Министерство образования Московской области

Государственное образовательное учреждение высшего образования Московской области

«Государственный гуманитарно-технологический университет»

**Ликино-Дулевский политехнический колледж – филиал ГГТУ**

**КУРСОВОЙ ПРОЕКТ**

«Разработка приложения по автоматизации учёта проектов и задач»

МДК 02.01 «Технология разработки программного обеспечения»

**Выполнил:**

Хапов Игорь Романович\_\_\_\_\_\_\_\_\_ \_\_\_

студент группы ИСП.18.2А\_\_\_\_\_\_\_\_\_\_\_\_\_

09.02.07 Информационные системы и программирование) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

очной формы обучения\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Руководитель:**

Селивёрстова Ольга Михайловна\_\_\_\_\_\_\_\_\_\_\_

Оценка

\_\_\_\_\_\_\_\_\_/\_\_\_­­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

Подпись руководителя

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ликино-Дулево

2022 год

Оглавление

[Введение 4](#_Toc93997204)

[1. Разработка системного проекта 6](#_Toc93997205)

[1.1 Назначение разработки 6](#_Toc93997206)

[1.2 Требования к функциональным характеристикам 6](#_Toc93997207)

[1.3 Требования к надежности и безопасности 6](#_Toc93997208)

[1.4 Требования к составу и параметрам технических средств 7](#_Toc93997209)

[1.5 Требования к информационной и программной совместимост7](#_Toc93997210)

[2. Разработка технического проекта 8](#_Toc93997211)

[2.1 Обоснование выбора CASE-средств 8](#_Toc93997212)

[2.1.1. Диаграмма прецедентов (таблица бизнес-процессов) 9](#_Toc93997213)

[2.1.2. Моделирование бизнес-процессов 9](#_Toc93997214)

[2.1.3. Словесный алгоритм бизнес-процесса 9](#_Toc93997215)

[2.1.4. Диаграмма действий бизнес-процесса 10](#_Toc93997216)

[2.1.5. Таблица операций 10](#_Toc93997217)

[3. Реализация 11](#_Toc93997218)

[3.1 Обоснование выбора средств разработки 11](#_Toc93997219)

[3.2 Руководство программиста 12](#_Toc93997220)

[3.3 Руководство пользователя 20](#_Toc93997221)

[3.3.1 Выполнение программного модуля 20](#_Toc93997222)

[4. Тестирование и отладка 25](#_Toc93997223)

[4.1 Таблица класса эквивалентности 27](#_Toc93997224)

[5. Разработка механизмов защиты данных 27](#_Toc93997225)

[Заключение 30](#_Toc93997226)

[Список литературы 31](#_Toc93997227)

[Приложения 32](#_Toc93997228)

# Введение

Управление проектами — деятельность по решению задач и достижению поставленных целей проекта.

Управление проектами есть применение знаний, навыков, инструментов и техник при выполнении проектной деятельности для достижения требований проекта и запланированных результатов. Проектные команды могут достигать результатов, используя широкий перечень подходов (предиктивный, гибридный, адаптивный). Управление проектами является частью системы менеджмента предприятия.

Существует множество подходов к управлению жизненным циклом проекта/продукта в зависимости от типа проекта:

Предположение о неизменности требований, низких рисках, критичности сроков завершения. В этом случае применяется водопадный жизненный цикл. Для планирования и контроля хорошо применимы методы PERT, метод критического пути, метод освоенного объема, диаграмма Ганта. Основная слабая сторона классического проектного менеджмента – нетолерантность к изменениям. Подход применим к строительным и инженерным проектам, в которых содержание проекта остаётся практически неизменным в течение всего проекта.

Предположение о критичности качества, при этом требования к сроку и ресурсам достаточно гибки (под качеством здесь понимается полнота удовлетворения потребностей, как известных, так и неизвестных заранее, часто создаваемых выходом нового продукта). В этом случае применяются спиральный жизненный цикл, гибкая методология разработки продукта, минимизация администрирования и неформальный подход к управлению проектом. К преимуществам относят гибкость и адаптивность под изменения требований. В качестве недостатков отмечают что гибкость может приводить к потере фокуса, усложнению внесения непредвиденных изменений.

Предположение о высоких неопределенностях и рисках проекта (для инновационных проектов и стартапов). В этом случае применяются подходы управления бережливый стартап, Phase–gate model, Управление реализацией преимуществ.

Менеджер проектов — программа для автоматизации процесса учёта выполнения проектов и задач, а также учёта занятости работников. Суть работы приложения заключается в отслеживании проектов с последующим учётом их выполнения путём выполнения задач, прикреплённых к этим проектам.

Процесс создания:

* Создание информационного наполнения;
* Создание системы управления проектами и задачами;
* Система отчётности о готовности проектов и задач;
* Система отслеживания сроков.

Целью курсовой работы является изучение предметной области решения проблем на предприятии и разработка менеджера проектов в соответствии с этапами жизненного цикла программного продукта.

При создании проекта предположительно будет использоваться следующее ПО:

* C# – как основной язык программирования;
* Интегрированная среда разработки Visual Studio 2019 (при разработке интерфейса приложения и логики его работы)
* Система управления базами данных SQL Server Management Studio (для организации структурированного хранения данных)
* Microsoft Word (для создания отчёта и формирования договора)
* Microsoft Visio (для уточнения бизнес-процессов проекта и составления диаграммы действий)

# Разработка системного проекта

## Назначение разработки

Руководитель отдела информационных технологий организует работу ИТ-подразделения, решая разнообразные технологические и кадровые вопросы, принимая тактические решения и осуществляя стратегическое планирование. Эту должность занимает опытный IT-специалист, который получает запросы на цифровое обслуживание от всех сотрудников, определяет схему их удовлетворения, отдает приказы исполнителям и контролирует конечные результаты, назначает, ведёт учёт проектов и задач.

## 1.2 Требования к функциональным характеристикам

Автоматизированная информационная система должна обеспечивать выполнение функций:

* Хранение, просмотр и добавление данных о проектах, задачах и работниках;
* Учёт данных о проектах и задачах
* Назначение сроков проектов и задач для них
* Экспорт отчётности о выполненных проектах

## Требования к надежности и безопасности

Разрабатываемое программное обеспечение должно нормально функционировать при бесперебойной работе компьютера пользователя.

При возникновении ошибок или сбоев в работе компьютера восстановление нормальной работы программы должно производиться после полной перезагрузки операционной системы, запуска исполняемого файла. При условии, если пользователь до сбоя работы сохранил внесенные данные, в таком случае данные сохранятся в базе данных.

Для защиты информации на компьютере пользователя должны быть предусмотрены необходимые меры: пароль на вход в компьютер, антивирусные программы, отсутствие на компьютере подозрительных программ, полученных с неофициальных источников.

## Требования к составу и параметрам технических средств

Минимальные системные требования для работы программного продукта должны быть следующими: частота процессора – 1 800 Ггц, объем оперативной памяти – 4 ГБ, объем свободного дискового пространства 500 МБ, разрешение монитора 1280х720. Для печати договора необходим принтер.

**Характеристика компьютера**

Процессор - Intel Core i3-4130

Частота процессора – 3.4 ГГц

Оперативная память - 8 Гб

Монитор – Philips LED, расширение 1440x900

Клавиатура - Oklick 250C

Мышь – A4Tech X718BK

## Требования к информационной и программной совместимости

Программа должна работать на операционной системе Windows 7. Все формируемые отчеты должны иметь возможность печати, а также экспорта в Microsoft Excel.

Для работы с базой данных и приложением, на рабочем компьютере должно быть установлено следующее программное обеспечение:

* База данных – Microsoft SQL Server 2019.
* Утилита для подключения и управления БД - SQL Server Management Studio 17.
* Программа для связи с БД – Visual Studio 2019.
* Для вывода информации – Microsoft Excel.

# Разработка технического проекта

## Обоснование выбора CASE-средств

Одним из основных этапов жизненного цикла программного продукта является этап проектирования предметной области с точки зрения заказчика, непосредственного участника бизнес-процессов. Цель: выявить участников, определить их взаимосвязь, автоматизировать процессы, наладить электронный документооборот в организации.

Use Case (вариант использования, ВИ, Прецедент, юскейс) — это сценарная техника описания взаимодействия. С помощью Use Case может быть описано и пользовательское требование, и требование к взаимодействию систем, и описание взаимодействия людей и компаний в реальной жизни. Проектирование предметной области с использованием Case-средства MS Visio и Case-технологии UML.

В разработке ПО эту технику часто применяют для проектирования и описания взаимодействия пользователя и системы, поэтому название Use Case часто воспринимает как синоним требования человека-пользователя к решению определенной задачи в системе.

Исторически требования к функционированию системы описывались в виде отдельных функций. Ивар Якобсон в середине 1990-х годов предложил UseCase как альтернативу и дополнение описания функциональности системы. Описание требований к системе не в виде отдельных функций, а в виде описания контекста и последовательности действий пользователя помогает сформировать набор функциональных требований, который будет обеспечивать полноту и не избыточность требований.

### 2.1.1. Диаграмма прецедентов (таблица бизнес-процессов)

Рис. 1 «Диаграмма прецедентов»

### 2.1.2. Моделирование бизнес-процессов

### 2.1.3. Словесный алгоритм бизнес-процесса

### 2.1.4. Диаграмма действий бизнес-процесса

### 2.1.5. Таблица операций

# Реализация

## Обоснование выбора средств разработки

При разработке программы использовался язык программирования c#, в среде разработки Visual Studio 2019 с хранением информации в базе данных MS SQL Server 2018.

**C# -** C# является объектно-ориентированным языком, но поддерживает также и компонентно-ориентированное программирование. Разработка современных приложений все больше тяготеет к созданию программных компонентов в форме автономных и само описательных пакетов, реализующих отдельные функциональные возможности. Главная особенность таких компонентов в том, что они представляют собой модель программирования со свойствами, методами и событиями. У них есть атрибуты, предоставляющие декларативные сведения о компоненте. Они включают в себя собственную документацию. C# предоставляет языковые конструкции, непосредственно поддерживающие такую концепцию работы. Благодаря этому C# подходит для создания и применения программных компонентов.

**Microsoft SQL Server** - система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Mic rosoft и Sybase.

**SQL Server Managem-ent Studio (SSMS)** — утилита для Microsoft SQL Server для конфигурирования, управления и администрирования компонентов базы данных. Данная утилита содержит редактор скриптов (который в основном и будет использоваться) и графическую программу, которая работает с объектами и настройками сервера. Главным инструментом SQL Server Management Studio является Object Explorer, который позволяет пользователю просматривать, извлекать объекты сервера, а также управлять ими.

**Visual Studio** — это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. Интегрированная среда разработки (IDE) представляет собой многофункциональную программу, которую можно использовать для различных аспектов разработки программного обеспечения. Также Visual Studio включает в себя компиляторы, средства автозавершения кода и графические конструкторы.

## Руководство программиста

Разработана база данных для «Менеджер проектов»

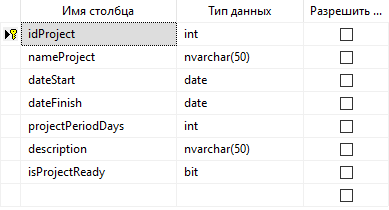


Рис. 3 «Таблица Проекты»

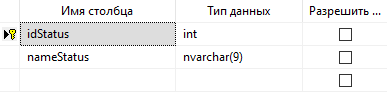


Рис. 4 «Таблица Статус»

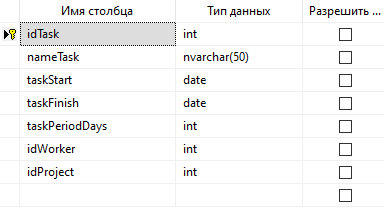


Рис. 5 «Таблица Задачи»

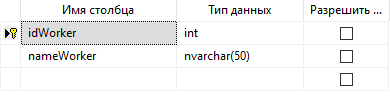


Рис. 6 «Таблица Работники»

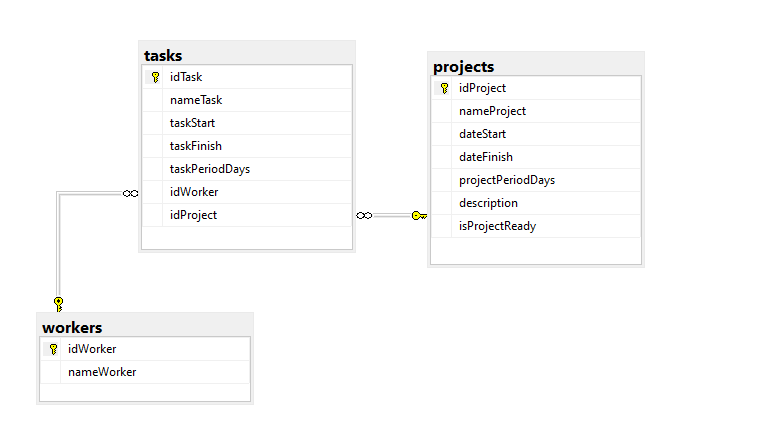


Рис. 12 «Структура базы данных»

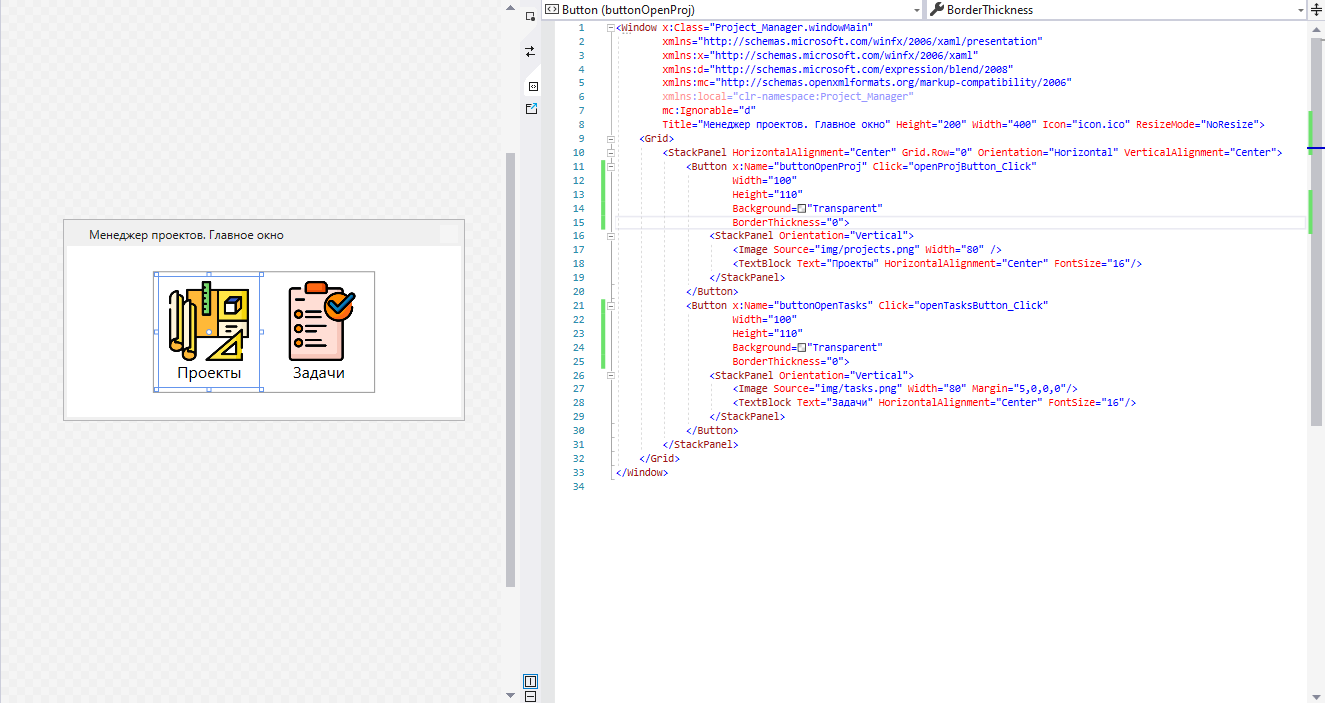


Рис. 13 «Конструктор главного окна»

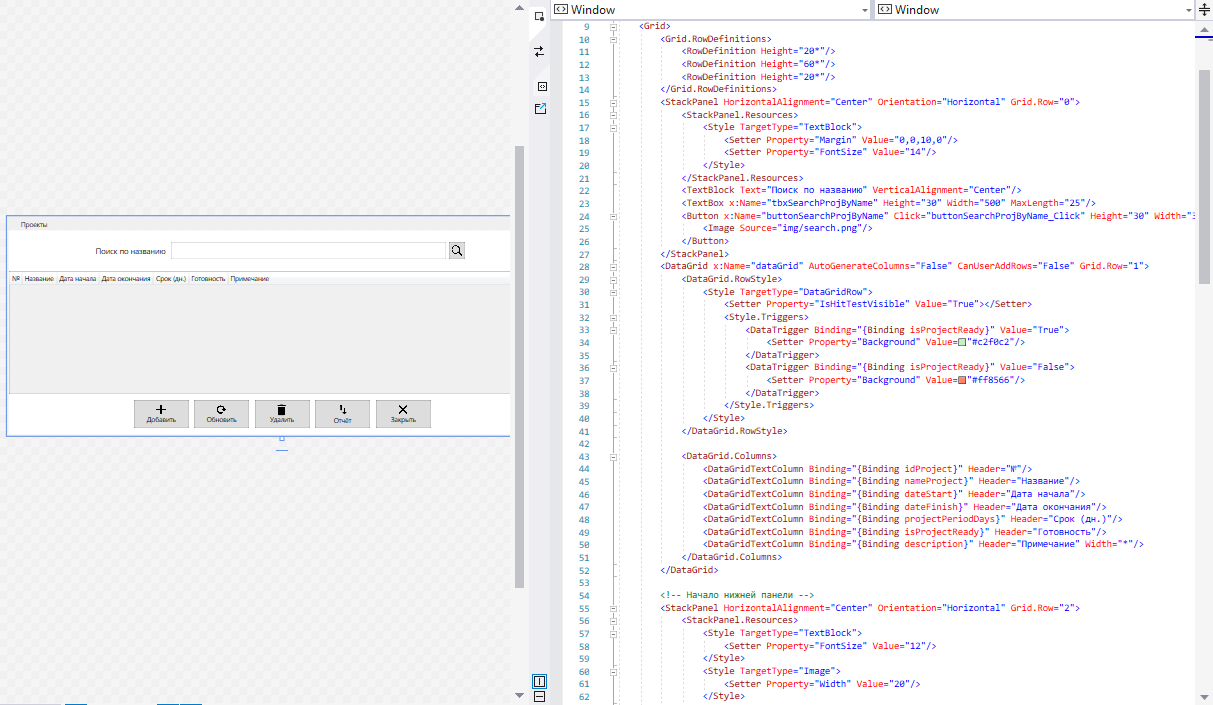


Рис. 14 «Конструктор окна с проектами»

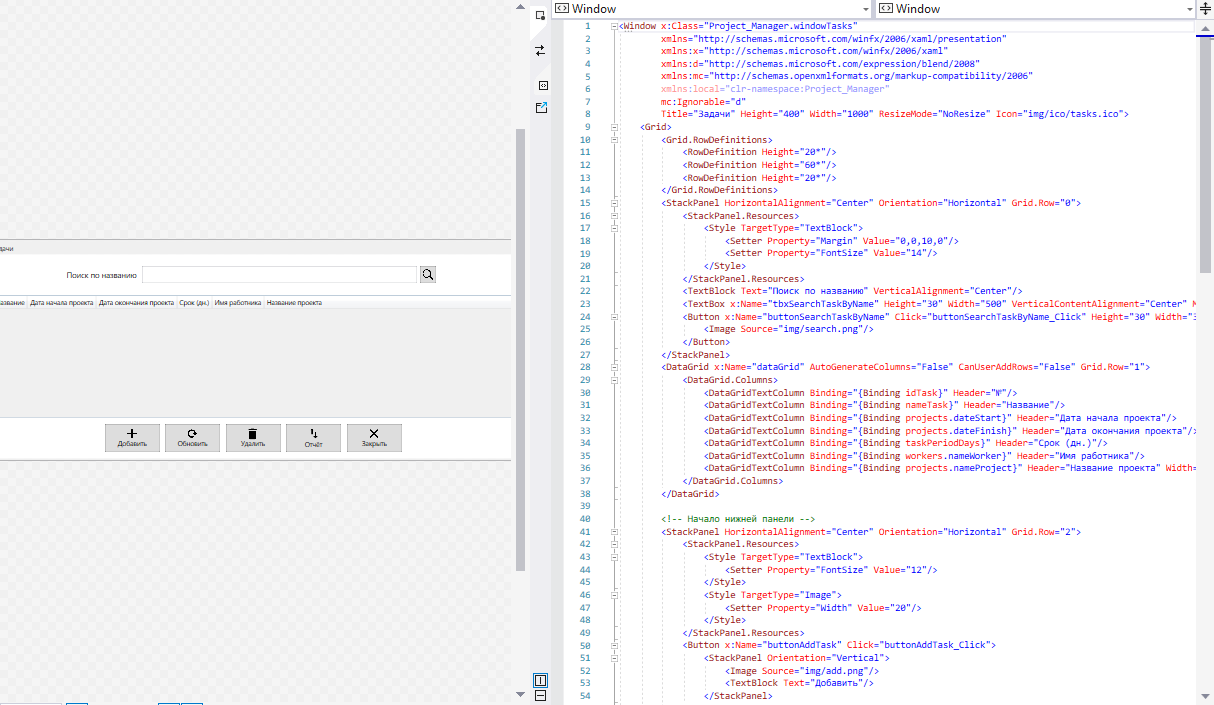


Рис. 15 «Конструктор окна с задачами»

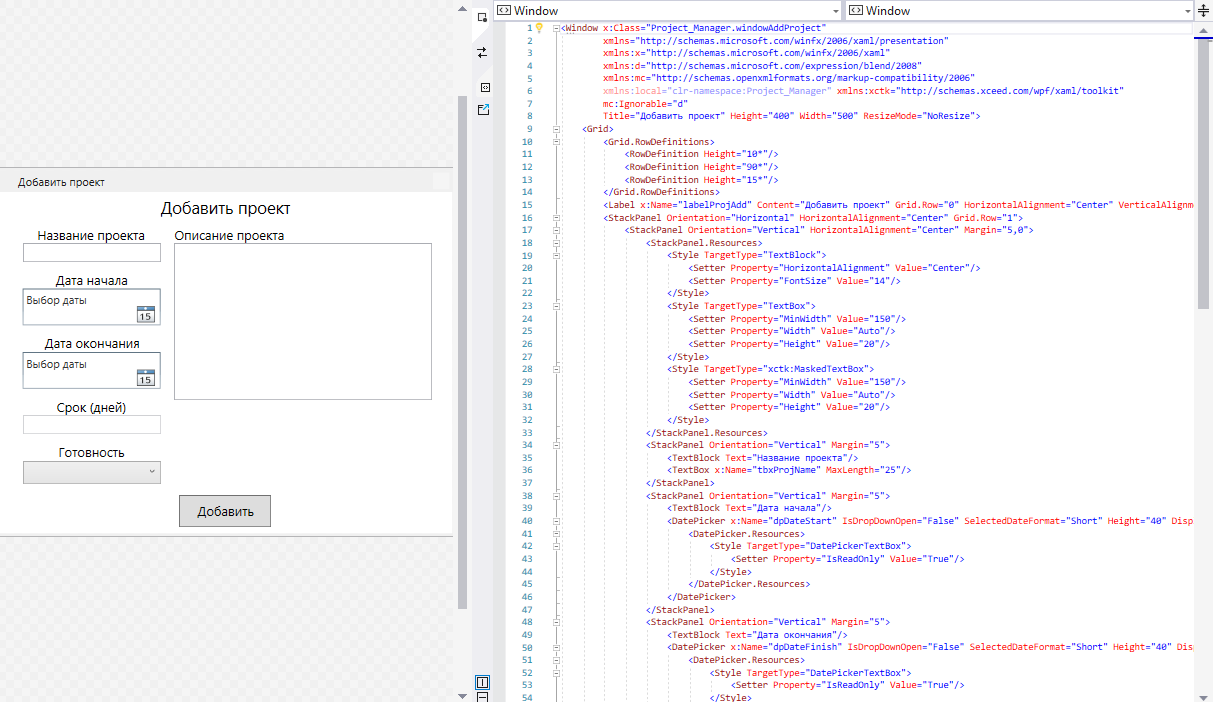


Рис. 16 «Конструктор окна с добавлением проекта»

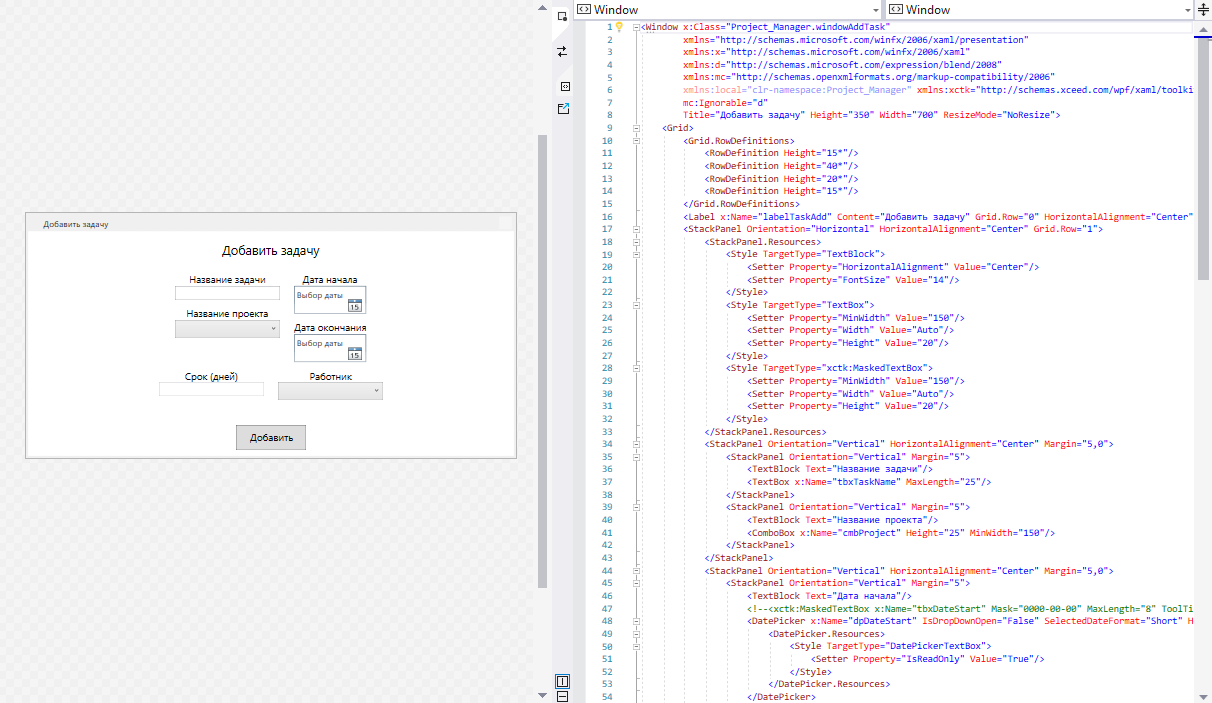


Рис. 17 «Конструктор окна с добавлением задачи»

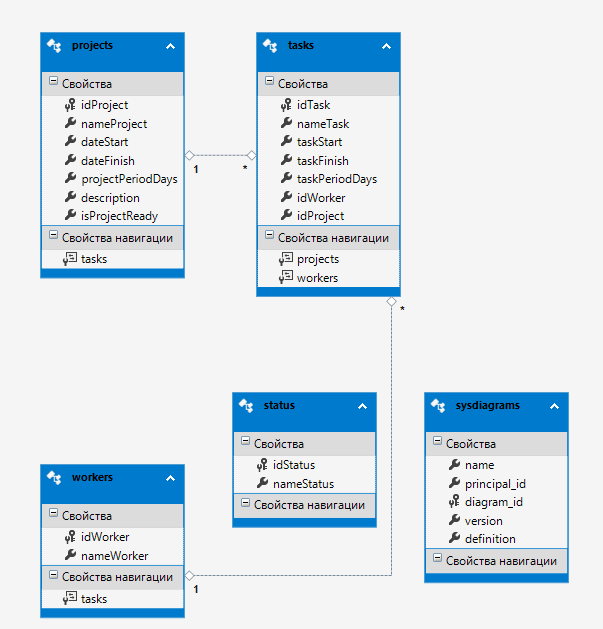


Рис. 26 «Классы форм»

## Руководство пользователя

### 3.3.1 Выполнение программного модуля

Для запуска программы надо запустить Project Manager.exe, который находится по пути:

\Project Manager\bin\Debug\ Project Manager.exe

После открытия исполняющего файла, открывается главное окно

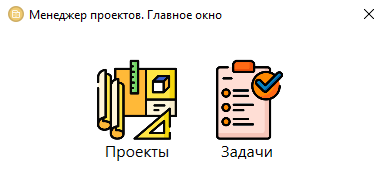


Рис. 27 «Главное окно»

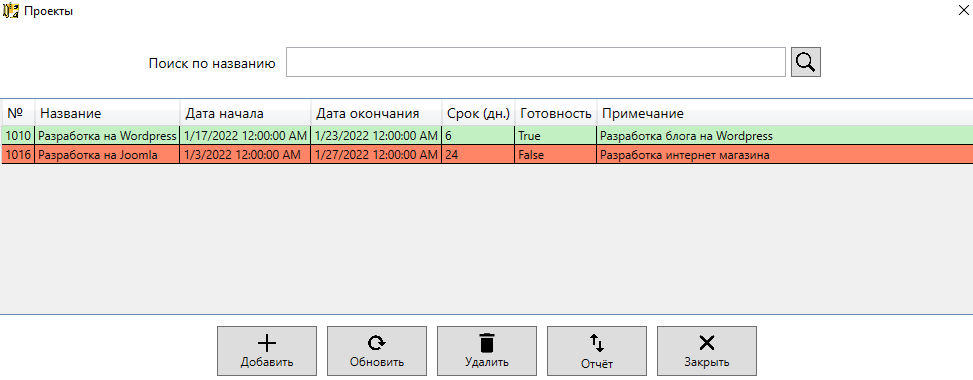


Рис. 28 «Страница просмотра проектов»

На данной странице реализована фильтрация по названию проекта

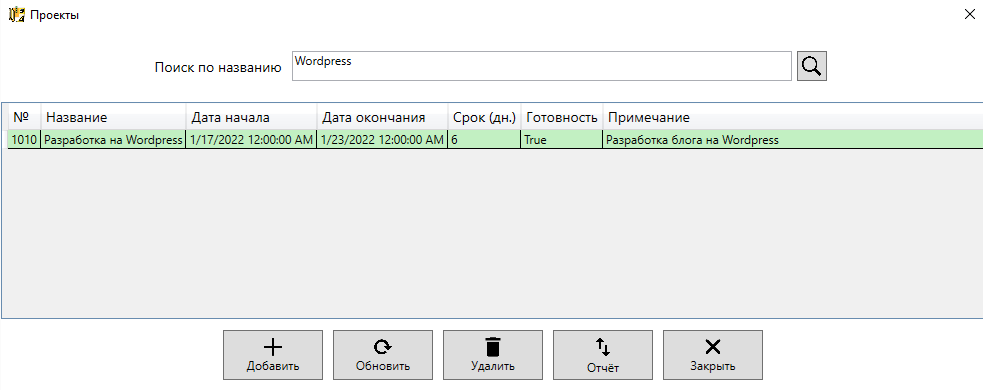


Рис. 29 «Страница просмотра проектов»

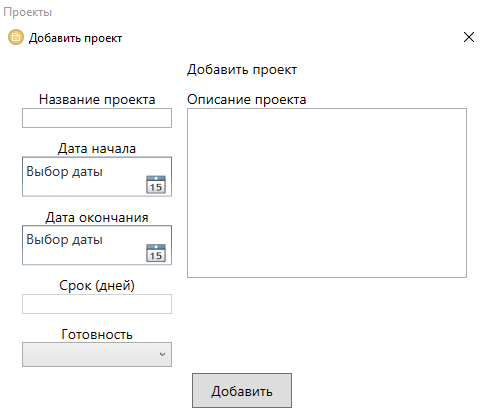


Рис. 30 «Страница добавления проекта»

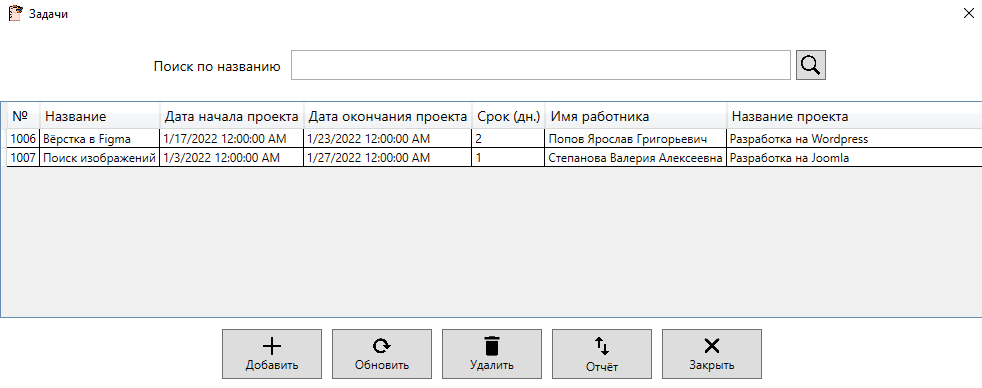


Рис. 31 «Страница просмотра задач»

На данной странице реализована фильтрация по названию задачи

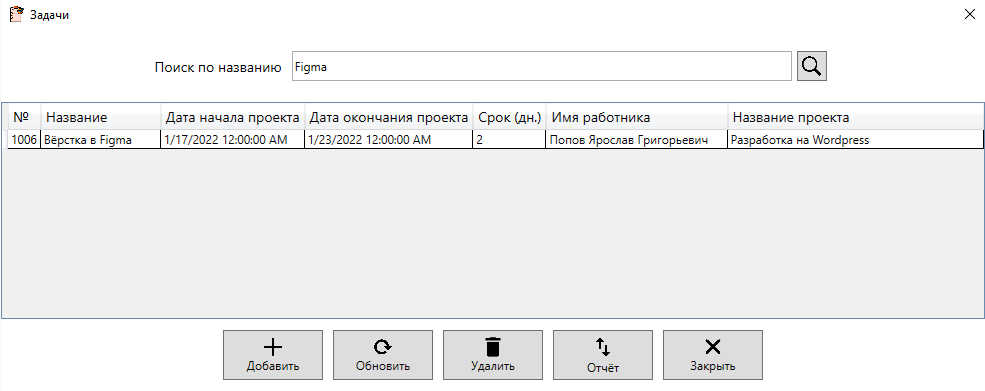


Рис. 32 «Фильтрация задач по названию»

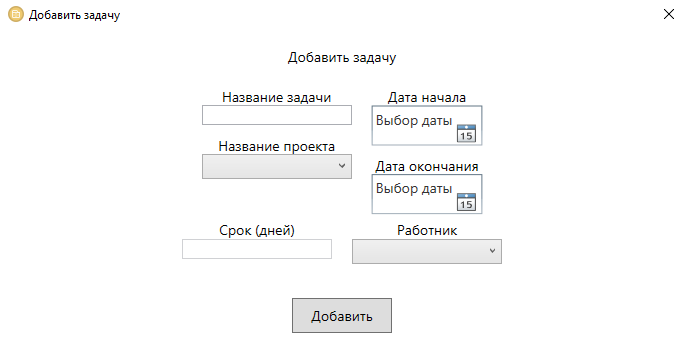


Рис. 33 «Страница добавления задачи»

# Тестирование и отладка

Отладка ПС - это деятельность, направленная на обнаружение и исправление ошибок в ПС с использованием процессов выполнения его программ. Тестирование ПС - это процесс выполнения его программ на некотором наборе данных, для которого заранее известен результат применения или известны правила поведения этих программ. Указанный набор данных называется тестовым или просто тестом. Таким образом, отладку можно представить в виде многократного повторения трех процессов: тестирования, в результате которого может быть констатировано наличие в ПС ошибки, поиска места ошибки в программах и документации ПС и редактирования программ и документации с целью устранения обнаруженной ошибки. Другими словами,

Отладка = Тестирование + Поиск ошибок + Редактирование.

Успех отладки ПС в значительной степени предопределяет рациональная организация тестирования. При отладке ПС отыскиваются и устраняются, в основном, те ошибки, наличие которых в ПС устанавливается при тестировании. Как уже отмечено, тестирование не может доказать правильность ПС, в лучшем случае оно может продемонстрировать наличие в нем ошибки. Другими словами, нельзя гарантировать, что тестированием ПС практически выполнимым набором тестов можно установить наличие каждой имеющейся в ПС ошибки. Поэтому возникает две задачи. Первая задача: подготовить такой набор тестов и применить к ним ПС, чтобы обнаружить в нем по возможности большее число ошибок. Однако чем дольше продолжается процесс тестирования (и отладки в целом), тем большей становится стоимость ПС. Отсюда вторая задача: определить момент окончания отладки ПС (или отдельной его компоненты). Признаком возможности окончания отладки является полнота охвата пропущенными через ПС тестами (т.е. тестами, к которым применено ПС) множества различных ситуаций, возникающих при выполнении программ ПС.

**Есть два крайних подхода к проектированию тестов:**

1. Тестирование черного ящика — это когда нету информации по работе программы внутри, нету доступа к коду

Поэтому ориентируемся только на внешнее поведение или ТЗ.

Примеры:

1.1 Программа добавляет новую заявки при использовании некорректной даты, а именно даты из прошлого.

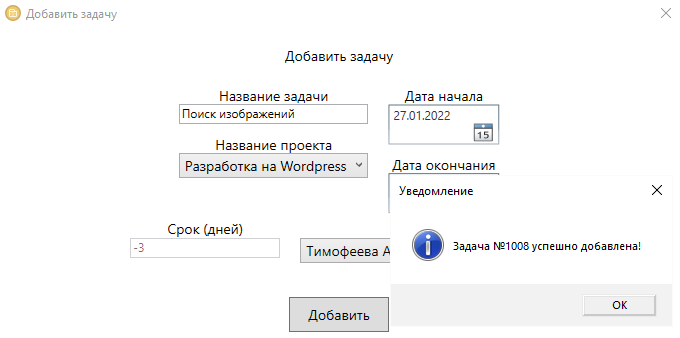


Рис. 40 «Добавление некорректной заявки»

1. Тестирование белого ящика — это когда у есть доступ к коду, можно его смотреть или тестировать. Есть возможность запуска в дебаге, или написать автотесты.

2.1 Фильтрация по мере ввода текста взамен на нажатие кнопки

До

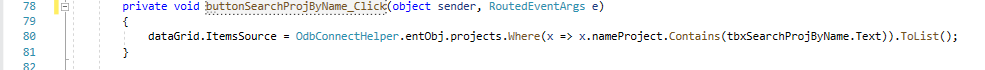


Рис. 42 «Фильтрация по нажатию на кнопку»

После

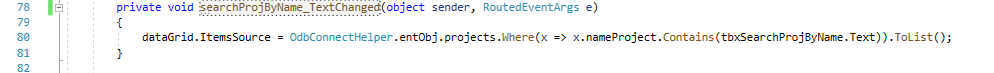


Рис. 43 «Фильтрация по мере набора текста»

## Таблица класса эквивалентности

Таблица 3 «класс эквивалентности»

|  |  |  |
| --- | --- | --- |
| Входное или выходное событие | Допустимые классы эквивалентности | Недопустимые классы эквивалентности |
| Ввод Название, Описание | Ввод всех букв | Длина поля не более 25 символа, включая 25ый  Пустое поле |
| Ввод Дата выполнения, Дата окончания | Только цифры от 0 до 12 для «День и месяц» и от 2020 до 2030 для «Года» так же Точка (.) или Слэш (/) | Пустое поле, длина поля не более 8 символов  Пустое поле |

# Разработка механизмов защиты данных

Защищаемым объектом является не сама БД или экспертная система, а компьютерная система (КС) в целом. Понятие защиты применимо не только к сохраняемым данным. Бреши в системе защиты могут возникать и в других частях системы, например, линий связи, что в свою очередь, подвергает опасности и собственно БД. Следовательно, защита БД должна охватывать используемое оборудование, программное обеспечение, персонал и собственно данные.

Информационная безопасностью (information security) – состояние рассматриваемой КС, при которой она, с одной стороны, способна противостоять дестабилизирующему воздействию внешних и внутренних информационных угроз, а с другой – функционирование системы и сам факт ее наличия не создают угроз для ее пользователей, для внешней среды и для элементов самой КС. Безопасность информации – состояние защищенности информации, при котором обеспечены ее конфиденциальность, целостность и доступность.

Защита информации - комплекс мероприятий, направленных на обеспечение информационной безопасности (целостности, доступности и, если нужно, конфиденциальности информации и ресурсов, используемых для ввода, хранения, обработки и передачи данных).

Защищаемая информация (sensitive information) – информация, являющаяся предметом собственности и подлежащая защите в соответствии с требованиями правовых документов или требованиями, устанавливаемыми собственником информации. Очень часто наряду с термином защищаемая информация употребляется и термин конфиденциальная информация – информация, доступ к которой ограничивается в соответствии с законодательством или требованиями владельца. Конфиденциальность (confidentiality или secrecy) - это свойство информации быть известной только допущенным и прошедшим проверку субъектам системы.

Защищаемая информационная система – система, предназначенная для обработки защищаемой информации с требуемым уровнем ее защищенности.

Система называется безопасной, если она, используя соответствующие аппаратные и программные средства, управляет доступом к информации так, что только должным образом авторизованные лица или же действующие от их имени процессы получают право читать, писать, создавать и удалять информацию. Система считается надежной (иначе доверенной), если она с использованием достаточных аппаратных и программных средств обеспечивает одновременную обработку информации разной степени секретности группой пользователей без нарушения прав доступа.

Так, например, в программе реализовано каскадное удаление, чтобы при удалении какой-либо записи не произошел критический сбой, который не позвонит в дальнейшем использовать программу

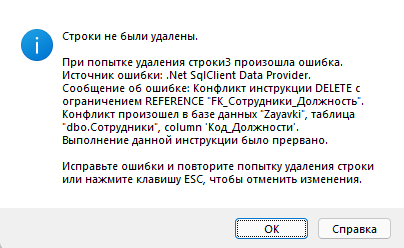


Рис. 44 «Пример каскадного удаления»

# Заключение

Для информационной системы «Менеджер проектов» разработано приложение.

В результате исследования предметной области разработано приложение, которое полностью соответствует всем требованиям к функциональным характеристикам, а также к требованию надежности и безопасности к составу параметрам технических средств и программной совместимости.

Разработана документация, в котором обоснование выбора Case-средств, путем создания диаграммы прецедентов, моделирование бизнес-процессов, словесный алгоритм, диаграмма действий и таблица операций.

Реализовано обоснование выбора средств разработки, почему выбор пал именно на те средства который использованы, а также составлены руководства пользователя и программиста, где описано поведение программы на работу с ней, и всяческие нюансы, и код программы, чтобы в если другому программисту попала программа, он смог про руководство программиста быстро сориентироваться и разобраться в приложении.

Приложение полностью протестировано, таким образом программа полностью отлажена.

В процессе разработки использовалась СУБД Sql Server 2018, применена технология Entity Framework, язык программирования C# Visual Studio. Приложение соответствует поставленным задачам, реализованы функции: просмотр, добавление, сортировка, выборка следующих таблиц: Сотрудники, Заявки, а также приложение может сделать экспорт данных в Excel по таблице Заявки. Реализованная программа полностью протестирована и отлажена в соответствии с документацией. Программный модуль может быть установлен на любую операционную систему, начиная с Windows 7 x32. В проекте используется язык запросов LINQ.

# Список литературы

1. C# 7.0. Справочник. Полное описание языка.: Пер. с англ. – СпБ.: ООО “Альфакнига”, 2018. – 1024 с. : ил. – Парал. тит. англ.
2. Microsoft Visual C#. Подробное руководство. 8-е изд. – СПб.: Питер, 2017. – 848 с.: ил. – (Серия «Библиотека программиста»).
3. Программирование на C# для начинающих. Основные сведения / Алексей Васильев. – Москва : Эксмо, 2018. – 592 с. – (Российский компьютерный бестселлер).
4. Язык программирования C# 7 и платформы .NET и .NET Core, 8-е изд. : Пер. с англ. – СПб. : ООО “Диалектика”, 2018 – 1328 с. : ил. – Парал. тит. англ.
5. Оппель, Эндрю Дж. SQL. Полное руководство / Оппель Эндрю Дж.. - М.: Диалектика / Вильямс, 2017. - 902 c.
6. Рудаков А. В. Р83 Технология разработки программных продуктов. Практикум: учеб. пособие для студ. учреждений сред. проф. образования / А. В.Рудаков, Г. Н. Федорова. — 4-е изд., стер. — М. : Издательский центр «Академия»; 2018. — 192 с
7. Грекул, В. И. Проектирование информационных систем В. И. Грекул, Г. Н. Денищенко, Н. Л. Коровкина. – Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), 2005. – 304 с. : ил.,табл., схем. – (Основы информационных технологий). –– с. 298 - 299.
8. Рудаков. А. В. Технология разработки программных продуктов: учеб. пособие для студ. Сред. Проф. Образования / А.В Рудаков -2-е изд., стер – М.: Издательский центр «Академия», 2018 – 208с.

# Приложения

Код создания заявки

projects projObj = new projects()

{

nameProject = tbxProjName.Text,

dateStart = DateTime.Parse(dpDateStart.Text),

dateFinish = DateTime.Parse(dpDateFinish.Text),

projectPeriodDays = Convert.ToInt32(tbxSrok.Text),

description = tbxDescription.Text,

isProjectReady = availability

};

if (tbxProjName.Text == "" || dpDateStart.Text == "" || dpDateFinish.Text == "" || tbxSrok.Text == "")

{

MessageBox.Show("Не все данные введены!", "Данные", MessageBoxButton.OK, MessageBoxImage.Warning);

} else

{

OdbConnectHelper.entObj.projects.Add(projObj);

OdbConnectHelper.entObj.SaveChanges();

MessageBox.Show("Проект №" + projObj.idProject + " успешно добавлен!", "Уведомление", MessageBoxButton.OK, MessageBoxImage.Information);

this.Close();

}

Сохраненный отчет по проектам в Excel

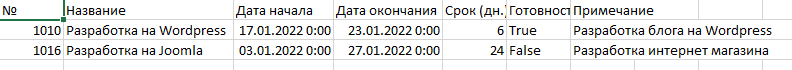


Рис. 45 «Отчет по проектам в Excel»