# ВВЕДЕНИЕ

Значимость темы курсового проекта заключается в проектировании и создании информационной системы для водительских курсов, предназначенную для упрощения ведения данных о преподавателях, учениках, автомобилях, платежей и расписания. Подобные системы позволяют на порядок повысить эффективность работы административного персонала водительских курсов, а также предоставить ученикам доступ к своевременной информации. Данная система позволит отказаться от бумажных носителей информации, а хранение данных в базе данных на серверах убережет их от потери в случае непредвиденных обстоятельств.

Цель работы: проектирование и создание информационной системы и ее администрирование для увеличения эффективности работы административного персонала водительских курсов.

Задачи работы:

* исследовать предметную область водительских курсов и определить основные проблемы, решаемые непосредственно в информационной системе;
* из описания предметной области вычленить функциональные требования, то есть, что должна делать система;
* определить будущих пользователей системы;
* спроектировать диаграмму классов информационной системы, макеты интерфейса пользователей и ER-модель;
* описать пользовательские истории;
* разработать проект базы данных для последующей разработки полной структуры её использования и создать физическую базу данных;
* реализовать информационную систему на основе объектно-ориентированного подхода, с подключенной базой данных, созданной в СУБД Postgesql.

# 1. ПРЕДМЕТНАЯ ОБЛАСТЬ

## 1.1 Исследование предметной области

Имеется некая компания, ведущая водительские курсы. Преподаватели (в некоторых случаях, по совместительству, администраторы системы), информация о которых представлена в виде их фамилии, имени, отчества и даты рождения, просматривают и, при необходимости, вносят изменения в данные, хранящиеся в системе. Информация, доступная преподавателям для просмотра, представлена в списках преподавателей, учеников, в свою очередь, представленным фамилией, именем, отчеством и датой рождения ученика, автомобилей, хранящем информацию о модели, номере и годе выпуска, а также расписанием, представляющем собой датированный список с записями по часам уроков. Пользователь с правами администратора имеет возможность редактировать все вышеперечисленные данные и просматривать данные о платежах, представляющими собой информацию об ученике, совершившем платеж, сумме и дате платежа. Пользователь с ролью Директор обладает всеми правами, что и администратор, а также имеет доступ к внесению изменений в платежи. Ученик же имеет доступ только к просмотру расписания.

## 1.2 Сравнение с существующими аналогами

Программа для автоматического составления расписания **«АРМ XXI»**: цель автоматизации работ по составлению расписания - повышение качества базового расписания, сокращение трудоёмкости его составления и текущих изменений.

Программа для составления расписания «**АСТРА»** (Автоматизированное СосТавление РАсписания) предназначена для применения в качестве инструмента при составлении расписания занятий в средних учебных заведениях различного профиля, которые работают в одну или две смены и в которых продолжительность цикла занятий в учебном процессе составляет одну неделю.

Программа **«Ректор»** предназначена для автоматизированного составления расписания занятий в учебных заведениях. Рекомендуется завучам учебных заведений и другим работникам сферы образования, занимающимся составлением учебных расписаний.

Таблица 1.1. Сравнительный анализ систем автоматизации составления расписаний

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Функциональные возможности | АРМ XXI | АСТРА | РЕКТОР | ВОДИТЕЛЬСКИЕ  КУРСЫ |
| Ведение БД учеников | + | + | + | + |
| Отчеты, аналитика | + | + | + | + |
| Ведение расписаний | + | + | + | + |
| Возможность настройки отчетов | - | + | + | - |
| Ведение прайс-листов | + | + | + | + |
| Система администрирования | + | + | + | + |
| Разграничение прав доступа | + | + | + | + |
| Возможность импорта/экспорта данных | + | + | - | - |
| Ведение расписания в классах | + | + | + | + |
| Ведение расписания индивидуальных занятиях на автотранспорте | - | - | - | + |
| Интеграция с программным обеспечением | + | - | - | - |
| Стоимость, рубли | 12000 | 23000 | 17000 | 0 |

## 1.3 Глоссарий

## 1.4 Функциональные требования

Программа «Водительские курсы» предназначена для частичной автоматизации учебного процесса автошкол. В ее задачи входит:

1. Ведение базы данных учеников и преподавателей, что подразумевает просмотр, добавление, изменение и удаление данных.
2. Составление отчетов по платежам.
3. Ведение учета платежей за курсы, редактирование записей о них.
4. Ведение учета учебных автомобилей, редактирование записей о них.
5. Составление расписания курсов.
6. Обеспечение доступа к программе персоналу автошколы и учеников.
7. Ограничение прав доступа к информации пользователей разных ролей.
8. Предоставление информации об актуальных правилах дорожного движения.

## 1.5 Действующие лица системы

В информационной системы «Водительские курсы» представлены следующие действующие лица:

* Директор (полные права) – имеет полные права на чтение и изменение данных информационной системы; только один пользователь системы может иметь данную роль.
* Администратор (права на изменение БД, прав доступа, расписания, учет платежей – только чтение) – может назначать роли пользователям (кроме директора), корректировать базы данных преподавателей и учеников, изменять расписания учебных групп и составлять отчеты.
* Преподаватель (права на изменение расписания своей группы, всё остальное, кроме учета платежей – только чтение) – может составлять и изменять расписание своей учебной группы, просматривать информацию о преподавателях и учениках; не имеет доступа к платежам.
* Ученик (только права на чтение своего расписания и информации) – может только смотреть своё расписание и справочную информацию, отображаемую в системе.

## 1.6 Глоссарий

**Вариант использования «Войти в систему»**

*Краткое описание:*

Данный вариант использования описывает вход пользователя в систему водительских курсов.

*Основной поток событий:*

Данный вариант использования начинает выполняться, когда пользователь хочет войти в систему водительских курсов.

1. Система запрашивает имя пользователя и пароль.
2. Пользователь вводит имя и пароль.
3. Система подтверждает имя и пароль, после чего открывается доступ в систему согласно правам доступа пользователя.

*Альтернативные потоки:*

1. Неправильное имя/пароль: если во время выполнения основного потока обнаружится, что пользователь ввел неправильное имя и/или пароль, система выводит сообщение об ошибке. Пользователь может вернуться к началу ос­новного потока или отказаться от входа в систему, при этом выпол­нение варианта использования завершается.

**Вариант использования «Управление платежами»**

*Краткое описание:*

Данный вариант использования позволяет вносить и изменять информацию о платежах.

*Основной поток событий:*

Данный вариант использования начинает выполняться, когда пользователь хочет внести или изменить информацию о платежах учеников водительских курсов.

1. Пользователь открывает окно платежей.
2. Система отображает в окне данные по платежам.
3. Пользователь выбирает необходимое действие, после чего запускается один из подчиненных потоков.

*Добавить:*

1. Система открывает окно добавления платежа.
2. Пользователь вводит необходимую информацию.
3. Система сохраняет данные и закрывает окно добавления платежа.

*Изменить:*

1. Система открывает окно изменения платежа.
2. Пользователь изменяет необходимую информацию.
3. Система сохраняет данные и закрывает окно изменения платежа.

*Удалить:*

1. Система спрашивает у пользователя, уверен ли он, что хочет удалить выбранную строку.
2. Пользователь подтверждает действие.
3. Система удаляет запись из базы данных.

*Альтернативные потоки:*

1. Пользователь пытается ввести дату платежа еще не наступившую. В данном случае, система выставит дату платежа как текущую.
2. Пользователь отменяет выбранное действие. В данном случае, система закрывает окно выбранного действия.
3. Пользователь пытается ввести в поле «Сумма» отрицательное число, или нечисловые символы. В данном случае, система не позволит этого сделать и не генерирует ошибку.

*Предусловия:*

Пользователь должен войти в систему как Директор.

**Вариант использования «Просмотр платежей»**

*Краткое описание:*

Данный вариант использования позволяет просматривать информацию о платежах.

*Основной поток событий:*

Данный вариант использования начинает выполняться, когда пользователь хочет посмотреть информацию о совершенных платежах.

1. Пользователь открывает окно платежей.
2. Система отображает в окне данные по платежам.

*Альтернативные потоки:*

*Предусловия:*

Пользователь должен войти в систему как Директор или Администратор.

**Вариант использования «Составление отчетов»**

*Краткое описание:*

Данный вариант использования позволяет сохранять в файле Excel отчет по платежам.

*Основной поток событий:*

Данный вариант использования начинает выполняться, когда пользователь хочет получить отчет.

1. Пользователь открывает окно платежей.
2. Система отображает в окне данные.
3. Система открывает диалоговое окно сохранения файла.
4. Пользователь выбирает место хранения файла и его название.
5. Система сохраняет составленный отчет.

*Альтернативные потоки:*

*Предусловия:*

Пользователь должен войти в систему как Директор или Администратор.

**Вариант использования «Управление базами данных»**

*Краткое описание:*

Данный вариант использования позволяет вносить и изменять данные в базах данных.

*Основной поток событий:*

Данный вариант использования начинает выполняться, когда пользователь хочет добавить или изменить информацию о преподавателях, учениках или автомобилях.

1. Пользователь открывает окно с необходимой ему информацией.
2. Система отображает запрошенные данные.
3. Пользователь выбирает необходимое действие, после чего запускается один из подчиненных потоков.

*Добавить:*

1. Система открывает окно добавления данных.
2. Пользователь вводит необходимую информацию.
3. Система сохраняет данные и закрывает окно добавления данных.

*Изменить:*

1. Система открывает окно изменения данных.
2. Пользователь изменяет необходимую информацию.
3. Система сохраняет данные и закрывает окно изменения данных.

*Удалить:*

1. Система спрашивает у пользователя, уверен ли он, что хочет удалить выбранную строку.
2. Пользователь подтверждает действие.
3. Система удаляет запись из базы данных.

*Альтернативные потоки:*

1. Пользователь пытается внести уже существующие объекты в БД (уникальность автомобилей определяется – по госномеру). В данном случае, система не позволит изменить/добавить данные и предупредит пользователя.
2. Пользователь пытается ввести в поля фамилии, имени или отчества некириллические символы. В данном случае, система не позволит этого сделать, не генерируя ошибку.
3. Пользователь пытается ввести в поля фамилии, имени, отчества или номера автомобиля буквы с некорректным регистром. В данном случае, система сама исправит символы, не генерируя ошибку.

*Предусловия:*

Пользователь должен войти в систему как Директор или Администратор.

**Вариант использования «Просмотр данных»**

*Краткое описание:*

Данный вариант использования позволяет просматривать базы данных системы.

*Основной поток событий:*

Данный вариант использования начинает выполняться, когда пользователь хочет просмотреть базы данных системы.

1. Пользователь открывает окно с необходимой ему информацией.
2. Система отображает запрошенные данные.

*Альтернативные потоки:*



*Предусловия:*

Пользователь должен войти в систему как Директор, Администратор или Преподаватель.

**Вариант использования «Изменение расписания»**

*Краткое описание:*

Данный вариант использования позволяет изменять расписания.

*Основной поток событий:*

Данный вариант использования начинает выполняться, когда пользователь хочет внести изменения в расписание.

1. Пользователь открывает окно расписаний.
2. Система отображает в окне расписание с начала текущего месяца.
3. Пользователь может выбрать интересующий его временной интервал расписания.
4. Пользователь выбирает необходимое действие, после чего запускается один из подчиненных потоков.

*Просмотр:*

1. Система отображает расписание согласно выбранному временному интервалу.

*Добавить:*

1. Система открывает окно добавления записи в расписании.
2. Пользователь устанавливает необходимую дату.
3. Пользователь выбирает преподавателей и одну из их групп в интересующее его время, нажимая кнопку «Записать».
4. Система сохраняет данные.

*Удалить:*

1. Система спрашивает у пользователя, уверен ли он, что хочет удалить выбранную строку.
2. Пользователь подтверждает действие.
3. Система удаляет запись.

*Альтернативные потоки:*

1. Пользователь пытается добавить запись в расписании за прошедшую дату. В данном случае, система предупредит пользователя о некорректности ввода данных.

*Предусловия:*

Пользователь должен войти в систему как Директор, Администратор.

**Вариант использования «Просмотр расписания»**

*Краткое описание:*

Данный вариант использования позволяет просматривать расписание.

*Основной поток событий:*

Данный вариант использования начинает выполняться, когда пользователь хочет просмотреть расписание.

1. Пользователь открывает окно расписаний.
2. Система отображает в окне расписание с начала текущего месяца.
3. Пользователь, по желанию, выбирает временной интервал для просмотра расписания.
4. Система отображает в окне расписание согласно выставленным пользователем ограничениям дат.

*Альтернативные потоки:*

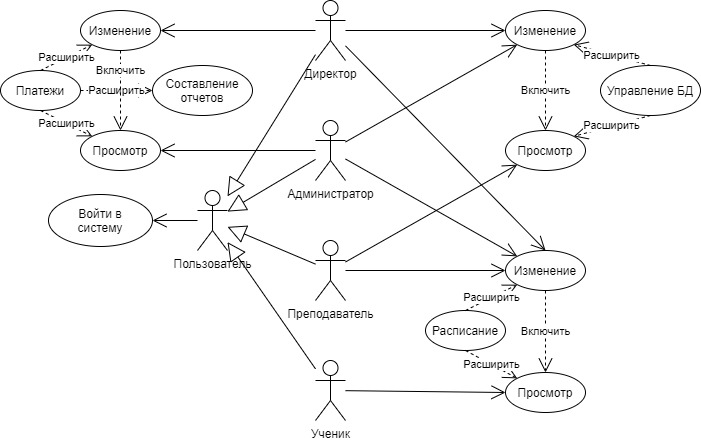


Рисунок 1.1 – Диаграмма вариантов использования

## 1.7 ER-модель

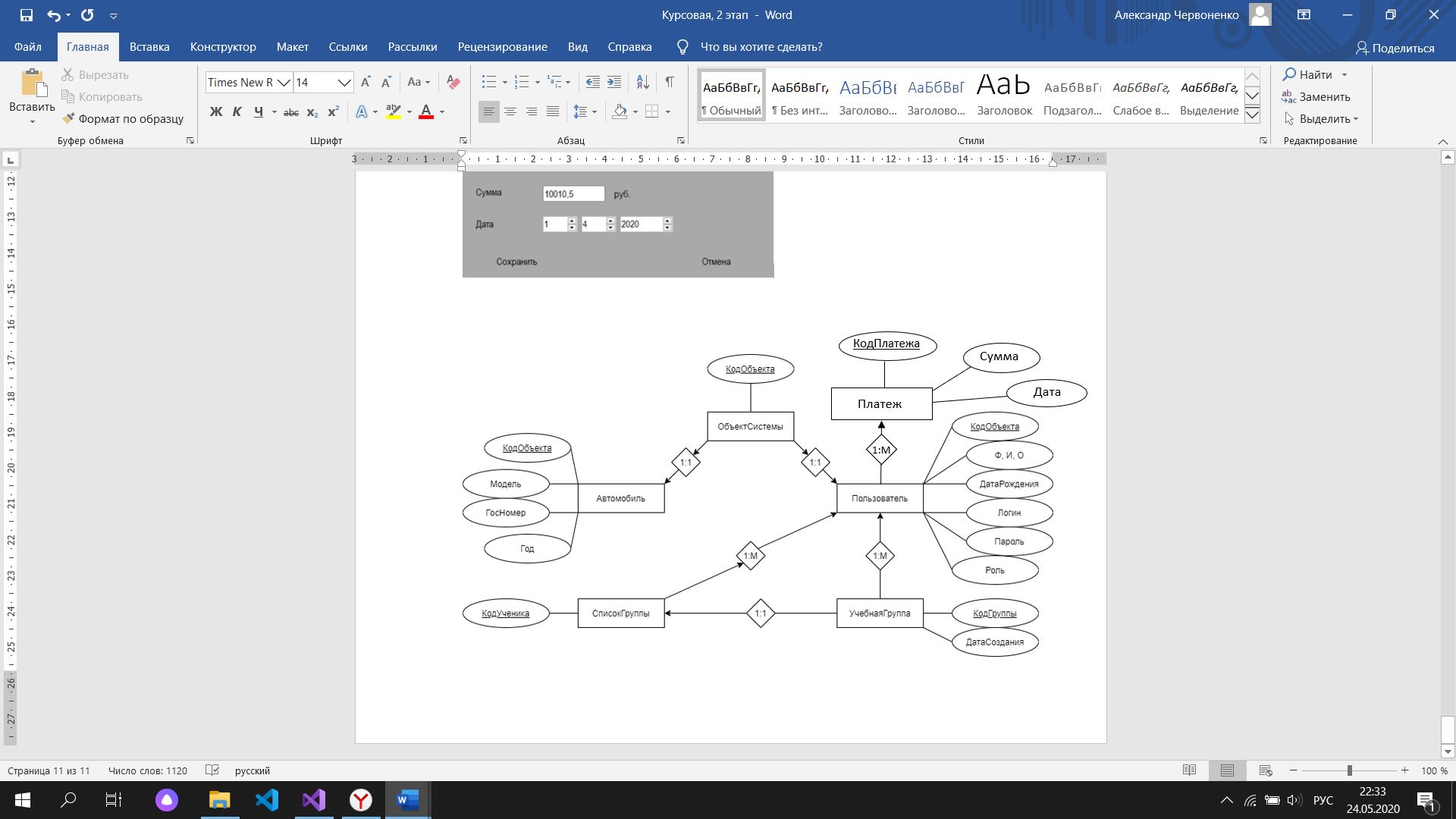


Рисунок 1.2 – ER-модель

Таблица 1.2 – сущность «Объект системы»

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип данных | Первичный ключ | Внешний ключ |
| Код объекта | Счетчик | Да | Да |
| Роль | Символьный | Нет | Нет |

Таблица 1.3 – сущность «Автомобиль»

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип данных | Первичный ключ | Внешний ключ |
| Код объекта | Числовой | Да | Да |
| Модель | Текстовый | Нет | Нет |
| ГосНомер | Текстовый | Нет | Нет |
| Год | Числовой | Нет | Нет |

Таблица 1.4 – сущность «Пользователь»

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип данных | Первичный ключ | Внешний ключ |
| Код объекта | Числовой | Да | Да |
| Фамилия | Текстовый | Нет | Нет |
| Имя | Текстовый | Нет | Нет |
| Отчество | Текстовый | Нет | Нет |
| Дата рождения | Дата | Нет | Нет |
| Логин | Текстовый | Нет | Нет |
| Пароль | Текстовый | Нет | Нет |

Таблица 1.5 – сущность «Платеж»

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип данных | Первичный ключ | Внешний ключ |
| Код платежа | Счетчик | Да | Нет |
| Код пользователя | Числовой | Нет | Да |
| Сумма | Числовой | Нет | Нет |
| Дата платежа | Дата | Нет | Нет |

Таблица 1.6 – сущность «Учебная группа»

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип данных | Первичный ключ | Внешний ключ |
| Код группы | Счетчик | Да | Нет |
| Код пользователя | Числовой | Нет | Да |
| Дата создания | Дата | Нет | Нет |

Таблица 1.7 – сущность «Учебная группа»

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип данных | Первичный ключ | Внешний ключ |
| Код группы | Числовой | Да | Да |
| Код ученика | Числовой | Да | Да |

## 1.8 Макеты системы

Рисунок 1.3 - форма входа в систему и страница неавторизованного пользователя

Рисунок 1.4 - главная страница системы

Рисунок 1.5 - страница добавления/изменения базы преподавателей (сходна с базами учеников, автомобилей и платежей).

Рисунок 1.6 - страница просмотра/изменения платежа.

Рисунок 1.7 - форма добавления Преподавателя (аналогична для Учеников, кроме параметра «Администратор»).

Рисунок 1.8 - форма добавления авто.

Рисунок 1.9 - форма изменения Преподавателя (аналогична для Учеников, кроме параметра «Администратор»).

Рисунок 1.10 - форма изменения авто.

Рисунок 1.11 - форма показа расписания.

Рисунок 1.12 - форма добавления строки расписания.

Рисунок 1.13 - форма изменения платежа.

# 2. РАЗРАБОТКА ПРОТОТИПА СИСТЕМЫ

## 2.1 Вход в систему

Вход в систему реализован путем сопоставления логина и пароля пользователя, желающего авторизоваться в системе.

На рисунке 2.1 представлен код метода «Check» класса «Open», который используется для проверки соответствия логина/пароля.

Рисунок 2.1 – листинг метода «Check»

Далее, на рисунке 2.2 показан вызов вышеописанной функции в коде программы по клику на linkLabel основной формы.

Рисунок 2.2 – листинг события «Click» для linkLabel1

## 2.2 Вывод данных

Вывод данных для преподавателей, учеников, платежей и авто реализовано по одному принципу. На рисунке 2.3 представлен код метода «ShowUser» класса «Show», обеспечивающего вывод необходимой информации из БД.

Рисунок 2.3 - листинг метода «ShowUser»

Далее, на рисунке 2.4 происходит вызов метода «ShowUser» в коде основной формы для преподавателей и учеников, происходящего по клику на преподавателиToolStripMenuItem и ученикиToolStripMenuItem соответсвенно.

Рисунок 2.4 – листинг вызова метода «ShowUser»

## 2.3 Добавление данных

Добавление данных для преподавателей, учеников, платежей и авто реализовано по одному принципу. На рисунке 2.5 представлен код метода «ShowAddForm» класса «Show»

Рисунок 2.5 – листинг метода «ShowAddForm»

Вызов вышеописанной функции для добавления представлен на рисунке 2.6.

Рисунок 2.6 – листинг вызова метода «ShowAddForm»

Далее, на рисунке 2.7 представлен код сохранения данных о преподавателе по клику на кнопку «Сохранить» формы добавления.

Рисунок 2.7 – листинг добавления данных в таблицу преподаватели

На рисунке 2.8 можно увидеть непосредственно код внесения новых данных в таблицу преподаватели через метод «AddTeacher» класса «TeacherForm».

Рисунок 2.8 – листинг метода «AddTeacher»

## 2.4 Изменение данных

Изменение данных для преподавателей, учеников, платежей и авто реализовано по одному принципу. На рисунке 2.9 представлен код метода «ShowEditForm» класса «Show» для вызова необходимой формы изменения данных.

Рисунок 2.9 – листинг метода «ShowEditForm»

Вышеописанный метод вызывается в коде программы, что можно увидеть на рисунке 2.10.

Рисунок 2.10 – листинг вызова метода «ShowEditForm»

Сам же код для изменения данных о преподавателях реализуется методом «EditTeacher» класса «TeacherForm», представленного на рисунке 2.11.

Рисунок 2.11 – листинг метода «EditTeacher»

## 2.5 Удаление данных

Удаление данных для преподавателей, учеников, платежей и авто реализовано по одному принципу. Для этого был реализован отдельный класс «DeleteData», методом которого и происходит удаление. Код данного метода можно увидеть на рисунке 2.12.

Рисунок 2.12 – листинг метода «Delete»

## 2.6 Расписание

Расписание в данной информационной системе хранится в файле Excel. Код метода «ShowSchedule» класса «ScheduleTable» для вывода расписания представлен на рисунке 2.13.

Рисунок 2.13 – листинг метода «ShowSchedule»

Удаление данных реализовано в том же классе методом «DeleteScheduleRow». Код данного метода можно увидеть на рисунке 2.14.

Рисунок 2.14 – листинг метода «DeleteScheduleRow»

Добавление данных в расписание реализовано в том же классе «ScheduleTabel» методом «AddScheduleString», код которого представлен на рисунке 2.15.

Рисунок 2.15 – листинг метода «AddScheduleString»

ЗАКЛЮЧЕНИЕ

В результате выполнения курсового проекта была создана, исследована и протестирована информационная система «Водительские курсы». Реализация проекта позволяет автоматизировать и упростить контроль за данными водительских курсов, направленными на его нужды, сокращая время на, в том числе, бухгалтерские расчеты и систематизируя поступающие данные.

Все задачи, поставленные в курсовом проекте, были успешно выполнены: была исследована предметная область водительских курсов, разработан проект информационной системы, создана физическая ее модель и проведены тестирования.

Основным достоинством представленной системы является возможность систематизации информации о сотрудниках и учениках курсов, автомобилях, ведение отчетности в по платежам, а также быстрое и эффективное оперирование данной информацией.