить одинаковую скорость вращения моторов, даже при различной нагрузке на каждый из них.

Задания лабораторной работы на тему «Программирование моторов» включают в себя интерактивное изучение возможностей управления серводвигателями:

Задание 1. Включение мотора.

Задание 2. Включение мотора и ожидание.

Задание 3. Остановка мотора.

Задание 4. Движение по квадрату.

Задание 5. Движение двумя моторами рулевым управлением.

Задание 6. Движение рулевым управлением.

Задание 7. Движение одним мотором при независимом управлении моторами.

Задание 8. Движение двумя моторами.

Задание 9. Движение по восьмерке.

Задание 10. Итоговое задание по теме: Чертежник.

В этом задании необходимо составить программу, чтобы робот, проехав по полю, начертил заданный рисунок.

ГЛАВА 2. ОПИСАНИЕ ПРОЕКТИРОВАНИЯ ПРОГРАММНОГО ПРИЛОЖЕНИЯ

2.1. Постановка задачи

Поддержка дисциплины «Основы робототехники» состоит из справочных материалов дисциплины, лабораторных работ по темам дисциплины и редактора для программирования движения виртуального робота.

Требуется создать программное приложение, в котором студенты направления «Педагогическое образование» при изучении дисциплины «Основы робототехники» могли пользоваться дополнительными справочными материалам по темам дисциплины, получать задания лабораторных работ, выполнять задания преподавателя по программированию движений робота в редакторе и отправлять выполненные задания преподавателю.

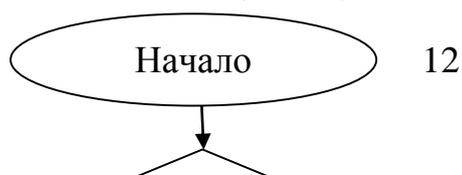
В редакторе программирования движения робота-тележки реализовать:

1. Блоки управлением движения робота-тележки: большой мотор, средний мотор, рулевое управление, независимое управление, ожидание.
2. Процесс составления программы из блоков и настройку параметров блоков.
3. Исполнение, составленной программы виртуальным роботом-тележкой.
4. Отправку ответа на задание преподавателю. В качестве ответа может быть составленная программа, траектория движения робота, текстовый ответ на задание.

Возможность выставления дополнительных справочных материалов по темам дисциплины, заданий лабораторных работ, просмотр ответов на задания реализовать на web – ресурсе.

2.2. Описание алгоритма работы web - ресурса

Рассмотрим алгоритм отображения информации на web – ресурсе для разных групп пользователей (Рис. 3).



—

Рис. 3 - Алгоритм отображения информации на web-ресурсе

2.3. Описание проектирования структуры базы данных web-ресурса

База данных web-ресурса создается для информационного обслуживания преподавателя, студентов и других пользователей среды. База данных должна содержать данные о студентах, справочные материалы дисциплины, задания лабораторных работ, данные о выполнении заданий и предоставлять возможность получать разнообразные отчеты.

В соответствии с предметной областью система строится с учётом следующих особенностей:

- каждое задание создается в рамках лабораторной работы;
- каждая лабораторная работа может содержать несколько заданий;
- у каждого задания может быть несколько вопросов;
- для каждого вопроса указывается справочный материал;
- каждый вопрос может быть одного типа;
- под типом вопроса подразумевается вопрос с текстовым ответом, вопрос с числовым ответом, однозначный выбор, не однозначный выбор из нескольких вариантов и т.п.
- студент может неограниченное количество раз выполнять одно задание;
- каждое задание выполняется одним студентом.

Выделим базовые сущности предметной области:

- **Пользователи** образовательной среды (студенты). Атрибуты пользователей – ФИО, Место учебы (работы), Телефон, E-mail, Логин, Пароль.
- **Справочный материал** образовательной среды включает в себя лекции по темам дисциплины. Атрибуты справочного материала – Тема, Название, Имя файла.
- **Задание** лабораторной работы для выполнения. Атрибуты задания – Название задания, Тема задания (лабораторная работа), Вопросы задания, Тип вопроса, Справочный материал.
- **Ответ** на задание лабораторной работы. Атрибуты ответа – Пользователь, давший ответ; Дата ответа; Вопрос; Ответ.

Рассмотрим сущность «Справочные материалы». Каждая лекция (файл) характеризуется названием и темой. Одной теме может принадлежать несколько лекций. Следовательно, из сущности «Справочный материал» необходимо вынести тему лекций в отдельную сущность «Тема» и сделать связь с помощью первичного ключа.

Аналогично сущность «Задание» разбивается на сущности «Тип вопроса», «Вопрос» и «Лабораторная работа».

В результате проведения процесса нормализации базовых сущностей базы данных web-ресурса получаем сущности: Тема (Таблица 1), Справочный материал (Таблица 2), Лабораторная работа (Таблица 3), Задание (Таблица 4), Вопрос (Таблица 5), Тип вопроса (Таблица 6), Ответ (Таблица 7), Пользователь (Таблица 8).

Таблица 1

Описание сущности «Тема»

Номер темы	Уникальный идентификатор темы
Название темы	Текст названия разделов лекций

Таблица 2

Описание сущности «Справочный материал»

Номер лекции	Уникальный идентификатор лекции
Номер темы	Идентификатор темы, которой принадлежит данная лекция
Название лекции	Текст названия лекции
Имя файла	Имя файла лекции

Таблица 3

Описание сущности «Лабораторная работа»

Номер лабораторной работы	Уникальный идентификатор лабораторной работы
Тема лабораторной работы	Текст названия лабораторной работы

Таблица 4

Описание сущности «Задание»

Номер задания	Уникальный идентификатор задания
Текст задания	Текст, содержащий задание для выполнения
Изображение	Графическое изображение, иллюстрирующее текст задания
Номер лабораторной работы	Идентификатор лабораторной работы, которой принадлежит данное задание

Таблица 5

Описание сущности «Вопрос»

Номер вопроса	Уникальный идентификатор вопроса
Текст вопроса	Текст вопроса
Номер лекции	Идентификатор лекции, где можно найти справочный материал по данному вопросу
Номер задания	Идентификатор задания, которому принадлежит данный вопрос
Номер типа вопроса	Идентификатор типа вопроса, которому принадлежит данный вопрос

Таблица 6

Описание сущности «Тип вопроса»

Номер типа вопроса	Уникальный идентификатор типа вопроса
Тип вопроса	Название типа вопроса

Таблица 7

Описание сущности «Ответ»

Номер ответа	Уникальный идентификатор ответа
Номер пользователя	Идентификатор пользователя, давшего ответ
Дата ответа	Дата ответа на вопрос
Номер вопроса	Идентификатор вопроса, на который данный ответ
Ответ	Текст ответа

Таблица 8

Описание сущности «Пользователь»

Номер пользователя	Уникальный идентификатор пользователя
ФИО	Фамилия Имя Отчество пользователя
Место учебы / работы	Место учебы или работы пользователя
Телефон	Телефон пользователя
E-mail	Адрес электронной почты пользователя
Логин	Логин пользователя
Пароль	Пароль пользователя

Изобразим данные сущности на диаграмме «сущность - связь» в нотации IDEF1X (Рисунок 4).

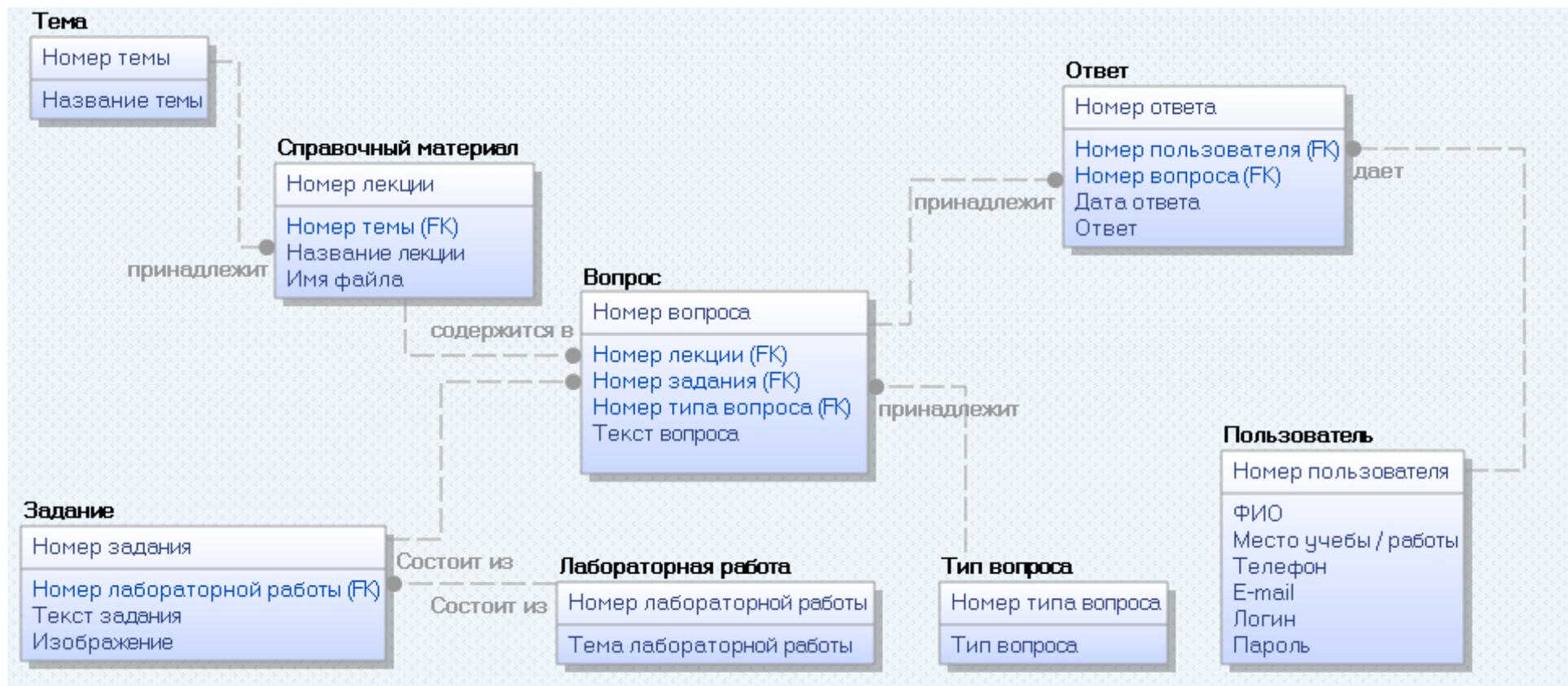


Рис. 4 - ER модель базы данных web-ресурса

Преобразуем ER модель в реляционную структуру базы данных.

На основании построенных отношений определим наборы данных, которые будут использоваться в организации базы данных.

Таблица 9

Состав базы данных web-ресурса

Наименование набора данных	Идентификатор	Тип набора данных	Функциональное назначение
Тема	Part	Таблица	Список тем, изучаемых в содержании дисциплины «Основы робототехники»
Справочный материал	Lecture	Таблица	Список файлов, в которых содержатся лекции
Лабораторная работа	Subject	Таблица	Список лабораторных работ
Задание	Task	Таблица	Список заданий для лабораторных работ
Вопрос	Question	Таблица	Список вопросов задания лабораторной работы
Тип вопроса	Type_question	Таблица	Тип ответа на вопрос (текст, число, изображение)
Ответ	Answer	Таблица	Ответ на вопрос
Пользователь	User	Таблица	Для хранения данных о студентах, выполняющих лабораторные работы

Определим характеристики данных, содержащиеся в каждой таблице.

Таблица 10

Описание базы данных web-ресурса

Наименование таблицы	Наименование поля	Идентификатор	Тип	Дополнительные характеристики
Part	Номер темы	Id_part	int(4)	Not null A_I Primary key
	Название темы	Part	varchar(255)	Not null
Lecture	Номер лекции	Id_lect	int(5)	Not null A_I Primary key
	Номер темы	Id_part	int(4)	Not null
	Название лекции	Lecture	varchar(255)	Not null
	Файл лекции	File	varchar(10)	Not null
Subject	Номер лабораторной работы	Id_subject	int(4)	Not null A_I Primary key
	Тема лабораторной работы	Subject	varchar(255)	Not null

Наименование таблицы	Наименование поля	Идентификатор	Тип	Дополнительные характеристики
Task	Номер задания	Id_task	int(5)	Not null A_I Primary key
	Текст задания	Task	varchar(255)	Not null
	Изображение	Picture	varchar(255)	Not null
	Номер лабораторной работы	Id_subject	int(5)	Not null
Question	Номер вопроса	Id_ques	int(5)	Not null A_I Primary key
	Текст вопроса	Question	varchar(255)	Not null
	Номер лекции	Id_lect	int(5)	Not null
	Номер задания	Id_task	int(5)	Not null
	Номер типа вопроса	Type_ques	Int(2)	Not null
Type_question	Номер типа вопроса	Id_type	int(2)	Not null A_I Primary key

Наименование таблицы	Наименование поля	Идентификатор	Тип	Дополнительные характеристики
	Тип вопроса	Part	varchar(50)	Not null
Answer	Номер ответа	Id_answer	int(6)	Not null A_I Primary key
	Номер пользователя	Id_user	int(4)	Not null
	Дата ответа	Date_answer	date	Not null
	Номер вопроса	Id_ques	int(5)	Not null
	Ответ	Answer	varchar(255)	Not null
User	Номер пользователя	Id_user	int(4)	Not null A_I Primary key
	ФИО	Name	varchar(255)	Not null
	Место учебы / работы	Job_user	varchar(255)	Not null
	Телефон	Phone_user	varchar(12)	Not null
	Е-mail	Email_user	varchar(50)	Not null
	Логин	Login	varchar(20)	Not null
	Пароль	Password	varchar(32)	Not null

2.4. Описание проектирования редактора программ

Основная цель редактора программирования движений виртуального робота заключается в создании программы управления движением роботом. Программа составляется из блоков: средний мотор, большой мотор, рулевое управление, независимое управление моторами.

Блоки имеют параметры настройки:

- «Включить»,
- «Выключить»,
- «Мощность»,
- «Включить на количество секунд»,
- «Включить на количество градусов»,
- «Поворот на угол»
- «Тормозить в конце».

В блоке «Рулевое управление» обязательно используется дополнительный атрибут «Синхронизация моторов», в этом случае на двигатели подается одинаковая мощность и робот двигается прямо.

В том случае, если «Синхронизация моторов» не используется, движение робота осуществляется с помощью блока «Независимое управление».

В блоке «Независимое управление» атрибут «Мощность» задается двумя различными значениями, для левого и правого двигателей. В результате такого движения робот совершает повороты. Поворот на определенный угол можно задать в настройках блока «Независимое управление» параметром «Поворот на угол».

Параметр «Тормозить в конце» управляет остановкой робота.

Для реализации данной структуры созданы классы: FileItem, WareCommand, TrWare, Pwr.

Класс Pwr является перечислением атрибутов, отвечающих за режимы работы мотора: «Выключить» (Off), «Включить» (On), «Поворот на угол» (OnAngle), «Количество секунд» (OnTime), «Количество градусов» (OnTurn).

Класс TrWare – это перечисление блоков программы: «Большой мотор» (BigDrive), «Средний мотор» (OvgDrive), «Рулевое управление» (Steer).

Класс WareCommand необходим для описания всех настроек отдельного блока программы. Данный класс имеет два конструктора: для блока, описывающего настройки одного мотора и для блока реализующего рулевое управление. При создании блока программы описывающего настройки одного мотора, конструктор заполняет соответствующие режиму (PowerSwitch) поля: градусы (degree), секунды (sec), выключить (stop), мощность (pwr). Для режима «Рулевое управление» добавляется поле, отвечающее за синхронизацию моторов (sync). При режиме «Независимое управление моторами» мощность левого мотора хранится в поле pwr, мощность правого мотора в поле pwrR.

Класс FileItem является родительским классом для классов WareCommand и TrWare и предназначен для хранения информации об одном шаге (блоке) программы.

Представим архитектуру редактора на диаграмме классов с помощью унифицированного языка моделирования UML (Рис. 5).

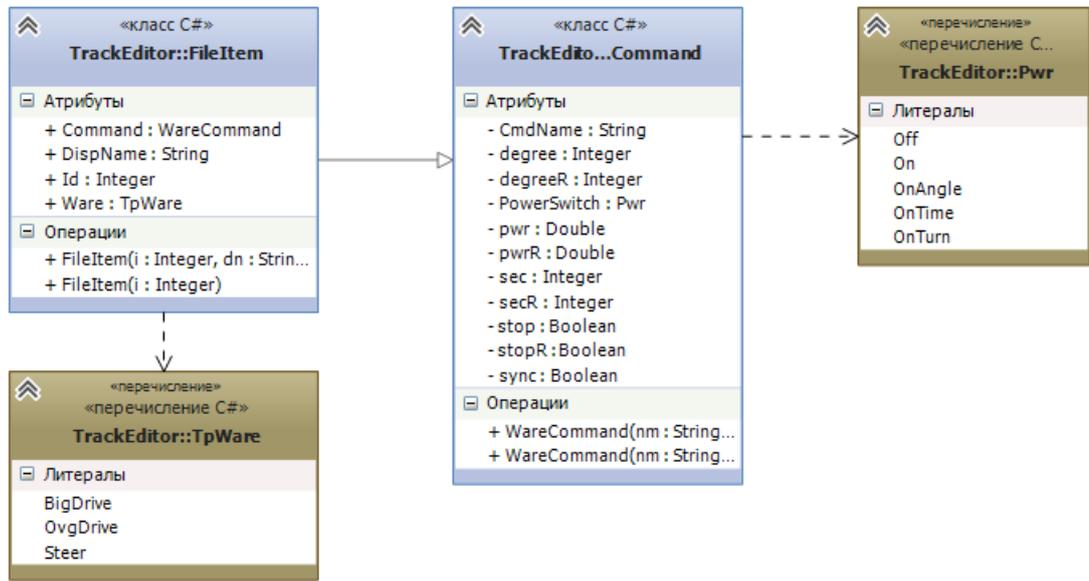


Рис. 5 – Диаграмма классов редактора программ

Следовательно, составленная программа для движения робота в редакторе программ это набор объектов класса FileItem.

ГЛАВА 3. ОПИСАНИЕ РАЗРАБОТКИ ПРОГРАММНОГО ПРИЛОЖЕНИЯ

3.1. Описание используемого программного обеспечения

Для проектирования логической и физической моделей базы данных web-ресурса использовалось CASE- средство CA Erwin Data Modeler 9.64.

CA Erwin Data Modeler имеет два уровня представления модели:

- Логический уровень, соответствующий инфологическому этапу проектирования и не привязанный к конкретной СУБД. Модели логического уровня оперируют с понятиями сущностей, атрибутов и связей, которые на этом уровне именуется на естественном языке.

- Физический уровень – это отображение логической модели на модель данных конкретной СУБД. Erwin автоматически поддерживает согласованность логической и физической схем модели данных и обеспечивает автоматическую генерацию файлов БД в различных форматах [5].

База данных реализована на реляционной системе управления базами данных MySQL. Версия программы: 5.5.25 – MySQL Community Server (GPL).

Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц:

- MyISAM, поддерживающие полнотекстовый поиск;
- InnoDB, поддерживающие транзакции на уровне отдельных записей;
- EXAMPLE, демонстрирующие принципы создания новых типов таблиц [4].

Web-ресурс создан с помощью скриптового языка программирования PHP.

PHP является одним из популярных сценарных языков в области веб-программирования. В данный язык встроены средства для разработки веб-приложений:

- автоматическое извлечение POST и GET-параметров, а также переменных окружения веб-сервера в predetermined массивы;
- взаимодействие с большим количеством различных систем управления базами данных;
- автоматическая отправка HTTP-заголовков;
- работа с HTTP – авторизацией;
- обработка файлов, загружаемых на сервер [8].

Редактор программирования движений робота-тележки разработан на объектно-ориентированном языке программирования C#. C# относится к семье языков с C-подобным синтаксисом. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов, события, свойства, комментарии в формате XML [13].

3.2. Разработка web-ресурса

Для реализации функциональных возможностей web-ресурса необходимо, чтобы web-ресурс имел модульную структуру. На web-ресурсе можно выделить несколько составляющих:

1. главная страница;
2. механизм регистрации, авторизации и модерации пользователей;
3. добавление и удаление справочных материалов (лекций);
4. добавление, удаление и изменение заданий лабораторных работ;
5. загрузка и отображение результатов выполнения лабораторных работ.

Результат разбиения программы на модули (скрипты) приводится в таблице 12, здесь же указываются идентификаторы, и описывается назначение каждого из них.

Таблица 11

Описание скриптов web-ресурса

Наименование скрипта	Назначение скрипта
index.php	Главная страница ресурса

Наименование скрипта	Назначение скрипта
config.php	Установка настройки соединения с БД (host, name, user, pass)
core.php	Старт сессии. Подключение к базе данных.
login.php	Форма для ввода логина и пароля. Авторизация пользователя
logout.php	Завершение сессии
registration.php	Форма для ввода данных нового пользователя
moderation.php	Модерация новых пользователей
user_add.php	Генерация логина и пароля для нового пользователя. Добавление в базу данных.
user_delete.php	Удаление из базы данных пользователя.
files_add.php	Добавление лекций
files_cat_delete.php	Удаление тем лекций
files_delete.php	Удаление лекции
files_upload.php	Загрузка файла - лекции
discipline_add.php	Добавление лабораторной работы
discipline_delete.php	Удаление лабораторной работы
task_add.php	Добавление задания лабораторной работы
task_delete.php	Удаление задания
task_edit.php	Изменение задания
question_add.php	Добавление вопроса
question_delete.php	Удаление вопроса
question_edit.php	Изменение вопроса

Наименование скрипта	Назначение скрипта
answer_upload.php	Загрузка файла с ответами
answer_list.php	Список ссылок на выполненные задания пользователей
answer_preview.php	Просмотр ответа на задание

3.3 Разработка редактора программ

Для реализации редактора программ на форме редактора размещены элементы управления:

TreeView – для отображения дерева программы;

MenuStrip – меню редактора;

ToolStripMenuItem – элементы меню;

TabControl – панель вкладок редактора;

TabPage – вкладки редактора;

ListBox – журнал действий;

Panel – для отображения изображения движения робота.

На вкладках редактора для задания режимов движения и их параметров размещены элементы управления: GroupBox, TextBox, RadioButton, CheckBox.

Для передачи в базу данных web-ресурса, дерева программы из редактора используется сохранение дерева в xml файл.

Файл имеет следующую структуру:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<Commands>
```

```
  <Step Id="1" Equipment="Steer" Sync="false" Pwe="50" Power="On"
  Brack=="true" />
```

```
</Commands>
```

Где тег:

<Commands> - хранит дерево программы;

<Step> - используется для передачи параметров отдельного шага программы.

Для хранения значений параметров в теге <Step> используются атрибуты:

Id – номер шага в дереве программы;

Equipment – тип оборудования;

Sync – использование синхронизации моторов;

Pwe – мощность двигателя;

Power – режим движения;

Time – время движения в секундах;

Angle – угол поворота в градусах;

Brack – тормозить в конце.

На данном этапе реализации приложения робот изображен на плоскости xOy в виде стрелки, показывающей направление движения для режима «Рулевое управление».

Для данного режима возможны следующие настройки: мощность двигателей, включить на количество секунд, включить на количество градусов, включить синхронизацию двигателей, поворот на градусы, тормозить в конце. При включенной синхронизации двигателей робот двигается прямо (левый и правый двигатели имеют одинаковую мощность), при выключенной – совершает поворот.

Отправной точкой движения робота является центр панели pnShow. Направление движения – положительное направление оси Ox.

Панель pnShow имеет размер 400X300 пикселей.

1 градус вращения двигателя равен 1 пикселю панели.

Перемещение робота на плоскости состоит из двух шагов: поворот на угол и движение вперед на определенное количество градусов.

Координаты размещения робота (x, y) после двух первых шагов программы рассчитываются по формулам:

$$x = c * \cos\alpha, \quad (1)$$

$$y = c * \sin\alpha, \quad (2)$$

Здесь c – количество градусов, введенных в поле tbGradStLeft режима «Движения вперед»,

α - количество градусов, введенных в поле tbGradStLeft режима «Поворот на угол».

Следующее движение робота зависит от значений координат предыдущих шагов. Координаты текущего шага (x_n, y_n) вычисляются по формулам:

$$x_n = x_{n-1} + c * \cos(\alpha_{n-1} - \alpha_n) \quad (3)$$

$$y_n = y_{n-1} + c * \sin(\alpha_{n-1} - \alpha_n) \quad (4)$$

Здесь x_{n-1} – координата x предыдущего шага;

y_{n-1} – координата y предыдущего шага;

c – количество градусов для движения вперед;

α_{n-1} – угол поворота предыдущего шага;

α_n – угол поворота текущего шага.

При режиме «Включить на количество секунд» длина пути рассчитывается, основываясь на скорости вращения двигателя 120 оборотов/минуту при мощности равной 100, по формуле:

$$c = \frac{720 * pwe}{100} * t \quad (5)$$

Здесь c - количество градусов для движения вперед;

pwe – значение мощности, считанное из поля tbPowerStLeft;

t – количество секунд, введенное в поле tbSecStLeft.

При выключенном режиме «Тормозить в конце» к длине пути (c) прибавляется 10% от мощности.

В том случае, если синхронизация двигателей не включена, мощность левого и правого двигателей может быть различна. В этом случае робот совершает поворот вокруг своей оси. Режим поворота задается при отключенной синхронизации включением переключателя rbOnAngleSteer «Поворот на угол» и заданием количества градусов в поле tbGradStLeft.

3.4 Алгоритм разбора лабораторной задачи

Рассмотрим задачу «Движение двумя моторами» из лабораторной работы «Программирование моторов». В данном задании требуется составить программу для движения роботом по указанным траекториям (Рис. 5).

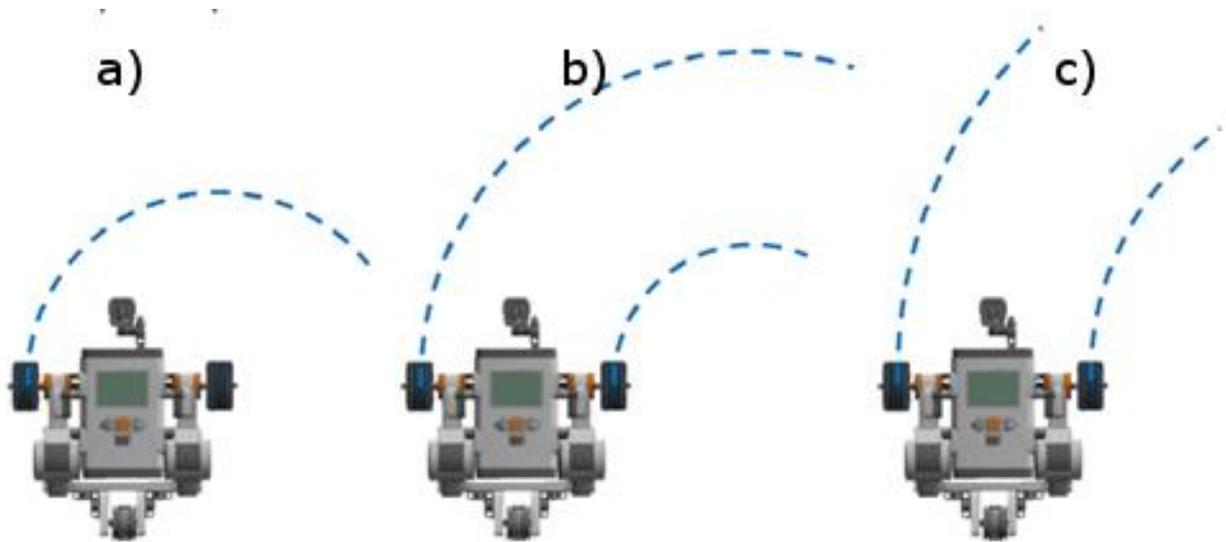


Рис. 5 – Траектории движения для задания «Движение двумя моторами»

Все три траектории движения робота а), б) и в) основываются на «Независимом управлении моторами», следовательно, функция «Синхронизация моторов» должна быть отключена.

Рассмотрим траекторию а). Одним из вариантов движения по данной траектории состоит из чередующихся движений поворота на угол и движения прямо. Для того чтобы траектория была плавной, функция «Тормозить в конце» должна быть выключена. В этом случае движение робота продолжается по инерции на некоторое количество градусов прямо пропорциональное мощности двигателей.

Траекторию а) разобьём на 10 шагов (Рис. 6):

1. поворот робота на 35 градусов;
2. движение вперед на 100 градусов;
3. поворот робота на -15 градусов;
4. движение вперед на 60 градусов;
5. поворот робота на -30 градусов;

6. движение вперед 50 градусов;
7. поворот робота на -20 градусов;
8. движение вперед 100 градусов;
9. поворот робота на -10 градусов;
10. движение вперед на 70 градусов.

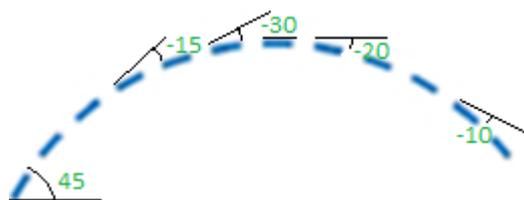


Рис. 6 – Разбивка траектории а) на участки

Результат выполнения данной программы приведен на рисунке 13 п. 4.2.

Другим вариантом организации движения робота по траектории а) является использование разных мощностей для левого и правого двигателей. В этом случае мощность для правого двигателя может быть равной 0 или меньше 0. Мощность левого двигателя подбирается экспериментально.

Аналогичное решение будет для траекторий б) и с). Единственное отличие заключается в том, что значение мощности для правого двигателя будет больше 0.

Такого рода задачи, когда нужно добиться от робота точной траектории движения содержат несколько технических нюансов, которые необходимо теоретически рассмотреть при выполнении задания.

1. Максимальная точность движения робота возможна на небольших мощностях.
2. Длина траектории зависит от уровня заряда батареи робота.
3. Отладка программы по отдельным шагам не эффективна.
4. Использование функции «Тормозить в конце» определяется экспериментальным путем.

Определение нужного количества градусов или мощностей двигателя, чтобы траектория была максимально точной, является кропотливой и сложной задачей. Не все учащиеся успешно справляются с такими задачами. Поэтому

данное задание знакомит будущих педагогов с трудностями, которые могут возникнуть в их профессиональной деятельности.

ГЛАВА 4. ОПИСАНИЕ РАБОТЫ ПРОГРАММНОГО ПРИЛОЖЕНИЯ

4.1. Описание работы web – ресурса

Отображение информации на главной странице web-ресурса зависит от авторизации пользователя ресурса.

Не авторизованные пользователи имеют возможность (Рис. 7):

1. просмотра справочных материалов;
2. регистрации на ресурсе.

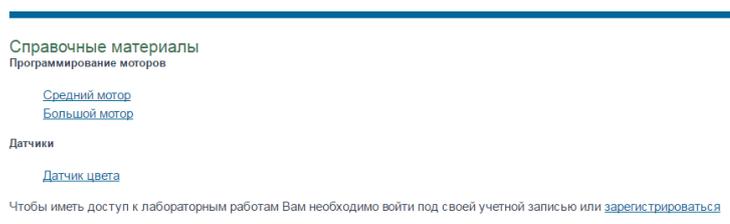


Рис. 7 – Отображение информации для не авторизованного пользователя

Авторизованные пользователи (студенты) (Рис. 8):

1. просмотр справочных материалов;
2. загрузка с ресурса установочного файла редактора программ;
3. просмотр заданий лабораторных работ;
4. загрузка в базу данных файла с ответами на задание (пиктограмма );
5. просмотр всех своих ответов (пиктограмма ).



Рис. 8 – Отображение информации для авторизованного пользователя
Преподаватель дисциплины «Основы робототехники» (Рис. 9):

1. добавление и удаление справочных материалов;
2. добавление, изменение и удаление заданий лабораторных работ;
3. просмотр выполненных лабораторных работ;
4. модерация новых пользователей.

Справочные материалы [Добавить](#)
 Программирование моторов

[Средний мотор](#) ✖
[Большой мотор](#) ✖

Датчики

[Датчик цвета](#) ✖

Лабораторные работы [Добавить](#)

[Программирование моторов](#) 🟢 ✖

1. [Включение моторов](#) 🟢 ✖
2. [Включение моторов и ожидание](#) 🟢 ✖
3. [Остановка моторов](#) 🟢 ✖
4. [Движение по квадрату](#) 🟢 ✖
5. [Блокирование движения](#) 🟢 ✖
6. [Движение двумя моторами рулевым управлением](#) 🟢 ✖
7. [Движение рулевым управлением](#) 🟢 ✖
8. [Движение одним мотором при независимом управлении](#) 🟢 ✖
9. [Движение двумя моторами](#) 🟢 ✖
10. [Движение по восьмерке](#) 🟢 ✖
11. [Чертежник](#) 🟢 ✖

[Добавить](#)

Новые пользователи

01.06.2016 [Иванов Иван Иванович](#) 🟢 ✖
 01.06.2016 [Петров Петр Петрович](#) 🟢 ✖

Рис. 9 – Отображение информации для преподавателя

При добавлении пользователя в базу данных ему автоматически генерируется логин и пароль, которые можно отправить на указанный при регистрации адрес электронной почты (Рис. 10).

Добавить пользователя

Пользователь успешно добавлен, запишите его логин и пароль 🟢

ФИО:

Логин:

Пароль:

[Добавить еще пользователя](#)

Рис. 10 – Добавление нового пользователя

При добавлении справочных материалов, указывается категория из списка или создается новая, название материала и выбирается файл для загрузки (Рис. 11).

Рис. 11 – Добавление справочного материала

При добавлении заданий лабораторной работы, необходимо сначала добавить название лабораторной работы, текст задания, иллюстрацию к заданию при необходимости и вопросы к заданию. Добавление вопроса состоит из ввода текста вопроса, выбора лекции, в которой содержится ответ на вопрос и выбор типа ответа (Рис.12).

Текст вопроса	Лекция	Тип ответа	Сохранить
Подберите значения мощности между двумя моторами для траектории а)	Средний мотор	Текст	🟢
Подберите значения мощности между двумя моторами для траектории б)	Средний мотор Большой мотор Рулевое управление	Изображение	🟢
Подберите значения мощности между двумя моторами для траектории с)	Независимое управление моторами	Программа	🟢

Рис. 12 – Страница изменения задания лабораторной работы

На главной странице преподавателя (Рис. 9), пиктограмма  справа от задания лабораторной работы означает, что в базе данных на данное задание имеются ответы студентов.

Ответы на задание фильтруются по дате (Рис. 13).

Текст вопроса	Лекция	Тип ответа	Сохранить
Подберите значения мощности между двумя моторами для траектории a)	Независимое управление моторами	Программа	✔
Подберите значения мощности между двумя моторами для траектории b)	Независимое управление моторами	Программа	✔
Подберите значения мощности между двумя моторами для траектории c)	Независимое управление моторами	Программа	✔

Ответы

Май 2016

ФИО	Дата	Вопрос 1	Вопрос 2	Вопрос 3
Иванов Иван Иванович	30.05.2016	Шаг 1 Рулевое управление Синхронизация отключена Установить мощность ЛД 50 ПД 20 Включить Отключить по завершению ЛД Отключить по завершению ПД	Шаг 1 Большой мотор Установить мощность 70 Включить Шаг 2 Большой мотор Установить мощность 30 Включить	Шаг 1 Рулевое управление Синхронизация отключена Установить мощность ЛД 70 ПД 10 Включить Отключить по завершению ЛД Отключить по завершению ПД

Рис. 13 – Просмотр ответов на задание лабораторной работы

4.2. Описание интерфейса редактора программ

На главной форме (Рис. 12) редактора для программирования движения робота четыре основных элемента управления: главное меню, дерево программы, панель команд и журнал действий.

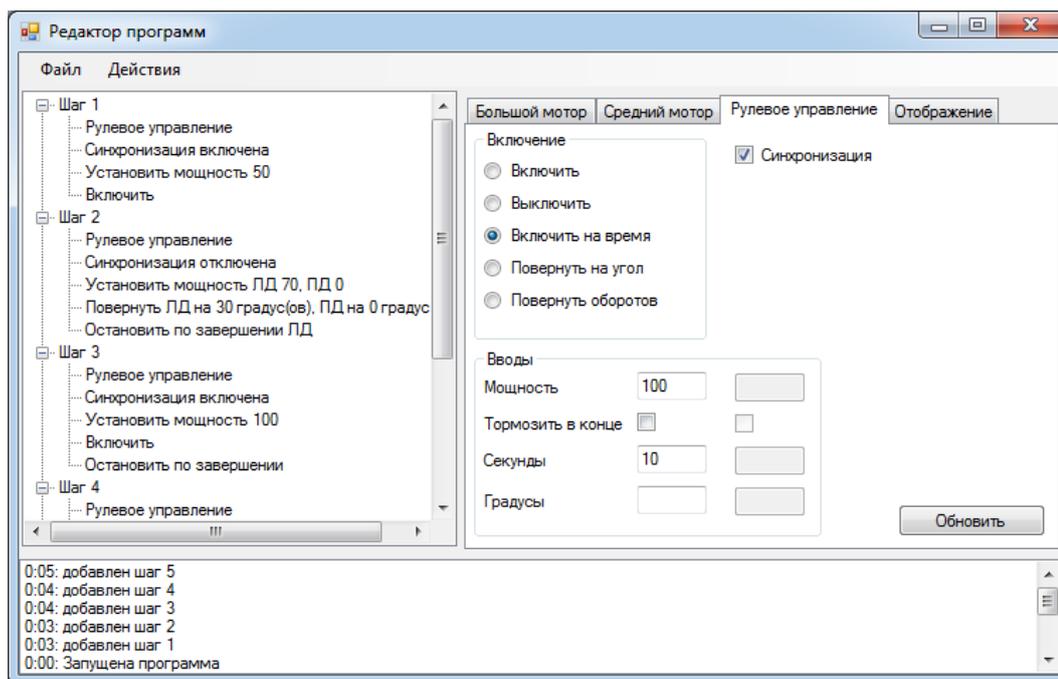


Рис. 12 – Окно редактора программ

Главное меню содержит два пункта:

1. Файл – для экспорта программы в XML файл.
2. Действия – команды для создания.

Меню «Файл» состоит из пунктов:

1. Открыть – позволяет открыть файл в формате xml;
2. Сохранить – предназначен для сохранения дерева программы в файл в формате xml;
3. Новый – для создания новой программы;
4. Выход – закрытие приложения.

Меню «Действия» содержит команды:

1. Добавить шаг – создание нового шага программы;
2. Удалить шаг – удаление выделенного шага из дерева программы;
3. Переместить вверх – перемещение шага на одну позицию вверх;
4. Переместить вниз – перемещение шага на одну позицию вниз;
5. Отобразить – отобразить траекторию движения робота.

Дерево программы состоит из двух уровней. На первом уровне отображается номер шага программы. На втором уровне название команды и список атрибутов команды.

Панель команд состоит из четырех вкладок. Вкладки «Средний мотор», «Большой мотор», «Рулевое управление» соответствуют определенной команде и содержат параметры для их настройки.

Журнал действий необходим для отображения последовательности действий.

Алгоритм создания программы в редакторе следующий: создается очередной шаг (Действия → Добавить шаг), и затем с помощью панели команд устанавливаются нужные параметры.

Доступ к параметрам «Рулевого управления» зависит от функции синхронизации двигателей. Представим алгоритм доступа параметров вкладки «Рулевое управление» (Рис. 13).

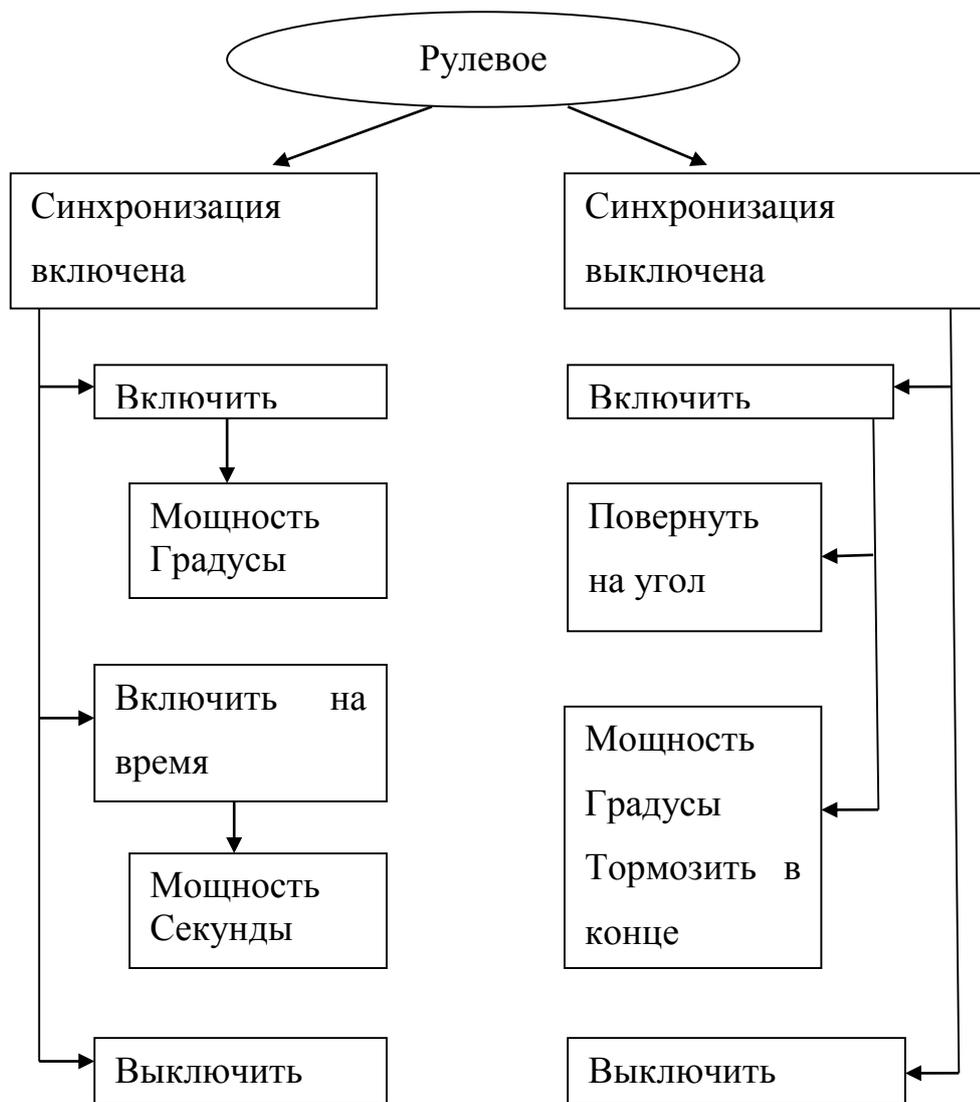


Рис. 13 – Алгоритм доступа к настройкам «Рулевого управления»
 Вкладка «Отображение» и одноименный пункт в меню «Действия»
 доступны после составления дерева программы (Рис.14).

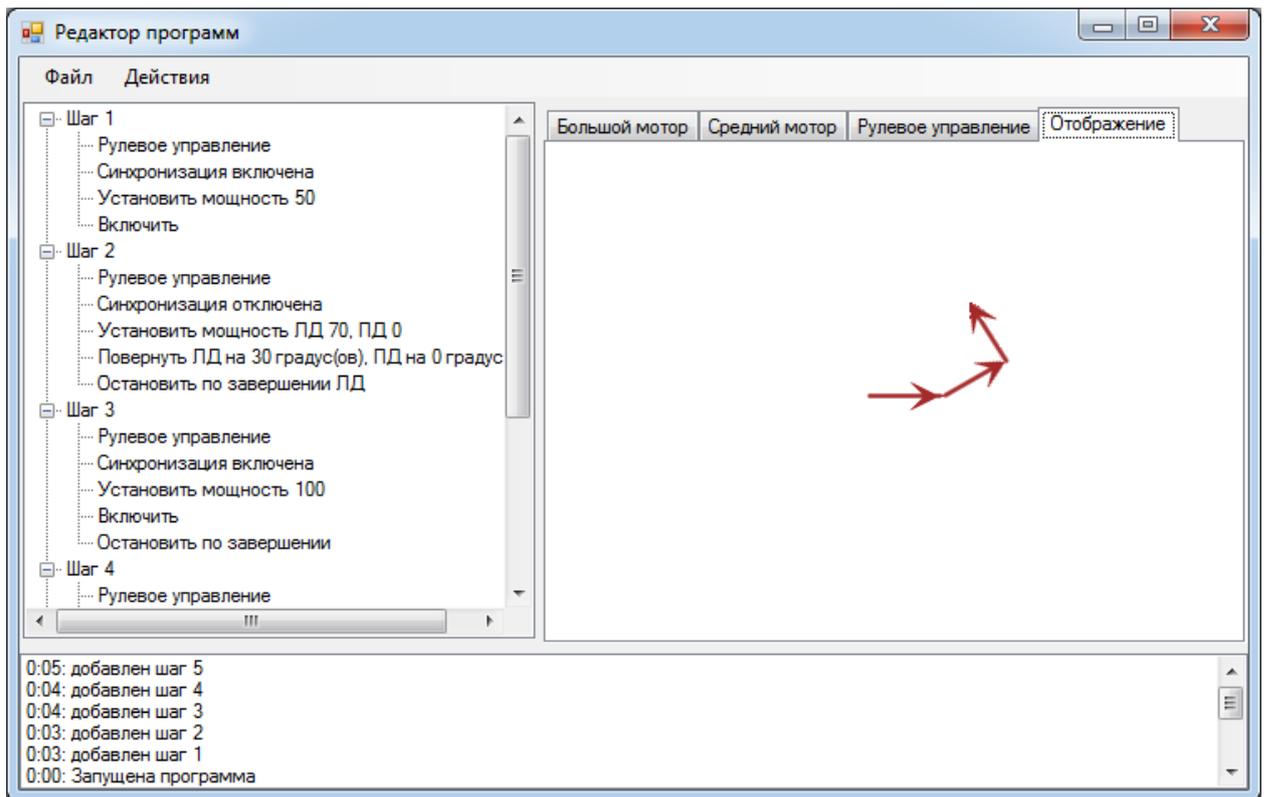


Рис. 13 – Отображение траектории движения

ЗАКЛЮЧЕНИЕ

Образовательная робототехника – это новое междисциплинарное направление в обучении, развитии школьников и новое направление в теории и методике обучения. Внедрение образовательной робототехники в школьное образование требует соответствующей квалификации педагогов. Педагог, занимающийся преподаванием основ робототехники школьникам, должен обладать соответствующими знаниями и навыками по теории и методике обучения частным вопросам таких предметов, как математика, технология, физика, информатика и ИКТ, а также иметь базовые знания по некоторым дисциплинам технических ВУЗов, в частности по теории автоматического управления, и при этом уметь программировать, поскольку каждая модель учебного робота, это точный автономный механизм с обратной связью, управляемый программой, содержащей ветвления, циклы, подпрограммы, параллельные программы и т.д.

Подготовка квалифицированных педагогов в области образовательной робототехники осуществляется по программе бакалавриата педагогического образования с двумя профилями подготовки (математика и информатика) в Институте математики и компьютерных наук Тюменского государственного университета. Вопросы робототехники рассматриваются в рамках дисциплины «Основы робототехники», в виде лекционных и лабораторных работ в объеме 108 часов.

В данной работе рассмотрен процесс разработки инструментального средства для организации и проведения лабораторных работ по дисциплине «Основы робототехники».

Первым этапом разработки являлось создание учебно-методического комплекса дисциплины «Основы робототехники». Для этого проанализированы учебные материалы и программы в области образовательной робототехники, определено содержание дисциплины, разработаны задания для лабораторных работ.

На втором этапе, при проектировании программного приложения определены функциональные требования к приложению, на основании которых приложение поделено на две части: web-ресурс, для размещения справочных материалов дисциплины и редактор для программирования движений робота.

Проектирование web-ресурса заключалось в представлении алгоритма отображения информации на web-ресурсе, описании сущностей базы данных, разработки логической модели базы данных.

При проектировании редактора программ создана система классов для хранения составленной программы.

Третий этап работы заключался в непосредственной разработке приложения. Web-ресурс реализован на языке PHP. База данных web-ресурса – MySQL. Редактор программ разработан на языке C#. Редактор программ позволяет создать программу для движения, сохранить ее в xml-файле и отобразить траекторию движения программы.

На четвертом этапе описан процесс работы с приложением.

Следует отметить, что все задачи, поставленные для достижения цели, выполнены. В результате проделанной работы, создано приложение для организации и проведения лабораторного практикума по дисциплине «Основы робототехники».

Кроме студентов направления «Педагогическое образование» данное приложение будет полезно педагогам начального, среднего, высшего и дополнительного образования, учащимся, студентам и всем, интересующимся вопросами робототехники.

СПИСОК ЛИТЕРАТУРЫ

1. Абушкин, Х. Х., Даданова, А. В. Межпредметные связи в робототехнике как средство формирования ключевых компетенций учащихся //Учебный эксперимент в образовании.-2014.-№ 3.-С.32-35
2. Андреев, Д. В. Повышение мотивации к изучению программирования у младших школьников в рамках курса робототехники /Д. В. Андреев, Е. В.Метелкин // Педагогическая информатика.-2015.-№1.-С.40-49
3. Вегнер, К. А. Внедрение основ робототехники в современной школе // Вестник Новгородского государственного университета им. Ярослава Мудрого.-2013.-№ 74 (Том 2).-С.17-19
4. Гольцман В. MySQL 5.0 – СПб: Питер, 2009. – 256 с.
5. Горбаченко В. И. Проектирование информационных систем с СА ERwin Modeling Suite 7.: учебное пособие / В. И. Горбаченко, Г. Ф. Убиенных, Г. В. Бобрышева – Пенза: Издательство ПГУ, 2012. – 154 с.
6. Дахин, А. Н. Педагогика робототехники как возникающая инновация школьной технологии // Народное образование.-2015.-34.-С.157-161
7. Камалов Р.Р. Использование элементов параллельного программирования для реализации методической системы дополнительного образования в области информатики / Р. Р. Камалов, К. А. Касаткин // Информатика и образование. - 2014. - № 8. - С. 65-67.
8. Кузнецов М., Симдянов И. РНР на примерах 2-е изд. – СПб: БХВ-Петербург.-2011. – 400 с.
9. Нетесова, О. С. Методические особенности реализации элективного курса по робототехнике на базе комплекта Lego Mindstorms NXT 2.0 / О. С. Нетесова // Информатика и образование. - 2013. - № 7. - С. 74-76.
10. Федосов А.Ю. Вопросы образовательной робототехники в подготовке педагогических кадров для начальной школы [Online]:http://www.makarenko.edu.ru/Publications/herz_chtenija_nach_obr_01_t6_2015_fedosov.pdf

11. Шимов И.В. Применение робототехнических устройств в обучении программированию школьников
[Online]:<http://elar.uspu.ru/bitstream/uspu/1174/1/povr-2013-01-32.pdf>
12. «Мускулы» работа - (Азбука робототехники) // Юный техник. - 2013. - № 9. - С. 65-71.
13. Интерактивный учебник по Visual C# [Online]:
[https://msdn.microsoft.com/ru-ru/library/bb383962\(v=vs.90\).aspx](https://msdn.microsoft.com/ru-ru/library/bb383962(v=vs.90).aspx)
14. Сервомоторы NXT [Online]: <http://lego56.ru/nxt-ext-capter4/>
15. Справочное руководство Lego Education Mindstorms EV3 [Online]:
<https://education.lego.com/ru-ru/learn/middle-school/mindstorms-ev3/teaching-resources>
16. Сравнительная характеристика систем дистанционного обучения (СДО)
[Online]: <http://www.infotechno.ru/analizSDO.htm>

ПРОГРАММНЫЙ КОД ПРИЛОЖЕНИЯ

Описание класса для загрузки изображений

```
Class Gallery {  
    var $imgDir = 'image/original/'; // Путь для хранения  
    var $thumbDir = 'image/';  
    var $imgWidth = 300; // Размер маленьких изображений  
    var $imgHeight = 500;  
    var $imgWidthB = 2500; //Размер большого изображения  
    var $imgHeightB = 2500;  
    var $imgName = "";  
    var $exts = array(2 => 'jpg'); // Разрешенные форматы изображений  
        1 => 'gif',  
        3 => 'png');  
    var $extList = array('image/gif', 'image/jpeg', 'image/png');  
    function __construct() {  
        if(!is_dir($this->imgDir) or !is_dir($this->thumbDir)) {  
            echo 'Директория не найдена.<br/>'. $this->imgDir . ' and ' . $this->thumbDir;  
        }  
    }  
    private function checkImg($img) { // Проверка изображения  
        if(!is_file($img['tmp_name'])) die('Переданные данные не корректны');  
        $data = getimagesize($img['tmp_name']);  
        $filesize = filesize($img['tmp_name']);  
        $format = explode('.', $img['name']);  
        if(!in_array(strtolower($format[count($format)-1]), $this->exts)) die('Не  
верный формат!');  
        if($filesize == 0) die('Ошибка записи изображения на сервер');  
        if($data[0] > $this->imgWidthB || $data[1] > $this->imgHeightB)
```

```

        die('Максимальный размер изображения ' . $this->imgWidthB . 'x' . $this->imgHeightB);
    if(!isset($this->exts[$data[2]])) die('Размер изображения не соответствует прописным в параметрах ' . $data[2]);
    if(!in_array($img['type'], $this->extList) || !in_array($data['mime'], $this->extList)) die('Файл не является изображением');
    $result = array('width' => $data[0],
        'height' => $data[1],
        'extension' => $this->exts[$data[2]],
        'size' => $filesize);
    return $result;
}

public function loadImg($img) {
    $data = $this->checkImg($img);
    if(!is_array($data)) die('Файл не является изображением');
    $this->imgName = $this->imgNewName($data['extension']);
    if(!move_uploaded_file($img['tmp_name'], $this->imgDir.$this->imgName))
die('Формат изображения не соответствует прописным в параметрах');
    $this->creatThumb($data['extension'], $this->imgName);
    return $this->imgName;
}

private function creatThumb($type, $name) {
    switch($type) {
        case 'gif':
            $src_img = imagecreatefromgif($this->imgDir.$name);
            break;
        case 'jpg':
            $src_img = imagecreatefromjpeg($this->imgDir.$name);
            break;
        case 'png':

```

```

    $src_img = imagecreatefrompng($this->imgDir.$name);
    break;
}
$old_w = imagesx($src_img);
$old_h = imagesy($src_img);
$ratio1 = $old_w/$this->imgWidth;
$ratio2 = $old_h/$this->imgHeight;
if($ratio1>$ratio2) {
    $thumb_w=$this->imgWidth;
    $thumb_h=$old_h/$ratio1;
}
else {
    $thumb_h=$this->imgHeight;
    $thumb_w=$old_w/$ratio2;
}
$im = imagecreatetruecolor($thumb_w, $thumb_h);
imagecopyresampled($im,$src_img, 0, 0, 0,
0,$thumb_w,$thumb_h,$old_w,$old_h);
switch($type) {
    case 'gif':
        imagegif($im, $this->thumbDir.$name);
        break;
    case 'jpg':
        imagejpeg($im, $this->thumbDir.$name);
        break;
    case 'png':
        imagepng($im, $this->thumbDir.$name);
        break;
}
}

```

```
private function imgNewName($type, $length=2) { // Метод для генерации
названий
```

```
    $chars = "abcdefghijklmnopqrstuvwxyz";
    $imgName = "";
    $slen = strlen($chars) - 1;
    while (strlen($imgName) < $length) {
        $imgName .= $chars[mt_rand(0,$slen)];
    }
    $imgName = $imgName . '_' . mt_rand(1000,9999) . '.' . $type;
    if(file_exists($this->imgDir.$imgName)) {
        $imgName = $this->imgNewName($type);
    }
    return $imgName;
}
}
```

Описание классов для хранения дерева команд

```
namespace TrackEditor
```

```
{
    public enum Pwr { On, Off, OnTime, OnAngle, OnTurn}
    public enum TpWare { BigDrive, OvgDrive, Steer}
    public class WareCommand //Произвольная команда, заготовка под набор
команд.
    {
        string CmdName;
        private Pwr PowerSwitch;
        private double pwr, pwrR;
        private bool stop, stopR;
        private bool sync;
        private int sec, secR;
        private int degree, degreeR;
```

```

public WareCommand(string nm, Pwr p, double pw, bool st, int sc, int dg)
{
    sync = true;
    CmdName = nm;
    PowerSwitch = p;
    pwr = pw;
    stop = st;
    sec = sc;
    degree = dg;
}

public WareCommand(string nm, Pwr p, double pw, double pwR, bool st,
bool stR, int sc, int scR, int dg, int dgR)
{
    sync = false;
    CmdName = nm;
    PowerSwitch = p;
    pwr = pw;
    pwrR = pwR;
    stop = st;
    stopR = stR;
    sec = sc;
    secR = scR;
    degree = dg;
    degreeR = dgR;
}
}

public class FileItem //Единичная запись - родительский класс
{
    public int Id;
    public string DispName; //Название

```

```
public TypeWare Ware; //Тип оборудования
public WareCommand Command; //Выполняемая команда
public FileItem(int i, string dn, TypeWare ware, WareCommand cmd)
{
    Id = i;
    DispName = dn;
    Ware = ware;
    Command = cmd;
}
public FileItem(int i)
{
    Id = i;
}
}
```