



**КОЛЛЕДЖ КОСМИЧЕСКОГО МАШИНОСТРОЕНИЯ И ТЕХНОЛОГИЙ**

«УТВЕРЖДАЮ»

Заместитель директора колледжа  
по учебной работе

\_\_\_\_\_ Е.В.Антропова

\_\_\_\_\_ 2019 г.

**ЗАДАНИЕ  
на дипломный проект**

студенту группы П1-15 **Соловьеву Игорю Максимовичу**

Тема задания **«Разработка программы для формирования отчетов о выполнении  
учебной нагрузки»**

Утверждена приказом по университету от «\_\_\_» апреля 2019 г. № \_\_\_\_\_

Руководитель дипломного проекта **Гусятинер Л.Б.**

**Содержание работы:**

Введение.

Глава 1. Теоретическая часть

Глава 2. Проектная часть

Глава 3. Организационно-экономическая часть

Заключение.

Список использованной литературы.

Приложения.

Срок выполнения дипломного проекта 15 июня 2019 года

Председатель цикловой комиссии 09.02.03 \_\_\_\_\_  
подпись, Ф.И.О.

Руководитель дипломного проекта \_\_\_\_\_  
подпись, Ф.И.О. руководителя

Задание принял к исполнению \_\_\_\_\_  
подпись студента

Дата выдачи задания «\_\_\_» апреля 2019 года



Государственное бюджетное образовательное учреждение высшего образования  
Московской области

**ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ**

---

**КОЛЛЕДЖ КОСМИЧЕСКОГО МАШИНОСТРОЕНИЯ И ТЕХНОЛОГИЙ**

---

«Допущен к защите»

Председатель цикловой комиссии

\_\_\_\_\_ А.А. Эшанов

« \_\_\_\_\_ » \_\_\_\_\_ 2019 г.

**РАЗРАБОТКА ПРОГРАММЫ ДЛЯ  
ФОРМИРОВАНИЯ ОТЧЕТОВ О  
ВЫПОЛНЕНИИ УЧЕБНОЙ  
НАГРУЗКИ** **Пояснительная записка**

**ДП.09.02.03.19.29ПЗ**

**Дипломный проект**

**Королёв, 2019**

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	4
1. Теоретическая часть .....	6
1.1. Математическая постановка задачи .....	6
1.1.1. Анализ предметной области.....	7
1.1.2. Необходимость автоматизации рабочего места преподавателя....	15
1.2. Анализ имеющихся программных решений.....	18
2. Проектная часть .....	20
2.1. Описание алгоритмов решения.....	20
2.2. Обоснование выбора инструментов, среды, языков программирования .....	22
2.3. Описание структуры программного комплекса .....	26
2.3.1. Логическое и физическое моделирование базы данных .....	28
2.3.2. Разработка автоматизированной системы .....	38
2.3.3. Описание тестовых наборов.....	48
2.4. Описание процедур и функций .....	51
3. Организационно-экономическая часть.....	54
3.1. Эксплуатационный раздел.....	54
3.1.1. Рекомендации по эксплуатации программного продукта .....	54
3.1.2. Руководство пользователя.....	55
3.1.3. Руководство системного администратора .....	57
3.2. Раздел техники безопасности.....	59
3.3. Расчет экономической эффективности применения программного комплекса.....	62
ЗАКЛЮЧЕНИЕ .....	73
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	74
ПРИЛОЖЕНИЯ .....	75

					<b>ДП.09.02.03.19.29 ПЗ</b>			
Изм.	Лист	№ докум.	Подпись	Дата				
Разработал	Соловьев И.М.				РАЗРАБОТКА ПРОГРАММЫ ДЛЯ ФОРМИРОВАНИЯ ОТЧЕТОВ О ВЫПОЛНЕНИИ УЧЕБНОЙ НАГРУЗКИ	Лит.	Лист	Листов
Консультант	Гусятинер Л.Б.						3	104
Рецензент						Группа П1-15		
Конс. по эк. ч	Капралов А.О.							
Норм. контро	Гусятинер О.В.							

## ВВЕДЕНИЕ

В наши дни, в связи с развитием компьютерной техники, появилась возможность автоматизации многих процессов, а также увеличился объем обрабатываемой информации. Вследствие этого, возникла объективная необходимость автоматизировать большую часть сферы человеческой деятельности.

Дипломный проект посвящен разработке модуля ведения фактической нагрузки преподавателей структурного подразделения МГОТУ ККМТ.

В связи с вышеизложенным, было принято решение разработать собственную программу ведения фактической нагрузки преподавателей колледжа.

Целью является разработка модуля ведения фактической нагрузки преподавателей колледжа, чтобы повысить комфортность и снизить трудоемкость при работе с документами для конечного пользователя. Сэкономить ресурсы, расходуемые на подготовку новых документов.

В рамках дипломного проекта были поставлены и решены следующие задачи:

- изучить и проанализировать соответствующую литературу;
- проанализировать деятельность организации;
- провести анализ существующих подходов к расчету нагрузки в средних учебных заведениях;
- разработать подробное техническое задание, в котором будет прописан функционал и системные требования к программному продукту;
- выбрать среду программирования путем выявления преимуществ и недостатков;
- спроектировать и разработать программный продукт;
- рассчитать экономические показатели;

– произвести отладку и тестирование программного продукта в организации.

В процессе проектирования изучены нормативные документы, регламентирующие деятельность учебного документооборота. Для проекта отобран модуль ведения фактической нагрузки преподавателей колледжа. При разработке модуля ведения фактической нагрузки преподавателей колледжа исследованы оптимизирующие алгоритмы. Разработана структура общей для всего комплекса задач базы данных. В проекте рассмотрены вопросы автоматизации управления учебным процессом, экономически обоснована целесообразность внедрения модуля.

Информационной базой исследования послужили описания процедур ведения организации учебного процесса в колледже и ряд документов, такие как годовой график учебного процесса, планы учебного процесса, планирование рабочего и учебного времени студентов.

## **1. Теоретическая часть**

### **1.1. Математическая постановка задачи**

Задачей дипломного проекта является разработка программы для формирования отчетов о выполнении учебной нагрузки.

Автоматизированная система формирования отчетов о выполнении учебной нагрузки должна выполнять следующие функции: работу с рабочими часами, составление расписания, а также систему регистрации и авторизации пользователей.

Цель: создать программное приложение для автоматизации работы преподавателей при подсчете рабочего времени.

Задачи:

- Провести анализ существующих систем, чтобы максимально грамотно внедрить программный продукт.
- Изучить процессы, документы, и информацию подлежащие автоматизации.
- Спроектировать программную модель для приложения.
- Спроектировать графический интерфейс.
- Написание кода.
- Отладка программы.

Приложение должно иметь следующий функционал:

- Наличие понятного и удобного интерфейса;
- Внесение выданного расписания;
- Распределение расписания на числитель и знаменатель с учетом праздников и выходных;
- Подсчет учебных часов в выбранном интервале между двумя датами;

- Вывод количества пар в выбранном интервале между двумя датами;
- Добавление дополнительных пар в сформированную базу данных;
- Удаление пар из сформированной базы данных;
- Формирование отчета с условиями фильтров и интервалом дат в документ Excel для дальнейшей печати;
- Отображение является ли неделя выбранного месяца числителем или знаменателем;
- Работа с личным кабинетом и настройкой аккаунта пользователя;
- Редактирование даты последнего рабочего дня в учебном году;
- Данные пользователя должны быть защищены паролем от несанкционированного доступа.

### **1.1.1. Анализ предметной области**

Под учебным процессом в колледже понимается целенаправленная совместная деятельность руководящего, преподавательского состава, учебно-вспомогательного, административно - хозяйственного персонала и обучающихся, по подготовке высококвалифицированных специалистов.

Основные задачи учебного процесса:

- Подготовка квалифицированных рабочих с начальным профессиональным образованием и квалифицированных специалистов со средним профессиональным образованием.
- Удовлетворение потребности личности в интеллектуальном, культурном и нравственном развитии, приобретении начального образования, специальности и квалификации.
- Реализация в ходе учебного процесса подхода освоения государственных образовательных стандартов и федеральных

государственных образовательных стандартов среднего профессионального образования.

Учебный процесс основывается на государственных образовательных стандартах начального и среднего профессионального образования и федеральных государственных образовательных стандартах начального и среднего профессионального образования по направлениям и специальностям, примерных учебных планах, примерных программах дисциплин, основных образовательных программах по направлениям и специальностям.

Освоение образовательных программ всех видов среднего профессионального образования завершается обязательной итоговой государственной аттестацией выпускников.

Организация учебного процесса призвана обеспечивать:

- Современный научный уровень подготовки специалистов, оптимальное соотношение теоретического и практического обучения.
- Логически правильное, научно и методически обоснованное соотношение, и последовательность преподавания дисциплин, планомерность и ритмичность учебного процесса.
- Органическое единство процесса обучения и воспитания.
- Внедрение в учебный процесс новейших достижений науки и техники, передового опыта.
- Рациональное сочетание традиционных методов передачи и закрепления научной информации с новейшими достижениями педагогики.
- Создание необходимых условий для педагогической деятельности преподавательского состава и освоения студентами профессиональных учебных программ, их творческой самостоятельной работы.

На данный момент сотрудники «МГОТУ» делятся на следующую структуру:



– ректорат включает в себя 19 и занимается управлением «МГОТУ». Ректорат возглавляет Старцева Татьяна Евгеньевна. Также председатель ученого совета, доктор педагогических наук, профессор

– ученый совет включает в себя 40 человек. Учёный совет является необходимым элементом структуры «МГОТУ». Возглавляет общее руководство университетом.;

– наблюдательный совет регионального научно-образовательного кластера «Северо-восток» является Наблюдательным советом базовой образовательной организации - Государственного бюджетного образовательного учреждения высшего образования Московской области «Технологический университет». Включает в себя 22 сотрудника. Председателем совета является глава города Королёв - Ходырев Александр Николаевич. Пятеро из них являются космонавтами.

– преподавательский состав, который в каждом филиале «МГОТУ» свой. В некоторых случаях они преподают в нескольких филиалах.

Структура предприятия:

– высшее образование

- 1) институт техники и цифровых технологий;
- 2) институт проектного менеджмента и инженерного бизнеса;
- 3) институт международного образования;
- 4) аспирантура;

5) инжиниринговый центр «Высокотемпературные композиционные материалы».

– среднее профессиональное образование:

1) Королёвский колледж космического машиностроения и технологии;

2) Королёвский государственный техникум технологии и дизайна.

– дополнительное образование:

1) институт дополнительного образования;

## 2) детский технопарк «Кванториум».

Далее рассмотрим организацию учебного процесса на примере МГОТУ ККМТ.

Государственное образовательное учреждение среднего профессионального образования «Колледж космического машиностроения и технологии», бывший Калининградский механический техникум - основано в 1947 году С. П. Королёвым.

В настоящее время колледж - это структурное подразделение МГОТУ. В нем студенты обучаются по семнадцати профессиональным образовательным программам среднего профессионального образования, базового и повышенного уровней на очной и заочной формах. После успешного усвоения этих программ каждый из выпускников колледжа проходит итоговую государственную аттестацию, и ему выдается диплом государственного образца. Государственная деятельность в колледже осуществляется по следующим специальностям:

- Технология машиностроения;
- Производство летательных аппаратов;
- Радиотехнические комплексы и системы управления космических летательных аппаратов;
- Программное обеспечение вычислительной техники и автоматизированных систем;
- Протезно-ортопедическая и реабилитационная техника;
- Биотехнические и медицинские аппараты и системы;
- Средства механизации и автоматизации;
- Техническое обслуживание и ремонт автомобильного транспорта;
- Автоматизированные системы обработки информации и управления (по отраслям);
- Строительство и эксплуатация зданий и сооружений;

- Монтаж, наладка и эксплуатация электрооборудования промышленных и гражданских зданий;

- Менеджмент;
- Коммерция;
- Экономика и бухгалтерский учет;
- Государственное и муниципальное управление;
- Право и организация социального обеспечения;
- Дизайн (графический).

На данный момент, колледж космического машиностроения и технологий находится под руководством 14 человек:

- Сысоев Дмитрий Васильевич - директор;
- Антропова Елена Викторовна - заместитель директора по учебной работе;
- Видова Галина Михайловна - заместитель директора по учебно-воспитательной работе;
- Гришанова Елена Сергеевна - заместитель директора по учебно-методической работе;
- Трухин Алексей Васильевич - заместитель директора по учебно-производственной работе;
- Капралов Алексей Олегович - заместитель директора по научно-методической работе;
- Никонова Дарья Николаевна - заведующий отделением информационных технологий;
- Потапова Надежда Викторовна - заведующий заочным отделением;
- Никишкина Ольга Вениаминовна - заведующий радиотехническим отделением;
- Шихиева Мария Тахировна - заведующий отделением технологий машиностроения;

- Рябушенко Елена Владимировна - заведующий отделением программирования;
- Ефимочкина Ирина Владимировна - заведующий отделением биотехнологии и ортопедии;
- Никитенко Влада Евгеньевна - заведующий отделением экономики и права;
- Панова Марина Валерьевна - заведующий отделением ракетостроения;

Также преподавательский состав включает в себя 117 преподавателей. Помимо преподавателей, некоторые из руководителей занимаются обучением студентов.

Ниже представлена схема структуры колледжа космического машиностроения и технологий:

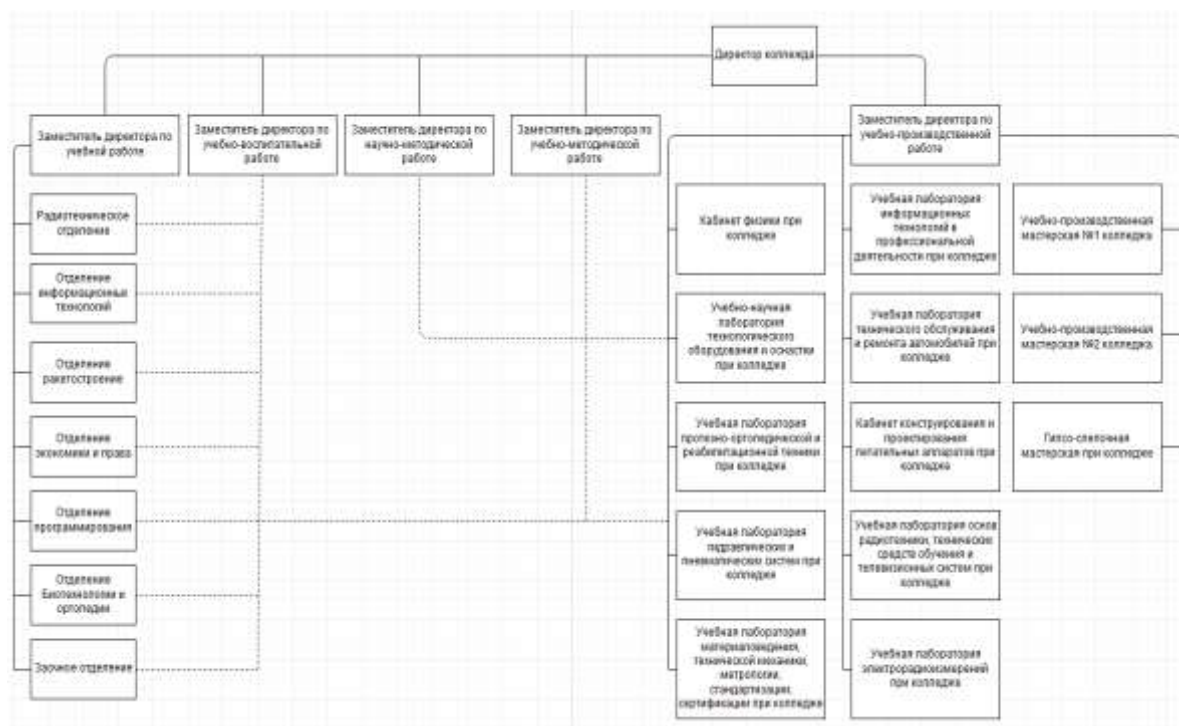


Рис. 1.1. Схематичное отображение структуры ККМТ

Областью разработки программного продукта является учебная часть колледжа космического машиностроения и технологий.

### Организация учебного процесса.

Учебный процесс в колледже организован по полугодиям. Учебный год начинается 1 сентября и заканчивается согласно учебному плану по конкретной специальности. Перечень дисциплин, выносимых на экзаменационную сессию и заканчивающихся итоговой семестровой оценкой, определяется учебным планом специальности. К сессии допускаются студенты, аттестованные предварительно по всем дисциплинам, выполнившие все практические, лабораторные, расчетно-графические и курсовые работы.

Учебный план – это документ, определяющий состав учебных дисциплин, изучаемых в учебном заведении, их распределение по годам в течение всего срока обучения. Включает 3 части:

- график учебного процесса – периоды теоретических занятий, учебной и производственной практики, экзаменационных сессий, дипломного проекта, каникул и их чередования в течение всего срока обучения;
- сводные данные по бюджету времени – общая продолжительность каждого периода учебного процесса по годам и за весь срок обучения;
- план учебного процесса – перечень обязательных, альтернативных и факультативных дисциплин с указанием объёма каждой из них в академических часах и распределения этих часов по неделям, семестрам, учебным годам, сроки сдачи экзаменов, зачётов и курсовых работ и количество часов, отводимых на лекции, семинары, лабораторные работы и упражнения по каждому предмету.

На рисунке 1.2 схематически представлен документооборот учебной нагрузки ККМТ:



Рис. 1.2. Схематичное отображение документооборота учебной нагрузки ККМТ

Председатель цикловой комиссии составляет учебный план в соответствии с федеральным государственным образовательным стандартом среднего профессионального образования. Данный план отправляется заместителю директора по учебной работе. Далее заместитель делит этот процесс на две части:

- работа с преподавателем;
- работа с методистом учебной части ККМТ;

Работа с преподавателем заключается в том, что заместитель директора по учебной работе в соответствии с профессиональным направлением преподавателя выделяет определенное количество пар по дисциплинам. Далее подписывается документ и передается методисту учебной части ККМТ. В учебной части методист составляет расписание согласно учебному плану и подписанным документам между преподавателем и заместителем директора по учебной работе.

После выполнения плана, преподаватель формирует отчет о проделанной работе и отправляет его председателю цикловой комиссии.

Председатель комиссии в свою очередь рассматривает его и отправляет заместителю директора по учебной работе.



среднего профессионального образования, все это не позволяет в полном объеме и с запланированной эффективностью использовать эти несколько устаревшие разработки. А учитывая явную тенденцию формирования в нашей стране информационного общества, необходимо учитывать еще и специфику информационной среды подразделения ККМТ и изменившуюся методологию формирования учебного процесса, поэтому число работ, описывающих управление учебным процессом с учетом современных условий и требований, соответственно уменьшается.

Кроме того, во многих работах, посвященных автоматизации управления учебным процессом, рассматриваются отдельные частные вопросы: составление расписания учебных занятий, формирование учебных планов по специальности, расчет нагрузки и т. д., без учета их взаимосвязей, что не позволяет достигнуть эффективного решения общей задачи в целом.

Автоматизация любой системы только тогда целесообразна и дает практический результат, когда будут автоматизированы наиболее сложные и трудоемкие процессы этой системы. Наиболее сложной подсистемой системы управления подразделением ККМТ является подсистема автоматизированного управления учебным процессом.

Автоматизированную систему управления учебным процессом в подразделении ККМТ предлагается скомпоновать из следующих относительно независимых автоматизированных подсистем:

- подсистемы планирования учебного процесса;
- подсистемы управления учебной деятельностью студентов;
- подсистемы информационного обеспечения учебного процесса;
- подсистемы системного анализа и аудита учебного процесса.

В свою очередь наиболее сложной, трудоемкой и подверженной, как показывает практика, наибольшему количеству дефектов является подсистема планирования учебного процесса. В то же время она является наиболее значимой составляющей в системе обеспечения качества



подготовки специалистов и в системе трудозатрат, при этом также необходимо учитывать, насколько совершенной будет ее структура, настолько успешной будет автоматизация и оптимизация других подсистем управления учебным процессом в подразделении ККМТ. Поэтому подсистему планирования учебного процесса необходимо разработать в таком виде, чтобы она обеспечила выполнение следующих операций и функций:

- автоматизированное планирование учебных планов по специальностям и направлениям;
- автоматизированный расчет учебной работы и планирование педагогической нагрузки подразделения ККМТ, кафедр, преподавателей;
- автоматизированная работа с праздничными днями, выходными и отпусками преподавателей;
- автоматизированное формирование расписания учебных занятий с учетом занятости преподавательского состава и ресурсов учебно-лабораторной базы подразделения ККМТ.
- Возможность редактирования сформированного учебного расписания без потери данных.

Кроме того, при разработке подсистемы планирования учебного процесса должны быть реализованы принципы связи теории с практикой, согласования объема учебной информации с определенным государственным образовательным стандартом среднего профессионального образования, последовательности изучения учебных предметов и их взаимосвязи, оформленных в виде структурно-логической схемы подготовки специалистов. Только в этом случае можно добиться оптимального планирования учебного процесса в подразделении ККМТ.

## 1.2. Анализ имеющихся программных решений

Прежде, чем приступить к проектированию необходимо обосновать целесообразность предстоящей разработки.

В настоящий момент на рынке можно найти большое количество аналогичных по профилю программных продуктов:

- «БИТ: Учёт нагрузки преподавателей». По информации БИТ, программа позволит сократить трудозатраты на учет нагрузки преподавателей. Кроме того, продукт поможет получать отчетность, обрабатывать данные и формировать результирующую информацию в реальном времени;
- Информационная система расчета нагрузки преподавателей как приложение платформы «1С: Предприятие 8». Информационная система расчета нагрузки преподавателей факультета открытого образования (ИСРНО) разработана на основании решения проректора по учебной работе Финакадемии при Правительстве РФ.

Как уже упоминалось выше, готовые системы автоматизирующие учебный процесс, либо обладают явно излишним функционалом, что негативно отражается на использовании системы. Либо требуют индивидуального подхода, что существенно увеличивает стоимость внедрения. В результате чего было принято решение создать самостоятельно необходимую программу и только с необходимым функционалом.

Проблемой автоматизации управления своими подразделениями занимаются многие колледжи. В ККМТ автоматизирован ряд задач, например, формирование объема учебной нагрузки, осуществляемое средствами программного пакета Microsoft Excel. К сожалению, несмотря на позитивные сдвиги, наметившиеся в последние годы, распределение и учет выполнения учебной нагрузки до сих пор выполняется вручную.

В процессе исследования организации управления учебным процессом в колледже, были выявлены недостатки:

- минимально используется персональный компьютер. ПК применяется в основном для подготовки отчетности, которая формируются вручную (MS Excel, MS Word);
- нерациональное распределение обязанностей между исполнителями, что приводит к потерям рабочего времени;
- отсутствие средств для оптимального решения основных операций;
- нерациональное использование средств информационных технологий.

Среди технических недостатков следует отметить отсутствие единой информационной системы, обеспечивающей централизованное хранение данных и автоматизированной подготовки всех необходимых печатных документов.

Потребность в разработке нового проекта обусловлена необходимостью обеспечить более высокую производительность труда, большую надежность и достоверность информации, лучшую ее сохранность.

Программный продукт также должен обеспечить возможность оперативного анализа учебного процесса в учебном заведении и оперативно выявлять все недостатки его ведения.

Поскольку разработка комплексной системы автоматизированного управления учебным процессом является сложным и дорогостоящим, требующим привлечения значительных ресурсов, в том числе - коллектива исполнителей, предлагается для начала ограничиться проектом, решающим ряд частных задач.

## 2. Проектная часть

### 2.1. Описание алгоритмов решения

Задача 1. Составление алгоритма работы программы.

Цель задачи: Алгоритм составляется на начальном этапе разработки проекта. Он предназначен для разбиения работы программы на более простые шаги, которые в последующем помогут решить поставленную задачу.

Функции: Объединение всех поставленных задач и целей в единую систему, с дальнейшим переносом на язык программирования.

Входные данные:

- Расписание преподавателя.

Выходные данные:

- Сформированный отчет о выполнении учебной нагрузки.

Задача 2. Выбор ПО для реализации проекта.

Цель задачи: Подобрать язык программирования и среду разработки для реализации проекта.

Функции: Инструментарий программиста для разработки ПО.

Входные данные: Информация о языках программирования и средах разработки.

Выходные данные: Анализ языков программирования с дальнейшим выбором для разработки.

Лицо, для которого ставится задача: Программист.

Задача 3. Разработка необходимой документации.

Цель задачи: Подготовить перечень документов, относящихся к разработке данного ПО.

Функции: Информирование пользователя или заказчика о возможностях программы и её функционале.

Входные данные: Информация по оформлению документов согласно ГОСТ.

Выходные данные: Готовая документация проекта.

Лицо, для которого ставится задача: программист.

Задача 4. Разработать логическую и физическую модели баз данных.

Цель задачи: Показать всевозможные связи и взаимодействия данных между собой внутри одной базы данных.

Функции: Аналитическая схема взаимодействия процессов внутри системы.

Входные данные: Информация для внесения в таблицы базы данных.

Выходные данные: Организационные связи между таблицами внутри одной базы данных.

Лицо, для которого ставится задача: программист.

Задача 5. Разработать базу данных.

Цель задачи: Объединить данные в единую систему.

Функции: Добавление, чтение, удаление.

Входные данные:

- Информация о преподавателе;
- Информация о расписании;
- Информация о праздниках;
- Информация о дополнительных рабочих днях.

Выходные данные: Таблица, содержащая все данные.

Лицо, для которого ставится задача: программист.

Задача 6. Разработать интерфейс.

Цель задачи: Разработать пользовательскую оболочку для данного проекта.

Функции: Связующее звено между программой и пользователем.

Входные данные: концепция создания и работы приложения.

Выходные данные: пользовательский интерфейс.

Лицо, для которого ставится задача: программист.

Задача 7. Запрограммировать необходимые компоненты программы.

Цель задачи: Реализовать отдельные компоненты для правильной работы программы.

Функции: Выполнение надлежащих функций для каждого из модулей программы.

Входные данные: Команды языка программирования.

Выходные данные: Готовые модули для работы программы.

Лицо, для которого ставится задача: программист.

Задача 8. Провести тестирование работоспособности программы.

Цель задачи: Произвести анализ работоспособности программы.

Функции: Поиск неисправностей на основе входных тестов программы.

Входные данные: Исходные тесты для проверки программы.

Выходные данные: Перечень найденных ошибок.

Лицо, для которого ставится задача: программист

Задача 9. Исправить возникшие ошибки.

Цель задачи: Довести продукт до правильного состояния.

Функции: Устранение неисправностей с целью лучшего обеспечения функциональности программы.

Входные данные: Перечень найденных ошибок.

Выходные данные: исправленный продукт.

Лицо, для которого ставится задача: программист.

## **2.2. Обоснование выбора инструментов, среды, языков программирования**

В рамках дипломного проекта было принято решение разработать автоматизированную систему, включающую базу данных по работе с учебной нагрузкой преподавателей. Учитывая возможности, имеющегося в

учебном заведении ККМТ оборудования и программного обеспечения, необходимо создать современный программный продукт, избегая таких недостатков существующих коммерческих предложений, как высокая стоимость внедрения и сопровождения и слабая ориентированность на пользователя с разной профессиональной подготовкой. Также необходимо уделить особое внимание надежности приложения и простоте его интерфейса.

Поэтому для разработки автоматизированной системы были выбраны:

- СУБД MySQL — предоставляет мощные средства для доступа, настройки, администрирования, разработки всех компонентов базы данных и управления ими. MySQL — это реляционная система управления базами данных. То есть данные в ее базах хранятся в виде логически связанных между собой таблиц, доступ к которым осуществляется с помощью языка запросов SQL. MySQL — свободно распространяемая система. Кроме того, это достаточно быстрая, надежная и, главное, простая в использовании СУБД. Работать с MySQL можно не только в текстовом режиме, но и в графическом. Существует очень популярный визуальный интерфейс для работы с этой СУБД — dbForgeStudio 2017 forMySQL. Этот интерфейс позволяет значительно упростить работу с базами данных в MySQL.

- dbForgeStudio 2017 forMySQL имеет следующие возможности:

- 1) Дизайнер таблиц включает в себя визуальные редакторы для столбцов, индексов, первичных ключей, внешних ключей, проверки целостности, статистики, свойств хранилищ. Дизайнер позволяет редактировать данные таблицы и синхронно перемещаться по визуальному редактору;

- 2) Создание диаграмму базы данных для визуализации ее структуры и построения логических отношений между таблицами;

- 3) Ускоренное написание SQL кода при помощи интеллектуального контекстно-зависимого дополнения, выполняет автоматическое

форматирование и рефакторинг, улучшает читаемость кода, повышает производительность и снижает затраты.

4) Функция поиска ошибок в сценариях SQL, хранимых процедурах, триггерах и функциях, наблюдение за их поведением во время выполнения, пошаговое выполнение, просмотр стека вызовов;

5) Перемещение схемы из песочницы в рабочую среду или обновление базы данных, разрабатываемую несколькими командами.

6) Создание полноценной документации базы данных SQL Server всего за несколько кликов. Широкий выбор опций позволяет настроить внешний вид документации.

– Среда Microsoft Visual Studio 2017 и язык программирования C#.Microsoft Visual Studio — линейка продуктов компании Майкрософт, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом. Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и как отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. VisualStudio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (например, Subversion и VisualSourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода). Главным преимуществом Visual Studio 2012 является производительность.



Обеспечивает возможность создания разнообразных приложений на основе одного набора навыков. [1, с. 47]

– C# — объектно-ориентированный язык программирования. C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML. [2, с. 76]

– Основные достоинства C#:

1) C# создавался параллельно с каркасом Framework .Net и в полной мере учитывает все его возможности — как FCL, так и CLR;

2) C# является полностью объектно-ориентированным языком, где даже типы, встроенные в язык, представлены классами;

3) C# является мощным объектным языком с возможностями наследования и универсализации;

4) C# является наследником языков C/C++, сохраняя лучшие черты этих популярных языков программирования. Общий с этими языками синтаксис, знакомые операторы языка облегчают переход программистов от C++ к C#;

5) сохранив основные черты своего великого родителя, язык стал проще и надежнее. Простота и надежность, главным образом, связаны с тем, что на C# хотя и допускаются, но не поощряются такие опасные свойства C++ как указатели, адресация, разыменование, адресная арифметика;

6) благодаря каркасу Framework .Net, ставшему надстройкой над операционной системой, программисты C# получают те же преимущества работы с виртуальной машиной, что и программисты Java. Эффективность кода даже повышается, поскольку исполнительная среда CLR представляет

собой компилятор промежуточного языка, в то время как виртуальная Java-машина является интерпретатором байт-кода;

7) мощная библиотека каркаса поддерживает удобство построения различных типов приложений на C#, позволяя легко строить Web-службы, другие виды компонентов, достаточно просто сохранять и получать информацию из базы данных и других хранилищ данных.

8) Реализация, сочетающая построение надежного и эффективного кода, является немаловажным фактором, способствующим успеху C#.

### **2.3. Описание структуры программного комплекса**

ПДФООВУН имеет следующую структуру проекта, состоящую из:

- Программы, состоящая из нескольких форм, отвечающая за подключение и работу с БД;
- базы данных, для хранения данных и слежение за процессом доставки.

Более детально структура изображена на рисунке (рис. 2.1.):

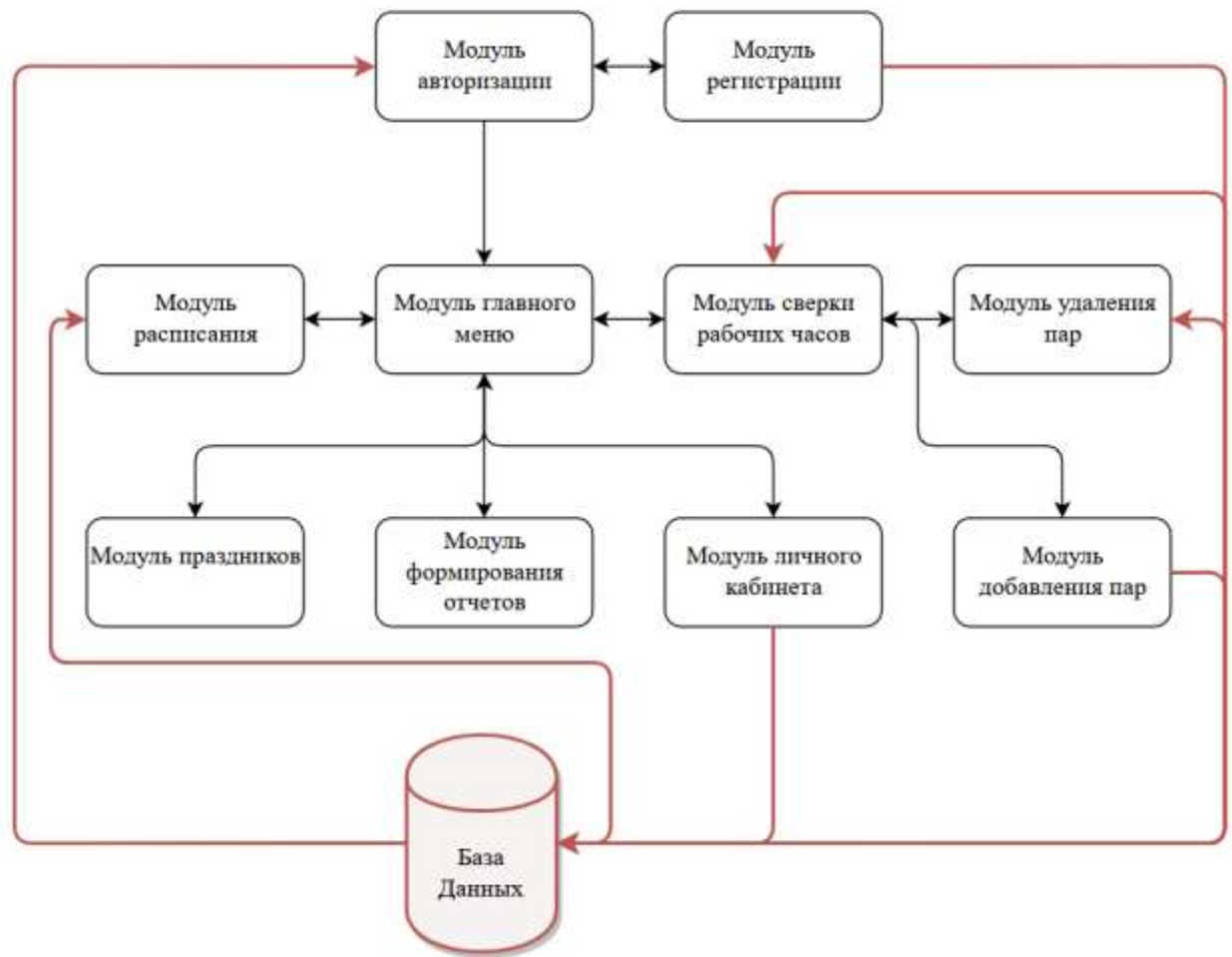


Рис. 2.1.Схематичное отображение структуры проекта

### 2.3.1. Логическое и физическое моделирование базы данных

Логическая организация БД — это представление пользователя о той предметной области, информация о которой должна храниться в БД.

Результатом логического проектирования является концептуальная схема БД. Эта модель отражает три вида информации:

- сведения об объектах предметной области;
- сведения о свойствах объектов;
- сведения об отношениях между объектами.

Такая модель не зависит от: физической среды, типа ЭВМ, операционной системы, СУБД. [5][6]

В предметной области «учебная нагрузка» можно выделить следующие объекты:

- объекты: праздники, преподаватели, дополнительные пары, расписание, рабочие дни;

В предметной области можно определить следующие сущности: праздники, преподаватели, дополнительные пары, расписание, рабочие дни;

Сущность **праздники** содержит следующие атрибуты: код праздника, дата праздника. В качестве ключевого атрибута был принят код товара.

Сущность **преподаватели** содержит следующие атрибуты: код преподавателя, фамилия, имя и отчество преподавателя. В качестве ключевого атрибута был принят код преподавателя.

Сущность **расписание** содержит следующие атрибуты: код расписания, день недели, номер пары, тип недели, код преподавателя, название группы, название дисциплины. В качестве ключевого атрибута был принят код расписания.

Сущность **дополнительные пары** содержит следующие атрибуты: код дополнительной пары, номер пары, название группы, код преподавателя,

дата дополнительной пары, название дисциплины. В качестве ключевого атрибута был принят код дополнительной пары.

Сущность рабочие дни содержит следующие атрибуты: код рабочего дня, дата рабочего дня, код преподавателя, название группы, номер пары, название дисциплины. В качестве ключевого атрибута был принят код рабочего дня.

Рассмотрим типы связей между сущностями преподаватель и расписание. Поскольку преподаватель должен иметь как минимум два расписания, на числитель и знаменатель, то связь один-ко-многим.

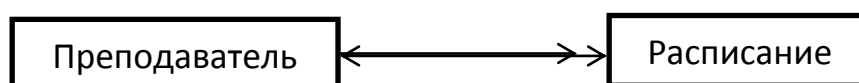


Рис. 2.2 - Связь между сущностями Преподаватель и Расписание

Рассмотрим типы связей между сущностями преподаватель и дополнительных пар. Поскольку один преподаватель может иметь несколько дополнительных пар, то связь один-ко-многим.

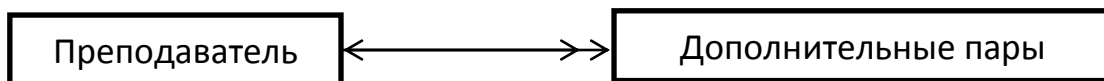


Рис. 2.3. - Связь между сущностями Преподаватель и Дополнительные пары

Рассмотрим типы связей между сущностями преподаватель и рабочие дни. Поскольку один преподаватель может иметь несколько рабочих дней, то связь один-ко-многим.

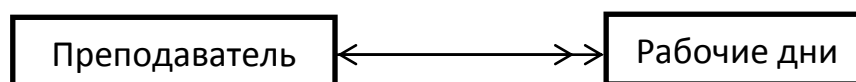


Рис. 2.4. - Связь между сущностями Преподаватель и Рабочие дни

Инфологическая модель предметной области в виде ER-диаграммы изображена на рисунке 2.5.

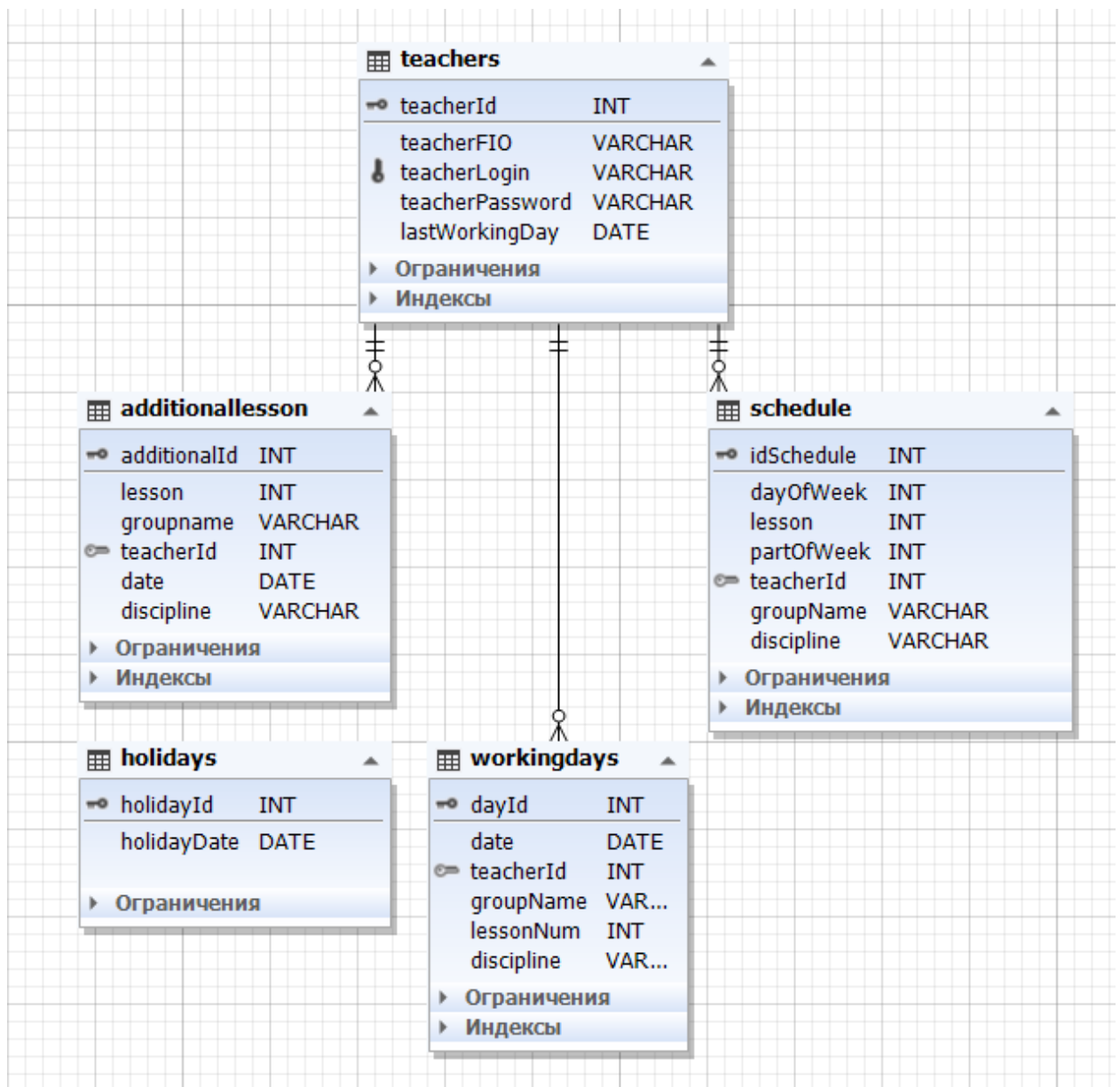


Рис. 2.5. ER-диаграмма «Учебная нагрузка»

На логическом этапе проектирования модели базы данных описывается перевод сущностей инфологической модели в отношения и доказывается, что отношения находятся в первой нормальной форме, второй нормальной форме и третьей нормальной форме.

Отношение «Преподаватель» находится в первой нормальной форме, так как на пересечении каждого столбца и строки находятся только элементарные значения атрибутов, не имеет повторяющихся записей. Отношение находится во второй нормальной форме, так как таблица

находится в первой нормальной форме и не содержит неполных функциональных зависимостей вторичных атрибутов от первичного ключа. Отношения находится в третьей нормальной форме, так как таблица находится во второй нормальной форме, и не содержат транзитивных зависимостей.

Структуру отношения «Преподаватель» можно представить в виде таблицы 2.1.

Таблица 2.1.

Структура отношения Преподаватель

Имя поля	Тип данных
Код преподавателя	Числовой
Фамилия, имя, отчество	Текстовый
Логин	Текстовый
Пароль	Текстовый
Последний рабочий день	Дата

Отношение «Расписание» находится в первой нормальной форме, так как на пересечении каждого столбца и строки находятся только элементарные значения атрибутов, не имеет повторяющихся записей. Отношение находится во второй нормальной форме, так как таблица находится в первой нормальной форме и не содержит неполных функциональных зависимостей вторичных атрибутов от первичного ключа. Отношения находится в третьей нормальной форме, так как таблица находится во второй нормальной форме, и не содержат транзитивных зависимостей.

Структуру отношения «Расписание» можно представить в виде таблицы 2.2.

Таблица 2.2.

## Структура отношения Расписание

Имя поля	Тип данных
Код расписания	Числовой
День недели	Числовой
Имя поля	Тип данных
Номер пары	Числовой
Часть недели	Числовой
Код преподавателя	Числовой
Название группы	Текстовой
Название дисциплины	Текстовый

Отношение «Дополнительные пары» находится в первой нормальной форме, потому что нет повторяющихся полей, в строках всех атрибутов находятся элементарные значения. Находится во второй нормальной форме, потому что находится в первой нормальной форме и не ключевое поле идентифицируется полным набором ключевых полей. Находится в третьей нормальной форме, потому что удовлетворяет условию второй нормальной формы и ни одно из не ключевых полей таблицы не идентифицируется с помощью другого не ключевого поля.

Структуру отношения «Дополнительные пары» можно представить в виде таблицы 2.3.

Отношение «Праздники» не имеет повторяющихся полей, на пересечении каждого столбца и каждой строки находятся только элементарные значения атрибутов, следовательно, отношение находится в первой нормальной форме. Кроме того, любое не ключевое поле однозначно идентифицируется полным набором ключевых полей. Поэтому отношение находится во второй нормальной форме.



Таблица 2.3.

## Структура отношения Дополнительные пары

Имя поля	Тип данных
Код дополнительной пары	Числовой
Название группы	Текстовый
Код преподавателя	Числовой
Имя поля	Тип данных
Дата	Дата
Название дисциплины	Текстовый

Отношение находится в третьей нормальной форме, потому что удовлетворяет условию второй нормальной формы и ни одно из не ключевых полей таблицы не идентифицируется с помощью другого не ключевого поля.

Структуру отношения «Праздники» можно представить в виде таблицы 2.4.

Таблица 2.4.

## Структура отношения Праздники

Имя поля	Тип данных
Код праздника	Числовой
Дата праздника	Дата

Отношение «Рабочие дни» не имеет повторяющихся записей, в нём отсутствуют повторяющиеся группы полей, строки не упорядочены и столбцы не упорядочены. Значит, отношение находится в первой нормальной форме. Кроме того, любое не ключевое поле однозначно идентифицируется полным набором ключевых полей. Поэтому отношение находится во второй нормальной форме. В отношении ни одно из не ключевых полей таблицы не

идентифицируется с помощью другого не ключевого поля. Следовательно, отношение находится в третьей нормальной форме.

Структуру отношения «Рабочие дни» можно представить в виде таблицы 2.5.

Таблица 2.5.

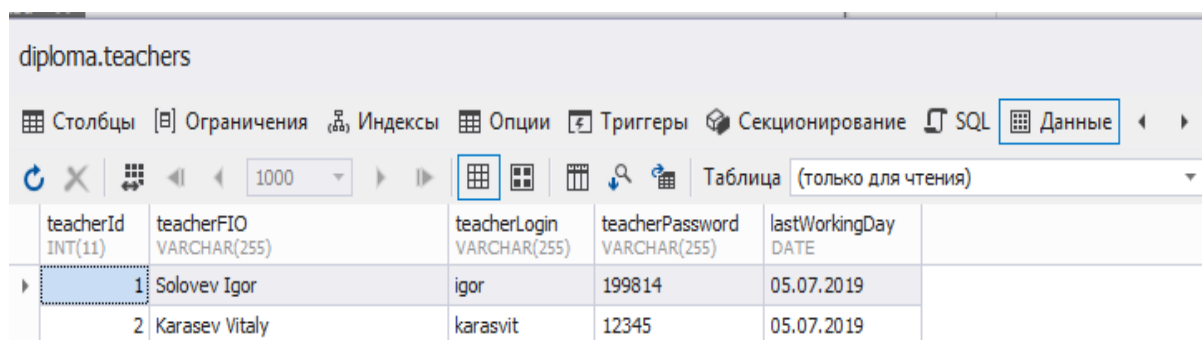
Структура отношения Рабочие дни

Имя поля	Тип данных
Код рабочего дня	Числовой
Дата	Дата
Код преподавателя	Числовой
Название группы	Текстовый
Номер пары	Числовой
Название дисциплины	Текстовый

Для разработки программного продукта используется интегрированная среда разработки VisualStudio 2017, язык C#. [3]

Для создания базы данных была выбрана система управления базами данных dbForgeStudio 2019 forMySQL. [8]

Таблица «Преподаватель» содержит общую информацию о преподавателях.



teacherId	teacherFIO	teacherLogin	teacherPassword	lastWorkingDay
INT(11)	VARCHAR(255)	VARCHAR(255)	VARCHAR(255)	DATE
1	Solovev Igor	igor	199814	05.07.2019
2	Karasev Vitaly	karasvit	12345	05.07.2019

Рисунок 2.6 – Структура таблицы «Преподаватель» в режиме конструктора

Таблица «Расписание» содержит общую информацию о расписании.

diploma.schedule

Столбцы [В] Ограничения Индексы Опции Триггеры Секционирование SQL Данные

1000 Таблица (только для чтения)

idSchedule INT(11)	dayOfWeek INT(11)	lesson INT(11)	partOfWeek INT(11)	teacherId INT(11)	groupName VARCHAR(255)	discipline VARCHAR(255)
241	0	0	1	1	P1-12	OOP
242	0	3	1	1	P1-12	OOP
243	2	1	1	1	P1-113	SQL
244	3	0	1	1	P2-14	Math
245	4	2	1	1	IS-11	1234
246	0	0	0	1	P2-14	Rus
247	0	3	0	1	P2-14	Documentation

Рис. 2.7. Структура таблицы «Расписание» в режиме конструктора

Таблица «Дополнительные пары» содержит информацию о дополнительных парах.

diploma.additionallesson

Столбцы [В] Ограничения Индексы Опции Триггеры Секционирование SQL Данные

1000 Таблица (только для чтения)

additionalId INT(11)	lesson INT(11)	groupname VARCHAR(255)	teacherId INT(11)	date DATE	discipline VARCHAR(255)
44	2	Group	1	01.09.2018	Discipline
45	3	P1-11	1	18.09.2018	Programming

Рис. 2.8. Структура таблицы «Дополнительные пары» в режиме конструктора

В таблице «Праздники» хранится информация о праздниках.

diploma.holidays

Столбцы | Ограничения | Индексы | Опции | Триггеры | Секционирование | SQL | **Данные**

1000 | Таблица (только для чтения)

holidayId INT(11)	holidayDate DATE
21	03.10.2018
22	04.10.2018
23	05.10.2018
24	31.12.2018
25	01.01.2019
26	02.01.2019
27	03.01.2019
28	04.01.2019
29	05.01.2019
30	06.01.2019
31	07.01.2019
32	08.01.2019
33	23.02.2019
34	24.02.2019
35	08.03.2019
36	09.03.2019
37	10.03.2019
38	01.05.2019
39	02.05.2019
40	03.05.2019
41	04.05.2019
42	05.05.2019
43	09.05.2019
44	10.05.2019

Рис. 2.9. Структура таблицы «Праздники» в режиме конструктора

В таблице «Рабочие дни» хранится информация о рабочих днях.

diploma.workingdays

Столбцы Ограничения Индексы Опции Триггеры Секционирование SQL Данные

1000 Таблица (только для чтения)

dayId INT(11)	date DATE	teacherId INT(11)	groupName VARCHAR(255)	lessonNum INT(11)	discipline VARCHAR(255)
7002	03.09.2018	1	P2-14	1	Rus
7003	03.09.2018	1	P2-14	4	Documentation
7004	10.09.2018	1	P1-12	1	OOP
7005	10.09.2018	1	P1-12	4	OOP
7006	12.09.2018	1	P1-113	2	SQL
7007	13.09.2018	1	P2-14	1	Math
7008	14.09.2018	1	IS-11	3	1234
7009	17.09.2018	1	P2-14	1	Rus
7010	17.09.2018	1	P2-14	4	Documentation
7011	24.09.2018	1	P1-12	1	OOP
7012	24.09.2018	1	P1-12	4	OOP
7013	26.09.2018	1	P1-113	2	SQL
7014	27.09.2018	1	P2-14	1	Math
7015	28.09.2018	1	IS-11	3	1234
7016	01.10.2018	1	P2-14	1	Rus
7017	01.10.2018	1	P2-14	4	Documentation
7018	08.10.2018	1	P1-12	1	OOP
7019	08.10.2018	1	P1-12	4	OOP
7020	10.10.2018	1	P1-113	2	SQL
7021	11.10.2018	1	P2-14	1	Math
7022	12.10.2018	1	IS-11	3	1234
7023	15.10.2018	1	P2-14	1	Rus
7024	15.10.2018	1	P2-14	4	Documentation
7025	22.10.2018	1	P1-12	1	OOP
7026	22.10.2018	1	P1-12	4	OOP

Рис. 2.10. Структура таблицы «Рабочие дни» в режиме конструктора

Между таблицами созданы связи и в БД обеспечивается целостность данных. На рисунке 2.11 представлена схема данных предметной области «Учебная нагрузка» в СУБД MySQL:

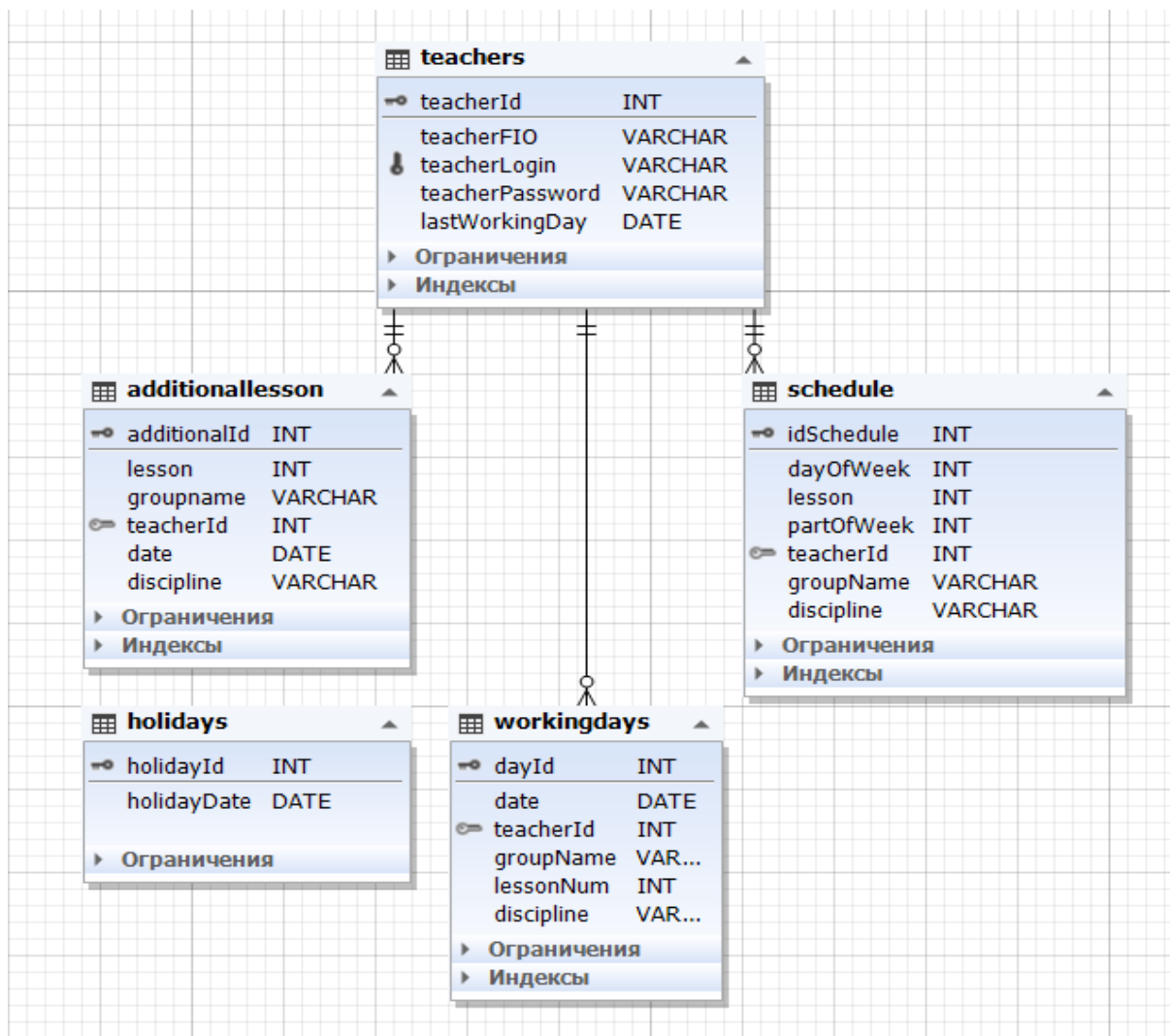


Рис. 2.11. Схема базы данных информационной системы «Учебная нагрузка»

### 2.3.2. Разработка автоматизированной системы

Описание структуры модулей программы

Модуль авторизации

На рисунке (рис. 2.14.) представлено окно авторизации:

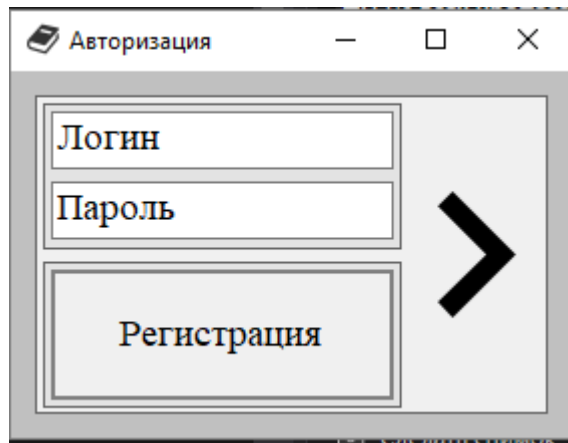
The image shows a window titled 'Авторизация' (Authorization). Inside the window, there are three input fields stacked vertically. The first field is labeled 'Логин' (Login), the second is labeled 'Пароль' (Password), and the third is a button labeled 'Регистрация' (Registration). To the right of these fields is a large, bold, black arrow pointing to the right.

Рис. 2.14. Авторизация

Чтобы успешно подключиться, необходимо заполнить все поля и нажать кнопку с изображением стрелки вправо.

#### Модуль регистрации

На рисунке (рис 2.15.) представлено окно регистрации, чтобы попасть в нее из формы регистрации, требуется нажать на кнопку «Регистрация» в форме авторизации. Далее требуется заполнить все поля и нажать на кнопку «Регистрация», чтобы в будущем была возможность авторизации в программе. Либо нажать на кнопку «Назад», чтобы вернуться к форме авторизации.

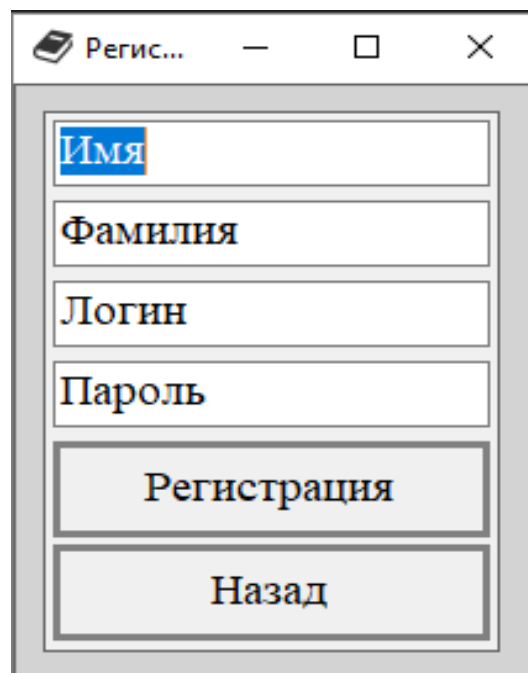
The image shows a window titled 'Регис...' (Registration). Inside the window, there are five input fields stacked vertically. The first field is labeled 'Имя' (Name), the second is labeled 'Фамилия' (Surname), the third is labeled 'Логин' (Login), the fourth is labeled 'Пароль' (Password), and the fifth is a button labeled 'Регистрация' (Registration). Below the 'Регистрация' button is another button labeled 'Назад' (Back).

Рис. 2.15. Форма регистрации

### Модуль главного меню

На рисунке (рис 2.16.) представлено окно главного меню, которое является связующим элементом между модулями программы. На выбор представлены пять кнопок для перехода между модулями:

- Выбрать расписание на месяц;
- Сверить рабочие часы;
- Редактирование праздников;
- Формирование отчета;
- Личный кабинет.

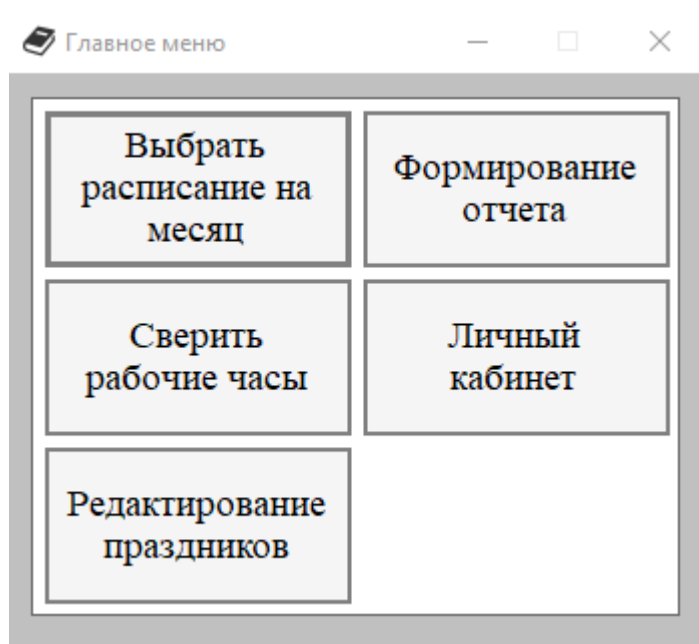


Рис. 2.16. Форма главного меню

На рисунке(рис. 2.17.) представлен модуль расписания, в котором представлены следующие функции для работы с расписанием:

- выбор дня недели для заполнения парами;
- выбор числителя или знаменателя для заполнения парами, либо для просмотра уже готового расписания;
- панель с элементами для заполнения расписания по номеру пары, в котором необходимо ввести пару и дисциплину;
- кнопка для очистки таблицы расписания;



- кнопка для применения изменений;
- схематичное отображенное расписания;
- нажатие на ячейку очищает ее от записей;
- Нажатие на кнопку применить, закрепляет в ячейках информацию о парах и дисциплинах;
- Нажатие на кнопку назад возвращает назад в главное меню.

Редактор расписания

Выбор дня недели

Понедельник ▾

Числитель/Знаменатель

Знаменатель ▾

Назад

Группа Дисциплина

Первая пара

Вторая пара

Третья пара

Четвертая пара

Пятая пара

Применить

Редактирование расписания

Очистить таблицу

Применить изменения

Дни/Пары	Первая пара	Вторая пара	Третья пара	Четвертая пара	Пятая пара
Понедельник	P1-12 OOP			P1-12 OOP	
Вторник					
Среда		P1-113 SQL			
Четверг	P2-14 Math				
Пятница			IS-11 1234		
Суббота					

Рис. 2.17. Модуль для работы с расписанием пар

### Модуль сверка рабочих часов

На рисунке (рис. 2.18.) представлен модуль сверки рабочих часов. Он включает в себя следующие функции:

- Выбор интервала для отображения рабочих часов с учетом праздников и выходных;
- Кнопка «добавить пару» для перехода в модуль добавления пары;
- Кнопка «удалить пару» для перехода в модуль удаления пары;
- Выбор месяца для отображения рабочих дней, выходных и праздников
- Панель, в которой показаны числители и знаменатели по неделям в выбранном месяце;
- Календарь, в котором отображены рабочие дни, выходные и праздники;
- При нажатии на ячейку календаря, будет отображено всплывающее окно с информацией:
  - 1) Если выбрана ячейка светло-зеленого цвета, то будет отображена информация о парах, группах и дисциплинах на выбранный день месяца;
  - 2) Если выбрана ячейка розового цвета, то будет отображена информация о том, что выбранный день месяца является выходным;
  - 3) Если выбрана ячейка светло-серого цвета, то ничего не произойдет.

Рабочее время

Выберите интервал дат для отображения часов

С 1 июня 2019 г. До 12 июня 2019 г.

Количество рабочих часов и пар составляет:  
Часов: 14  
Пар: 7

Добавить пару

Назад Удалить пару

Месяц: Сентябрь, 2018

Пн	Вт	Ср	Чт	Пт	Сб	Вс
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Числитель/Знаменатель

Числитель  
01.09.2018 - 07.09.2018  
Знаменатель  
08.09.2018 - 14.09.2018  
Числитель  
15.09.2018 - 21.09.2018  
Знаменатель  
22.09.2018 - 28.09.2018  
Числитель  
29.09.2018 - 05.10.2018

Рис. 2.18. Добавление и изменение данных в таблице

Модуль добавления дополнительных пар

На рисунке (рис. 2.19.) представлен модуль добавления дополнительных пар. Данный модуль включает в себя следующие функции:

- выбор месяца дополнительной пары;
- выбор дня дополнительной пары;
- выбор номера дополнительной пары;
- выбор группы дополнительной пары;
- выбор дисциплины дополнительной пары;
- кнопка «добавить пару» для добавления дополнительной пары к рабочим дням, учитывая вышеперечисленную информацию;

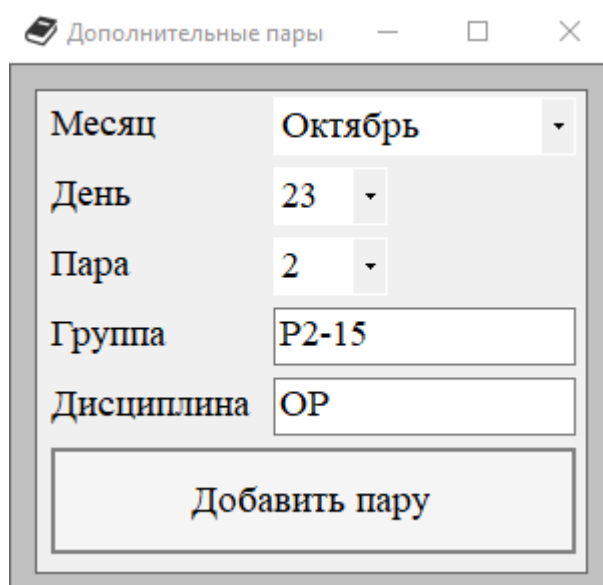


Рис. 2.19. Добавление дополнительной пары

#### Модуль удаления пар

На рисунке (рис. 2.20.) представлен модуль удаления пар. Данный модуль включает в себя следующие функции:

- Просмотр всех пар на выбранную дату;
- Выбор пары из перечня пар на данную дату;
- Кнопка «удалить» для удаления пары;
- Кнопка назад для возвращения к модулю сверки рабочих часов.

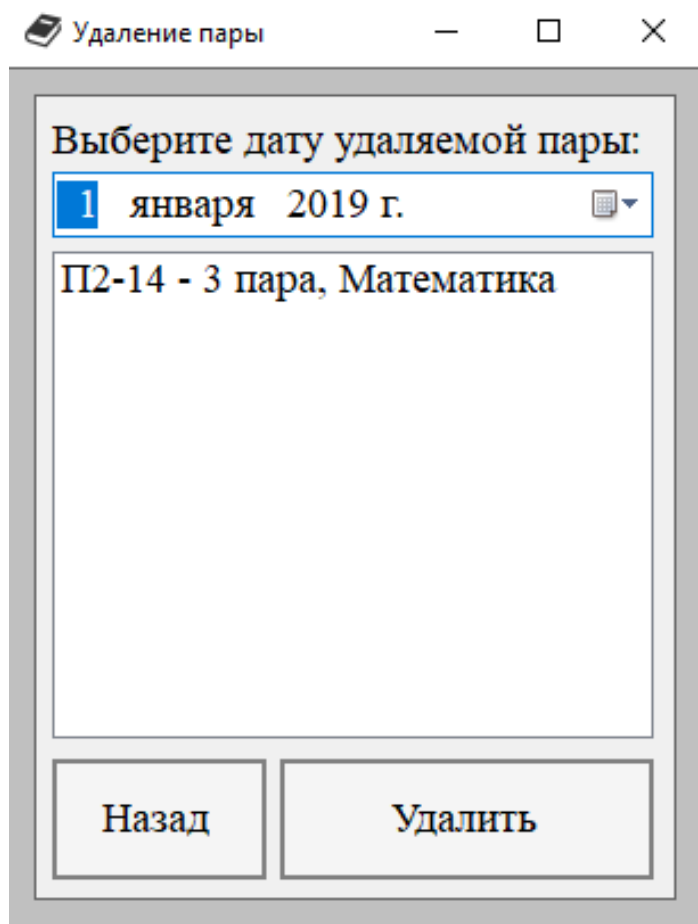


Рис. 2.20. Удаление пары

#### Модуль праздников и дополнительных выходных

На рисунке (рис. 2.21.) представлен модуль для редактирования праздников и дополнительных выходных. Данный модуль включает в себя следующие функции:

- Мини-календарь для выбора даты праздника или дополнительного выходного дня;
- Кнопка с рисунком стрелки направленной пары для перемещения даты из календаря в окно с перечнем праздников и дополнительных выходных;
- Кнопка с рисунком стрелки влево для удаления даты праздника или дополнительного выходного из перечня дат;
- Элемент управления «ListBox» с перечнем праздников и дополнительных выходных;

- Кнопка «Назад» для возврата обратно в модуль главного меню;

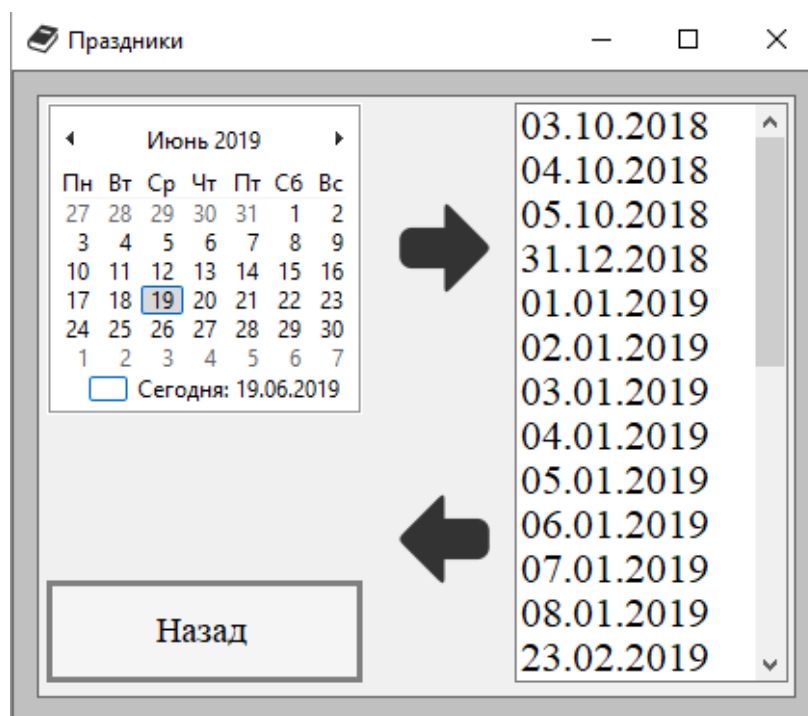


Рис. 2.21. Модуль праздников и дополнительных выходных

#### Модуль формирования отчета

На рисунке (рис. 2.22.) представлен модуль для формирования отчетов об учебной нагрузке. Он предназначен для окончательной стадии работы с подсчетом количества пар с учетом праздников и дополнительных выходных. Данный модуль включает в себя следующие функции:

- Выбор интервала дат для вывода пар, находящихся в этом интервале, в таблицу;
- Фильтр «Группа» для вывода пар с исключительно выбранным названием группы;
- Фильтр «Дисциплина» для вывода пар с исключительно выбранным названием дисциплины;
- Кнопка «Применить» для формирования таблицы с фильтрами интервала дат, названием группы и названием дисциплины;
- Кнопка «Сформировать» для открытия таблицы MicrosoftExcel с данными из таблицы DataGridView;

- Таблица DataGridView для предварительного просмотра отчета;
- Кнопка «Назад» для возврата в модуль главного меню

**Формирование отчета за интервал дат**

С 1 июня 2019 г.

До 18 июня 2019 г.

**Условия выборки**

Группа  
TM2-16

Дисциплина

Применить

Назад Сформировать

Дата	Группа	Пара	Дисциплина
03.06.2019	TM2-16	3	МДК 02.01
03.06.2019	TM2-16	4	МДК 02.01
05.06.2019	TM2-16	1	МДК 02.01
05.06.2019	TM2-16	4	
06.06.2019	TM2-16	2	МДК 02.01
07.06.2019	TM2-16	3	МДК 02.01
10.06.2019	TM2-16	3	МДК 02.01
10.06.2019	TM2-16	4	МДК 02.01
13.06.2019	TM2-16	2	МДК 02.01
14.06.2019	TM2-16	3	МДК 02.01
17.06.2019	TM2-16	3	МДК 02.01
17.06.2019	TM2-16	4	МДК 02.01

Рис. 2.22. Модуль праздников и дополнительных выходных

На рисунке (рис. 2.23.) представлен отчет в виде таблицы MicrosoftExcel, сформированный программой.

	A	B	C	D
1	Дата	Группа	Пара	Дисциплина
2	03.06.2019	TM2-16		3 МДК 02.01
3	03.06.2019	TM2-16		4 МДК 02.01
4	05.06.2019	TM2-16		1 МДК 02.01
5	05.06.2019	TM2-16		4
6	06.06.2019	TM2-16		2 МДК 02.01
7	07.06.2019	TM2-16		3 МДК 02.01
8	10.06.2019	TM2-16		3 МДК 02.01
9	10.06.2019	TM2-16		4 МДК 02.01
10	13.06.2019	TM2-16		2 МДК 02.01
11	14.06.2019	TM2-16		3 МДК 02.01
12	17.06.2019	TM2-16		3 МДК 02.01
13	17.06.2019	TM2-16		4 МДК 02.01

Рис. 2.23. Сформированный отчет об учебной нагрузке

#### Модуль личного кабинета

На рисунке (рис. 2.22.) представлен модуль личного кабинета, в котором находится информация о пользователе (преподавателе) и настройки личной информации. Функции модуля:

- Отображение имени и фамилии пользователя(преподавателя);
- Выбор даты последнего рабочего дня в учебном году;
- Кнопка «применить» для сохранения информации и перестройки рабочих дней согласно выбранной даты;
- Кнопка «назад» для возврата обратно в главное меню;

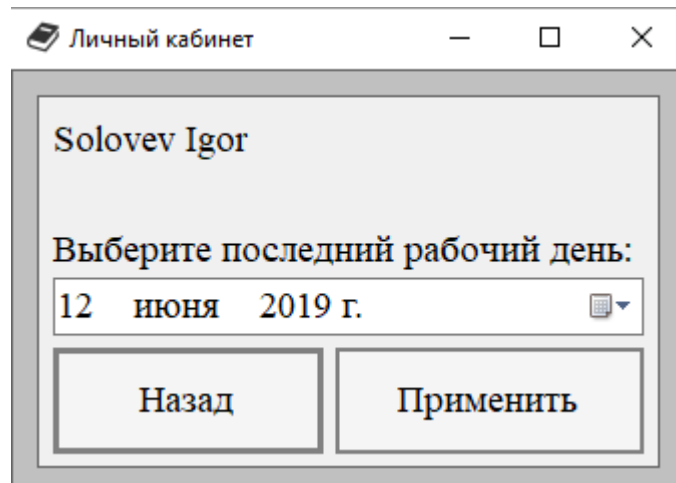


Рис. 2.24. Модуль личного кабинета

### 2.3.3. Описание тестовых наборов

Цель:

Разработать тест кейсы для проверки работы приложения.

Выполнение:

Тест-кейс №1. Авторизация пользователя, проверка поля «Логин».

Тест-кейс создан для проверки вводимых значений в окне авторизации поля «Логин».

Шаги:

- Запустить исполняемый файл Автоматизированная система ресторанного бизнеса.exe
- В открывшемся окне заполнить поле: Логин и Пароль (см. «Ожидаемый результат»)
- Нажать на кнопку «Вход».

Ожидаемый результат

Таблица 2.6.

Ожидаемый результат

Вводимое значение	Ожидаемый результат
Igor - 199814	Добро пожаловать, Соловьев Игорь



Продолжение таблицы 2.6

Вводимое значение	Ожидаемый результат
Admin - 0430	Неверное имя пользователя или пароль

Тестирование, отладка и оптимизация функций.

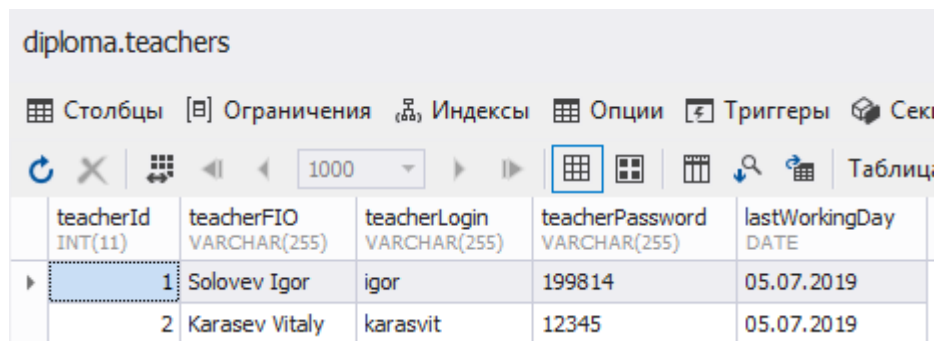
Цель:

Провести тестирование, отладку и оптимизацию разработанных функций.

Выполнение:

Тестирование функции «черным ящиком»

При авторизации пользователь вводит свой логин или пароль если имя пользователя или пароль не совпадают, то авторизация не должна быть пройдена и пользователю должна быть выведена ошибка на экран.



teacherId INT(11)	teacherFIO VARCHAR(255)	teacherLogin VARCHAR(255)	teacherPassword VARCHAR(255)	lastWorkingDay DATE
1	Solovev Igor	igor	199814	05.07.2019
2	Karasev Vitaly	karasvit	12345	05.07.2019

Рис. 2.25. Информация о пользователе в базе данных

Тестирование

Тест #1

Входные данные:

Логин: admin

Пароль: 12345

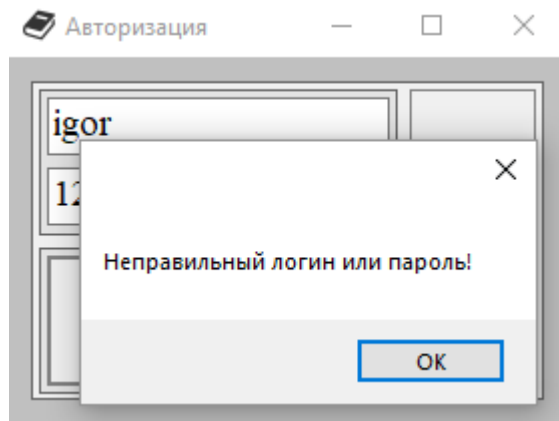


Рис. 2.26. Неверные логин и пароль

Выходные данные: Неверный логин или пароль!

Тест пройден

Тест #2

Входные данные:

Логин: sony002

Пароль: 3456;

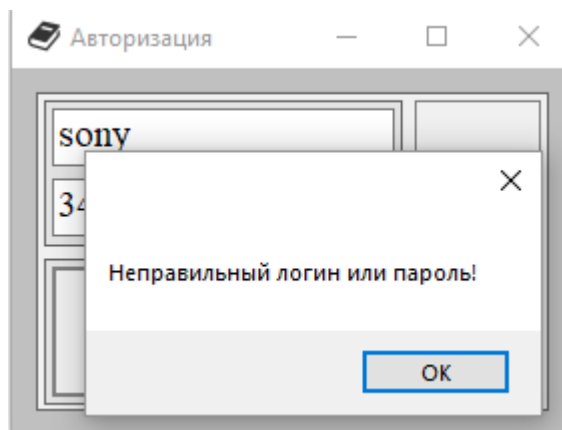


Рис. 2.27. Неверные логин и пароль

Тест пройден

Тест #3

Входные данные:

Логин: igor

Пароль: 199814

Выходные данные: Добро пожаловать, Solovev Igor!

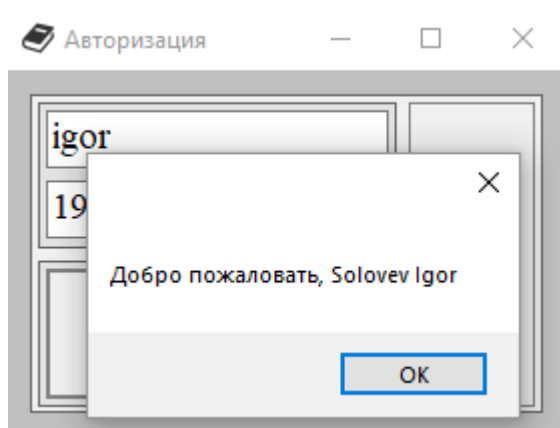


Рис. 2.28. Добро пожаловать, SolovevIgor!

Тест пройден

Результаты тестов показали:

- Высокую скорость, а также корректность выполнения поставленной задачи.

Итог:

Задача перед программистом была выполнена.

## 2.4. Описание процедур и функций

Данная программа состоит из 10 основных форм, класса для перемещения данных между формами и класса, который отвечает за запуск программного продукта. Каждая форма имеет функции и методы для полноценной работы программы. Далее рассмотрим формы и их методы и функции:

- **AutorisationWin** – это окно для авторизации пользователя. Имеет только методы, вызываемые элементами формы;
- **RegistrationWin** - это окно для регистрации пользователя. Имеет только методы, вызываемые элементами формы;
- **MainMenu** - это окно главного меню. Имеет только методы, вызываемые элементами формы;

– ScheduleSet – это окно для работы с расписанием. Помимо методов, вызываемых элементами формы, имеет еще и созданные вручную:

- 1) ClearTable – очистка таблицы;
- 2) ImportFromSql – импорт расписания из MySql;
- 3) AddToMysql – экспорт расписания в MySql;
- 4) ScheduleSet - Заполнение расписания с помощью ImportFromMySql;

– DateWork – это окно для работы с учебными часами и датами. Помимо методов, вызываемых элементами формы, имеет еще и созданные вручную:

- 1) ImportFromMySQL - Импорт выходных из MySQL;
- 2) AddAdditionalLessons - Добавить дополнительные пары в список рабочих дней;
- 3) setPartOfWeek - Деление недели на числитель и знаменатель по датам;
- 4) SetStudyYear - Создание учебного года;
- 5) CountOfDays - Количество дней в учебном году;
- 6) SetDayOfWeek1 - возврат дня недели в текстовом виде;
- 7) SetDayOfWeek - возвращает день недели, только в числовом виде;
- 8) SetAllDays - количество учебных дней в году;
- 9) CreateCustomCalendar - Создание кастомного календаря;
- 10) FillCustomCalendarWithDates - Заполнение кастомного календаря днями
- 11) ClearCalendar - Очистка календаря;
- 12) Helper - Вспомогательный метод для начала заполнения календаря с определенного дня недели;
- 13) WordEnding - Добавление правильного окончания к слову "час" в зависимости от числа;

- 14) InsertAllWorkDays - Синхронизация рабочих дней и базы данных;
- 15) ImportFromWorkingDays - Заполнение календаря рабочими днями;
- 16) ShowChislitelZnamenatel - Показывать числитель и знаменатель в отдельном окошке;
- 17) CreateWorkingDaysList – метод для работы со списками;
  - DeleteLesson - это окно для удаления пар. Имеет только методы, вызываемые элементами формы;
  - AdditionalLessons - это окно для добавления пар. Имеет только методы, вызываемые элементами формы;
  - Holidays - это окно для работы с праздничными днями и выходными. Имеет только методы, вызываемые элементами формы;
  - Otchet - это окно для формирования отчета. Имеет только методы, вызываемые элементами формы;
  - UserSettings - это окно для работы с личной информацией пользователя. Имеет только методы, вызываемые элементами формы;

### 3. Организационно-экономическая часть

#### 3.1. Эксплуатационный раздел

##### 3.1.1. Рекомендации по эксплуатации программного продукта

Данная программа не требует большого объема памяти для своего выполнения. Для запуска программы необходимы следующие технические и операционные средства ЭВМ, они представлены в таблице 3.1

Таблица 3.1

Технические характеристики устройства

Тип оборудования	Наименование характеристики	Значение
Процессор	Количество, не менее	1
	Количество ядер, не менее	1
	Частота, не менее	1000 МГц
Память	Тип, не хуже	DDR3
	Количество слотов, не меньше	1 DIMM
	Частота, не менее	1600 МГц
Сетевой контроллер	Требований не предъявляется	-
Жесткие диски	Объем, не менее	2 ГБ
Видеопамять	Тип, не хуже	GDDR2
	Частота, не менее	1200 МГц
	Объем, не менее	64 Мб
Оптический привод	Тип, не хуже	DVD-RW
Операционная система	Версия, не хуже	Windows 7
Экран	Разрешение, не менее	640x480
Клавиатура	не обязательно	Мембранная
Мышь	не обязательно	Лазерная

Не должно быть запущено никаких приложений, блокирующих работу с программой или доступы к пунктам меню.

### **3.1.2. Руководство пользователя**

Порядок загрузки данных и программ

Перед началом работы с ПДФООВУН на рабочем месте пользователя необходимо выполнить следующие действия:

- необходимо открыть файл программы с расширением EXE.

Порядок проверки работоспособности

Для проверки доступности ПДФООВУН с рабочего места пользователя необходимо выполнить следующие действия:

- открыть программу, для этого необходимо кликнуть по файлу программы;
- В форме аутентификации ввести пользовательский логин и пароль. Нажать кнопку «Далее»;
- Убедиться, что в окне открылось приложение;
- В случае если приложение не запускается, то следует обратиться в службу поддержки;

Описание операций технологического процесса обработки данных, необходимых для выполнения задач.

Ниже приведено описание пользовательских операций для выполнения каждой из задач.

Условия, при соблюдении которых возможно выполнение операций:

- компьютер пользователя подключен к сети;
- сервер базы данных ПДФООВУН доступен;
- ПДФООВУН функционирует в штатном режиме.

Задача: «Вход в главную форму»

Операция 1: Авторизация в ПДФООВУН

Основные действия в требуемой последовательности:

- на файле exe произвести двойной щелчок левой кнопкой мышки;
- в открывшемся окне в поле «Логин» ввести имя пользователя, в поле «Пароль» ввести пароль пользователя. Нажать кнопку «Войти».

Заключительные действия:

- не требуются.

Ресурсы, расходуемые на операцию:

- 15-30 секунд.

Задача: «Регистрация в системе»

Операция 1: регистрация в ПДФООВУН

Основные действия в требуемой последовательности:

- На форме авторизации необходимо нажать на кнопку «Регистрация»;
- После открытия формы регистрации, требуется заполнить поля имя, фамилия, логин, пароль;
- Нажать кнопку «Регистрация»;

Заключительные действия:

- не требуются.

Ресурсы, расходуемые на операцию:

- 15-30 секунд.

Задача: «Заполнение расписания»

Операция 1: Изменение данных

Основные действия в требуемой последовательности:

- выбрать в главном меню кнопку «Выбрать расписание на месяц»;
- после открытия нужной формы, выбрать день недели и тип недели. Учитывая данные условия необходимо заполнить поля группы и ввести название дисциплины;
- Нажать кнопку применить изменения;



- если ПДФООВУН не заполнит ячейки в приложении или базе данных, то стоит обратиться в службу поддержки.

Заключительные действия:

- не требуются.

Ресурсы, расходуемые на операцию:

- 15-30 секунд.

Задача: «Удаление пар»

Операция 1: удаление пары

Основные действия в требуемой последовательности:

- кликнуть по кнопке с названием «Сверить рабочие часы»;
- в открывшейся форме нажать на кнопку «Удалить пару»;
- Выбрать дату удаляемой пары;
- Выбрать пару из выпавшего списка;

Повторить, при необходимости:

- если ПДФООВУН не удалит информацию из базы данных, то стоит обратиться в службу поддержки.

Заключительные действия:

- не требуются.

Ресурсы, расходуемые на операцию:

- 15-30 секунд.

### **3.1.3. Руководство системного администратора**

Общие сведения о программе:

Программа для формирования отчетов о выполнении учебной нагрузки предназначена для автоматизации и упрощения работы преподавателей при работе с подсчетом учебных часов.

Требования к программному и аппаратному обеспечению.

Минимальный состав аппаратных средств:

– Сервер:

3.1.4. процессор Intel Core i3-2.0GHzs, не менее;

3.1.5. оперативную память объемом 4 Гигабайта, не менее;

3.1.6. HDD, 40 Гигабайт, не менее;

3.1.7. операционную систему Windows 7/8/10.

– Клиент:

1) процессор Intel Core i3-1.8GHzs, не менее;

2) оперативную память объемом 2 Гигабайта, не менее;

3) HDD, 40 Гигабайт, не менее;

4) операционную систему Windows 7/8/10.

Минимальный состав программных средств:

– Сервер:

1) MySQL Server 5.7;

2) Workbench 6.3.

– Клиент:

1) Microsoft Visual C++ 2017 Redistributable (x86) - 14.13.26020.

Структура программы.

Программа представляет собой клиент-серверное приложение. На серверной части находятся сервер базы данных. На клиентской части необходимо настроить подключение к серверу.

Настройка сервера.

Для работы базы данных необходим MySQLServer 5.7. Для удобства можно использовать графическую среду по работе с данной СУБД dbForgestudio 2017 forMySQL.

Для того чтобы загрузить базу данных на сервер необходимо сделать следующее:

– Открыть dbForgestudio 2017 forMySQL (информацию по установке и настройке читайте на сайте производителя).

– Открыть подключение и ввести пароль.

### **3.2. Раздел техники безопасности**

Компьютерная техника в настоящее время используется практически во всех организациях. Применение работниками данного оборудования способно оказывать негативное влияние на их здоровье и даже привести к чрезвычайным ситуациям и несчастным случаям на производстве. Для того, чтобы этого избежать, работники должны соблюдать технику безопасности при работе с компьютером.

Негативные факторы при работе за компьютером.

Работник, использующий в своей трудовой деятельности компьютер, может испытывать на себе влияние следующих негативных факторов:

- электромагнитное и инфракрасное излучения;
- шум работающего компьютера (или нескольких компьютеров);
- риск поражения электрическим током в случае замыкания;
- возможность возникновения возгорания.

Правила работы за компьютером.

Статья 212 ТК РФ обязывает работодателя разработать и утвердить инструкции по охране труда и обеспечить ознакомление с ними работников. Таким образом, в организации может быть утвержден специальный документ, предусматривающий правила выполнения трудовых операций с компьютерной техникой, который может называться, например, инструкцией по технике безопасности при работе за компьютером. Отдельные положения по данному вопросу могут содержаться в инструкции по охране труда для офисных работников.

В некоторых отраслях хозяйственной деятельности утверждаются типовые инструкции по охране труда. Например, Приказом Минсвязи РФ от 02.07.2001 N 162 утверждена Инструкция по охране труда при работе на персональном компьютере ТОИ Р-45-084-01.

Рассмотрим правила безопасности по каждому этапу работы с компьютером.

– До начала работы: проверить исправность электропроводки, розеток и вилок компьютера, заземление ПК.

– Во время работы:

1) необходимо аккуратно обращаться с проводами;  
2) запрещается работать с неисправным компьютером;  
3) нельзя заниматься очисткой компьютера, когда он находится под напряжением;

4) недопустимо самостоятельно проводить ремонт оборудования при отсутствии специальных навыков;

5) нельзя располагать рядом с компьютером жидкости, а также работать с мокрыми руками;

6) нельзя в процессе работы с ПК прикасаться к другим металлическим конструкциям (например, батареям);

7) не допускается курение и употребление пищи в непосредственной близости с ПК и др.

– В аварийных ситуациях:

1) при любых неполадках необходимо сразу отсоединить ПК от сети;

2) в случае обнаружения оголенного провода незамедлительно оповестить всех работников и исключить контакт с проводом;

3) в случае возникновения пожара принять меры по его тушению с использованием огнетушителей (работники должны знать, где они находятся);

4) в случае поражения человека током оказать первую помощь и вызвать скорую медицинскую помощь.

– По окончании работы:

- 1) выключить компьютер;
- 2) желательно провести влажную уборку рабочего места;
- 3) отключить электропитание.

Организация рабочего места.

Постоянная работа за компьютером вызывает отклонения в здоровье работника, в частности:

- нагрузка на зрение приводит к его ухудшению, покраснениям глаз, возникновению «синдрома сухого глаза»;
- несоблюдение нормативов организации рабочего места может привести к искривлению позвоночника, заболеваниям суставов и болям различного характера;
- длительная концентрация внимания на экране вызывает переутомление.

Порядок организации работы с компьютером определяется СанПиН 2.2.2/2.4.1340-03, введенными в действие с 30.06.2003.

Приведем некоторые требования, предъявляемые СанПиН к рабочему месту пользователя ПК:

- расстояние от монитора до глаз должно составлять от 600 до 700 мм, но не меньше 500;
- стул работника должен быть регулируемым по высоте и обеспечивать возможность поворота и изменения позы во время работы;
- высота стола – от 680 до 800 мм;
- поверхность стола должна позволять оптимально разместить на ней все необходимое для работы и др.

Перерывы в работе за компьютером.

С целью избежать переутомления работника СанПиН рекомендуют делать перерывы длительностью от 10 до 15 минут после 45 — 60 минут работы. Во время перерыва работнику следует выполнять гимнастику для глаз и физические упражнения, предусмотренные приложениями 8 – 10 к СанПиНу.

Соблюдение правил работы за компьютером позволит снизить негативное воздействие компьютера на здоровье работника. Однако чаще всего именно работники пренебрегают данными правилами, и задача работодателя в данном случае – постоянно доводить до сведения своих сотрудников информацию о последствиях несоблюдения вышеизложенных требований и своими распоряжениями организовывать обязательные перерывы в работе.

### **3.3. Расчет экономической эффективности применения программного комплекса**

Расчёт стоимости одного часа работы ПЭВМ.

Стоимость одного часа работы ПЭВМ производится по формуле:

$$C_{\text{ЭВМ}} = C / (T \times n) \quad (1)$$

Где:

- С – затраты на содержание и обслуживание ПЭВМ
- Т – полезный фонд времени работы на ПЭВМ
- n – количество машин

$$C = C_{\text{зп}} + C_{\text{свзн}} + C_{\text{н}} + C_{\text{ам}} + C_{\text{рем}} + C_{\text{эл}} + C_{\text{м}} \quad (2)$$

Где:

- Сзп - зарплата обслуживающего персонала,
- Ссвзн - страховые взносы, зачисляемые в государственные внебюджетные фонды,
- Сн - накладные расходы,
- Сам - амортизационные отчисления,
- Срем - затраты на ремонт оборудования,
- Сэл - затраты на электроэнергию,
- См - затраты на материалы.

Расчет зарплат обслуживающего персонала.

Таблица 3.1

## Обслуживающий персонал

Обслуживающий персонал	Заработная плата в месяц (руб.)	Количество работников (чел.)	Фонд заработной платы (руб.)
Инженер по снабжению	20000	1	20000
Оператор ЭВМ	15000	1	15000
ИТОГО			35000

$$ЗП_{\text{осн}} = \sum ЗП_i, i=1 \quad (3)$$

$$ЗП_i = (ЗП_1 / Q) \times q \quad (4)$$

Где:

- ЗП – заработная плата работника в месяц,
- Q – количество обслуживаемых работником машин,
- q – количество машин, на которых установлена программа.
- Q = 10 (шт.),
- q = 8 (шт.),

$$ЗП_1 = (20000 / 10) \times 8 = 16\,000 \text{ (руб.)}$$

$$ЗП_2 = (15000 / 10) \times 8 = 12\,000 \text{ (руб.)}$$

$$ЗП_{\text{осн}} = 16\,000 + 8\,000 = 24\,000 \text{ (руб.)}$$

Так как дополнительная заработная плата отсутствует, то  $C_{\text{зп}} = ЗП_{\text{осн}}$ .

Расчет страховых взносов, зачисляемых в государственные внебюджетные фонды

$$C_{\text{свзн}} = (C_{\text{зп}} \times \%СВзн) / 100 \% \quad (5)$$

Где

- $\%СВзн$  – взносы на социальное страхование и обеспечение (%).

Таблица 3.2

Ставки страховых взносов, зачисляемых в государственные  
внебюджетные фонды

Государственные внебюджетные фонды	Ставка налога (%)
Фонд обязательного медицинского страхования РФ	5,1
Страховые взносы на социальное страхование от несчастных случаев на производстве и профзаболеваний	0,2
ИТОГО	30.2

$$C_{\text{свзн}} = (24\,000 \times 30,2) / 100\% = 7248 \text{ (руб.)}$$

Расчет накладных расходов

$$C_{\text{н}} = (3\Pi_{\text{осн}} \times \%N) / 100$$

$$C_{\text{н}} = (3\Pi_{\text{осн}} \times \%N) / 100 \quad (6)$$

Где:

- $\%N$  – ставка накладных расходов
- $\%N = 150$

$$C_{\text{н}} = (24\,000 \times 150) / 100 = 36000 \text{ (руб.)}$$

Расчет амортизационных отчислений

$$C_{\text{ам}} = (C_{\text{п}} \times N_{\text{ам}}) / (100 \times 12) \quad (7)$$

Где:

- $C_{\text{п}}$  - первоначальная стоимость оборудования,
- $N_{\text{ам}}$  - ставка годовой нормы амортизации (%)

Таблица 3.3

Состав оборудования

Наименование оборудования	Цена за единицу (руб.)	Количество единиц оборудования	Стоимость оборудования (руб.)
Компьютер	19000	10	190000



Продолжение таблицы 3.3

Наименование оборудования	Цена за единицу (руб.)	Количество единиц оборудования	Стоимость оборудования (руб.)
Принтер HP LaserJet 1020	5500	1	5500
ИТОГО			<b>195500</b>

$$N_{ам}=12,5\%$$

$$C_{ам}=(195500 \times 12,5)/(100 \times 12)=2036.45 \text{ (руб.)}$$

Расчет на ремонт оборудования

$$C_{рем}=(C_{п} \times \%P)/(100 \times 12)$$

$$C_{рем}=(C_{п} \times \%P)/(100 \times 12) \quad (8)$$

Где:

–  $\%P$  – ставка годовых плановых затрат на ремонт оборудования (%).

$$– \quad \%P = 15 \%$$

$$C_{рем} = (195500 \times 15) / (100 \times 15) = 1955 \text{ (руб.)}$$

Расчет затрат на электроэнергию

$$C_{эл} = P \times T \times Z \quad (9)$$

Где:

- $P$  – общая мощность оборудования,
- $T$  – полезный фонд времени работы ПЭВМ,
- $Z$  – цена 1 Квт/ч (руб.)

$$T = T_{реж} - T_{рем} \quad (10)$$

$$T_{рем} = (T_{реж} \times \%Рем) / 100 \quad (11)$$

Где:

- $T_{реж}$  – режимный фонд работы оборудования (час)

– %Рем – ставка плановых потерь на профилактику и ремонт оборудования (%)

$$T_{\text{реж}} = (D - D_{\text{в}} - D_{\text{пр}}) \times A \times S - D'_{\text{пр}} \times t \quad (12)$$

Где:

- D – количество календарных дней за отчетный период,
- D<sub>в</sub> – количество выходных дней за отчетный период,
- D<sub>пр</sub> – количество праздничных дней за отчетный период,
- D' пр – количество предпраздничных дней за отчетный период,
- A – длительность смены (час),
- s – количество смен,
- t – количество часов, на которые сокращается рабочий день в предпраздничные дни.

Таблица 3.4

Общая мощность оборудования

Наименование оборудования	Мощность единицы оборудования (Квт/час)	Количество оборудования	Суммарная мощность (Квт/час)
Компьютер	0,45	15	6,75
Принтер HP LaserJet 1020	0,04	1	0,04
ИТОГО			<b>6,79</b>

Расчет ведется за май,

Z = 3,38,

% Рем = 15,

A = 8 (часов),

S = 2,

t = 1 (час).

$T_{\text{реж}} = (31 - 5 - 2) \times 8 \times 2 - 0 \times 1 = 384$  (часов)

$$T_{\text{рем}} = (384 \times 15) / 100 = 57,6 \text{ (часов)}$$

$$T = 384 - 57,6 = 326,4 \text{ (часов)}$$

$$C_{\text{эл}} = 6,79 \times 326,4 \times 3,38 = 7\,490,94 \text{ (руб.)}$$

Расчет затрат на материалы

По данным бухгалтерии затраты на материалы в месяц составили 1200 руб.

Расчет стоимости часа работы на ПЭВМ

$$C = 24000 + 7248 + 36000 + 2036,45 + 1955 + 7490,94 + 1200 = 79930,39 \text{ (руб.)}$$

$$C_{\text{эвм}} = 79930,39 / (345,6 \times 10) = 23,12 \text{ (руб.)}$$

Таблица 3.5

### Структура стоимости 1 часа работы на ПЭВМ

№	Элемент затрат	Сумма (руб.)	Удельный вес %
1	Заработная плата обслуживающего персонала	24000	30,03
2	Страховые взносы, зачисляемые в государственные внебюджетные фонды	7248	9,07
3	Накладные расходы	36000	45,04
4	Амортизационные отчисления	2036,45	2,55
5	Затраты на ремонт оборудования	1955	2,45
6	Затраты на электроэнергию	7490,94	9,37
7	Затраты на материалы	1200	1,5
ИТОГО		79930,39	100,00

Расчет стоимости решения задачи на ПЭВМ

Стоимость решения задачи на ПЭВМ рассчитывается по формуле:

$$S_o = 3П_{\text{рук}} + 3П_{\text{студ}} + C'_{\text{свзн}} + C'_n + C'_m + T_{\text{отл}} \times C_{\text{эвм}} \quad (13)$$

Где:

- $3П_{\text{рук}}$  - заработная плата руководителя,
- $3П_{\text{студ}}$  - заработная плата студента,

- С`свзн - страховые взносы, зачисляемые в государственные внебюджетные фонды,
- С`н - накладные расходы,
- С`м - затраты на материалы, используемые при составлении и отладке программы,
- Тотл - затраты времени на отладку программы и решения ее на ПЭВМ,
- Сэвм – стоимость одного часа работы ПЭВМ.

Расчет заработной платы руководителя

$$ЗП_{рук} = (N_q \times T_q) / N_{ст} \quad (14)$$

Где:

- $N_q$  - время, отводимое на консультации по дипломному проектированию,

- $T_q$  - часовая тарифная ставка руководителя,

- $N_{ст}$  - количество студентов в группе,

$$N_q = 16 \text{ (часов)}, 17$$

$$T_q = 560 \text{ (руб.)}$$

$$N_{ст} = 1 \text{ (чел.)},$$

$$ЗП_{рук} = (16 \times 560) / 1 = 8960 \text{ (руб.)}$$

Расчет заработной платы студента

$$ЗП_{студ} = (C_{ст} \times T_{отл}) / T_y \quad (15)$$

Где:

- $C_{ст}$  - стипендия студента,

- $T_y$  - количество учебных часов в месяце,

- Тотл – затраты времени на отладку программы и решение задачи на ПЭВМ.

$$T_{отл} = S \times d \quad (16)$$

Где:

- $S$  - средняя занятость в течении дня,

– d- количество дней для отладки программы,

$$C_{\text{ст}} = 1000,$$

$$T_y = D_p \times S = 24 \times 6 = 144 \text{ (часов)},$$

$D_p$  – количество рабочих дней в месяце,

$$S = 4 \text{ (часа)},$$

$$d = 5 \text{ (дн.)},$$

$$T_{\text{отл}} = 4 \times 5 = 20 \text{ (часов)},$$

$$ЗП_{\text{студ}} = (1000 \times 20) / 144 = 138,88 \text{ (руб.)}$$

Расчет страховых взносов, зачисляемых в государственные внебюджетные фонды

$$C'_{\text{сн}} = (ЗП_{\text{рук}} \times \%СВЗН) / 100 \quad (17)$$

Где:

–  $\%СВЗН$  – взносы на социальное страхование и обеспечение (%),

$$\%СВЗН = 30,2;$$

$$C'_{\text{сн}} = (8960 \times 30,2) / 100 = 2705,92 \text{ (руб.)}$$

Расчет накладных расходов

$$C'_n = ((ЗП_{\text{рук}} + ЗП_{\text{студ}}) \times \%Н) / 100 \quad (18)$$

Где:

–  $\%Н$  – ставка накладных расходов,

$$\%Н = 120$$

$$C'_n = ((8960 + 138,88) \times 120) / 100 = 10918,656 \text{ (руб.)}$$

Затраты на материалы, используемые при составлении и отладке программы

Таблица 3.6

Затраты на материалы, используемые при составлении и отладке  
программы

Наименование материалов	Цена за единицу (руб.)	Количество	Стоимость (руб.)
Бумага	1	120	120
Катридж	1100	1	1100
ИТОГО			1220

$$C'_m = 1220 \text{ (руб.)},$$

$$S_o = 8960 + 138,88 + 7248 + 10918,656 + 1220 + 20 + 23,12 = 28526,656 \text{ (руб.)}$$

Расчет стоимости решения задачи вручную

Стоимость решения задачи вручную рассчитывается по формуле:

$$S_p = T_p \times Ч \times (1 + (\%CH/100) + (\%H/100)) \quad (19)$$

Где:

- $T_p$  - затраты времени на решение задачи вручную,
- $Ч$  – часовая ставка студента, решающего задачу (руб.),

$$T_p = 5 \text{ (часа)},$$

$$Ч = C_{ст}/T_y = 1000/144 = 6,94 \text{ (руб.)},$$

$$S_p = 5 \times 6,94 \times (1 + (30,2/100) + (120/100)) = 86,81 \text{ (руб.)}$$

Расчет показателей эффективности применения ПЭВМ

Расчет экономии труда за счет применения ПЭВМ

$$\mathcal{E}_T = T_p \times N - (T_o + N \times T_{эвм}) \quad (20)$$

Где:

- $T_p$  – затраты времени на решение задачи вручную,
- $N$  - предполагаемое число обращений к программе,
- $T_{отл}$  - затраты времени на составление и отладку программы на ПЭВМ,
- $T_{эвм}$  - время решения задачи на ПЭВМ,

Расчет затрат времени на составление и отладку программы

$$T_o = S \times K \quad (21)$$

Где:

- S- средняя занятость в течение дня,
- K- количество дней на составление и отладку программы на ПЭВМ,

$$T_o = 5 \times 30 = 150 \text{ (часов)}.$$

Расчет времени решения задачи на ПЭВМ

$$T_{\text{ЭВМ}} = T_v + T_m + T_{\text{п}} \quad (22)$$

Где:

- $T_v$  - время ввода исходной информации,
- $T_m$  - машинное время,
- $T_{\text{п}}$  - время вывода информации на печать,

$$T_v = 0,025 \text{ (часа)},$$

$$T_m = 0,00043 \text{ (часа)},$$

$$T_{\text{п}} = 0,01 \text{ (часа)},$$

$$T_{\text{ЭВМ}} = 0,025 + 0,00043 + 0,01 = 0,03543 \text{ (часа)}$$

Расчет предполагаемого числа обращений к программе

$$N = (N_{\text{СТ}} \times 1) + (N_{\text{СТ}}/2) \times 1,$$

$$N = (120 \times 1) + (120/2) \times 1 = 180,$$

Таким образом,

$$\text{Эт} = 5 \times 180 - (150 + 180 \times 0,03543) = 743.62 \text{ (руб.)}$$

Расчет денежной экономической эффективности применения ПЭВМ

$$\text{Э}_д = S_p \times N - (S_o + N \times S_{\text{ЭВМ}}) \quad (23)$$

Где:

- $S_p$  - стоимость решения задачи вручную,
- N - предполагаемое число обращений к программе,
- $S_o$  - стоимость решения задачи на ПЭВМ,
- $S_{\text{ЭВМ}}$  - затраты на работу программы.

Расчет затрат на работу программы

$$S_{\text{ЭВМ}} = T_{\text{ЭВМ}} \times C_{\text{ЭВМ}} \quad (24)$$

Где:

- $T_{\text{ЭВМ}}$  - время решения задачи на ПЭВМ,
- $C_{\text{ЭВМ}}$  - стоимость часа работы ПЭВМ.

$$S_{\text{ЭВМ}} = 0,03543 \times 23,12 = 0,81 \text{ (руб.)},$$

$$\text{Э}_\text{д} = 86,81 \times 180 - (28526,656 + 180 \times 0,81) = -13046,656 \text{ (руб.)}$$

Расчет роста производительности труда от применения ПЭВМ

$$\Delta\P = (\text{Э}_\text{т} / (T_\text{р} \times N)) \times 100\% \quad (25)$$

Где:

- $\text{Э}_\text{т}$  - показатель экономии труда за счет применения ПЭВМ,
- $T_\text{р}$  - затраты времени на решение задачи вручную,
- $N$  - Предполагаемое число обращений к программе,

$$\Delta\P = (743.62 / (5 \times 180)) \times 100\% = 82.6\%.$$

Таким образом, стоимость одного часа работы ЭВМ – 23,12 (руб.); Затраты на составление и отладку программы – 28526,656(руб.); Рост производительности труда составляет –82.6%.

Внедрение программы повысит оперативность, точность и эффективность решения задач. Внедрение данной программы сократит число кадров и позволит сократить расходы на зарплаты сотрудникам.



## **ЗАКЛЮЧЕНИЕ**

В ходе дипломной работы было разработано приложение для преподавателей «МГОТУ», разработка подобного рода приложений, объясняется только нужным функционалом, узкой направленностью и лаконичностью, что значительно снижает её стоимость, количество ненужных функций, а только самые необходимые, для быстрой и упрощенной работы. А также всесторонним распространением информационных технологий и внедрением их в большинство сфер человеческой деятельности.

Функции, выполняемые программным продуктом, позволяют автоматизировать работу пользователя, проводить анализ данных, хранить большое количество информации, учитывая структуризацию предметной области.

Приложение разработано с применением среды разработки Visual Studio 2017 и СУБД MySQL.

Приложение работает в операционных системах Windows 7, Windows 10, WindowsXP.

Были рассчитаны следующие экономические показатели: стоимость одного часа работы ЭВМ – 23,12 (руб.); затраты на составление и отладку программы 28526,656(руб.);рост производительности труда составляет – 82.6%.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Стиллмен, Э. «Изучаем С#. С#. 3-е издание»/ Э. Стиллмен. – М.: Питер СПб, 2014. – 816 с.
2. Албахари, Д. «С# 7.0. Карманный справочник. Скорая помощь для программистов на С# 7.0»/ Д. Албахари, Б. Албахари. – М.: Альфа-Книга, 2017. – 224 с.
3. С# .NET // С# .NET - Форум программистов С# URL: <http://www.cyberforum.ru/csharp-net/> (дата обращения: 19.06.2019).
4. Полное руководство по языку программирования С# 7.0 и платформе .NET 4.7 // Metanit URL: <https://metanit.com/sharp/tutorial/> (дата обращения: 18.06.2019).
5. Объектно-ориентированный язык программирования // habr URL: <https://habr.com/ru/hub/csharp/> (дата обращения: 15.06.2019).
6. Документация по .NET // Microsoft URL: <https://docs.microsoft.com/ru-ru/dotnet/> (дата обращения: 14.05.2019).
7. С# // Progopedia URL: <http://progopedia.ru/language/csharp/> (дата обращения: 10.06.2019).
8. Полезные материалы по С#, уроки для начинающих // Tproger URL: <https://tproger.ru/tag/c-sharp/> (дата обращения: 05.06.2019).
9. Практическое владение языком SQL // Sql-ex URL: <http://www.sql-ex.ru/> (дата обращения: 16.05.2019).
10. Язык манипуляции данными в SQL // Sql-Tutorial URL: [http://www.sql-tutorial.ru/ru/book\\_sql\\_dml.html](http://www.sql-tutorial.ru/ru/book_sql_dml.html) (дата обращения: 10.05.2019).
11. Why Learn C#? // Codecademy URL: <https://www.codecademy.com/learn/learn-c-sharp> (дата обращения: 12.06.2019).
12. С# Programming Language // GeeksForGeeks URL: <https://www.geeksforgeeks.org/csharp-programming-language/> (дата обращения: 25.05.2019).

# ПРИЛОЖЕНИЯ

## Приложение 1.

### Листинг «Авторизация»

Листинг программы:

```
using System;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

namespace Диплом
{
    public partial class AutorisationWin : Form
    {

        public static string connectionString = "User Id=root; Password = 199814;
        Host=localhost;Database=diploma;Character Set=utf8";
        MySqlConnection con = new MySqlConnection(connectionString);
        public bool ValidUser;

        public AutorisationWin()
        {
            InitializeComponent();
            con.Open();
        }

        private void scheduleBtn_Click(object sender, EventArgs e)
        {
            MySqlCommand command = new MySqlCommand("SELECT * FROM teachers WHERE
            teacherLogin=@Username AND teacherPassword=@Password", con);
            command.Parameters.AddWithValue("@Username", loginBox.Text);
            command.Parameters.AddWithValue("@Password", passwordBox.Text);
            using (var dataReader = command.ExecuteReader())
            {
                ValidUser = dataReader.Read();
            }
            if (ValidUser == true)
            {
                command = new MySqlCommand("SELECT teacherId FROM teachers WHERE
            teacherLogin='" + loginBox.Text + "'", con);
                DateKeeper.teacherId = Convert.ToInt16(command.ExecuteScalar());
                command = new MySqlCommand("SELECT teacherFIO FROM teachers WHERE
            teacherLogin='" + loginBox.Text + "'", con);
                MessageBox.Show("Добро пожаловать, " +
            command.ExecuteScalar().ToString());
                mainMenu m_win = new mainMenu();
                m_win.Show();
                Hide();
            }
            else
            {
                MessageBox.Show("Неправильный логин или пароль!");
                return;
            }
        }

        private void loginBox_Click(object sender, EventArgs e)
        {
            loginBox.Text = "";
        }
    }
}
```

## Листинг «Главное меню»

Листинг программы:

```
using System;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

namespace Диплом
{
    public partial class mainMenu : Form
    {
        public mainMenu()
        {
            InitializeComponent();

            //Коннект к мускулу
            public static string connectionString = "User Id=root; Password = 199814;
            Host=localhost; Database=diploma; Character Set=utf8";
            MySqlConnection con = new MySqlConnection(connectionString);

            private void scheduleBtn_Click(object sender, EventArgs e)
            {
                ScheduleSet s_win = new ScheduleSet();
                s_win.Show();
                Hide();
            }

            private void button1_Click(object sender, EventArgs e)
            {
                DateWork d_win = new DateWork();
                d_win.Show();
                Hide();
            }

            private void button2_Click(object sender, EventArgs e)
            {
                Holidays holiday_win = new Holidays();
                holiday_win.Show();
                Hide();
            }

            private void button4_Click(object sender, EventArgs e)
            {
                UserSettings user_win = new UserSettings();
                user_win.Show();
                Hide();
            }

            private void button3_Click(object sender, EventArgs e)
            {
                Otchet ot_win = new Otchet();
                ot_win.Show();
            }
        }
    }
}
```

## Листинг «Расписание»

Листинг программы:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

namespace Диплом
{
    public partial class ScheduleSet : Form
    {
        //Коннект
        public static string connectionString = "User Id=root; Password = 199814;
        Host=localhost;Database=diploma;Character Set=utf8";
        MySqlConnection con = new MySqlConnection(connectionString);

        //Переменные для заполнения расписания
        public int dayOfWeek, partOfWeekInt = 0;
        public int hours = 0;
        //Сетка кнопок. Расписание
        public Button[,] buttonCoords = new Button[6, 5];
        public Button[,] buttonRemastered = new Button[6, 5];

        public List<int> dayOfWeekSql = new List<int>();
        public List<int> lesson = new List<int>();
        public List<string> groupName = new List<string>();
        public List<string> discipline = new List<string>();

        //Очистить таблицу
        void ClearTable()
        {
            for (int i = 0; i < 6; i++)
            {
                for (int j = 0; j < 5; j++)
                {
                    buttonRemastered[i, j].Text = "";
                }
            }
        }

        //Импорт расписания из MySQL
        void ImportFromSql()
        {
            con.Open();
            dayOfWeekSql = new List<int>();
            lesson = new List<int>();
            groupName = new List<string>();
            MySqlDataReader reader;
            MySqlCommand cmd = new MySqlCommand("Select lesson from schedule where
            partOfWeek = '" + partOfWeekInt + "' and teacherId = 1", con);
            reader = cmd.ExecuteReader();
            while (reader.Read())
            {
                for (int temp = 0; temp < reader.FieldCount; temp++)
                {
                    lesson.Add(Convert.ToInt16(reader[temp]));
                }
                cmd = new MySqlCommand("Select dayOfWeek from schedule where partOfWeek = '"
                + partOfWeekInt + "' and teacherId = 1", con);
                reader.Close();
            }
        }
    }
}
```

```

        reader = cmd.ExecuteReader();
while (reader.Read())
{
    for (int temp = 0; temp < reader.FieldCount; temp++)
        dayOfWeekSql.Add(Convert.ToInt16(reader[temp]));
    }
    reader.Close();

    cmd = new MySqlCommand("Select groupName from schedule where partOfWeek = '"
+ partOfWeekInt + "' and teacherId = 1", con);
    reader = cmd.ExecuteReader();
while (reader.Read())
{
    for (int temp = 0; temp < reader.FieldCount; temp++)
        groupName.Add(reader[temp].ToString());
    }
    reader.Close();
    discipline.Clear();
    cmd = new MySqlCommand("Select discipline from schedule where partOfWeek = '"
+ partOfWeekInt + "' and teacherId = 1", con);
    reader = cmd.ExecuteReader();
while (reader.Read())
{
    for (int temp = 0; temp < reader.FieldCount; temp++)
        discipline.Add(reader[temp].ToString());
    }
    reader.Close();

    for (int temp = 0; temp < lesson.Count; temp++)
    {
        buttonRemastered[dayOfWeekSql[temp], lesson[temp]].Text = groupName[temp]
+ " " + discipline[temp];
    }
    con.Close();
}

//Экспорт расписания в MySql
void AddToMysql()
{
    con.Open();
    MySqlCommand cmd = new MySqlCommand("", con);
    int n = 0;
    for (int i = 0; i < 6; i++)
    {
        for (int j = 0; j < 5; j++)
        {
            if (buttonRemastered[i, j].Text != "")
                n++;
        }
    }
    int[] temp = new int[n];
    int[] temp1 = new int[n];
    n = 0;
    for (int i = 0; i < 6; i++)
    {
        for (int j = 0; j < 5; j++)
        {
            if (buttonRemastered[i, j].Text != "")
            {
                temp[n] = i;
                temp1[n] = j;
                n++;
            }
        }
    }
}

```

```

    }
}
for(n=0; n<temp.Length;n++)
{
    cmd = new MySqlCommand("INSERT INTO schedule (idSchedule, dayOfWeek,
groupName, lesson, partOfWeek, teacherId, discipline) VALUES (0, '"+temp[n]+
"', '"+buttonRemastered[temp[n],temp1[n]].Text.Substring(0, buttonRemastered[temp[n],
temp1[n]].Text.IndexOf(" ")) + "' , '"+temp1[n]+'', '"+partOfWeekInt+ "', 1, '"+
buttonRemastered[temp[n], temp1[n]].Text.Substring(buttonRemastered[temp[n],
temp1[n]].Text.IndexOf(" ") + 1) + "')"; con);
cmd.ExecuteNonQuery();
}
con.Close();
}

//Заполнение расписания с помощью ImportFromMySQL
public ScheduleSet()
{
    InitializeComponent();

    panel13.Enabled = false;
    panel15.Enabled = true;

    int i = 0, j=0;
    foreach(Button but in tableLayoutPanel12.Controls.OfType<Button>())
    {
        buttonCoords[i,j] = but;
        j++;
    if (j == 5)
        {
            i++;
            j = 0;
        }
    }

    int a = 5, b = 4;

    for (i = 0; i < 6; i++)
    {
        for (j = 0; j < 5; j++)
        {
            buttonRemastered[a, b] = buttonCoords[i, j];
            b--;
        }
        b = 4;
        a--;
    }
    ImportFromSql();
}

//Визуальное заполнение расписания
privatevoid scheduleBtn_Click(object sender, EventArgs e)
{
    if (firstGroup.Text != "")
        buttonRemastered[dayOfWeek, 0].Text = firstGroup.Text + " " +
disBox1.Text;
    if (secondGroup.Text != "")
        buttonRemastered[dayOfWeek, 1].Text = secondGroup.Text + " " +
disBox2.Text;
    if (thirdGroup.Text != "")
        buttonRemastered[dayOfWeek, 2].Text = thirdGroup.Text + " " +
disBox3.Text;
    if (fourthGroup.Text != "")

```

```

        buttonRemastered[dayOfWeek, 3].Text = fourthGroup.Text + " " +
disBox4.Text;
if (disBox5.Text != "")
    buttonRemastered[dayOfWeek, 4].Text = fifthGroup.Text + " " +
disBox5.Text;
    firstGroup.Text = "";
    secondGroup.Text = "";
    thirdGroup.Text = "";
    fourthGroup.Text = "";
    fifthGroup.Text = "";
    disBox1.Text = "";
    disBox2.Text = "";
    disBox3.Text = "";
    disBox4.Text = "";
disBox5.Text = "";
    }

//Выбор дня недели и присвоение ему личного Id
privatevoid comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    if (dayOfWeekBox.SelectedItem.ToString() == "Понедельник")
        dayOfWeek = 0;
    elseif (dayOfWeekBox.SelectedItem.ToString() == "Вторник")
        dayOfWeek = 1;
    elseif (dayOfWeekBox.SelectedItem.ToString() == "Среда")
        dayOfWeek = 2;
    elseif (dayOfWeekBox.SelectedItem.ToString() == "Четверг")
        dayOfWeek = 3;
    elseif (dayOfWeekBox.SelectedItem.ToString() == "Пятница")
        dayOfWeek = 4;
    elseif (dayOfWeekBox.SelectedItem.ToString() == "Суббота")
        dayOfWeek = 5;
    if(partOfWeek.SelectedItem != null)
    {
        panel3.Enabled = true;
        panel5.Enabled = true;
    }
}

//Очистка ячейки таблицы при нажатии на нее
privatevoid button2_Click(object sender, EventArgs e)
{
    Button temp = (Button)sender;
    temp.Text = "";
}

//Применение нового расписания
privatevoid button1_Click(object sender, EventArgs e)
{
    con.Open();
    MySqlCommand cmd = new MySqlCommand("DELETE FROM schedule where partOfWeek =
'" + partOfWeekInt + "' and teacherId = 1;", con);
    cmd.ExecuteNonQuery();
    con.Close();

    AddToMysql();
    ClearTable();
    ImportFromSql();
    DateWork dt_w = new DateWork();
    con.Close();
    dt_w.InsertAllWorkDays();
    MessageBox.Show("Данные успешно обновлены");
}

```



```

//Визуальнаяочисткатаблицы
privatevoid button32_Click_2(object sender, EventArgs e)
{
    ClearTable();
}

//Визуальноеобновлениетаблицы
privatevoid updateTable_Click(object sender, EventArgs e)
{
    ImportFromSql();
}

//Возвратглавноеменю
privatevoid backBtn_Click(object sender, EventArgs e)
{
    mainMenu s_main = new mainMenu();
    s_main.Show();
    Hide();
}

privatevoid button1_Click_1(object sender, EventArgs e)
{
}

//Выбор числителя или знаменателя и присвоение ему id
privatevoid partOfWeek_SelectedIndexChanged(object sender, EventArgs e)
{
    if (partOfWeek.SelectedItem.ToString() == "Числитель")
        partOfWeekInt = 0;
    elseif (partOfWeek.SelectedItem.ToString() == "Знаменатель")
        partOfWeekInt = 1;
        ClearTable();
        ImportFromSql();
    if(dayOfWeekBox.SelectedItem != null)
    {
        panel13.Enabled = true;
        panel15.Enabled = true;
    }
}
}
}

```

## Приложение 4.

## Листинг «Сверка рабочих часов»

## Листинг программы:

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

namespace Диплом
{
    public partial class DateWork : Form
    {
        //Коннект к мускулу
        public static string connectionString = "User Id=root;
        Password = 199814; Host=localhost;Database=diploma;Character
        Set=utf8";
        MySqlConnection con = new MySqlConnection(connectionString);

        //Инициализация формы и методов
        public DateWork()
        {
            InitializeComponent();
            SetStudyYear();
            CountOfDays();
            SetAllDays();
            setPartOfWeek();
            CreateWorkingDaysList();
            for(int i = 0; i < allDays.Count(); i++)
            {
                monthSelect.Add(allDays[i].Month);
            }
            List<int> temp = monthSelect.Distinct().ToList();
            for(int i = 0; i < temp.Count(); i++)
            {
                if (temp[i] == 1)
                    monthBox.Items.Add("Январь");
                else if (temp[i] == 2)
                    monthBox.Items.Add("Февраль");
                else if (temp[i] == 3)
                    monthBox.Items.Add("Март");
                else if (temp[i] == 4)
                    monthBox.Items.Add("Апрель");
                else if (temp[i] == 5)
                    monthBox.Items.Add("Май");
                else if (temp[i] == 6)
                    monthBox.Items.Add("Июнь");
                else if (temp[i] == 7)
                    monthBox.Items.Add("Июль");
                else if (temp[i] == 9)

```

```

        monthBox.Items.Add("Сентябрь");
    else if (temp[i] == 10)
        monthBox.Items.Add("Октябрь");
    else if (temp[i] == 11)
        monthBox.Items.Add("Ноябрь");
    else if (temp[i] == 12)
        monthBox.Items.Add("Декабрь");
    }
    fromDatePicker.MinDate = studyYear[0];
    fromDatePicker.MaxDate = lastWorkingDay;
    toDatePicker.MinDate = studyYear[0];
    toDatePicker.MaxDate = lastWorkingDay;
}

//Учебный год
public DateTime[] studyYear = new DateTime[2];
//Количество дней в году
public TimeSpan countOfWeeks = new TimeSpan();
//Месяц год
public int monthNum, year;
//Числитель - знаменатель
public List<int> partOfWeekToBinary = new List<int>();
//Праздники
public List<DateTime> weekends = new List<DateTime>();
//Дополнительные пары
List<DateTime> additionalLessons = new List<DateTime>();
//Группы дополнительных пар
List<string> additionalGroups = new List<string>();
//Номер пары дополнительных пар
List<int> additionalLessonsNum = new List<int>();
//Вседни
public DateTime[] allDays;
//Только рабочие дни
//public List<DateTime> onlyWorkDays = new List<DateTime>();
//Дополнение к рабочим дням в виде группы
public List<string> groupName = new List<string>();
//Основные номера пар
public List<int> lessonNum = new List<int>();
//Кнопочки кастомного календаря
public Button[,] calendar;
public Button[,] calendarReversed;
//Количество дней в учебном году
public int daysCount;
//Массив месяцев
public List<int> monthSelect = new List<int>();
//Последний рабочий день
public DateTime lastWorkingDay = new DateTime();
//Дисциплины
public List<string> discipline = new List<string>();

//Импорт выходных из MySQL
void ImportFromMySQL()
{
    con.Open();
    weekends.Clear();
    MySqlCommand cmd = new MySqlCommand("select distinct
holidaydate from holidays;", con);

```

```

        MySqlDataAdapter adapter = new MySqlDataAdapter(cmd);
        DataTable table = new DataTable();
        adapter.Fill(table);
        for (int i = 0; i < table.Rows.Count; i++)
        {
            DateTime temp =
DateTime.Parse(table.Rows[i]["holidaydate"].ToString());
            weekends.Add(temp);
        }

        for(int i = 0; i < weekends.Count(); i++)
        {
            for(int j = 0; j < DateKeeper.onlyWorkDays.Count();
j++)
            {
                DateKeeper.onlyWorkDays.Remove(weekends[i]);
            }
        }
        con.Close();
    }

    //Добавить дополнительные пары в список рабочих дней
    void AddAdditionalLessons()
    {
        con.Open();
        MySqlCommand cmd = new MySqlCommand("select * from
additionallesson where teacherid = 1;", con);
        using (MySqlDataReader reader = cmd.ExecuteReader())
        {
            while (reader.Read())
            {
                DateKeeper.onlyWorkDays.Add(DateTime.Parse(reader["date"].ToString()))
;
            }
            reader.Close();
        }
        con.Close();
    }

    //Деление недели на числитель и знаменатель по датам
    void setPartOfWeek()
    {
        int temp = 1;
        for (int i = 0; i < allDays.Count(); i++)
        {
            if (i % 7 == 0)
            {
                if (temp == 1)
                    temp = 0;
                else
                    temp = 1;
            }
            partOfWeekToBinary.Add(temp);
        }
        PartOfWeek.PartOfWeekToBinary = new
int[partOfWeekToBinary.Count()];
    }

```

```

        for (int i = 0; i < partOfWeekToBinary.Count(); i++)
        {
            PartOfWeek.PartOfWeekToBinary[i] =
partOfWeekToBinary[i];
        }
    }

    //Создание учебного года
void SetStudyYear()
{
    studyYear[0] = new DateTime(2018, 09, 1);
    con.Open();
    MySqlCommand cmd = new MySqlCommand("select lastworkingday
from teachers where teacherid = 1;", con);
    studyYear[1] =
DateTime.Parse(cmd.ExecuteScalar().ToString());
    lastWorkingDay =
DateTime.Parse(cmd.ExecuteScalar().ToString());
    con.Close();
}

    //Количество дней в учебном году
void CountOfDays()
{
    countOfWeeks = studyYear[1] - studyYear[0];
    daysCount = Convert.ToInt16(countOfWeeks.Days) + 1;
}

    //Выбор месяца в combobox и возврат года и месяца в виде int
    //а также запуск системы заполнения календаря и некоторых
методов
    /* CreateCustomCalendar() - Создание календаря
ClearCalendar() - Очистка текста календаря
yearLabel.Text = year.ToString() - Присвоение label'у год
FillCustomCalendarWithDates() - Заполнение кастомного календаря днями
setWorkDays() - Запуск работы метода и присвоение цвета рабочим дням*/
private void monthBox_SelectedIndexChanged(object sender, EventArgs e)
{
    if (monthBox.SelectedItem.ToString() == "Сентябрь")
    {
        monthNum = 9;
        year = 2018;
    }
    else if (monthBox.SelectedItem.ToString() == "Октябрь")
    {
        monthNum = 10;
        year = 2018;
    }
    else if (monthBox.SelectedItem.ToString() == "Ноябрь")
    {
        monthNum = 11;
        year = 2018;
    }
    else if (monthBox.SelectedItem.ToString() == "Декабрь")
    {
        monthNum = 12;
        year = 2018;
    }
}

```

```

    }
    else if (monthBox.SelectedItem.ToString() == "Январь")
    {
        monthNum = 1;
        year = 2019;
    }
    else if (monthBox.SelectedItem.ToString() == "Февраль")
    {
        monthNum = 2;
        year = 2019;
    }
    else if (monthBox.SelectedItem.ToString() == "Март")
    {
        monthNum = 3;
        year = 2019;
    }
    else if (monthBox.SelectedItem.ToString() == "Апрель")
    {
        monthNum = 4;
        year = 2019;
    }
    else if (monthBox.SelectedItem.ToString() == "Май")
    {
        monthNum = 5;
        year = 2019;
    }
    else if (monthBox.SelectedItem.ToString() == "Июнь")
    {
        monthNum = 6;
        year = 2019;
    }
    else if (monthBox.SelectedItem.ToString() == "Июль")
    {
        monthNum = 7;
        year = 2019;
    }
    CreateCustomCalendar();
    ClearCalendar();
    yearLabel.Text = year.ToString();
    FillCustomCalendarWithDates();
    ImportFromWorkingDays();
    ShowChislitelZnamenatel();
}

//Возвратднянеделивтекстовомвиде(навсякийслучай)
string SetDayOfWeek1(DateTime dayOfWeek)
{
    if (dayOfWeek.DayOfWeek == DayOfWeek.Monday)
        return "Пн";
    else if (dayOfWeek.DayOfWeek == DayOfWeek.Tuesday)
        return "Вт";
    else if (dayOfWeek.DayOfWeek == DayOfWeek.Wednesday)
        return "Ср";
    else if (dayOfWeek.DayOfWeek == DayOfWeek.Thursday)
        return "Чт";
    else if (dayOfWeek.DayOfWeek == DayOfWeek.Friday)
        return "Пт";
}

```

```

        else if (dayOfWeek.DayOfWeek == DayOfWeek.Saturday)
            return "Сб";
        else
            return "Вс";
    }

    //Аналогично предыдущему методу возвращает день недели, только
    в числовом виде
    int SetDayOfWeek(DateTime dayOfWeek)
    {
        if (dayOfWeek.DayOfWeek == DayOfWeek.Monday)
            return 0;
        else if (dayOfWeek.DayOfWeek == DayOfWeek.Tuesday)
            return 1;
        else if (dayOfWeek.DayOfWeek == DayOfWeek.Wednesday)
            return 2;
        else if (dayOfWeek.DayOfWeek == DayOfWeek.Thursday)
            return 3;
        else if (dayOfWeek.DayOfWeek == DayOfWeek.Friday)
            return 4;
        else if (dayOfWeek.DayOfWeek == DayOfWeek.Saturday)
            return 5;
    }
    else
        return 6;
}

//Количество учебных дней в году
void SetAllDays()
{
    allDays = new DateTime[daysCount];
    allDays[0] = studyYear[0];
    for (int i = 1; i < daysCount; i++)
    {
        allDays[i] = allDays[i - 1].AddDays(1);
    }
}

//Создание кастомного календаря
void CreateCustomCalendar()
{
    calendar = new System.Windows.Forms.Button[6, 7];
    int i = 0, j = 0;
    foreach (System.Windows.Forms.Button but in
tableLayoutPanel1.Controls.OfType<System.Windows.Forms.Button>())
    {
        calendar[i, j] = but;
        j++;
        if (j == 7)
        {
            i++;
            j = 0;
        }
    }

    calendarReversed = new System.Windows.Forms.Button[6, 7];

    int a = 5, b = 6;

```

```

        for (i = 0; i < 6; i++)
        {
            for (j = 0; j < 7; j++)
            {
                calendarReversed[a, b] = calendar[i, j];
                b--;
            }
        }
    }

    b = 6;
    a--;

    //Заполнение кастомного календаря днями
    void FillCustomCalendarWithDates()
    {
        List<DateTime> thisMonthCalendar = new List<DateTime>();
        for (int m = 0; m < allDays.Count(); m++)
        {
            if (allDays[m].Month == monthNum)
            {
                thisMonthCalendar.Add(allDays[m]);
            }
        }

        int start = SetDayOfWeek(thisMonthCalendar[0]);
        int n = 0;
        bool check = false;

        for (int i = 0; i < 6; i++)
        {
            for (int j = Helper(i, start); j < 7; j++)
            {
                calendarReversed[i, j].Text =
thisMonthCalendar[n].Day.ToString();
                if (n == thisMonthCalendar.Count() - 1)
                {
                    check = true;
                    break;
                }
                n++;
            }
            if (check == true)
            {
                break;
            }
        }

        DateTime weekendsCalendar;
        for (int i = 0; i < 6; i++)
        {
            for (int j = 0; j < 7; j++)
            {
                calendarReversed[i, j].BackColor =
SystemColors.Control;
            }
        }
    }

```



```

        for (int i = 0; i < 6; i++)
        {
            for (int j = 0; j < 7; j++)
            {
                if (calendarReversed[i, j].Text == "")
                {
                    continue;
                }
                else
                {
                    weekendsCalendar = new DateTime(year,
monthNum, Convert.ToInt16(calendarReversed[i, j].Text));
                    for (int m = 0; m < weekends.Count(); m++)
                    {
                        if (weekendsCalendar == weekends[m])
                        {
                            calendarReversed[i, j].BackColor =
Color.MistyRose;
                        }
                    }
                }
            }
        }
    }

    //Кнопка возврата
    private void scheduleBtn_Click_1(object sender, EventArgs e)
    {
        mainMenu w_main = new mainMenu();
        w_main.Show();
    }

    Hide();

    //Возврат нажатой кнопки из массива кнопушек и вывод
    количества пар или же праздней
    private void button1_Click(object sender, EventArgs e)
    {
        Button tempBut = (Button)sender;
        if (tempBut.Text == "")
            return;

        DateTime selectedDay = new DateTime(year, monthNum,
Convert.ToInt16(tempBut.Text));
        string output = "";

        con.Open();
        MySqlCommand cmd = new MySqlCommand("select groupname,
lessonnum, discipline from workingdays where date =
'" + selectedDay.ToString("yyyy-MM-dd") + "' and teacherid = 1;", con);
        using (MySqlDataReader reader = cmd.ExecuteReader())
        {
            while (reader.Read())
            {
                output += reader["groupname"].ToString() + " - " +
reader["lessonnum"].ToString() + " пара, " +
reader["discipline"].ToString() + "\r\n";
            }
        }
    }

```

```

        }
    }
    con.Close();

    if(tempBut.BackColor == Color.LightGreen)
        MessageBox.Show(output);
    else
        MessageBox.Show("Выходной");
}

//Очистка календаря
void ClearCalendar()
{
    for (int i = 0; i < 6; i++)
    {
        for (int j = 0; j < 7; j++)
        {
            calendarReversed[i, j].Text = "";
        }
    }
}

//Вспомогательный метод для начала заполнения календаря с
определенного дня недели
int Helper(int a, int n)
{
    int b = 0;
    if (a > 0)
        b = 0;
    else
        b = n;
return b;
}

//Открытие окна дополнительных пар
private void button43_Click(object sender, EventArgs e)
{
    AdditionalLessons addl_win = new AdditionalLessons();
    addl_win.Show();
}

//Добавление правильного окончания к слову "час" в зависимости
от числа
string WordEnding(int number)
{
    if (((number % 100) > 10) && ((number % 100) < 20))
        return "часов";
    if (number % 10 == 1)
        return "час";
    if ((number % 10 == 2) || (number % 10 == 3) || (number %
10 == 4))
        return "часа";
    return "часов";
}

//Синхронизация рабочих дней и базы данных
public void InsertAllWorkDays()

```

```

    {
        DateKeeper.onlyWorkDays.Clear();
        con.Open();
        int tempCount = 0;
        MySqlCommand cmd = new MySqlCommand("delete from
workingdays where teacherid = 1", con);
        cmd.ExecuteNonQuery();
        discipline.Clear();
        for (int j = 0; j < allDays.Count(); j++)
        {
            cmd = new MySqlCommand("SELECT COUNT(*) FROM schedule
WHERE dayOfWeek = '" + SetDayOfWeek(allDays[j]) + "' AND partOfWeek =
'" + partOfWeekToBinary[j] + "' AND teacherId = 1;", con);
            tempCount = Convert.ToInt16(cmd.ExecuteScalar());
            if (tempCount != 0)
            {
                cmd = new MySqlCommand("SELECT * FROM schedule
WHERE dayOfWeek = '" + SetDayOfWeek(allDays[j]) + "' AND partOfWeek =
'" + partOfWeekToBinary[j] + "' AND teacherId = 1;", con);
                using (MySqlDataReader reader =
cmd.ExecuteReader())
                {
                    while(reader.Read())
                    {
                        DateKeeper.onlyWorkDays.Add(allDays[j]);

                        groupName.Add(reader["groupname"].ToString());

                        lessonNum.Add(Convert.ToInt16(reader["lesson"]));

                        discipline.Add(reader["discipline"].ToString());
                    }
                    reader.Close();
                }
            }
        }
        for(int i = 0; i < DateKeeper.onlyWorkDays.Count(); i++)
        {
            MySqlCommand cmd1 = new MySqlCommand("INSERT INTO
workingdays (dayId, date, teacherId, groupName, lessonNum, discipline)
VALUES (0, '" + DateKeeper.onlyWorkDays[i].ToString("yyyy-MM-dd") +
"', 1, '" + groupName[i].ToString() + "', '" +
(lessonNum[i]+1).ToString() + "', '"+discipline[i].ToString()+"');" ,
con);

            cmd1.ExecuteNonQuery();
        }

        weekends.Clear();
        cmd = new MySqlCommand("select * from holidays;", con);
        using (MySqlDataReader reader = cmd.ExecuteReader())
        {
            while(reader.Read())
            {
                weekends.Add(DateTime.Parse(reader["holidayDate"].ToString()));
            }
            reader.Close();
        }
    }

```

```

    }

    for(int i = 0; i < weekends.Count(); i++)
    {
        cmd = new MySqlCommand("delete from workingdays where
date = '"+weekends[i].ToString("yyyy-MM-dd")+"' and teacherid = 1;",
con);
        cmd.ExecuteNonQuery();
    }

    con.Close();
}

//Вывод часов и пар за интервал
private void button44_Click(object sender, EventArgs e)
{
    CreateWorkingDaysList();
    con.Open();
    DateTime tempFrom =
DateTime.Parse(fromDatePicker.Value.ToShortDateString());
    DateTime tempTo =
DateTime.Parse(toDatePicker.Value.ToShortDateString());
    int lessonCount = 0;
    for (int i = 0; i < DateKeeper.onlyWorkDays.Count(); i++)
    {
        if(DateKeeper.onlyWorkDays[i] >= tempFrom &&
DateKeeper.onlyWorkDays[i] <= tempTo)
        {
            lessonCount++;
        }
    }
    int HoursCount = 0;
    HoursCount = lessonCount * 2;
    lessonLabel.Text = lessonCount.ToString();
    hoursLabel.Text = HoursCount.ToString();
    con.Close();
}

//Открыть удаление пар
private void button45_Click(object sender, EventArgs e)
{
    DeleteLesson del_win = new DeleteLesson();
    del_win.Show();
}

//Синхронизация
private void button46_Click(object sender, EventArgs e)
{
    InsertAllWorkDays();
}

//Заполнение календаря рабочими днями
void ImportFromWorkingDays()
{
    for (int i = 0; i < DateKeeper.onlyWorkDays.Count(); i++)
    {
        foreach (Button b in calendarReversed)

```

```

        {
            if (DateKeeper.onlyWorkDays[i].Month == monthNum
&& b.Text == DateKeeper.onlyWorkDays[i].Day.ToString())
            {
                b.BackColor = Color.LightGreen;
            }
        }
    }
    foreach (Button b in calendarReversed)
    {
        if (b.BackColor != Color.LightGreen && b.Text != "")
        {
            b.BackColor = Color.MistyRose;
        }
    }
}

//Показывать числитель и знаменатель в отдельном окошке
void ShowChislitelZnamenatel()
{
    richTextBox1.Clear();
    for(int i = 0; i < allDays.Count(); i++)
    {
        if(allDays[i].Month == monthNum)
        {
            if(partOfWeekToBinary[i] == 0)
            {
                richTextBox1.Text += "Числитель " + "\r\n" +
allDays[i].ToShortDateString();
                if(i+6 >= allDays.Count()-1)
                {
                    i = allDays.Count() - 1;
                }
                else
                    i += 6;
                richTextBox1.Text += " - " +
allDays[i].ToShortDateString() + "\r\n";
            }
            else
            {
                richTextBox1.Text += "Знаменатель " + "\r\n" +
allDays[i].ToShortDateString();
                if (i + 6 >= allDays.Count() - 1)
                {
                    i = allDays.Count() - 1;
                }
                else
                    i += 6;
                richTextBox1.Text += " - " +
allDays[i].ToShortDateString() + "\r\n";
            }
        }
    }
}

public void CreateWorkingDaysList()

```

```

    {
        DateKeeper.onlyWorkDays.Clear();
        groupName.Clear();
        lessonNum.Clear();
        con.Open();
        MySqlCommand cmd = new MySqlCommand("select * from
workingdays where teacherid = 1;", con);
        using (MySqlDataReader reader = cmd.ExecuteReader())
        {
            while (reader.Read())
            {
                DateKeeper.onlyWorkDays.Add(DateTime.Parse(reader["date"].ToString()))
;
            }
            reader.Close();
        }
        con.Close();
        ImportFromMySQL();
        AddAdditionalLessons();
    }
}

```

## Листинг «Добавить пару»

```

using System;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

namespace Диплом
{
    public partial class AdditionalLessons : Form
    {
        public static string connectionString = "User Id=root; Password = 199814;
        Host=localhost;Database=diploma;Character Set=utf8";
        MySqlConnection con = new MySqlConnection(connectionString);

        public string day, lesson, groupname;

        public AdditionalLessons()
        {
            InitializeComponent();
        }

        private void button43_Click(object sender, EventArgs e)
        {
            if (monthBox.SelectedItem == null)
                MessageBox.Show("Выберите месяц!");
            elseif (selectDayBox.SelectedItem == null)
                MessageBox.Show("Выберите день!");
            elseif (lessonBox.SelectedItem == null)
                MessageBox.Show("Выберите пару!");
            elseif (groupNameBox.Text == "")
                MessageBox.Show("Введите группу!");
            day = selectDayBox.SelectedItem.ToString();
            DateTime temp = new DateTime(year, monthNum, Convert.ToInt16(day));
            con.Open();
            MySqlCommand cmd = new MySqlCommand("INSERT INTO additionallesson
            (additionalId, lesson, groupname, teacherId, date, discipline) VALUES (0,
            '"+Convert.ToInt16(lessonBox.SelectedItem)+'', '"+groupNameBox.Text.ToString()+"', 1,
            '"+temp.ToString("yyyy-MM-dd")+'', '"+disciplineBox.Text +"'");", con);
            cmd.ExecuteNonQuery();
            cmd = new MySqlCommand("INSERT INTO workingdays (dayId, date, teacherId,
            groupName, lessonNum, discipline) VALUES (0, '" + temp.ToString("yyyy-MM-dd")+'', 1, '"+
            groupNameBox.Text.ToString() + "', '"+ Convert.ToInt16(lessonBox.SelectedItem) + "', '" +
            disciplineBox.Text + "'");", con);
            cmd.ExecuteNonQuery();
            con.Close();
            DateWork d_work = new DateWork();
            d_work.CreateWorkingDaysList();
            d_work.Hide();
            d_work.Show();
            Hide();
        }

        public int monthNum, year;

        private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
        {
        }

        private void monthBox_SelectedIndexChanged(object sender, EventArgs e)
        {
            selectDayBox.Items.Clear();
        }
    }
}

```

```

if (monthBox.SelectedItem.ToString() == "Сентябрь")
{
    monthNum = 9;
    year = 2018;
}
elseif (monthBox.SelectedItem.ToString() == "Октябрь")
{
    monthNum = 10;
    year = 2018;
}
elseif (monthBox.SelectedItem.ToString() == "Ноябрь")
{
    monthNum = 11;
    year = 2018;
}
elseif (monthBox.SelectedItem.ToString() == "Декабрь")
{
    monthNum = 12;
    year = 2018;
}
elseif (monthBox.SelectedItem.ToString() == "Январь")
{
    monthNum = 1;
    year = 2019;
}
elseif (monthBox.SelectedItem.ToString() == "Февраль")
{
    monthNum = 2;
    year = 2019;
}
elseif (monthBox.SelectedItem.ToString() == "Март")
{
    monthNum = 3;
    year = 2019;
}
elseif (monthBox.SelectedItem.ToString() == "Апрель")
{
    monthNum = 4;
    year = 2019;
}
elseif (monthBox.SelectedItem.ToString() == "Май")
{
    monthNum = 5;
    year = 2019;
}
elseif (monthBox.SelectedItem.ToString() == "Июнь")
{
    monthNum = 6;
    year = 2019;
}
elseif (monthBox.SelectedItem.ToString() == "Июль")
{
    monthNum = 7;
    year = 2019;
}

for(int i = 1; i < DateTime.DaysInMonth(year,monthNum)+1; i++)
{
    selectDayBox.Items.Add(i);
}
}
}

```



## Приложение 6.

## Листинг «Удалить пару»

```

using System;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

namespace Диплом
{
    public partial class DeleteLesson : Form
    {
        public static string connectionString = "User Id=root; Password = 199814;
        Host=localhost;Database=diploma;Character Set=utf8";
        MySqlConnection con = new MySqlConnection(connectionString);

        public string num = "", name = "";
        public DateTime tempDate;

        public DeleteLesson()
        {
            InitializeComponent();
        }

        private void fromDatePicker_ValueChanged(object sender, EventArgs e)
        {
            lessonBox.Items.Clear();
            con.Open();
            tempDate = fromDatePicker.Value;
            string temp = "";
            MySqlCommand cmd = new MySqlCommand("SELECT groupname, lessonnum, discipline
            FROM workingdays WHERE date='" + tempDate.ToString("yyyy-MM-dd") + "';", con);
            using (MySqlDataReader reader = cmd.ExecuteReader())
            {
                while (reader.Read())
                {
                    temp = reader["groupname"].ToString() + " - " +
                    reader["lessonnum"].ToString() + " пара, " + reader["discipline"].ToString() + "\r\n";
                    lessonBox.Items.Add(temp);
                    temp = "";
                }
            }
            con.Close();
        }

        private void returnBtn_Click(object sender, EventArgs e)
        {
            Hide();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            name = lessonBox.SelectedItem.ToString().Substring(0,
            lessonBox.SelectedItem.ToString().IndexOf(" "));
            num =
            lessonBox.SelectedItem.ToString().Substring(lessonBox.SelectedItem.ToString().IndexOf("
            ") + 3, 1);
            con.Open();
            MySqlCommand cmd = new MySqlCommand("delete from workingdays where date='" +
            tempDate.ToString("yyyy-MM-dd") + "' and lessonnum = '" + num + "' and groupname = '" +
            name + "'", con);
            cmd.ExecuteNonQuery();
        }
    }
}

```

```

        cmd = new MySqlCommand("DELETE FROM additionallesson WHERE date = '" +
tempDate.ToString("yyyy-MM-dd") + "' AND groupname = '" + name + "' AND lesson = '" + num
+ "' AND teacherId = 1;", con);
        cmd.ExecuteNonQuery();
        MessageBox.Show("Парауспешноудалена!");
        con.Close();
        fromDatePicker_ValueChanged(sender, e);
        DateWork d_work = new DateWork();
        d_work.CreateWorkingDaysList();
    }
}
}

```

## Листинг «Редактирование праздников»

```

using System;
using System.Data;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

namespace Диплом
{
    public partial class Holidays : Form
    {
        public static string connectionString = "User Id=root; Password = 199814;
        Host=localhost;Database=diploma;Character Set=utf8";
        MySqlConnection con = new MySqlConnection(connectionString);

        public Holidays()
        {
            InitializeComponent();
            con.Open();
            MySqlCommand cmd = new MySqlCommand("select distinct holidaydate from
            holidays;", con);
            MySqlDataAdapter reader = new MySqlDataAdapter(cmd);
            DataTable table = new DataTable();
            reader.Fill(table);
            for (int i = 0; i < table.Rows.Count; i++)
            {
                listBox1.Items.Add(table.Rows[i]["holidaydate"]);
            }
            con.Close();
        }

        private void returnBtn_Click(object sender, EventArgs e)
        {
            con.Open();
            for (int i = 0; i < listBox1.Items.Count; i++)
            {
                DateTime temp = DateTime.Parse(listBox1.Items[i].ToString());
                MySqlCommand cmd = new MySqlCommand("INSERT INTO holidays (holidayId,
            holidayDate) VALUES (0, '"+temp.ToString("yyyy-MM-dd") + "')", con);
                cmd.ExecuteNonQuery();
            }
            con.Close();
            mainMenu main_win = new mainMenu();
            main_win.Show();
            Hide();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            listBox1.Items.Add(monthCalendar1.SelectionStart.ToShortDateString());
        }

        private void button2_Click(object sender, EventArgs e)
        {
            listBox1.Items.Remove(listBox1.SelectedItem);
        }
    }
}

```

## Листинг «Редактирование праздников»

```

using System;
using System.Windows.Forms;
using MySql.Data.MySqlClient;
using Microsoft.Office.Interop.Excel;

namespace Диплом
{
    public partial class Otchet : Form
    {
        public Otchet()
        {
            InitializeComponent();
            con.Open();
            MySqlCommand cmd = new MySqlCommand("select distinct groupname from
workingdays where teacherid = 1;", con);
            using (MySqlDataReader reader = cmd.ExecuteReader())
            {
                while (reader.Read())
                {
                    groupBox.Items.Add(reader["groupname"].ToString());
                }
            }
            cmd = new MySqlCommand("select distinct discipline from workingdays where
teacherid = 1;", con);
            using (MySqlDataReader reader = cmd.ExecuteReader())
            {
                while (reader.Read())
                {
                    disBox.Items.Add(reader["discipline"].ToString());
                }
            }
            finalFormationBtn.Enabled = false;
            con.Close();
        }

        public static string connectionString = "User Id=root; Password = 199814;
Host=localhost;Database=diploma;Character Set=utf8";
        MySqlConnection con = new MySqlConnection(connectionString);

        private void acceptBtn_Click(object sender, EventArgs e)
        {
            System.Data.DataTable table = new System.Data.DataTable();
            dataGridView1.DataSource = table;
            con.Open();
            MySqlCommand cmd = new MySqlCommand();
            if (groupBox.SelectedItem != null && disBox.SelectedItem == null)
            {
                cmd = new MySqlCommand("SELECT workingdays.date as Дата, groupname as
Группа, lessonnum as Пара, discipline as Дисциплина FROM workingdays WHERE date>='" +
fromDatePicker.Value.ToString("yyyy-MM-dd") + "' AND date<='" +
toDatePicker.Value.ToString("yyyy-MM-dd") + "' and teacherid = 1 and groupname='" +
groupBox.Text + "';", con);
            }
            elseif (groupBox.SelectedItem == null && disBox.SelectedItem != null)
            {
                cmd = new MySqlCommand("SELECT workingdays.date as Дата, groupname as
Группа, lessonnum as Пара, discipline as Дисциплина FROM workingdays WHERE date>='" +
fromDatePicker.Value.ToString("yyyy-MM-dd") + "' AND date<='" +
toDatePicker.Value.ToString("yyyy-MM-dd") + "' and teacherid = 1 and discipline='" +
disBox.Text + "';", con);
            }
        }
    }
}

```

```

elseif (groupBox.SelectedItem != null&& disBox.SelectedItem != null)
{
    cmd = new MySqlCommand("SELECT workingdays.date as Дата, groupname as
Группа, lessonnum as Пара, discipline as Дисциплина FROM workingdays WHERE date>='" +
fromDatePicker.Value.ToString("yyyy-MM-dd") + "' AND date<='" +
toDatePicker.Value.ToString("yyyy-MM-dd") + "' and teacherid = 1 and discipline='" +
disBox.Text + "' and groupname='" + groupBox.Text + "';", con);
}
else
{
    cmd = new MySqlCommand("SELECT workingdays.date as Дата, groupname as
Группа, lessonnum as Пара, discipline as Дисциплина FROM workingdays WHERE date>='" +
fromDatePicker.Value.ToString("yyyy-MM-dd") + "' AND date<='" +
toDatePicker.Value.ToString("yyyy-MM-dd") + "' and teacherid = 1;", con);
}
MySQLDataAdapter adapter = new MySQLDataAdapter(cmd);
adapter.Fill(table);
dataGridView1.DataSource = table;
con.Close();
finalFormationBtn.Enabled = true;
}

privatevoid returnBtn_Click(object sender, EventArgs e)
{
    mainMenu main_win = new mainMenu();
    main_win.Show();
    Hide();
}

privatevoid finalFormationBtn_Click(object sender, EventArgs e)
{
    _Application app = new Microsoft.Office.Interop.Excel.Application();
    _Workbook workbook = app.Workbooks.Add(Type.Missing);
    _Worksheet worksheet = null;
    app.Visible = true;
    worksheet = workbook.Sheets["Лист1"];
    worksheet = workbook.ActiveSheet;
    worksheet.Name = "Отчетоработе";
    worksheet.StandardWidth = 12;
    for (int i = 1; i < dataGridView1.Columns.Count + 1; i++)
    {
        worksheet.Cells[1, i] = dataGridView1.Columns[i - 1].HeaderText;
    }
    for (int i = 0; i < dataGridView1.Rows.Count - 1; i++)
    {
        for (int j = 0; j < dataGridView1.Columns.Count; j++)
        {
            if(j == 0)
            {
                worksheet.Cells[i + 2, j + 1] =
DateTime.Parse(dataGridView1.Rows[i].Cells[j].Value.ToString()).ToShortDateString();
            }
            else
            {
                worksheet.Cells[i + 2, j + 1] =
dataGridView1.Rows[i].Cells[j].Value.ToString();
            }
        }
    }
}
}

```

## Листинг «Отчет»

```

using System;
using System.Windows.Forms;
using MySql.Data.MySqlClient;
using Microsoft.Office.Interop.Excel;

namespaceДиплом
{
    publicpartialclassОтчет : Form
    {
        public Отчет()
        {
            InitializeComponent();
            con.Open();
            MySqlCommand cmd = new MySqlCommand("select distinct groupname from
workingdays where teacherid = 1;", con);
            using (MySqlDataReader reader = cmd.ExecuteReader())
            {
                while(reader.Read())
                {
                    groupBox.Items.Add(reader["groupname"].ToString());
                }
                cmd = new MySqlCommand("select distinct discipline from workingdays where
teacherid = 1;", con);
                using (MySqlDataReader reader = cmd.ExecuteReader())
                {
                    while (reader.Read())
                    {
                        disBox.Items.Add(reader["discipline"].ToString());
                    }
                }
                finalFormationBtn.Enabled = false;
                con.Close();
            }

            publicstaticstring connectionString = "User Id=root; Password = 199814;
Host=localhost;Database=diploma;Character Set=utf8";
            MySqlConnection con = new MySqlConnection(connectionString);

            privatevoid acceptBtn_Click(object sender, EventArgs e)
            {
                System.Data.DataTable table = new System.Data.DataTable();
                dataGridView1.DataSource = table;
                con.Open();
                MySqlCommand cmd = new MySqlCommand();
                if(groupBox.SelectedItem != null&& disBox.SelectedItem == null)
                {
                    cmd = new MySqlCommand("SELECT workingdays.date as Дата, groupname as
Группа, lessonnum as Пара, discipline as Дисциплина FROM workingdays WHERE date>='" +
fromDatePicker.Value.ToString("yyyy-MM-dd") + "' AND date<='" +
toDatePicker.Value.ToString("yyyy-MM-dd") + "' and teacherid = 1 and groupname='" +
groupBox.Text + "';", con);
                }
                elseif(groupBox.SelectedItem == null&& disBox.SelectedItem != null)
                {
                    cmd = new MySqlCommand("SELECT workingdays.date as Дата, groupname as
Группа, lessonnum as Пара,discipline as Дисциплина FROM workingdays WHERE date>='" +
fromDatePicker.Value.ToString("yyyy-MM-dd") + "' AND date<='" +
toDatePicker.Value.ToString("yyyy-MM-dd") + "' and teacherid = 1 and discipline='" +
disBox.Text + "';", con);
                }
            }
        }
    }
}

```

```

elseif (groupBox.SelectedItem != null&& disBox.SelectedItem != null)
{
    cmd = new MySqlCommand("SELECT workingdays.date as Дата, groupname as
Группа, lessonnum as Пара,discipline as Дисциплина FROM workingdays WHERE date>='" +
fromDatePicker.Value.ToString("yyyy-MM-dd") + "' AND date<='" +
toDatePicker.Value.ToString("yyyy-MM-dd") + "' and teacherid = 1 and discipline='" +
disBox.Text + "' and groupname='" + groupBox.Text + "';", con);
}
else
{
    cmd = new MySqlCommand("SELECT workingdays.date as Дата, groupname as
Группа, lessonnum as Пара,discipline as Дисциплина FROM workingdays WHERE date>='" +
fromDatePicker.Value.ToString("yyyy-MM-dd") + "' AND date<='" +
toDatePicker.Value.ToString("yyyy-MM-dd") + "' and teacherid = 1;", con);
}
MySQLDataAdapter adapter = new MySQLDataAdapter(cmd);
adapter.Fill(table);
dataGridView1.DataSource = table;
con.Close();
finalFormationBtn.Enabled = true;
}

privatevoid returnBtn_Click(object sender, EventArgs e)
{
    mainMenu main_win = new mainMenu();
    main_win.Show();
    Hide();
}

privatevoid finalFormationBtn_Click(object sender, EventArgs e)
{
    _Application app = new Microsoft.Office.Interop.Excel.Application();
    _Workbook workbook = app.Workbooks.Add(Type.Missing);
    _Worksheet worksheet = null;
    app.Visible = true;
    worksheet = workbook.Sheets["Лист1"];
    worksheet = workbook.ActiveSheet;
    worksheet.Name = "Отчетоработе";
    worksheet.StandardWidth = 12;
    for (int i = 1; i < dataGridView1.Columns.Count + 1; i++)
    {
        worksheet.Cells[1, i] = dataGridView1.Columns[i - 1].HeaderText;
    }
    for (int i = 0; i < dataGridView1.Rows.Count - 1; i++)
    {
        for (int j = 0; j < dataGridView1.Columns.Count; j++)
        {
            if(j == 0)
            {
                worksheet.Cells[i + 2, j + 1] =
DateTime.Parse(dataGridView1.Rows[i].Cells[j].Value.ToString()).ToShortDateString();
            }
            else
            {
                worksheet.Cells[i + 2, j + 1] =
dataGridView1.Rows[i].Cells[j].Value.ToString();
            }
        }
    }
}

```

## Листинг «Личный кабинет»

```

using System;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

namespace Диплом
{
    public partial class UserSettings : Form
    {

        public static string connectionString = "User Id=root; Password = 199814;
        Host=localhost;Database=diploma;Character Set=utf8";
        MySqlConnection con = new MySqlConnection(connectionString);

        public UserSettings()
        {
            InitializeComponent();
            con.Open();
            MySqlCommand cmd = new MySqlCommand("select teacherfio from teachers where
teacherid = 1", con);
            nameLabel.Text = cmd.ExecuteScalar().ToString();
            con.Close();
        }

        private void returnBtn_Click(object sender, EventArgs e)
        {
            mainMenu main_win = new mainMenu();
            main_win.Show();
            Hide();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            DateTime temp = fromDatePicker.Value;
            con.Open();
            MySqlCommand cmd = new MySqlCommand("UPDATE teachers SET lastWorkingDay =
'" + temp.ToString("yyyy-MM-dd") + "' WHERE teacherId = 1;", con);
            cmd.ExecuteNonQuery();
            con.Close();
            DateWork dt_w = new DateWork();
            dt_w.InsertAllWorkDays();
            MessageBox.Show("Данные обновлены!");
        }
    }
}

```