

Содержание

Введение.....	5
1 Анализ исходных данных и постановка задачи.....	6
1.1 Описание предметной области	6
1.2 Функциональные и нефункциональные требования.....	7
1.3 Техническое задание.....	7
2 Проектирование программного обеспечения	10
2.1 Проектирование функциональной структуры	10
2.2 Проектирование логической модели данных	12
2.3 Выбор инструментальных средств.....	13
2.4 Проектирование физической модели данных.....	15
3 Реализация и тестирование	19
3.1 Разработка бизнес-логики	19
3.2 Используемые компоненты при разработке программы	21
3.3 Результат реализации функциональных частей	21
3.4 Руководство пользователя.....	23
3.5 Тестирование программного продукта.....	28
4 Экономическая часть	30
4.1 Обоснование необходимости разработки программного обеспечения	30
4.2 Этапы разработки программного обеспечения.....	30
4.3 Расчет стоимости программного средства	31
4.4 Расчёт отпускной цены программного средства	36
5 Охрана труда.....	37
5.1 Идентификация и анализ вредных и опасных факторов в проектируемом объекте	37
5.2 Технические, технологические, организационные решения по устранению опасных и вредных факторов, разработка защитных средств.....	39

Изм.	Лист	№ докум.	Подпись	Дата	Информационная система «Архив учащихся кружка по программированию»		
Разраб.							
Провер.							
Реценз.							
Н. Контр.							
Утверд.							
					Лит.	Лист	Листов
						3	

5.3 Разработка мер безопасности при эксплуатации объекта проектирования.	40
Заключение	45
Список использованных источников	46
ПРИЛОЖЕНИЕ А (Обязательное) Код создания базы данных.....	47

Введение

В настоящее время в связи с развитием компьютерной техники появилась возможность автоматизировать многие процессы, также увеличился объем обрабатываемой информации. И возникла объективная необходимость автоматизировать большую часть сферы человеческой деятельности.

Для автоматизации обработки данных в начале 70-х годов были предложены программы, специально предназначенные для управления данными – системы управления базами данных (СУБД).

В самом общем смысле база данных – совокупность специальным образом организованных данных, хранимых в памяти вычислительной системы и отображающих состояние объекта и их взаимосвязей в рассматриваемой предметной области [1].

Задача базы данных состоит в хранении всех представляющих интерес данных в одном или нескольких местах, причём таким способом, который заведомо исключает ненужную избыточность. Создание баз данных преследует две основные цели: понизить избыточность данных и повысить их надёжность.

В рамках дипломного проектирования разработана Информационная система «Архив учащихся кружка по программированию». Данная система позволит оптимизировать процесс формирования отчётов, автоматизирует процесс ведения журнала тем и оценок учащихся в кружке по программированию. Также облегчит всю работу, которая до этого велась на бумажных носителях и решит проблему повторения и потери данных.

В первой главе записки к дипломному проекту приводится описание предметной области, поставлены цели к разрабатываемой системе, описано техническое задание к разработке.

Во второй главе приводится функциональная модель системы, разрабатывается структура базы данных, определяются инструментальные средства разработки и пользовательский интерфейс.

В третьей главе представлены результаты реализации функциональных частей и результаты тестирования системы.

В четвертой главе выполнены расчеты экономической эффективности внедрения разрабатываемой автоматизированной системы.

В пятой главе рассмотрены вопросы охраны труда.

1 Анализ исходных данных и постановка задачи

1.1 Описание предметной области

Кружок по программированию занимается обучением приходящих учащихся программированию, работой с компьютером в целом. Кружок по программированию состоит из множества учащихся, которые занимаются в этом кружке. Эти учащиеся могут приходить из разных школ, и, соответственно, ходить в разные классы.

При приходе нового учащегося в кружок по программированию, его данные записываются в журнал, в котором есть данные всех членов кружка.

Каждое занятие преподаватель с учащимися разбирает какую-то тему, которая также должна быть отображена в журнале. По темам может выдаваться практическая работа, за которую будет поставлена оценка, чтобы следить за успеваемостью учащегося. Также, на темах может присутствовать или отсутствовать тот или иной учащийся, что также при необходимости может быть задокументировано.

Время от времени проходят соревнования, которые по своему виду делятся на городские, областные и республиканские. Учащие кружка по программированию могут принимать участия, занимая определенные места в соревновании. Важно помнить, что соревнования могут проходить как по месту проведения кружка, так и в любом другом.

Так же ученик может внести денежный вклад в развитие кружка, что фиксируется в отдельном месте. Кроме того, один и тот же участник может делать это неоднократно.

Периодически составляются отчеты следующего вида:

- отчет по оценкам – общая таблица, на которой отображены все оценки всех учащихся по всем темам с выводением среднего бала
- отчет по вкладам – содержит информацию об учащихся, которые внесли денежный вклад с размером вносимого вклада и высчитыванием всех полученных денег.
- отчет по соревнованиям – содержащий информацию по каждому соревнованию, такую как список участвовавших учащихся и занятых ими мест.

К основным целям функционирования автоматизированной системы относятся:

- повышение качества работы кружка в целом за счёт своевременного и точного предоставления информации;
- увеличение производительности заполнения и хранения данных;
- уменьшение количества ошибок при вводе данных;
- оптимизация процесса формирования отчётов.

						Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		

1.2 Функциональные и нефункциональные требования

В рамках дипломного проекта необходимо разработать информационную систему «Архив учащихся кружка по программированию», которое позволило бы преподавателю кружка:

- Хранить данные об учащемся;
- Хранить данные об соревнованиях;
- Хранить данные об вкладах в развитие кружка;
- Добавлять, изменять и удалять данные об учащихсья;
- Добавлять, изменять и удалять данные об соревнованиях;
- Добавлять, изменять и удалять данные об вкладах в развитие кружка;
- Осуществлять сортировку по всем столбцам таблиц;
- Осуществлять поиск учащего или соревнования;
- Выставлять оценки учащимся по выбранным темам;
- Автоматизировать формирование отчётности.

Разрабатываемый программный продукт должен удовлетворять следующим требованиям к нефункциональным характеристикам:

- Работать на персональных компьютерах с объёмом оперативной памяти не менее 2048 Мб;
- Следить за правильностью ввода;
- Иметь русский интерфейс.

Аппаратные требования

Для полноценной работы системы компьютер пользователя должен удовлетворять следующим аппаратным требованиям:

- процессор AMD Athlon X4 840 3.10 GHz;
- ОЗУ: не менее 2Гб;
- HDD: не менее 250Гб;
- видеоадаптер: DirectX9;
- видеопамять: не менее 32 МБ;

1.3 Техническое задание

Введение [2]

Настоящее техническое задание распространяется на разработку информационной системы «Архив учащихся кружка по программированию».

Тема разработки: Информационная система «Архив учащихся кружка по программированию» для предприятия «ГУДО дворец детей и молодёжи».

Целевая аудитория: программный продукт предназначен для преподавателя кружка по программированию.

1 Основание для разработки

Данный проект разрабатывается на основании задания дипломного проектирования для предприятия «ГУДО дворец детей и молодёжи» для автоматизации работы кружка по программированию

					Лист
					7
Изм.	Лист	№ докум.	Подпись	Дата	

2 Назначение разработки

2.1 Функциональное назначение

Программное средство предназначено для ведения хранилищ данных видов соревнований, соревнований, участвующих в соревнованиях, учащихся, тем, журнала оценок. Также для выставления учащимся оценок по темам кружка, ведение журнала посещений соревнований и вести учёт денежного вклада в кружок. Программа сделана для организации «ГУДО дворец детей и молодёжи», которое облегчит работу преподавателю.

2.2 Эксплуатационное назначение

Разрабатываемый программный продукт предназначен для использования преподавателем.

3 Требования к программе

3.1 Требования к функциональным характеристикам

Для разрабатываемого программного продукта были предъявлены следующие требования к функциональным характеристикам:

- добавление, удаление и редактирование данных всех таблиц базы данных;
- поиск данных по фамилиям учащихся или названию соревнований;
- сортировка данных таблиц по всем столбцам;
- формирование отчёта.

3.2 Требования к надёжности

Разработанное программное средство обеспечит нормальное функционирование, контроль входных данных, сохранность информации.

3.3 Условия эксплуатации

Программный продукт может эффективно функционировать на любом персональном компьютере.

3.4 Требования к составу и параметрам технических средств

Для обеспечения полноценной работы программы на персональном компьютере требуется:

- объём оперативной памяти не менее 2048 Мб;
- не менее 256 мегабайт свободной памяти;
- наличие мыши и клавиатуры;
- наличие принтера.
- наличие установленного .NET Framework 4.6 и сопутствующих библиотек
- наличие установленной базы данных MySQL 8.

3.5 Требования к маркировке и упаковке

Разработанный программный продукт должен поставляться на CD – диске, укомплектованный пакетом документации.

3.6 Требования к транспортировке и хранению

						Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

Разработанный программный продукт должен транспортироваться и храниться в специально-предназначенной коробке или конверте.

4 Требования к программной документации

Состав программной документации должен включать в себя следующие программные документы:

- пояснительная записка;
- техническое задание;
- программа и методика испытаний;
- руководство оператора.

Программная документация должна быть оформлена в соответствии со следующими нормативными документами:

ГОСТ 19.201 – 78ЕСПД. Техническое задание. – Издательство стандартов, 1982 г.

ГОСТ 19.505 – 79 ЕСПД. Программа и методика испытаний. – Издательство стандартов, 1982г.

ГОСТ 19.301 – 2000 ЕСПД. Руководство оператора. – Издательство стандартов, 2001г.

5 Техничко-экономические показатели

Улучшение технико-экономических показателей достигается за счёт следующих характеристик:

- простота использования по сравнению с аналогичными программами;
- программа предназначена для учёта учащихся, выставления оценок, внесение учащимся денежного вклада в развитие кружка, внесение достижений ученика.

6 Стадии и этапы разработки

Разработка программного продукта делится на следующие этапы:

- анализ поставленной задачи;
- определение требований к системе;
- проектирование системы;
- реализация системы;
- тестирование разработанного ПП;
- составление проектной документации.

7 Порядок контроля и приёмки

Для контроля выполнения требований индивидуального задания необходимо провести испытания. Порядок и состав испытаний определяется документом «Программа и методика испытаний», разработанным по ГОСТ 19.301-2000. Приём проекта осуществляется руководителем дипломного проекта.

						Лист
						9
Изм.	Лист	№ докум.	Подпись	Дата		

2 Проектирование программного обеспечения

2.1 Проектирование функциональной структуры

Учитывая требования к проектируемому программному продукту, разработана функциональная структура, которая состоит из подсистем:

- подсистема для работы с темами и оценками. Предназначена для добавления, удаления и изменения информации темах кружка, а также выставления оценок по выбранной теме или обозначение отсутствующих лиц.
- подсистема для работы с учащимися. Используется для внесения в систему, редактирования и удаления информации об учащихся, а также работы в связанных таблицах с информацией по вкладам выбранного учащегося и просмотра списка его участия в соревнованиях.
- подсистема для работы с соревнованиями. Предназначена для внесения в систему, просмотра и редактирования информации о видах соревнований, внесения в систему, просмотра и редактирования информации о соревнованиях выбранного вида и внесения в систему, просмотра и редактирования информации об учащихся, принимающих участие в выбранном соревновании..
- подсистема для работы с отчётами. Предназначена для удобного и быстрого формирования различных отчётов.
- подсистема для работы с вкладом в развитие кружка. Предназначена для внесения в систему, просмотра и редактирования информации о денежных вкладах в кружок по программированию.

Для наглядности взаимодействия подсистем разработана функциональная структура программного средства (рис. 2.1):



Рисунок 2.1 – Функциональная структура

Диаграмма вариантов использования является исходным концептуальным представлением или концептуальной моделью системы в процессе ее проектирования и разработки. Разработка диаграммы вариантов использования преследует цели [3]:

- определить общие границы и контекст моделируемой предметной области;

						Лист
						10
Изм.	Лист	№ докум.	Подпись	Дата		

- сформулировать общие требования к функциональному поведению проектируемой системы;
- разработать исходную концептуальную модель системы для ее последующей детализации в форме логических и физических моделей;
- подготовить исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями.

Суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью так называемых вариантов использования. При этом актером (actor) или действующим лицом называется любая сущность, взаимодействующая с системой извне.

Это может быть человек, техническое устройство, программа или любая другая система, которая может служить источником воздействия на моделируемую систему так, как определит сам разработчик. В свою очередь, вариант использования (use case) служит для описания сервисов, которые система предоставляет актеру. Другими словами, каждый вариант использования определяет некоторый набор действий, совершаемый системой при диалоге с актером. При этом ничего не говорится о том, каким образом будет реализовано взаимодействие актеров с системой. Диаграмма вариантов использования отображает все действия, которые могут быть выполнены в программе, а также иерархию распределения ролей в системе.

Диаграмма представлена на рисунке 2.2:

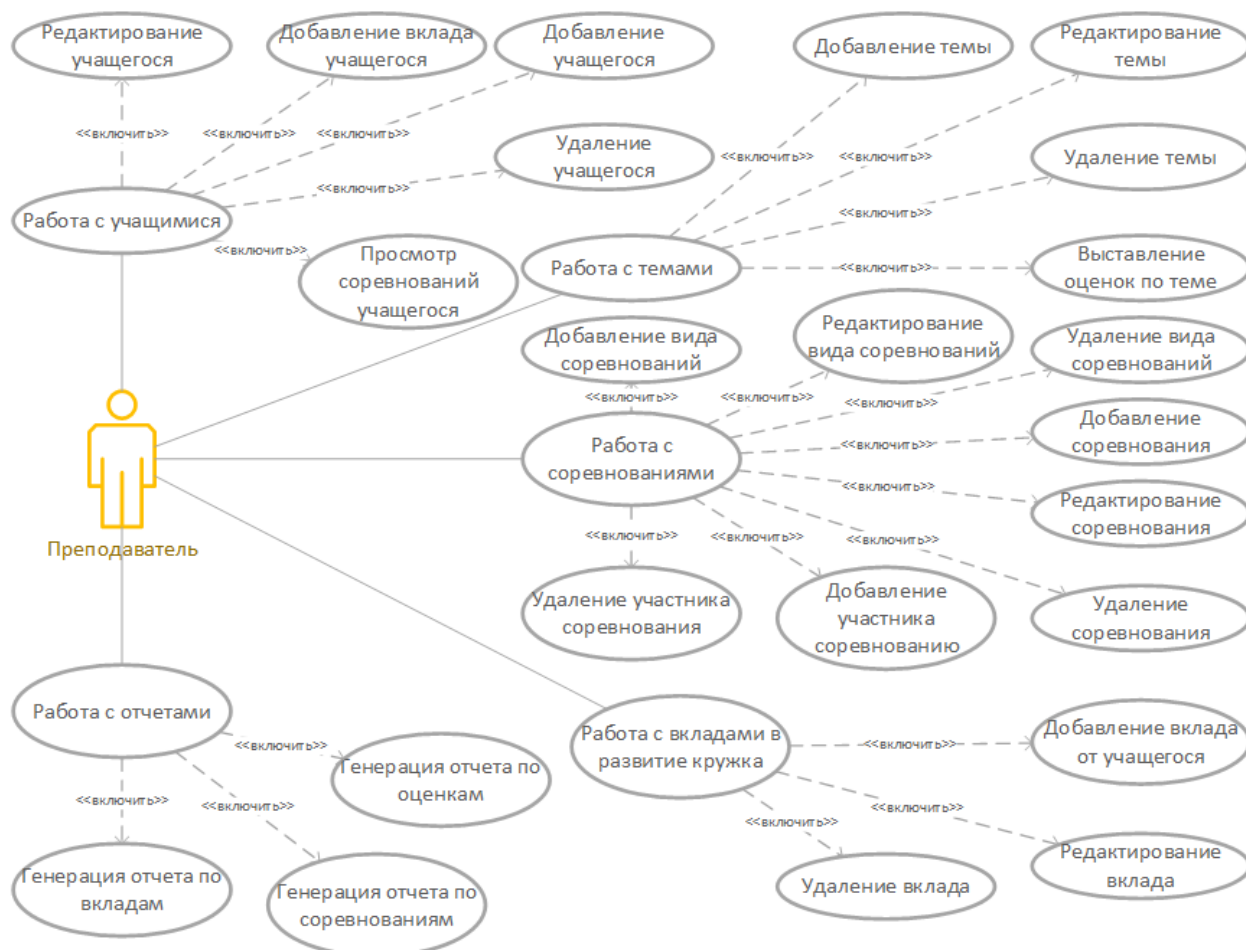


Рисунок 2.2 – Диаграмма вариантов использования

2.2 Проектирование логической модели данных

Для любой предметной области существует множество вариантов проектных решений ее отображения в логической модели. Логический уровень — это абстрактный взгляд на данные, когда данные представляются так, как выглядят в реальном мире. Объекты модели, представляемые на логическом уровне, называются сущностями и атрибутами [4].

Этап создания логической модели называется логическим проектированием. Логическая модель данных разрабатывается на основе существующих моделей данных, но не связана с конкретной реализацией системы управления базой данных (СУБД) и прочих физических условий реализации.

Логическая модель данных является источником информации для фазы физического проектирования. Она предоставляет разработчику физической модели данных средства проведения всестороннего анализа различных аспектов работы с данными, что имеет важное значение для выбора эффективного проектного решения.

Ниже будет представлен список сущностей с их атрибутами, разработанные в результате логического моделирования.

- Учащийся (Код, Фамилия, Имя, Отчество, Школа, Класс, Дата рождения, Дата регистрации);
- Виды соревнований (Название);
- Соревнование (Код, Название, Вид, Описание, Дата, Время, Адрес);
- Участники соревнований (Код, Соревнование, Участник, Место);
- Темы (Код, Название, Описание, Дата);
- Журнал оценок (Код, Учащийся, Тема, Оценка);
- Вклад в развитие (Код, Учащийся, Дата, Вклад).

2.3 Выбор инструментальных средств

Для разработки программного средства организации и функционирования программы необходимо выбрать среду разработки, с помощью которой будет производиться проектирование. При разработке ПО важное значение занимает проблема производительности. Выбор языка программирования играет большую роль в решении этой проблемы.

СУБД MicrosoftAccess – наиболее широко используемая в среде MicrosoftWindows реляционная СУБД. MicrosoftAccess – типичная СУБД для персональных компьютеров, обеспечивающая хранение, сортировку и поиск данных для множества приложений [5]. В СУБД MicrosoftAccess для создания таблиц, запросов, форм и отчётов предусмотрен графический интерфейс пользователя (GraphicalUserInterface – GUI). Кроме того, в СУБД Access предусмотрены программы, называемые мастерами (Wizards), которые упрощают многие из процессов формирования приложений с базой данных. В СУБД Access предусмотрены также конструкторы (Builders), которые могут помочь пользователю сформировать синтаксически правильные выражения. СУБД MicrosoftAccess поддерживает значительную часть стандарта языка SQL, а также стандарт MicrosoftODBC (OpenDatabaseConnectivity – открытый интерфейс доступа к базам данных), обеспечивающий общий интерфейс для доступа к разнотипным базам данных SQL.

MySQL является реляционной СУБД и в SQL-запросах позволяет связывать таблицы по общим полям, поддерживает индексы, автоинкрементные поля, а также множество функций для преобразования данных. MySQL является бесплатной и свободно распространяемой системой. Данная СУБД обладает достаточно функциональным набором для создания простых приложений. Также на сайте разработчика можно найти много свободно распространяемых компонентов и утилит для работы СУБД. Ведётся работа по дальнейшему развитию MySQL [6].

MicrosoftSQLServer 2000 – система управления реляционными базами данных. Применяется для сред клиент/сервер. В SQLServer реализовано множество функциональных возможностей, включая новый способ организации хранения данных, утилиты администрирования, поддержку технологий ActiveX, COM/DCOM.

Реализация технологий MicrosoftSearch, OLAP, MicrosoftRepository и др. Для обращения к базам данных SQLServer можно используются интерфейсы OLEDB и ODBC. Если необходимо обратиться к данным из приложения, написанного на языке высокого уровня, можно использовать специализированные API:ActiveXObject (ADO), DNA, RDO. SQLServer, является продуктом, масштабируемым от ноутбука до многопроцессорного кластера. Сервер постоянно отслеживает потребность в тех или иных ресурсах и динамически изменяет параметры своей настройки. В обработчике запросов SQLServer реализованы новые методы поиска, повышающие скорость обработки комплексных запросов.

В результате анализа можно сделать вывод, что при наличии в каждой из выбранных СУБД требуемого набора возможностей, достаточной производительности СУБД MySQL является наиболее простой в использовании и при этом полностью соответствует всем предъявляемым требованиям. Таким образом, для учебного проекта наиболее соответствует требованиям СУБД MySQL.

Для создания приложения используется платформа Microsoft Visual Studio Professional 2019.

C# является объектно-ориентированным и в этом плане много перенял у Java и C++. Поддерживает полиморфизм, наследование, перегрузку операторов, статическую типизацию. Объектно-ориентированный подход позволяет решить задачи по построению крупных, но в тоже время гибких, масштабируемых и расширяемых приложений [7].

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, разыменованние, 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.

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		14

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#.

2.4 Проектирование физической модели данных

В соответствии с разработанной моделью данных разработана база данных, которая состоит из группы таблиц, связанных между собой.

Таблица «Вклад в развитие» хранит информацию о том, кто когда и сколько денег внес в развитие. Таблица имеет следующую структуру (табл. 2.1):

Таблица 2.1 – Атрибуты сущности «Вклад в развитие»

Название атрибута	Описание	Тип данных
Код	Код сущности	Числовой, обязательное поле, внешний ключ
Учащийся	Ссылка на ключ учащихся	Числовой
Дата	Дата вклада	Дата и время
Вклад	Сумма вклада	Денежный

Таблица «Учащийся» хранит информацию об учащиххся кружка. Таблица имеет следующую структуру (табл. 2.2):

Таблица 2.2 – Атрибуты сущности «Учащийся»

Название атрибута	Описание	Тип данных
Код	Код сущности	Числовой, обязательное поле, внешний ключ
Фамилия	Фамилия учащегося	Текстовое поле
Имя	Имя учащегося	Текстовое поле
Отчество	Отчество учащегося	Текстовое поле
Школа	Название его школы	Текстовое поле
Класс	В каком он классе	Текстовое поле
Дата регистрации	Дата прихода в кружок	Дата и время

Таблица «Темы» хранит информацию о темах занятия. Таблица имеет следующую структуру (табл. 2.3):

Таблица 2.3 – Атрибуты сущности «Темы»

Название атрибута	Описание	Тип данных
Код	Код сущности	Числовой, обязательное поле, внешний ключ
Название	Название темы урока	Текстовое поле
Дата	Дата проведения урока	Дата и время
Описание	Краткие пояснения	Текстовое поле

Таблица «Соревнования» хранит информацию о проходимых соревнованиях. Таблица имеет следующую структуру (табл. 2.4):

Таблица 2.4 – Атрибуты сущности «Соревнования»

Название атрибута	Описание	Тип данных
Код	Код сущности	Числовой, обязательное поле, внешний ключ
Название	Название соревнования	Текстовое поле
Вид	Тип, ссылка на первичный ключ таблицы видов соревнований	Текстовое поле
Описание	Краткое описание	Текстовое поле
Дата	Когда проходит	Дата и время
Время	В какое время начало	Дата и время
Адрес	По какому адресу	Текстовое поле

Таблица «Журнал оценок» хранит информацию об оценках учеников по темам. Таблица имеет следующую структуру (табл. 2.5):

Таблица 2.5 – Атрибуты сущности «Журнал оценок»

Название атрибута	Описание	Тип данных
Код	Код сущности	Числовой, обязательное поле, внешний ключ
Учащийся	Код учащегося	Числовой
Тема	Код темы	Числовой
Оценка	Оценка текущего учащего за выбранную тему	Числовой

Присутствовал	Присутствовал ли он на занятии	Логический
---------------	--------------------------------	------------

Таблица «Виды соревнований» хранит информацию о видах соревнований. Таблица имеет следующую структуру (табл. 2.6):

Таблица 2.6 – Атрибуты сущности «Виды соревнований»

Название атрибута	Описание	Тип данных
Название	Название вида	Текстовое поле, обязательное поле, внешний ключ

Таблица «Участники соревнований» хранит информацию о том, кто в каком соревновании участвовал и какое место занял. Таблица имеет следующую структуру (табл. 2.7):

Таблица 2.7 – Атрибуты сущности «Участники соревнований»

Название атрибута	Описание	Тип данных
Код	Код сущности	Числовой, обязательное поле, внешний ключ
Соревнование	Код соревнования	Числовой
Участник	Код учащегося	Числовой
Место	Занятое место в соревновании	Числовой

Схема данных имеет вид (рис. 2.4):

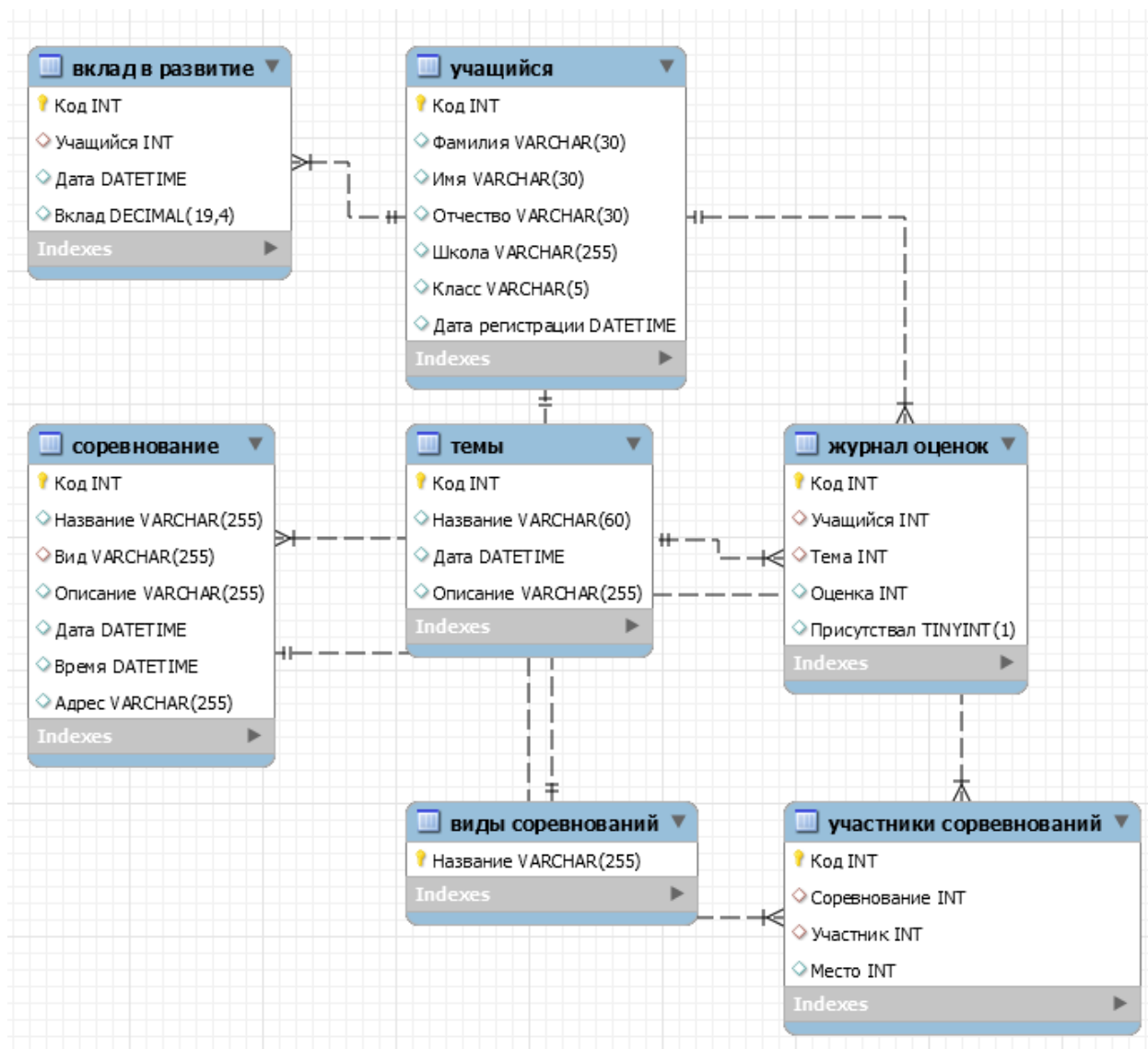


Рисунок 2.4 – Схема данных

3 Реализация и тестирование

3.1 Разработка бизнес-логики

Для использования данных в приложении необходимо установить соединение. Для этого необходимо создать новый источник данных на вкладке «Источники данных». Запустится окно мастера (рис. 3.1):

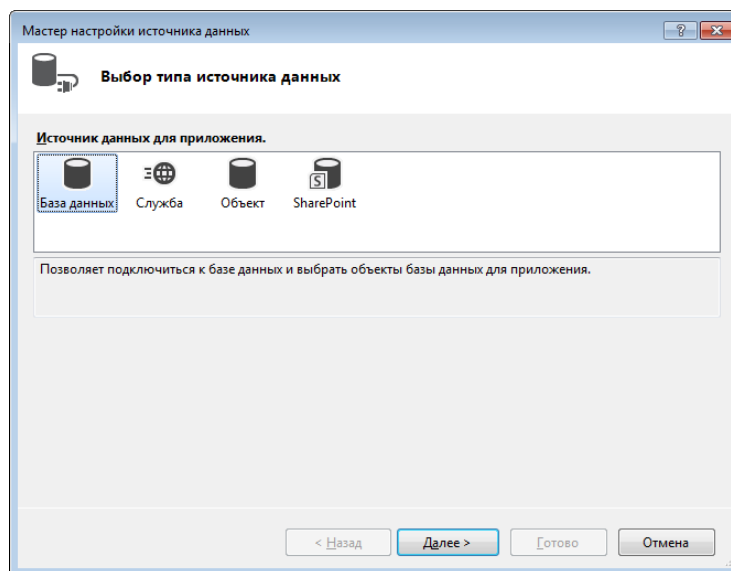


Рисунок 3.1 – Мастер создания нового источника данных.

В окне выберем нужный тип - «База данных» - нажмем кнопку «Далее». Следующий шаг предлагает выбрать тип модели данных (рис. 3.2):

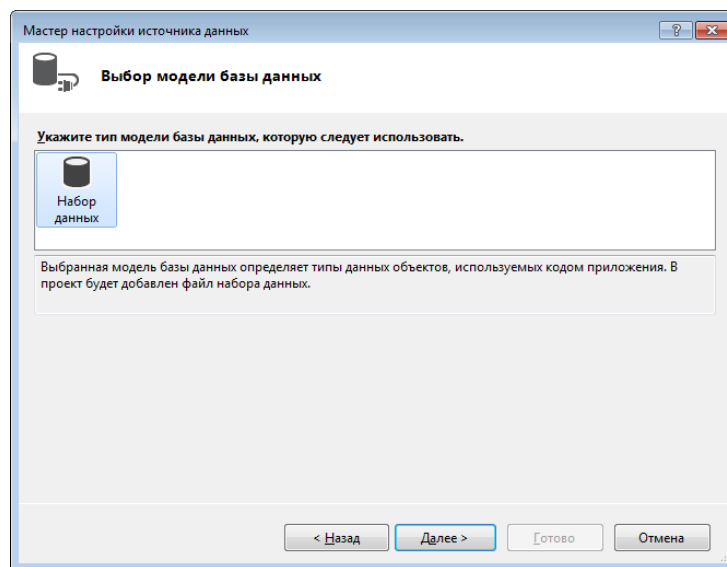


Рисунок 3.2 – Выбор модели данных.

Выбрав «Набор данных», переходим к следующему шагу (рис. 3.3):

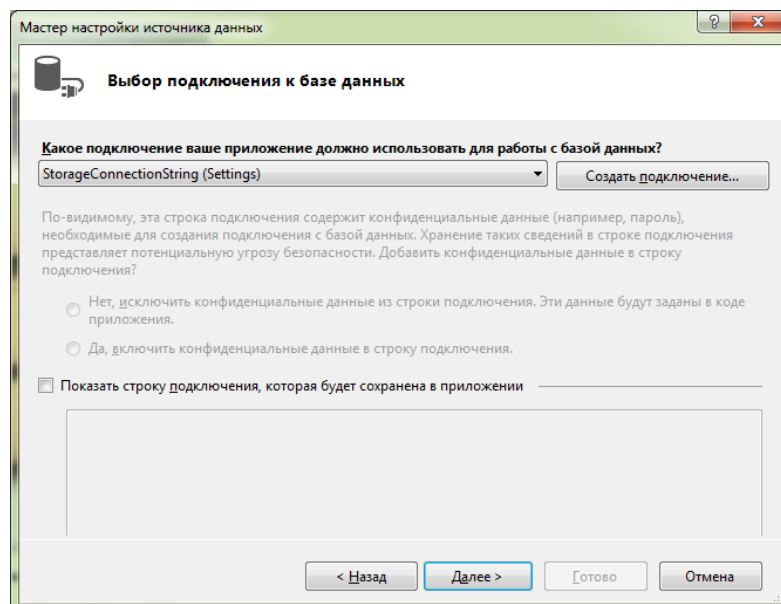


Рисунок 3.3 – Выбор подключения базы данных.

В открывшемся окне создаем подключение, если его не было создано ранее. Для создания необходимо нажать соответствующую кнопку. Откроется окно (рис. 3.4):

Рисунок 3.4 – Окно добавления нового подключения.

На данном этапе заполняем (выбираем) все данные. Нажимаем кнопку «ОК». На этом процесс подключения данных завершен. Все объекты можно использовать для работы в приложении.

3.2 Используемые компоненты при разработке программы

Для отображения данных в приложении используются следующие объекты:

- datagridview – таблица для отображения данных. Может использоваться как для отображения данных из базы, так и в качестве самостоятельной единицы;
- textbox – используется для ввода/вывода текста, который в последующем используется в программе, или для вывода результатов расчетов программы;
- combobox – позволяет пользователю выбрать значение из списка или задать собственное. Заполнить данный список можно во время разработки формы, а также элемент может отображать данные из базы;
- groupbox – предназначен для группирования объектов в определенных местах на форме
- bindingsource – предназначен для получения данных из базы данных и вывода их в адаптеры
- bindingNavigator – позволяет переходить по записям последовательно и работать с ними как с обычной таблицей

3.3 Результат реализации функциональных частей

Диаграммы классов являются одной из форм статического описания системы с точки зрения ее проектирования, показывая ее структуру. Диаграмма классов не отображает динамическое поведение объектов, изображенных на ней классов [8]. На диаграммах классов показываются классы, интерфейсы и отношения между ними. Диаграмма классов может отражать, в частности, различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывать их внутреннюю структуру и типы отношений. Каждый блок на диаграмме классов соответствует разработанному классу в программе. Атрибутами являются элементы, отображаемые на форме (поле, надпись, список), а также объявленные при необходимости переменные. Операции представляют собой функции и процедуры класса.

Диаграмма классов представлена на рисунке 3.5:

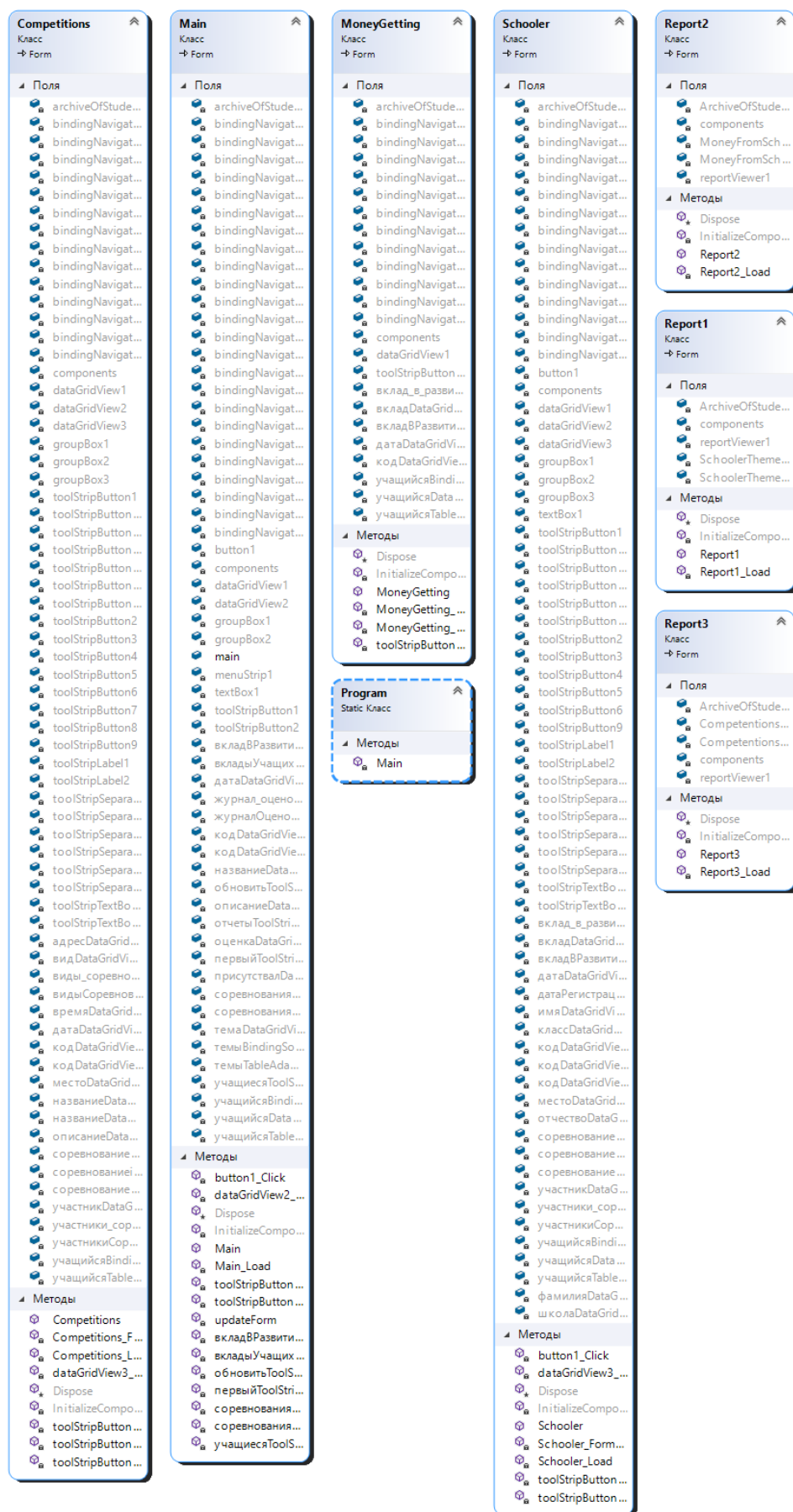


Рисунок 3.5 – Диаграмма классов

3.4 Руководство пользователя

Для начала работы с программой необходимо запустить файл «ArchiveOfStudentsOfTheProgrammingCircle.exe» и загрузится главная форма (рис. 3.6):

The screenshot shows the main window of the application. At the top, there is a menu bar with the following items: 'Учащиеся', 'Соревнования', 'Вклад в развитие кружка', 'Отчеты', and 'Обновить'. Below the menu bar, there is a search bar with a 'Найти тему' button. The main content area is divided into two sections. The first section, titled 'Темы', contains a table with the following data:

Название	Дата	Описание
Уроки рисования Photoshop 1	3/1/2021	Учимся рисовать
Уроки рисования Photoshop 2	3/2/2021	Учимся рисовать
Уроки рисования Photoshop 3	3/3/2021	Учимся рисовать
Pascal 1	3/4/2021	Программируем
Pascal 2	3/5/2021	Программируем
Opera	3/6/2021	Уроки браузера
Chrome	3/7/2021	Просмотр браузера
What is Internet	3/8/2021	Общение интернетом

The second section, titled 'Оценки за темы', contains a table with the following data:

Учащийся	Оценка	Присутствовал
Алистратова	8	<input checked="" type="checkbox"/>
Мосякова	7	<input checked="" type="checkbox"/>
Николюк	6	<input checked="" type="checkbox"/>
Кочергов	8	<input checked="" type="checkbox"/>
Косоруков	7	<input checked="" type="checkbox"/>
Хромченко	9	<input checked="" type="checkbox"/>
Бондарев	4	<input checked="" type="checkbox"/>
Дубиновский	8	<input checked="" type="checkbox"/>
Пургина	9	<input checked="" type="checkbox"/>
*		<input type="checkbox"/>

Рисунок 3.6– Главная форма

В верхней части главной формы отображается меню для навигации по системе. Первая таблица представляет собой список проходимых тем по датам. При нажатии на любую из них, во второй таблице, что находится ниже, появляются данные об отметках учащихся по этой теме. При необходимости можно добавить нового учащегося для выбранной темы, выбрав его из выпадающего списка и указать его оценку. Или просто можно добавить учащегося и указать его отсутствие на занятии. Для сохранения данных после манипуляции с таблицами, нужно нажать кнопку “Сохранить изменения” в изменяемой таблице.

Также можно найти и вывести определенную тему, используя поиск и соответствующую кнопку вверху формы.

При выборе «Учащиеся» открывает окно для работы со списком учащихся (рис. 3.7). Помимо списка учащихся и возможности их поиска по

ФИО, под таблицей со спискам есть еще две – с вкладами и соревнованиями. При выборе учащегося, в таблице вкладов отображены все вклады этого учащегося с возможностью их редактирования. Также, при выборе учащегося в таблице соревнований указаны все соревнования и занятые места этим учащимся.

The screenshot shows a window titled "Учащиеся" with a search bar and a "Найти учащегося" button. Below is a table of students with columns: Фамилия, Имя, Отчество, Школа, Класс, and Дата регистрации. The first row is highlighted. Below the main table are two sub-tables: "Вклад в развитие" and "Участие в соревнованиях".

Фамилия	Имя	Отчество	Школа	Класс	Дата регистрации
Алистратова	Варвара	Владиленовна	33	7	7/21/2004
Мосякова	Арина	Владиленовна	44	7	2/18/2004
Николюк	Максим	Ефремович	55	7	3/13/2004
Кочергов	Филипп	Тарасович	33	11	10/23/2004
Косоруков	Агафон	Агапович	25	11	7/5/2004
Хромченко	Людмила	Василиевна	84	8	12/23/2004

Дата	Вклад
3/11/2021	20.0000

Соревнование	Место
Рисование Photoshop	1
Рисование Photoshop	2
Рисование Photoshop	1

Рисунок 3.7 – Список учащихся

При выборе пункта меню «Соревнования», открывается форма для работы с видами соревнований, самими соревнованиями и участниками соревнований (рис. 3.8).

При выборе вида соревнования, во второй таблице показаны все соревнования данного вида. Затем, при нажатии на любое соревнование, в последней таблице показаны все учащиеся, которые участвуют в этом соревновании и занятые ими места. Во все эти таблицы можно добавлять данные, редактировать их и удалять, после чего также нужно нажать кнопку сохранения.

Соревнования

Виды соревнований

1 для 3 Сохранить изменения

Название
Городской
Областной
Республиканский

Соревнования выбранного вида

3 для 3 Сохранить изменения

Название	Описание	Дата	Время	Адрес
Рисование Photoshop	Первый этап рисования	3/7/2021	12/30/1899 9:00 AM	Школа 12
Pascal word	Мир программирования	3/8/2021	12/30/1899 10:00 AM	Оперный театр
Internet review	Знание интернета	3/9/2021	12/30/1899 1:45 PM	Гимназия 2

Участники выбранного соревнования

1 для 4 Сохранить изменения

Участник	Место
Косоруков	8
Хромченко	6
Бондарев	14
Дубиновский	22

Рисунок 3.8 – Соревнования

При выборе «Вклад в развитие кружка» загружается форма для просмотра и редактирования данных о вкладах. Тут отображена информация о том, какой учащийся, когда и сколько денек вкладывал в кружок (рис. 3.9).

Вклад в развитие кружка

1 для 6 Сохранить изменения

Учащийся	Дата	Вклад
Кочергов	3/1/2021	10.0000
Косоруков	3/9/2021	20.0000
Дубиновский	3/10/2021	30.0000
Алистратова	3/11/2021	20.0000
Кочергов	3/11/2021	122.0000
Мосякова	5/3/2021	158.0000

Рисунок 3.9 – Добавление или изменение вклада в кружок

При нажатии «Отчеты» на главной форме появляется окно для выбора выводимого отчета (рис. 3.10).

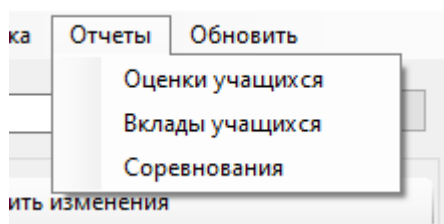


Рисунок 3.10 – Меню выбора отчетов

В зависимости от выбора того или иного отчета (рис. 3.11-3.13), на экран будет выведен соответствующий отчет по оценкам, вкладам или соревнованиям. Каждый отчет можно масштабировать, сохранять, печатать.

Оценки учащихся по темам									
Фамилия	Имя	Отчество	Chrome	Opera	Pascal 1	Pascal 2	Уроки рисования Photoshop 1	Уроки рисования Photoshop 2	Средний балл
Алистратова	Варвара	Владименовна			0		8		4
Бондарев	Архип	Всеволодович				8	4		6
Дубинский	Григорий	Андреевич		8			8		8
Косоруков	Агафон	Агапович					7		7
Кочергов	Филипп	Тарасович			0		8	10	6
Мослякова	Арина	Владименовна					7	7	7
Николюк	Максим	Ефремович			9		6	10	8,333333333333333
Пургина	Анфиса	Ивановна	6				9	8	7,666666666666667
Хромченко	Людмила	Василиевна			8		9	10	9

Рисунок 3.11 – Отчет по оценкам

Вклады учащихся в развитие кружка

Фамилия Кочергов
Имя Филипп
Отчество Тарасович

Вклады

Дата	Вклад
01.03.2021 0:00:00	10,0000
11.03.2021 0:00:00	122,0000

Фамилия Косоруков
Имя Агафон
Отчество Агапович

Вклады

Дата	Вклад
09.03.2021 0:00:00	20,0000

Фамилия Дубиновский
Имя Григорий
Отчество Андреевич

Вклады

Дата	Вклад
10.03.2021 0:00:00	30,0000

Рисунок 3.12 – Отчет по вкладам

Отчет по соревнованиям

1 из 2 ?

100%

Найти | Следующий

Отчет о соревнованиях

Вид соревнования: Городской

Рисование Photoshop

Первы этап рисования

Проводится: 07.03.2021 09:00

По адресу: Школа 12

Участники

Фамилия	Имя	Отчество	Класс	Место
Алистратова	Варвара	Владиленовна	7	1

Вид соревнования: Городской

Pascal word

Мир программирования

Проводится: 08.03.2021 10:00

По адресу: Оперный театр

Участники

Фамилия	Имя	Отчество	Класс	Место
Николюк	Максим	Ефремович	7	18

Вид соревнования: Городской

Internet review

Знание интернета

Проводится: 09.03.2021 13:45

Рисунок 3.13 – Отчет по соревнованиям

3.5 Тестирование программного продукта

Тестирование программного обеспечения – это процесс проверки соответствия заявленных к продукту требований и реально реализованной функциональности, осуществляемый путем наблюдения за его работой в искусственно созданных ситуациях и на ограниченном наборе тестов, выбранных определенным образом [9].

Целью тестирования является обнаружение дефектов, проверка соответствия программы заявленным требованиям, а также предоставление обратной связи разработчикам, менеджерам и другим заинтересованным лицам. Чтобы сократить время тестирования и сделать его наиболее эффективным перед проведением тестирования специальным образом создаются сценарии тестирования приложения. Все виды тестирования

программного обеспечения, в зависимости от преследуемых целей, можно разделить на группы:

Таблица 3.1 – Тестирование системы

4 Экономическая часть

4.1 Обоснование необходимости разработки программного обеспечения

Целью дипломного проектирования является разработка информационной системы «Архив учащихся кружка по программированию».

Программный продукт автоматизирует набор действий, которые ежедневно выполняет преподаватель кружка на рабочем месте. Большое количество наименований тем, различные ФИО учащихся и оценок за рабочую смену могут быть причиной частых ошибок сотрудников при работе с данными. Автоматизированная система позволит облегчить работу средствами автоматизации. Программа хранит большой объем необходимой информации, защищена от случайных ошибок, а также является удобным инструментом для разработки отчетной документации.

4.2 Этапы разработки программного обеспечения

Программное обеспечение, в сравнении с иными промышленными изделиями, имеет жизненный цикл – период от начала разработки нового программного средства до снятия его с эксплуатации потребителем. Сам цикл делится на три основных этапа – разработка, производство и использование. Все стадии делятся на фазы или этапы [11].

Стадия разработки программного обеспечения делится на четыре этапа: проектирование, реализация, тестирование системы и доработка.

Суммарное время разработки эквивалентно 12 неделям (3 месяца).

К этапу «Проектирование» относится анализ технического задания на разработку и проектирование. На его проведение отведено 3 недели (25% общего времени).

К этапу «Реализация» относятся непосредственно разработка программного продукта. На его проведение отведено 6 недель (50% общего времени).

На этапе «Тестирование и испытание системы» проводится тестирование разработанного программного обеспечения и доработка программного обеспечения. На его проведение отведено 1 недели (8% общего времени).

На этапе «Доработка» проводится заключительные приемо-сдаточные испытания и оформление заключительных глав пояснительной записки. На его проведение отведено 2 недели (17% общего времени).

Диаграмма, отображающая стадии проектирования программного обеспечения, приведена на рисунке 4.1.

Помимо деления на этапы, стадии и фазы жизненного цикла программного обеспечения, большое внимание уделяется составу работ, которые организуются в процессе создания программного продукта. В основном выделяется семь основных видов работ:

- проектирование изделий. Этап подразумевает определения, спецификации, анализ и модификации программной архитектуры, проекта программы;
- программирование. На данном этапе определяется детальное проектирование, кодирование, автономная отладка и комплексирование отдельных частей программы, а также планирование работ программистов, разработка базы данных, документирование отдельных компонентов и организация программирования на уровне компонентов;
- планирование отладки. На данном этапе формируются критерии выделения маршрутов для отладки и выбор стратегий упорядочения выделенных маршрутов;
- верификация. Представляет собой процесс проверки корректности требований, отладки изделий и приёмные испытания;
- управление проектом. Включает планирование и контроль проекта, контроль и регулирование договоров, связь с пользователями;
- контроль качества. Включает разработку и контроль стандартов, а также технические проверки программных средств и процессов разработки;
- документирование. Это последний вид работы, включает разработку и корректировку руководства для пользователей и операторов.



Рисунок 4.1 – Этапы разработки программного обеспечения

Все работы, относящиеся к делению на стадии, этапы и фазы жизненного цикла, а также выполняемые работы в процессе разработки, производства и эксплуатации программного продукта, требуют детального определения затрат, требуемых на эти цели.

4.3 Расчет стоимости программного средства

При расчете стоимости разработки учитываются следующие статьи затрат:

- статья «Материалы и комплектующие» (М) отражает сумму затрат, связанных с расходом материалов на производство результатной информации (бумага, носители памяти и т.д.);

- статья «Электроэнергия» (Ξ) включает затраты на электроэнергию;
- к статье «Основная заработная плата исполнителей» ($З_о$) относят затраты на заработную плату работников, непосредственно участвующих в разработке;
- статья «Дополнительная заработная плата исполнителей» ($З_д$) включает оплату отпусков, времени выполнения государственных обязанностей и т.д. Принимается в размере 15% от основной заработной платы (по данным организации);
- статья «Отчисления на социальные нужды» ($З_{сз}$) включает в себя отчисления в фонд социальной защиты населения ($Н_{сз} - 35\%$) и отчисления на обязательное страхование от несчастных случаев ($Н_{стр} - 0,1-0,6\%$);
- по статье «Амортизация» (A) рассчитываются амортизационные отчисления, исходя из стоимости основных средств, используемых в процессе разработки программного обеспечения, сроков эксплуатации оборудования и годовой нормы амортизации;
- статья «Расходы на спецоборудование» (P_c) включает расходы на специальное оборудование;
- статья «Накладные расходы» (P_n) включает в себя затраты, связанные с необходимостью содержания аппарата управления и расходами на общехозяйственные нужды;
- статья «Прочие прямые расходы» ($П_з$) на конкретное программное обеспечение включает затраты на приобретение и подготовку научно-технической информации и специальной литературы.

Затраты по статье «Материалы и комплектующие» учитывают расходы на разработку программного обеспечения и представлены в таблице 4.1.

Таблица 4.1 – Расходы на разработку программного обеспечения

Материал	Цена, рублей	Количество, штук	Стоимость, рублей
Бумага формата А4	0,02	100	2,00
Бумага формата А1	0,60	5	3,00
Флеш-носитель (2 Гб)	8,00	1	8,00
Компакт-диск CD-RW	2,00	1	2,00
Всего:			15,00

Таким образом, стоимость эксплуатационных материалов (M) составляет 15,00 рублей.

Затраты на электроэнергию определяются по формуле:

$$\Xi = K_{\Xi} \times T_p \times t_{пс}, \quad (4.1),$$

где K_{Ξ} – стоимость 1 кВт/ч. Базовый тариф для бюджетных организаций с 01.01.2020 г. составляет 0,26871 руб. за 1 кВт/ч (согласно Приказа

Департамента ценовой политики Министерства экономики Республики Беларусь от 30.09.2016 №95).;

T_p – расход электроэнергии за один месяц (100 кВт/ч);

$t_{пс}$ – период разработки программы, месяцев. Определяется в соответствии с длительностью дипломного проектирования и составляет 3 месяца. Тогда затраты на электроэнергию составляют:

$$\mathcal{E} = 0,26871 \times 100 \times 3 = 80,61 \text{ (рублей)}$$

Расчет по статье «Основная заработная плата» осуществляется по формуле:

$$\mathcal{Z}_o = T_m \times K \times n \times t_{пс}, \quad (4.2)$$

где T_m – месячная тарифная ставка;

K – коэффициент премирования;

n – количество исполнителей, занятых разработкой конкретного ПО;

$t_{пс}$ – период разработки программы, месяцев.

Предположим, что специалист, занимающийся разработкой программного обеспечения, имеет 12 тарифный разряд. Поправочный коэффициент для 12 разряда равен 1,47. Тарифный коэффициент единой тарифной сетки для 12 разряда равен 2,84. От 01.09.2019 тарифная ставка первого разряда равна 41 рубль. Повышающий коэффициент равен 2,5. Таким образом, месячная тарифная ставка специалиста T_m составляет 427,92 рублей в месяц.

Премия составляет 20% от средней заработной платы.

Таким образом, основная заработная плата для одного исполнителя за 3 месяца составляет:

$$\mathcal{Z}_o = 427,92 \times 3 \times 1,2 = 1540,51 \text{ (рублей)}.$$

Дополнительная заработная плата зависит от основной и её значение вычисляется по формуле:

$$\mathcal{Z}_d = \mathcal{Z}_o \times \frac{H_d}{100\%}, \quad (4.3)$$

где \mathcal{Z}_d – дополнительная заработная плата;

\mathcal{Z}_o – основная заработная плата;

H_d – норматив дополнительной заработной платы на предприятии, выраженный в процентах (принимается равным 10%).

$$\mathcal{Z}_d = 1540,51 \times 0,1 = 154,05 \text{ (рублей)}.$$

Затраты по статье «Отчисления на социальные нужды» определяются в процентах от затрат на заработную плату. Сумма отчислений на социальные нужды рассчитывается по формуле:

$$\mathcal{Z}_{сз} = (\mathcal{Z}_o + \mathcal{Z}_d) \times \frac{H_{сз}}{100\%}, \quad (4.4)$$

где $\mathcal{Z}_{сз}$ – затраты на отчисления в Фонд социальной защиты населения и на обязательное страхование;

\mathcal{Z}_o – основная заработная плата;

\mathcal{Z}_d – дополнительная заработная плата;

						Лист
						33
Изм.	Лист	№ докум.	Подпись	Дата		

$H_{сз}$ – норматив отчислений в Фонд социальной защиты населения и отчислений на обязательное страхование, в процентах.

На данный момент согласно норматив отчислений в Фонд социальной защиты населения законодательству составляет 35%, а норматив отчислений на обязательное страхование от несчастных случаев примем в размере 0,3%. Следовательно, сумма отчислений составит:

$$З_{сз} = (1540,51 + 154,05) \times (0,35 + 0,3) = 1101,46 \text{ (рублей).}$$

По статье «Амортизация оборудования» рассчитываются амортизационные отчисления исходя из стоимости основных фондов, используемых в процессе разработки программного обеспечения, годовой нормы амортизации, сроков эксплуатации оборудования.

В процессе написания дипломной работы и разработки автоматизированной информационной системы складского учета интернет-магазина использовался средний по мощности персональный компьютер стоимостью 600 руб.

В таблице 4.2 приведен перечень со стоимостью использованного программного обеспечения.

Таблица 4.2 – Стоимость программного обеспечения

Наименование нематериальных средств	Первоначальная стоимость, рублей
Windows 7 Home Basic	200
Microsoft Office 2016	80
Microsoft Visual Studio 2019 Professional	90
Microsoft SQL Server 2012 Express	80
Итого	450

Годовая норма амортизации равна 20% для основных и нематериальных фондов. Действительное время работы оборудования равно трем месяцам. Амортизационные отчисления для основных и нематериальных фондов рассчитаны в таблице 4.3.

Таблица 4.3 – Расходы на амортизацию

Оборудование	Первоначальная стоимость, рублей	Норма амортизации, %	Годовая сумма амортизации, рублей	Сумма амортизации на все время работы, рублей
Основные материалы	600	20	120,0	30,0
Дополнительные материалы	450	20	90,0	22,5
Всего:				52,5

Накладные расходы составляют 10% и рассчитываются по формуле:

$$P_n = Z_o \times \frac{H_p}{100\%}, \quad (4.5)$$

где P_n – накладные расходы;

Z_o – основная заработная плата исполнителей;

H_p – норматив накладных расходов.

$$P_n = 1540,51 \times \frac{10}{100} = 154,05 \text{ (рублей)}.$$

На основании полученных данных по отдельным статьям затрат рассчитана плановая себестоимость программного обеспечения (таблица 4.4).

Таблица 4.4 – Расчет плановой себестоимости программного обеспечения

Статья затрат	Сумма затрат, рублей
Материалы и комплектующие	15,00
Электроэнергия	80,61
Основная заработная плата	1540,51
Дополнительная заработная плата	154,05
Отчисления на социальные нужды	1101,46
Амортизация	52,50
Накладные расходы	154,05
Общая сумма расходов по смете (плановая себестоимость)	3098,18

Таблицу 4.4 удобно представить в виде диаграммы, отображающей влияние статей затрат на разработку программного обеспечения на окончательную себестоимость программы. Данная диаграмма представлена на рисунке 4.2.

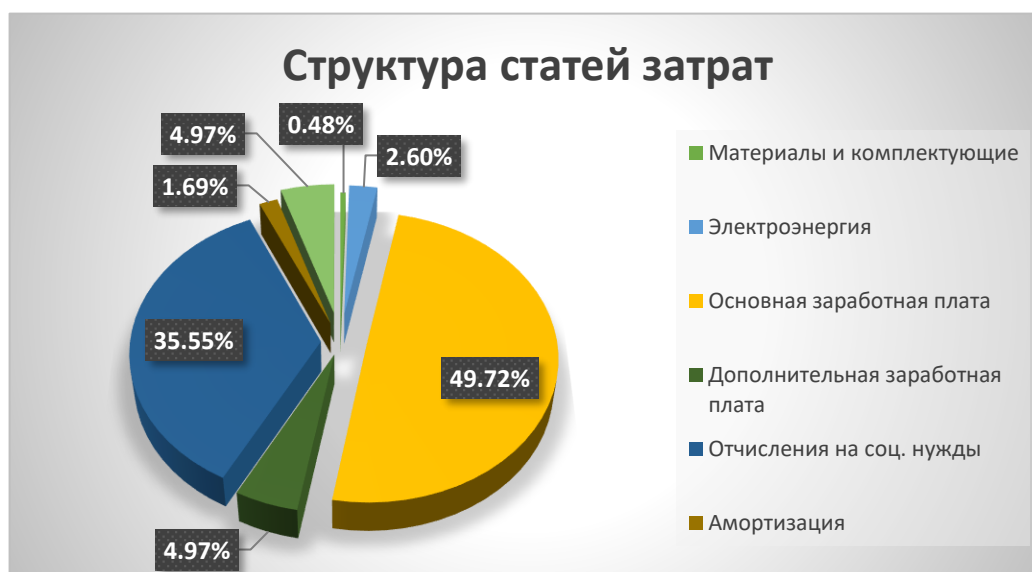


Рисунок 4.2 – Диаграмма статей затрат

Таким образом, результатом данного раздела дипломного проекта является полный расчет количества денежных средств, затраченных на разработку программного обеспечения. Получаем, что полная (плановая) себестоимость равна сумме вышеперечисленных расходов, то есть равна 3098,18 рублей.

4.4 Расчёт отпускной цены программного средства

Отпускная цена продукции формируется из плановой себестоимости, всех видов установленных налогов и прибыли, качества, потребительских свойств и конъюнктуры рынка. Отпускная цена продукции рассчитывается по формуле:

$$ОЦ = С + П \quad (4.6)$$

где $ОЦ$ – отпускная цена; $С$ – плановая себестоимость; $П$ – прибыль.

Прибыль рассчитывается как произведение себестоимости и уровня рентабельности. Примем коэффициент рентабельности 20%. Следовательно, прибыль составит:

$$П = 3098,18 \times 0,2 = 619,64 \text{ (рублей).}$$

Отпускная цена продукции составит:

$$ОЦ = 3098,18 + 619,64 = 3717,82 \text{ (рублей).}$$

Стоимость проекта с учетом налога на добавленную стоимость представляет собой сумму отпускной цены и НДС. НДС рассчитывается по формуле:

$$НДС = \frac{(С + П) \times \text{СтавкаНДС}\%}{100\%}, \quad (4.7)$$

где НДС – величина налога на добавленную стоимость;

Ставка НДС % – ставка налога на добавленную стоимость в процентах.

Ставка НДС установлена в размере 20%. Величина налога на добавленную стоимость для программного средства составляет:

$$НДС = \frac{3717,82 \times 20\%}{100\%} = 743,56 \text{ (рублей).}$$

Таким образом, цена программного средства с учетом НДС рассчитывается как сумма отпускной цены и величины налога на добавленную стоимость. Для разработанного приложения цена с учетом НДС составляет:

$$ОЦ_{НДС} = 3717,82 + 743,56 = 4461,38 \text{ (рублей).}$$

Величина $ОЦ_{НДС}$, равная 4461,38 рублям, покрывает затраты и обеспечивает прибыль за разработку проекта.

5 Охрана труда

5.1 Идентификация и анализ вредных и опасных факторов в проектируемом объекте

Вредные факторы, воздействующие на человека при работе с компьютерной техникой, классифицируемых в соответствии с государственным стандартом ГОСТ 12.0.003-74 ССБТ. "Опасные и вредные производственные факторы. Классификация". К опасным и вредным производственным факторам относятся:

- повышенные уровни электромагнитного излучения. Источниками излучения могут быть системный блок, беспроводные системы передачи информации на расстояния, клавиатура, мышь. Действие электромагнитных полей на организм человека в основном определяется поглощенной в нем энергией. Эта энергия преобразуется в тепло, в результате чего наступает опасный перегрев. Кроме того, электромагнитные поля вызывают нарушения в работе сердечно-сосудистой и нервной систем.

- повышенные уровни ультрафиолетового излучения. Источником излучения является монитор. Повышенные дозы ультрафиолетового излучения могут вызывать головную боль, головокружение, повышение температуры тела, повышение утомляемости и др.

- повышенные уровни инфракрасного излучения. Инфракрасное излучение негативно влияет на зрительные органы человека. Может вызвать нарушения в работе сетчатки, хрусталика, радужной оболочки глаза. Инфракрасное излучение коротковолнового диапазона может воздействовать на внутренние органы, мозговую ткань, а при воздействии на глаза может вызвать появление инфракрасной катаракты.

- повышенные уровни статического электричества. При воздействии повышенного уровня статического электричества возможно появление головной боли, нарушение сна, снижение аппетита и др. Вредное воздействие на организм человека статическое электричество оказывает не только при непосредственном его контакте с зарядом, но и за счет действия электрического поля, возникающего вокруг заряженных поверхностей. В вычислительных центрах разрядные токи статического электричества чаще всего возникают при прикосновении к любому из элементов компьютера. Такие разряды могут привести к выходу из строя техники.

- пониженная или повышенная влажность воздуха рабочей зоны. При относительной влажности более 75-80 % снижается сопротивление изоляции, изменяются рабочие характеристики, и повышается интенсивность отказов элементов ПК. Повышенная влажность воздуха приводит к нарушению терморегуляции, уменьшается испарение пота, что приводит к резкому ухудшению состояния здоровья человека и снижению его работоспособности. Понижение влажности воздуха вызывает неприятные ощущения сухости слизистых оболочек дыхательных путей.

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		37

- пониженная или повышенная подвижность воздуха рабочей зоны. Скорость движения воздуха оказывает влияние на распределение вредных частиц в помещении и влияет на отдачу теплоты организмом. Повышенная скорость движения воздуха, особенно в условиях низких температур, вызывает увеличение теплопотерь и ведет к сильному охлаждению организма.

- пониженная или повышенная температура воздуха рабочей зоны. Повышенная температура приводит к быстрой утомляемости, повышению артериального давления, ослаблению внимания и др. При пониженной температуре организм человека переохлаждается, что ведет к простудным заболеваниям.

- повышенный уровень шума. Среди многочисленных проявлений воздействия шума на организм можно выделить снижение разборчивости речи, неприятные ощущения, развитие утомления, снижение производительности труда и возникновение шумовой патологии. При действии интенсивного шума изменения со стороны нервной системы значительно более выражены и предшествуют развитию патологии органа слуха. У рабочих преобладают жалобы на головные боли, несистематические головокружения, снижение памяти, повышенную утомляемость, эмоциональную неустойчивость, нарушение сна, сердцебиение и боли в области сердца, снижение аппетита и др. Источниками шума на рабочем месте являются: вентиляторы систем охлаждения, накопители, монитор, клавиатура, принтер, преобразователи напряжения.

- повышенный или пониженный уровень освещенности. Недостаточное и чрезмерное освещение рабочего места оказывает отрицательное воздействие на зрение человека, вызывает эмоциональное напряжение, что ведет к снижению производительности труда.

- повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека. Поражение электрическим током опасно для здоровья и жизни человека и обусловлено тем, что проходящий ток не видим человеком и зачастую не воспринимается как источник опасности. Степень поражения человека электрическим током зависит от схемы включения человека в сеть, степени изоляции токоведущих частей, напряжения в сети, схемы сети, индивидуальных свойств человека. Проходя через организм, электрический ток оказывает термическое, электролитическое и биологическое воздействие. Это многообразие действий электрического тока приводит к различным травмам от легкого пощипывания до остановки сердца.

- статические перегрузки костно-мышечного аппарата и динамические локальные перегрузки мышц кистей рук. Основной причиной перенапряжения мышц спины и ног являются нерациональная высота рабочей поверхности стола и сидения, отсутствие опорной спинки и подлокотников, неудобное размещение монитора и клавиатуры, отсутствие подставки для ног. Длительная монотонная работа с клавиатурой может привести к развитию воспалительных процессов тканей сухожилий.

- умственное перенапряжение и эмоциональные перегрузки. Работа за компьютером требует большой концентрации и внимания. Последствиями воздействия такого рода нагрузок приводят к быстрому переутомлению, головным болям, плохому сну и снижению работоспособности.

- монотонность труда. Может вызвать быстрое развитие усталости в связи с локализацией мышечных и нервных нагрузок, гиподинамию, развитие неврозов, недовольство работой и снижение творческой активности.

- пожароопасность. Источниками возгорания могут быть электронные схемы, приборы, применяемые для технического обслуживания, устройства электропитания, кондиционирования воздуха, где в результате различных нарушений образуются перегретые элементы, электрические искры и дуги. Для отвода избыточной теплоты служат системы вентиляции и кондиционирования воздуха. Но при постоянном действии эти системы представляют собой дополнительную пожарную опасность. Опасными для человека факторами пожара являются: огонь (ожоги кожи), токсичные продукты горения, повышенная температура среды (вызывают потерю сознания, поражение и некроз верхних дыхательных путей), дым (вызывает кашель, отек легких), пониженная концентрация кислорода (вызывает ухудшение двигательной функции организма), взрывы, вытекание опасных веществ (вызывает ранения обломками, химические ожоги и отравления), разрушения строительных конструкций, паника.

5.2 Технические, технологические, организационные решения по устранению опасных и вредных факторов, разработка защитных средств.

Характеристики используемого компьютера:

- 1) процессор AMD Athlon X4 840 3.10 GHz;
- 2) Материнская плата ASUS M48-EH AMD 230A;
- 3) Модуль оперативной памяти Hynix DDR2 800 DIMM 1Gb;
- 4) Жесткий диск WD Caviar Blue 1TB (WD10EZEX);
- 5) Жидкокристаллический монитор Philips 19S4QAB/00

Параметры факторов производственной среды на рабочих местах с использованием ЭВМ регламентируются в СанПиН, утвержденных Постановлением Минздрава РБ от 28.06.2013 № 59 «Требования при работе с видеодисплейными терминалами и электронно-вычислительными машинами» и гигиеническом нормативе «Предельно допустимые уровни нормируемых параметров при работе с видеодисплейными терминалами (ВДТ) и электронно-вычислительными машинами» [12].

Для работы используется жидкокристаллический монитор, что обеспечивает защиту от инфракрасного, ультрафиолетового, ионизирующего и электромагнитного излучения. Используемые мониторы соответствуют СанПиН, т.к. их напряженность электромагнитного поля по электрической составляющей не превышает 25 В/м. Интенсивность инфракрасного и видимого излучения от экрана монитора не превышает 0,1 Вт/м² в видимом (400 – 760 нм) диапазоне. Для защиты от электромагнитного излучения устанавливается рациональное время работы оборудования и работников

						Лист
						39
Изм.	Лист	№ докум.	Подпись	Дата		

С целью снижения уровня статического электричества пол в помещении покрыт антистатическим материалом.

Показатели микроклимата на рабочем месте соответствуют требованиям СанПиН, утвержденным Постановлением минздрава РБ от 28.06.2013 №59 «Требования при работе с видеодисплейными терминалами и электронно-вычислительными машинами». Для обеспечения нормативных параметров микроклимата используются как организационные методы (рациональная организация проведения работ в зависимости от времени года и суток, чередование труда и отдыха), так и технические средства (вентиляция, кондиционирование воздуха, отопительная система).

В помещении используется совмещенное освещение. Искусственное освещение выполняется в виде общего освещения люминесцентными лампами. Светильники располагаются над рабочей поверхностью в равномерно-прямоугольном порядке. Такой вариант освещения обеспечивает необходимый уровень освещенности (300лк), не оказывает слепящего действия на оператора. Для предотвращения образования бликов на экране, применяются светильники общего освещения, при этом линии светильников располагаются параллельно светопроемам.

Электробезопасность в проектируемом объекте обеспечивается в соответствии с ГОСТ 12.1.019-2017 ССБТ. «Электробезопасность. Общие требования и номенклатура видов защиты». В качестве основного способа защиты используется защитное заземление.

Для того чтобы снизить риск переутомления, истощения нервно-психических ресурсов организма, нужно соблюдать регламентированные перерывы в работе.

Для обнаружения начальной стадии возгорания и оповещения службы пожарной охраны используется система автоматической пожарной сигнализации. В качестве основного средства пожаротушения используются порошковые огнетушители ОП-4, которые позволяют тушить электроустановки.

5.3 Разработка мер безопасности при эксплуатации объекта проектирования.

Для безопасной работы с ПЭВМ разработана "Инструкция по охране труда для оператора ПЭВМ".

Глава 1

ОБЩИЕ ТРЕБОВАНИЯ ПО ОХРАНЕ ТРУДА

1. Настоящей Типовой инструкцией устанавливаются требования по охране труда при работе с персональными электронно-вычислительными машинами (далее – ПЭВМ) [13].

						Лист
						40
Изм.	Лист	№ докум.	Подпись	Дата		

ПЭВМ – техническое средство, предназначенное для индивидуального использования с целью автоматической обработки информации в процессе решения вычислительных и информационных задач, средство коммуникации с помощью телекоммуникационных сетей.

2. К выполнению работ с ПЭВМ допускаются работающие, не имеющие медицинских противопоказаний, прошедшие в установленном законодательством порядке медосмотр, инструктаж по охране труда (далее – работающие).

3. Женщинам со дня установления беременности и в период кормления ребенка грудью следует ограничить время работы с ПЭВМ до 3 часов за рабочую смену с учетом обеспечения оптимальных условий труда, и регламентированных перерывов в соответствии с законодательством Республики Беларусь.

При невозможности организации работ в соответствии с требованиями части первой настоящего пункта по причинам, связанным с особенностями технологического процесса, женщины со времени установления беременности и в период кормления ребенка грудью должны быть переведены на работы, не связанные с использованием ПЭВМ.

4. Работающие обязаны:

соблюдать режим труда и отдыха, установленный законодательством, правилами внутреннего трудового распорядка организации, трудовую дисциплину, выполнять требования по охране труда, правила личной гигиены; выполнять требования пожарной безопасности;

курить только в специально предназначенных для курения местах;

заботиться о личной безопасности и личном здоровье, а также о безопасности окружающих в процессе выполнения работ либо во время нахождения на территории организации;

содержать рабочее место в надлежащем санитарном состоянии;

знать местонахождение аптечки первой медицинской помощи универсальной и уметь оказывать необходимую помощь потерпевшим при несчастном случае на производстве;

сообщать непосредственному руководителю или иному уполномоченному должностному лицу нанимателя о неисправности ПЭВМ и периферийных устройств (принтера, сканера, клавиатуры ПЭВМ, электрических компьютерных сетевых устройств, блока бесперебойного питания и других устройств) (далее – оборудование) и иных замечаниях, препятствующих выполнению работы, и не приступать к работе до их устранения;

немедленно сообщать непосредственному руководителю или иному уполномоченному должностному лицу нанимателя о любой ситуации, угрожающей жизни или здоровью работающих и окружающих.

5. Работающие имеют право отказаться от выполнения порученной работы в случае возникновения непосредственной опасности для жизни и здоровья их и окружающих до устранения этой опасности.

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		41

6. Не допускается нахождение работающих в состоянии алкогольного, наркотического или токсического опьянения, а также распитие спиртных напитков, потребление наркотических средств, психотропных веществ, их аналогов, токсических средств на рабочем месте и в рабочее время.

7. За невыполнение требований настоящей Типовой инструкции, работающие несут ответственность в соответствии с законодательными актами.

Глава 2

ТРЕБОВАНИЯ ПО ОХРАНЕ ТРУДА ПЕРЕД НАЧАЛОМ РАБОТЫ

8. Перед началом работы с ПЭВМ работающий обязан:
осмотреть и привести в порядок рабочее место:

отрегулировать освещенность на рабочем месте, убедиться в достаточности освещенности, отсутствии отражений на экране, отсутствии встречного светового потока.

проверить правильность подключения оборудования в электросеть;

проверить правильность установки стола, стула, подставки для ног, положения оборудования, угла наклона экрана, положение клавиатуры и, при необходимости, произвести регулировку рабочего стола и кресла, а также расположение элементов компьютера в соответствии с требованиями эргономики и в целях исключения неудобных поз и длительных напряжений тела.

9. При включении компьютера необходимо соблюдать следующую последовательность включения оборудования:

9.1. Включить блок бесперебойного питания (если имеется).

9.2. Включить периферийные устройства (принтер, монитор, сканер и др.).

9.3. Включить системный блок.

10. Работающему запрещается:

устанавливать системный блок в закрытых объемах мебели, непосредственно на полу;

использовать для подключения оборудования розетки, удлинители, не оснащенные заземляющим контактом (шиной);

включать охлажденное (принесенное с улицы в зимнее время) оборудование;

располагать экраны видеомониторов навстречу друг другу при рядном размещении рабочих столов в целях исключения их взаимного отражения;

приступать к работе с ПЭВМ при мелькании изображения на экране видеомонитора, в случае обнаружения неисправности оборудования, кабелей или проводов, разъемов, штепсельных соединений, при отсутствии или неисправности защитного заземления (зануления) оборудования.

Глава 3

						Лист
						42
Изм.	Лист	№ докум.	Подпись	Дата		

ТРЕБОВАНИЯ ПО ОХРАНЕ ТРУДА ПРИ ВЫПОЛНЕНИИ РАБОТЫ

11. При работе необходимо:

11.1. В течение всего рабочего дня содержать в порядке и чистоте рабочее место.

11.2. Держать открытыми все вентиляционные отверстия устройств.

11.3. При необходимости прекращения работы на некоторое время корректно закрыть все активные задачи.

11.4. Выполнять санитарные нормы и соблюдать режимы работы и отдыха.

11.5. Соблюдать правила эксплуатации вычислительной техники в соответствии с инструкциями по эксплуатации.

11.6. Соблюдать установленные режимом рабочего времени регламентированные перерывы в работе и выполнять в физкультпаузах и физкультминутках рекомендованные упражнения для глаз, шеи, рук, туловища, ног.

11.7. Соблюдать расстояние от глаз до экрана в пределах 60 - 80 см.

12. Во время работы запрещается:

12.1. Прикасаться к задней панели системного блока при включенном питании.

12.2. Переключать разъемы интерфейсных кабелей периферийных устройств при включенном питании.

12.3. Загромождать верхние панели устройств бумагами и посторонними предметами.

12.4. Допускать захламленность рабочего места бумагой в целях недопущения накопления органической пыли.

12.5. Производить отключение питания во время выполнения активной задачи.

12.6. Производить частые переключения питания.

12.7. Допускать попадание влаги на поверхность системного блока, монитора, рабочую поверхность клавиатуры, дисководов, принтеров и др. устройств.

12.8. Включать сильно охлажденное (принесенное с улицы в зимнее время) оборудование.

12.9. Производить самостоятельно вскрытие и ремонт оборудования.

Глава 4

ТРЕБОВАНИЯ ПО ОХРАНЕ ТРУДА ПО ОКОНЧАНИИ РАБОТЫ

13. По окончании работ необходимо соблюдать следующую последовательность выключения вычислительной техники:

13.1. Произвести закрытие всех активных задач.

13.2. Выключить питание системного блока.

13.3. Выключить питание всех периферийных устройств.

13.4. Отключить блок бесперебойного питания (при наличии).

						Лист
						43
Изм.	Лист	№ докум.	Подпись	Дата		

14. По окончании работ необходимо осмотреть и привести в порядок рабочее место.

15. Обо всех недостатках, возникших во время выполнения работ сообщить руководителю.

Глава 5

ТРЕБОВАНИЯ ПО ОХРАНЕ ТРУДА В АВАРИЙНЫХ СИТУАЦИЯХ

16. Сотрудник обязан немедленно сообщить руководителю подразделения о происшедшем с ним или по его вине несчастном случае, а также о любом несчастном случае с участием других работников, свидетелем которого он был.

17. Сотрудник должен оказать пострадавшему при несчастном случае первую доврачебную помощь, помочь доставить его в здравпункт или ближайшее медицинское учреждение или, при необходимости, вызвать медицинских работников на место происшествия.

18. Во всех случаях обнаружения обрыва проводов питания, неисправности заземления и других повреждений электрооборудования, появления запаха гари немедленно отключить питание и сообщить об аварийной ситуации руководителю.

19. При обнаружении человека, попавшего под напряжение, немедленно освободить его от действия тока путем отключения электропитания и до прибытия врача оказать потерпевшему первую медицинскую помощь;

20. При любых случаях сбоя в работе технического оборудования или программного обеспечения немедленно вызвать представителя соответствующих служб.

21. В случае появления рези в глазах, резком ухудшении видимости - невозможности сфокусировать взгляд или навести его на резкость, появлении боли в пальцах и кистях рук, усилении сердцебиения немедленно покинуть рабочее место, сообщить о происшедшем руководителю и обратиться к врачу.

22. При возгорании оборудования, отключить питание и принять меры к тушению очага пожара при помощи углекислотного огнетушителя, вызвать пожарную команду и сообщить о происшествии руководителю.

Заключение

В ходе выполнения данного дипломного проекта была спроектирована и разработана информационная система «Архив учащихся кружка по программированию»

Разработанное программное средство представляет собой систему, предназначенную для облегчения работы преподавателя по информатики в выставление оценок, денежных вкладов и достижений.

В ходе разработки программного средства был выполнен анализ предметной области и выявлены функциональные и нефункциональные требования, на основании чего был сформирован документ «Техническое задание». На этапе проектирования была разработана диаграмма вариантов использования. Была выполнена реализация функциональных частей программы, пользовательского интерфейса и тестирование реализованного приложения.

Данная программа полностью соответствует требованиям технического задания. Программный продукт может эффективно функционировать на персональном компьютере.

Результаты выполнения дипломного проектирования представлены в виде программы на машинном носителе (оригинал, дубликаты и копии в необходимом количестве), эксплуатационной программной документации в целом и рабочей документации информационного обеспечения.

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		45

Список использованных источников

- 1 Общие сведения о базах данных. История развития систем управления базами данных [Электронный ресурс]. – Режим доступа: https://www.bestprog.net/ru/2017/09/14/general-information-about-databases-history-of-the-development-of-database-management-systems_ru/. – Дата доступа: 16.03.2021.
- 2 Разработка технического задания. [Электронный ресурс]. – Режим доступа: <https://intuit.ru/studies/courses/2195/55/lecture/15047>. – Дата доступа: 16.03.2021.
- 3 Элементы графической нотации диаграммы вариантов использования [Электронный ресурс]. – Режим доступа: <https://intuit.ru/studies/courses/32/32/lecture/1004>. – Дата доступа: 16.03.2021.
- 4 Проектирование Логической модели базы данных [Электронный ресурс]. – Режим доступа: https://studbooks.net/2195440/informatika/proektirovanie_logicheskoy_modeli_bazy_dannyh. – Дата доступа: 16.03.2021.
- 5 СУБД Microsoft Access [Электронный ресурс]. – Режим доступа: https://studme.org/54436/informatika/subd_microsoft_access. – Дата доступа: 16.03.2021.
- 6 MySQL — система управления базами данных [Электронный ресурс]. – Режим доступа: <https://web-creator.ru/articles/mysql>. – Дата доступа: 16.03.2021.
- 7 Язык C# и платформа .NET Core [Электронный ресурс]. – Режим доступа: <https://metanit.com/sharp/tutorial/1.1.php>. – Дата доступа: 16.03.2021.
- 8 Диаграммы классов UML. Логическое моделирование [Электронный ресурс]. – Режим доступа: <https://www.informicus.ru/default.aspx?SECTION=6&id=73&subdivisionid=3>. – Дата доступа: 16.03.2021.
- 9 Тестирование программного обеспечения [Электронный ресурс]. – Режим доступа: <https://present5.com/testirovanie-programmnogo-obespecheniya-test-dizajn-2008-v-2-6/>. – Дата доступа: 16.03.2021.
- 10 Тестирование и обеспечение качества [Электронный ресурс]. – Режим доступа: <http://mellarius.ru/testing>. – Дата доступа: 16.03.2021.
- 11 Этапы жизненного цикла программного обеспечения [Электронный ресурс]. – Режим доступа: <https://infopedia.su/3xbae4.html>. – Дата доступа: 16.03.2021.
- 12 Постановление №59 от 28.06.2013 [Электронный ресурс]. – Режим доступа: https://kodeksy-by.com/norm_akt/source-Минздрав%20РБ/type-Постановление/59-28.06.2013.htm. – Дата доступа: 16.03.2021.
- 13 Об утверждении Типовой инструкции по охране труда при работе с персональными электронными вычислительными машинами [Электронный ресурс]. – Режим доступа: https://pravo.by/upload/docs/op/W21428532_1397077200.pdf. – Дата доступа: 16.03.2021.

ПРИЛОЖЕНИЕ А

(Обязательное)

Код создания базы данных

```
DROP DATABASE IF EXISTS
`ArchiveOfStudentsOfTheProgrammingCircle`;
CREATE DATABASE IF NOT EXISTS
`ArchiveOfStudentsOfTheProgrammingCircle`;
USE `ArchiveOfStudentsOfTheProgrammingCircle`;
```

```
DROP TABLE IF EXISTS `Учащийся`;
CREATE TABLE `Учащийся` (
  `Код` INTEGER NOT NULL AUTO_INCREMENT,
  `Фамилия` VARCHAR(30),
  `Имя` VARCHAR(30),
  `Отчество` VARCHAR(30),
  `Школа` VARCHAR(255),
  `Класс` VARCHAR(5),
  `Дата регистрации` DATETIME,
  PRIMARY KEY (`Код`)
) ENGINE=innodb DEFAULT CHARSET=utf8;
SET autocommit=1;
```

```
DROP TABLE IF EXISTS `Виды соревнований`;
CREATE TABLE `Виды соревнований` (
  `Название` VARCHAR(255) NOT NULL,
  PRIMARY KEY (`Название`)
) ENGINE=innodb DEFAULT CHARSET=utf8;
SET autocommit=1;
```

```
DROP TABLE IF EXISTS `Соревнование`;
CREATE TABLE `Соревнование` (
  `Код` INTEGER NOT NULL AUTO_INCREMENT,
  `Название` VARCHAR(255),
  `Вид` VARCHAR(255),
  `Описание` VARCHAR(255),
  `Дата` DATETIME,
  `Время` DATETIME,
  `Адрес` VARCHAR(255),
  PRIMARY KEY (`Код`),
  FOREIGN KEY (`Вид`) REFERENCES `Виды соревнований`
  (`Название`)
) ENGINE=innodb DEFAULT CHARSET=utf8;
SET autocommit=1;
```

```

DROP TABLE IF EXISTS `Участники соревнований`;
CREATE TABLE `Участники соревнований` (
  `Код` INTEGER NOT NULL AUTO_INCREMENT,
  `Соревнование` INTEGER,
  `Участник` INTEGER,
  `Место` INTEGER DEFAULT 0,
  PRIMARY KEY (`Код`),
  FOREIGN KEY (`Соревнование`) REFERENCES `Соревнование`
  (`Код`),
  FOREIGN KEY (`Участник`) REFERENCES `Учащийся` (`Код`)
) ENGINE=innodb DEFAULT CHARSET=utf8;
SET autocommit=1;

```

```

DROP TABLE IF EXISTS `Темы`;
CREATE TABLE `Темы` (
  `Код` INTEGER NOT NULL AUTO_INCREMENT,
  `Название` VARCHAR(60),
  `Дата` DATETIME,
  `Описание` VARCHAR(255),
  PRIMARY KEY (`Код`)
) ENGINE=innodb DEFAULT CHARSET=utf8;
SET autocommit=1;

```

```

DROP TABLE IF EXISTS `Журнал оценок`;
CREATE TABLE `Журнал оценок` (
  `Код` INTEGER NOT NULL AUTO_INCREMENT,
  `Учащийся` INTEGER,
  `Тема` INTEGER,
  `Оценка` INTEGER DEFAULT 0,
  `Присутствал` TINYINT(1) DEFAULT 1,
  PRIMARY KEY (`Код`),
  FOREIGN KEY (`Учащийся`) REFERENCES `Учащийся` (`Код`),
  FOREIGN KEY (`Тема`) REFERENCES `Темы` (`Код`)
) ENGINE=innodb DEFAULT CHARSET=utf8;
SET autocommit=1;

```

```

DROP TABLE IF EXISTS `Вклад в развитие`;
CREATE TABLE `Вклад в развитие` (
  `Код` INTEGER NOT NULL AUTO_INCREMENT,
  `Учащийся` INTEGER,
  `Дата` DATETIME,
  `Вклад` DECIMAL(19,4) DEFAULT 0,
  PRIMARY KEY (`Код`),
  FOREIGN KEY (`Учащийся`) REFERENCES `Учащийся` (`Код`)
) ENGINE=innodb DEFAULT CHARSET=utf8;
SET autocommit=1;

```

						Лист
						48
Изм.	Лист	№ докум.	Подпись	Дата		

```
) ENGINE=innodb DEFAULT CHARSET=utf8;
SET autocommit=1;
```

```
INSERT INTO `Учащийся` (`Код`, `Фамилия`, `Имя`, `Отчество`,
`Школа`, `Класс`, `Дата регистрации`) VALUES (1, 'Алистратова', 'Варвара',
'Владиленовна', '33', '7', '2004-07-21 00:00:00');
```

```
INSERT INTO `Учащийся` (`Код`, `Фамилия`, `Имя`, `Отчество`,
`Школа`, `Класс`, `Дата регистрации`) VALUES (2, 'Мосякова', 'Арина',
'Владиленовна', '44', '7', '2004-02-18 00:00:00');
```

```
INSERT INTO `Учащийся` (`Код`, `Фамилия`, `Имя`, `Отчество`,
`Школа`, `Класс`, `Дата регистрации`) VALUES (3, 'Николюк', 'Максим',
'Ефремович', '55', '7', '2004-03-13 00:00:00');
```

```
INSERT INTO `Учащийся` (`Код`, `Фамилия`, `Имя`, `Отчество`,
`Школа`, `Класс`, `Дата регистрации`) VALUES (4, 'Кочергов', 'Филипп',
'Тарасович', '33', '11', '2004-10-23 00:00:00');
```

```
INSERT INTO `Учащийся` (`Код`, `Фамилия`, `Имя`, `Отчество`,
`Школа`, `Класс`, `Дата регистрации`) VALUES (5, 'Косоруков', 'Агафон',
'Агапович', '25', '11', '2004-07-05 00:00:00');
```

```
INSERT INTO `Учащийся` (`Код`, `Фамилия`, `Имя`, `Отчество`,
`Школа`, `Класс`, `Дата регистрации`) VALUES (6, 'Хромченко', 'Людмила',
'Василиевна', '84', '8', '2004-12-23 00:00:00');
```

```
INSERT INTO `Учащийся` (`Код`, `Фамилия`, `Имя`, `Отчество`,
`Школа`, `Класс`, `Дата регистрации`) VALUES (7, 'Бондарев', 'Архип',
'Всеволоодович', '98', '8', '2004-03-18 00:00:00');
```

```
INSERT INTO `Учащийся` (`Код`, `Фамилия`, `Имя`, `Отчество`,
`Школа`, `Класс`, `Дата регистрации`) VALUES (8, 'Дубиновский', 'Григорий',
'Андреевич', '4', '9', '2004-10-29 00:00:00');
```

```
INSERT INTO `Учащийся` (`Код`, `Фамилия`, `Имя`, `Отчество`,
`Школа`, `Класс`, `Дата регистрации`) VALUES (9, 'Пургина', 'Анфиса',
'Ивановна', '6', '10', '2003-07-18 00:00:00');
```

9 records

#

Dumping data for table 'Виды соревнований'

#

```
INSERT INTO `Виды соревнований` (`Название`) VALUES
('Городской');
```

```
INSERT INTO `Виды соревнований` (`Название`) VALUES
('Областной');
```

```
INSERT INTO `Виды соревнований` (`Название`) VALUES
('Республиканский');
```

						Лист
						49
Изм.	Лист	№ докум.	Подпись	Дата		

3 records

#

Dumping data for table 'Соревнование'

#

```
INSERT INTO `Соревнование` (`Код`, `Название`, `Вид`, `Описание`,
`Дата`, `Время`, `Адрес`) VALUES (1, 'Рисование Photoshop', 'Городской',
'Первы этап рисования', '2021-03-07 00:00:00', '1899-12-30 09:00:00', 'Школа
12');
```

```
INSERT INTO `Соревнование` (`Код`, `Название`, `Вид`, `Описание`,
`Дата`, `Время`, `Адрес`) VALUES (2, 'Pascal word', 'Городской', 'Мир
программирования', '2021-03-08 00:00:00', '1899-12-30 10:00:00', 'Оперный
театр');
```

```
INSERT INTO `Соревнование` (`Код`, `Название`, `Вид`, `Описание`,
`Дата`, `Время`, `Адрес`) VALUES (3, 'Interner review', 'Городской', 'Знание
интернета', '2021-03-09 00:00:00', '1899-12-30 13:45:00', 'Гимназия 2');
```

```
INSERT INTO `Соревнование` (`Код`, `Название`, `Вид`, `Описание`,
`Дата`, `Время`, `Адрес`) VALUES (4, 'Рисование Photoshop', 'Областной',
'Второй зап рисования', '2021-03-10 00:00:00', '1899-12-30 08:00:00', 'Школа
12');
```

```
INSERT INTO `Соревнование` (`Код`, `Название`, `Вид`, `Описание`,
`Дата`, `Время`, `Адрес`) VALUES (5, 'Pascal word', 'Областной', 'Мир
программирования часть 2', '2021-03-11 00:00:00', '1899-12-30 11:00:00',
'Гимназия 2');
```

```
INSERT INTO `Соревнование` (`Код`, `Название`, `Вид`, `Описание`,
`Дата`, `Время`, `Адрес`) VALUES (6, 'Рисование Photoshop',
'Республиканский', 'Финальный этап рисования', '2021-03-12 00:00:00', '1899-
12-30 11:00:00', 'Школа 12');
```

6 records

#

Dumping data for table 'Участники соревнований'

#

```
INSERT INTO `Участники соревнований` (`Код`, `Соревнование`,
`Участник`, `Место`) VALUES (1, 1, 1, 1);
```

```
INSERT INTO `Участники соревнований` (`Код`, `Соревнование`,
`Участник`, `Место`) VALUES (2, 4, 1, 2);
```

```
INSERT INTO `Участники соревнований` (`Код`, `Соревнование`,
`Участник`, `Место`) VALUES (3, 6, 1, 1);
```

```
INSERT INTO `Участники соревнований` (`Код`, `Соревнование`,
`Участник`, `Место`) VALUES (4, 2, 3, 18);
```

```
INSERT INTO `Участники соревнований` (`Код`, `Соревнование`,
`Участник`, `Место`) VALUES (5, 5, 4, 3);
```

						Лист
						50
Изм.	Лист	№ докум.	Подпись	Дата		


```

INSERT INTO `Участники соревнований` (`Код`, `Соревнование`,
`Участник`, `Место`) VALUES (6, 5, 4, 13);
INSERT INTO `Участники соревнований` (`Код`, `Соревнование`,
`Участник`, `Место`) VALUES (7, 3, 5, 8);
INSERT INTO `Участники соревнований` (`Код`, `Соревнование`,
`Участник`, `Место`) VALUES (8, 3, 6, 6);
INSERT INTO `Участники соревнований` (`Код`, `Соревнование`,
`Участник`, `Место`) VALUES (9, 3, 7, 14);
INSERT INTO `Участники соревнований` (`Код`, `Соревнование`,
`Участник`, `Место`) VALUES (10, 3, 8, 22);
# 10 records

```

```

#
# Dumping data for table `Темы`
#

```

```

INSERT INTO `Темы` (`Код`, `Название`, `Дата`, `Описание`) VALUES
(1, 'Уроки рисования Photoshop 1', '2021-03-01 00:00:00', 'Учимся рисовать');
INSERT INTO `Темы` (`Код`, `Название`, `Дата`, `Описание`) VALUES
(2, 'Уроки рисования Photoshop 2', '2021-03-02 00:00:00', 'Учимся рисовать');
INSERT INTO `Темы` (`Код`, `Название`, `Дата`, `Описание`) VALUES
(3, 'Уроки рисования Photoshop 3', '2021-03-03 00:00:00', 'Учимся рисовать');
INSERT INTO `Темы` (`Код`, `Название`, `Дата`, `Описание`) VALUES
(4, 'Pascal 1', '2021-03-04 00:00:00', 'Программируем');
INSERT INTO `Темы` (`Код`, `Название`, `Дата`, `Описание`) VALUES
(5, 'Pascal 2', '2021-03-05 00:00:00', 'Программируем');
INSERT INTO `Темы` (`Код`, `Название`, `Дата`, `Описание`) VALUES
(6, 'Opera', '2021-03-06 00:00:00', 'Уроки браузера');
INSERT INTO `Темы` (`Код`, `Название`, `Дата`, `Описание`) VALUES
(7, 'Chrome', '2021-03-07 00:00:00', 'Просмотр браузера');
INSERT INTO `Темы` (`Код`, `Название`, `Дата`, `Описание`) VALUES
(8, 'What is Internet', '2021-03-08 00:00:00', 'Обучение интернету');
# 8 records

```

```

#
# Dumping data for table `Журнал оценок`
#

```

```

INSERT INTO `Журнал оценок` (`Код`, `Учащийся`, `Тема`, `Оценка`,
`Присутствовал`) VALUES (1, 1, 1, 8, 1);
INSERT INTO `Журнал оценок` (`Код`, `Учащийся`, `Тема`, `Оценка`,
`Присутствовал`) VALUES (2, 2, 1, 7, 1);
INSERT INTO `Журнал оценок` (`Код`, `Учащийся`, `Тема`, `Оценка`,
`Присутствовал`) VALUES (3, 3, 1, 6, 1);

```

					Лист
					51
Изм.	Лист	№ докум.	Подпись	Дата	

```

INSERT INTO `Журнал оценок` (`Код`, `Учащийся`, `Тема`, `Оценка`,
`Присутствовал`) VALUES (4, 4, 1, 8, 1);
INSERT INTO `Журнал оценок` (`Код`, `Учащийся`, `Тема`, `Оценка`,
`Присутствовал`) VALUES (5, 5, 1, 7, 1);
INSERT INTO `Журнал оценок` (`Код`, `Учащийся`, `Тема`, `Оценка`,
`Присутствовал`) VALUES (6, 6, 1, 9, 1);
INSERT INTO `Журнал оценок` (`Код`, `Учащийся`, `Тема`, `Оценка`,
`Присутствовал`) VALUES (7, 7, 1, 4, 1);
INSERT INTO `Журнал оценок` (`Код`, `Учащийся`, `Тема`, `Оценка`,
`Присутствовал`) VALUES (8, 8, 1, 8, 1);
INSERT INTO `Журнал оценок` (`Код`, `Учащийся`, `Тема`, `Оценка`,
`Присутствовал`) VALUES (9, 9, 1, 9, 1);
INSERT INTO `Журнал оценок` (`Код`, `Учащийся`, `Тема`, `Оценка`,
`Присутствовал`) VALUES (10, 2, 2, 7, 1);
INSERT INTO `Журнал оценок` (`Код`, `Учащийся`, `Тема`, `Оценка`,
`Присутствовал`) VALUES (11, 3, 2, 10, 1);
INSERT INTO `Журнал оценок` (`Код`, `Учащийся`, `Тема`, `Оценка`,
`Присутствовал`) VALUES (12, 4, 2, 10, 1);
INSERT INTO `Журнал оценок` (`Код`, `Учащийся`, `Тема`, `Оценка`,
`Присутствовал`) VALUES (13, 6, 2, 10, 1);
INSERT INTO `Журнал оценок` (`Код`, `Учащийся`, `Тема`, `Оценка`,
`Присутствовал`) VALUES (14, 6, 2, 9, 1);
INSERT INTO `Журнал оценок` (`Код`, `Учащийся`, `Тема`, `Оценка`,
`Присутствовал`) VALUES (15, 9, 2, 8, 1);
INSERT INTO `Журнал оценок` (`Код`, `Учащийся`, `Тема`, `Оценка`,
`Присутствовал`) VALUES (16, 1, 4, 0, 0);
INSERT INTO `Журнал оценок` (`Код`, `Учащийся`, `Тема`, `Оценка`,
`Присутствовал`) VALUES (17, 3, 4, 9, 0);
INSERT INTO `Журнал оценок` (`Код`, `Учащийся`, `Тема`, `Оценка`,
`Присутствовал`) VALUES (18, 4, 4, 0, 0);
INSERT INTO `Журнал оценок` (`Код`, `Учащийся`, `Тема`, `Оценка`,
`Присутствовал`) VALUES (19, 6, 4, 8, 1);
INSERT INTO `Журнал оценок` (`Код`, `Учащийся`, `Тема`, `Оценка`,
`Присутствовал`) VALUES (20, 7, 5, 9, 1);
INSERT INTO `Журнал оценок` (`Код`, `Учащийся`, `Тема`, `Оценка`,
`Присутствовал`) VALUES (21, 8, 6, 0, 0);
INSERT INTO `Журнал оценок` (`Код`, `Учащийся`, `Тема`, `Оценка`,
`Присутствовал`) VALUES (22, 9, 7, 6, 1);
# 22 records

```

```

#
# Dumping data for table 'Вклад в развитие'
#

```

						Лист
						52
Изм.	Лист	№ докум.	Подпись	Дата		

INSERT INTO `Вклад в развитие` (`Код`, `Учащийся`, `Дата`, `Вклад`)
VALUES (1, 4, '2021-03-01 00:00:00', 10);

INSERT INTO `Вклад в развитие` (`Код`, `Учащийся`, `Дата`, `Вклад`)
VALUES (2, 5, '2021-03-09 00:00:00', 20);

INSERT INTO `Вклад в развитие` (`Код`, `Учащийся`, `Дата`, `Вклад`)
VALUES (3, 8, '2021-03-10 00:00:00', 30);

INSERT INTO `Вклад в развитие` (`Код`, `Учащийся`, `Дата`, `Вклад`)
VALUES (4, 1, '2021-03-11 00:00:00', 20);

INSERT INTO `Вклад в развитие` (`Код`, `Учащийся`, `Дата`, `Вклад`)
VALUES (5, 4, '2021-03-11 00:00:00', 12);

5 records

						Лист
						53
Изм.	Лист	№ докум.	Подпись	Дата		