# Содержание

[Содержание 2](#_Toc27321892)

[1. Техническое задание 3](#_Toc27321893)

[1.1. Введение 3](#_Toc27321894)

[1.2. Основания для разработки 4](#_Toc27321895)

[1.3. Назначение разработки 4](#_Toc27321896)

[1.4. Требования к программе или программному изделию 5](#_Toc27321897)

[1.4.1. Требования к функциональным характеристикам 5](#_Toc27321898)

[1.4.2. Требования к надежности 6](#_Toc27321899)

[1.4.3. Условия эксплуатации 6](#_Toc27321900)

[1.4.4. Требования к составу и параметрам технических средств 7](#_Toc27321901)

[1.4.5. Требования к информационной и программной совместимости 7](#_Toc27321902)

[1.4.6. Требования к программной документации 7](#_Toc27321903)

[1.5. Технико-экономические показатели 7](#_Toc27321904)

[1.6. Стадии и этапы разработки 8](#_Toc27321905)

[1.7. Порядок контроля и приемки 8](#_Toc27321906)

[2. Исследовательская часть 9](#_Toc27321907)

[2.1 Постановка задачи проектирования. 9](#_Toc27321908)

[2.2. Описание предметной области. 9](#_Toc27321909)

[2.3 Перечень задач, подлежащих решению в процессе разработки. 13](#_Toc27321910)

[3. Проектно-конструкторская часть 14](#_Toc27321911)

[3.1 Разработка структуры программной системы 14](#_Toc27321912)

[3.2 Разработка интерфейса взаимодействия пользователя с системой 15](#_Toc27321913)

[3.3 Разработка архитектуры приложения 19](#_Toc27321914)

[4. Проектно-технологическая часть 21](#_Toc27321915)

[4.1 Проектирование начального и тестового наполнения базы данных 21](#_Toc27321916)

[4.2 Технологические решения, поддерживающие эксплуатационный цикл программы 22](#_Toc27321917)

[Список литературы 24](#_Toc27321918)

[Приложение 1 – Карта трансформатора 26](#_Toc27321919)

[Приложение 2 – Тестовое наполнение таблицы карт 27](#_Toc27321920)

# Техническое задание

# Введение

Предметом разработки является программная система оптимизации работы ООО «Электротехнический завод».

На указанном предприятии представлена система хранения информации о трансформаторах в виде картотеки, содержащей карты намотки двух- и трехфазных трансформаторов, реализуемых данным предприятием. Карта намотки представляет собой лист формата А4, содержащий следующие данные:

* первичное и вторичное напряжение трансформатора;
* мощность;
* тип трансформатора;
* количество витков на обмотках;
* наличие или отсутствие экранирования;
* сечение провода;
* перечень номенклатуры для сборки;
* графическая схема намотки трансформатора;
* информация о выводах;
* тип соединения для трехфазных трансформаторов и прочая информация.

С примером карты намотки трансформатора можно ознакомиться в приложении 1. Карты собраны в сборники по мощности

Требуется разработать программную систему, при помощи которой можно будет оцифровать имеющееся хранилище данных о трансформаторах, а также поддерживать базу в актуальном состоянии: вносить изменения, добавлять новые карты, автоматически обрабатывать поступающие в систему заказы.

# Основания для разработки

Разработка ведется на основании данного технического задания и приложений к нему. Этим документом подтверждается процесс разработки программного обеспечения на тему «Исследование и разработка системы хранения и обработки технических данных о трансформаторах на производственном предприятии»

# Назначение разработки

Целью разработки указанного программного обеспечения является повышение эффективности работы данного предприятия, что приведет к увеличению прибыли. Критериями повышения эффективности будут служить:

* уменьшение затрат времени на поиск нужной карты намотки трансформатора работниками предприятия;
* автоматическое формирование перечня карт трансформаторов в наличии для поступающих заказов, что сократит время на обработку заказа;
* уменьшение затрат времени на разработку и внедрение новых карт за счет системы актуализации базы данных: ввода новых и редактирования существующих карт;
* снижение трудовых и временных затрат на расчет требуемой для выполнения заказа номенклатуры за счет реализуемой системы учета.

# Требования к программе или программному изделию

# Требования к функциональным характеристикам

Программная система должна представлять собой работоспособное приложение, реализующее следующие функции:

* функция добавления в базу данных карты намотки трансформатора с бумажного носителя с корректным заполнением всех ключевых полей записи в базе данных;
* функция автоматической разработки новых карт;
* поиск требующейся карты намотки по ключевым полям: типу трансформатора (ОСМ, ОСЗМ, ОСМ1, ОСВМ для двухфазных, ТСМ, ТСЗМ, ТСВМ для трехфазных трансформаторов), мощности, первичному и вторичному напряжению, наличию или отсутствию экрана, наличию или отсутствию заявки на изготовление трансформаторов нового типа, по типу соединения трехфазных трансформаторов[11];
* функция автоматической обработки заказа и расчета требуемой на его выполнение номенклатуры;

В качестве входных данных в систему поступают заказы в виде документов в формате exel, содержащие перечень необходимых трансформаторов и их количество. Также в систему необходимо вносить информацию о новых картах, как при помощи сканирования существующих бумажных носителей, так и при автоматизированной разработке новой карты.

В системе должны быть реализованы роли инженера-технолога, специалиста по логистике, а также менеджера. Менеджер принимает заказы, инженер-технолог проектирует новые карты и проверяет существующие, а специалист по логистике отправляет в цех информацию о необходимых трансформаторах и номенклатуре для них.

Поиск трансформаторов должен производиться не зависимо от их типа, а также порядка цифр в первичном и вторичном напряжениях. То есть запись  во вторичном напряжении "110-220" для поиска эквивалентна записи "110-220". Для трехфазных трансформаторов стоит также учитывать различие в 1,73 раза между соединением "звезда" и "треугольник". То есть запись карты намотки трансформатора напряжением 380/380 типа "звезда"-"звезда" эквивалентна записи 380/220 для типа соединения "звезда"-"треугольник", а также 220/220 для типа "треугольник"-"треугольник"[11].

Система должна отражать актуальные данные о трансформаторах, реализуемых данным предприятием.

# Требования к надежности

Не допускается потеря хранимой в базе информации в случае экстренного отключения электроэнергии. Информация должна присутствовать в полном объеме и целостности за исключением случаев физического повреждения накопителей информации, хранящих базу данных[5].

Система должна работать стабильно, не допускается прерываний работы системы, вызванных ее внутренними ошибками. Интерфейс системы не должен содержать элементов, назначение которых неочевидно, которые вводят пользователя в заблуждение или не несут функционального значения.

В случае ввода пользователем некорректной информации система должна показать соответствующее предупреждение, прерывание работы программы в таком случае не допускается[9].

# Условия эксплуатации

Система предназначена для использования в нормальных условиях. Специальных требований по эксплуатации системе не предъявляется. Персонал обладает базовыми навыками работы с персональным компьютером, дополнительного обучения программа требовать не должна, кроме знакомства с конструкторской документацией.

# Требования к составу и параметрам технических средств

Рабочее место для установки программы снабжено операционной системой Windows 10 и 8 гигабайтами оперативной памяти. Для функционирования системы можно использовать до 50 Гб памяти жесткого диска.

# Требования к информационной и программной совместимости

Программа должна быть совместима с операционной системой Windows, реализовать интерфейс пользователя при помощи платформы Windows Presentation Foundation, в качестве системы управления базой данных рекомендуется использовать PostgreSQL. Программу реализовать на языке программирования C#[5].

# Требования к программной документации

Разрабатываемое программное обеспечение должно быть снабжено руководством пользователя. Также в пакет программной документации входит данное техническое задание. Вся техническая документация должна быть оформлена в соответствии с требованиями ГОСТ.

# Технико-экономические показатели

Разрабатываемая система не имеет аналогов. В результате использования программы эффективность работы данного предприятия должна возрасти за счет снижение временных и производственных затрат.

# Стадии и этапы разработки

Порядок разработки программной системы разбит на следующие этапы:

* Согласование и утверждение технического задания на разработку системы. Срок – 11.09.2019
* Согласование прототипа интерфейса, сущностей для хранения в базе данных. Срок – 24.09.2019
* Демонстрация рабочего приложения с системой поиска и оцифровки базы данных. Срок – 17.10.2019
* Демонстрация функций добавления новой карты намотки, а также внедрение системы в производственный процесс. Срок – 03.11.2019
* Добавление функции принятия заказа и подсчета номенклатуры. Сопровождение и отладка системы. Покрытие тестами. Срок – 01.12.2019

# Порядок контроля и приемки

Система в процессе разработки должна быть протестирована. Приемка работы осуществляется на предприятии уполномоченными на это лицами.

# Исследовательская часть

# Постановка задачи проектирования.

Целью написания данной курсовой работы является повышение эффективности работы производственного предприятия за счет автоматизации процесса хранения и обработки информации о выпускаемых трансформаторах.

Задача проектирования такой системы может быть решена при помощи следующих средств разработки: языка программирования C#, системы управления базами данных PostgreSQL, средством реализации пользовательских интерфейсов Windows Presentation Foundation[7].

Структурное проектирование системы осуществляется с помощью составления концептуальной, логической и физической моделей хранимых данных, а также IDEF0 диаграмме процессов системы. Описание этапов проектирования архитектуры приложения сопровождается описанием на естественном языке[9].

# **Описание предметной области**.

На предприятии хранение информации о реализуемых трансформаторах производится в печатном виде, что затрудняет поиск информации о трансформаторах во время оформления заказа. Информация о трансформаторе оформляется в виде карт трансформаторов, где перечислены основные его характеристики, а также отображена его схема.

Добавление информации о трансформаторах в картотеку производится сотрудниками предприятия. Также в их полномочия входит оформление поступающего заказа, подбор трансформаторов по параметрам и отправление собранной информации в цех вместе с перечнем требующейся для производства трансформатора номенклатуры.

На основании представленной информации в предметной области можно выделить несколько сущностей (лист 1): Трансформатор, Сотрудник, Заказ, Номенклатура. Ниже представлено описание каждой из этих сущностей с характеристикой их полей.

Первой и основной сущностью хранимых данных является Трансформатор. Он воплощает электронный эквивалент карты трансформатора и содержит ключевые для поиска поля:

* Тип: предприятие реализует трансформаторы следующих типов: ОСМ, ОСМ1, ОСЗМ, ОСВМ для однофазных трансформаторов; ТСМ, ТСМ1, ТСЗМ, ТСВМ для трехфазных трансформаторов. О – однофазный, Т – трехфазный, С – сухой при открытом исполнении, СВ – при водозащищенном исполнении, СЗ – при каплезащищенном исполнении, М – морской[11].
* Мощность: мощность реализуемых трансформаторов на превосходит 4кВа.
* Первичное напряжение и Вторичное напряжение: в картах намотки трансформатора эти поля представлены в виде строки следующего содержания: «380-660-1140/36-110/42/8», где до первой косой черты представлено первичное напряжение трансформатора (380-660-1140 в данном примере), а все, что идет после нее – вторичное напряжение трансформатора (36-110/42/8 в данном примере). Необходимо учитывать, что порядок цифр в записи напряжений для поиска безразличен. То есть два трансформатора с аналогичными характеристиками, но с различным порядком цифр в напряжениях представляют одну и ту же карту намотки трансформатора.
* Экран: поле, характеризующее наличие или отсутствие экранирования.
* Дата создания: поле, по которому работники предприятия могут судить об актуальности той или иной карты намотки.
* Заявка: поле, отражающее наличие или отсутствие особых требований к производству трансформатора.
* Ссылка на документ: поле, хранящее ссылку на документ карты трансформатора в виде оцифрованного pdf или doc файла. Это поле необходимо для получения дополнительной информации о трансформаторе, не задействованной при поиске.
* Ссылка на схему: поле, хранящее ссылку на изображение схемы трансформатора в формате png.
* Примечание: поле, хранящее дополнительную информацию, не входящую в рамки описанных выше полей, однако важную для поиска трансформатора по его особенностям.
* Тип соединения: поле, актуальное только для трехфазных трансформаторов. Представляет краткую характеристику соединения, где буквой Д обозначается соединение типа «треугольник», У – типа «звезда». Для трехфазных трансформаторов стоит также учитывать различие в 1,73 раза между соединением "звезда" и "треугольник". То есть запись карты намотки трансформатора напряжением 380/380 типа "звезда"-"звезда" эквивалентна записи 380/220 для типа соединения "звезда"-"треугольник", а также 220/220 для типа "треугольник"-"треугольник".
* Единицы измерения мощности: по умолчанию мощность трансформаторов измеряется в кВа, однако иногда встречаются исключения с единицами измерения мощности кВт.

Следующей сущностью разрабатываемой системы является Сотрудник, реализующий поля для хранения данных о сотруднике: его имени, фамилии и отчестве, занимаемой им должности, что влияет на круг его обязанностей, как следствие и на функциональность системы, а также о его логине и пароле для входа в систему и получения прав доступа к определенным ее частям.

Извне в систему поступают заказы на производство трансформаторов в виде электронных таблиц формата exel, состоящие из двух колонок. В первой колонке представлены краткие описания требуемых трансформаторов по характеристикам, а во втором – количество в штуках. Этот объект предметной области реализует сущность Заказ с полями даты поступления заказа, его номера, перечня необходимых трансформаторов, а также их количества.

Наконец, для производства определенного трансформатора на производстве потребуется потратить определенное количество различных материалов. Виду их неоднородности и для их учета в системе реализована сущность Номенклатура, состоящая из полей Наименование, Характеристика, Значение характеристики и Единицы измерения. Так, например, описание кабеля диаметром 2 мм будет представлено в виде наименования Кабель, типа характеристики: Диаметр, значением 2 и единицами измерения в миллиметрах.

Как можно увидеть из схемы, сущности связывают следующие отношения:

* Один сотрудник может создать множество карт намотки, поэтому отношение Трансформатор-Сотрудник – многие к одному;
* Трансформатор может иметь большое число номенклатуры, но и одна и та же стандартная номенклатура требуется для создания множества трансформаторов, поэтому отношение Трансформатор-Номенклатура – многие ко многим;
* Один заказ может содержать множество трансформаторов, но и одна и та же карта намотки может оказаться нужной в различных заказах, поэтому отношение Трансформатор-Заказ – многие ко многим;
* Наконец, один сотрудник может создавать и редактировать множество номенклатурных единиц и принимать множество заказов, поэтому отношение Сотрудник-Номенклатура и Сотрудник-Заказ – многие к одному[4].

Представление данных предметной области можно увидеть на рисунке 2 в виде логической модели базы данных.

В перечень функций, подлежащих автоматизации, входят:

* функция добавления в базу данных карты намотки трансформатора с бумажного носителя с корректным заполнением всех ключевых полей записи в базе данных;
* функция автоматической разработки новых карт;
* поиск требующейся карты намотки по ключевым полям: типу трансформатора, мощности, первичному и вторичному напряжению, наличию или отсутствию экрана, наличию или отсутствию заявки на изготовление трансформаторов нового типа и проч.;
* функция автоматической обработки заказа и расчета требуемой на его выполнение номенклатуры[10].

# 2.3 Перечень задач, подлежащих решению в процессе разработки.

Задачи, подлежащие решению процессе разработки:

* Провести исследование предметной области;
* Сформировать концептуальную и логическую модели хранимых данных;
* Подготовить перечень необходимого функционала;
* Оформить описание процессов, протекающих в системе;
* Спроектировать интерфейс пользователей в зависимости от исполняемых ими обязанностей;
* Организовать структуру хранения данных и реализовать тестовое наполнение базы данных,
* Реализовать систему:
* Произвести тестирование и отладку приложения;

# Проектно-конструкторская часть

# 3.1 **Разработка структуры программной системы**

Рабочий процесс на предприятии начинается с поступления в систему заказа. Из заказа извлекается информация о параметрах трансформаторов, которые необходимо произвести. После этого в конструкторском отделе находят карты таких трансформаторов, если они когда-то уже были произведены, или разрабатывают новые карты, после чего карты направляются в технический отдел для расчета требуемой для производства номенклатуры. Как только в соответствии с нормами расхода рассчитывается потребность в той или иной номенклатуре, эта информация поступает в отдел закупок, если всего на складе не хватает материалов. Финальным шагом является передача всей накопленной информации в цех намотки[12].

Заказ поступает в систему в виде документа в формате exel, соответственно, первым шагом при обработке заказа будет извлечение данных о каждом трансформаторе. После того, как данные были извлечены, система производит поиск трансформаторов с такими параметрами в базе данных. По нахождению информация передается в технологический отдел для расчета номенклатуры. Если карту найти не удалось, предлагается разработать новую карту в полуавтоматическом режиме.

В технологическом отделе производится расчет потребности в номенклатуре, а также редактируются показатели нормы расхода материала или его характеристик, если это необходимо, после чего посчитанные карты передается в намоточный цех, автоматизация которого не входит в задачи этой курсовой работы[8].

Более подробно с процессами, протекающими в системе можно ознакомиться на листе 1 в виде диаграммы IDEF0, где в качестве точки зрения выбрана точка зрения директора предприятия.

IDEF0 - нотация графического моделирования, используемая для создания функциональной модели, отображающей структуру и функции системы, а также потоки информации и материальных объектов, связывающих эти функции. Стандарт IDEF0 (Integration Definition for Function Modeling) утвержден в США в 1993 как Федеральный стандарт обработки информации. В России находится в статусе руководящего документа с 2000 года и в настоящее время в качестве стандарта не утвержден. Тем не менее методология IDEF0 является одним из популярных подходов для описания бизнес-процессов. К ее особенностям можно отнести:

* использование контекстной диаграммы;
* поддержка декомпозиции;
* доминирование;
* выделение 4 типов стрелок[2].

# 3.2 Разработка интерфейса взаимодействия пользователя с системой

На рисунке 1 представлен прототип интерфейса окна поиска трансформатора по параметрам в ручном режиме. С левой стороны окна представлены поля для ввода характеристик, сверху предоставлена возможность фильтрации карт по дате создания для случаев, когда для каждой карты требуется выбрать наиболее актуальный аналог.

В центральной части окна в качестве рабочей области представлен перечень трансформаторов, имеющих требующиеся характеристики. Для подбора нескольких трансформаторов предусмотрены специальные поля. Если специалист видит в перечне карту трансформатора с требуемыми характеристиками, он выбирает его и нажимает на кнопку «Создать заказ».

При нажатии на кнопку «Отмена» процесс поиска трансформатора завершается.

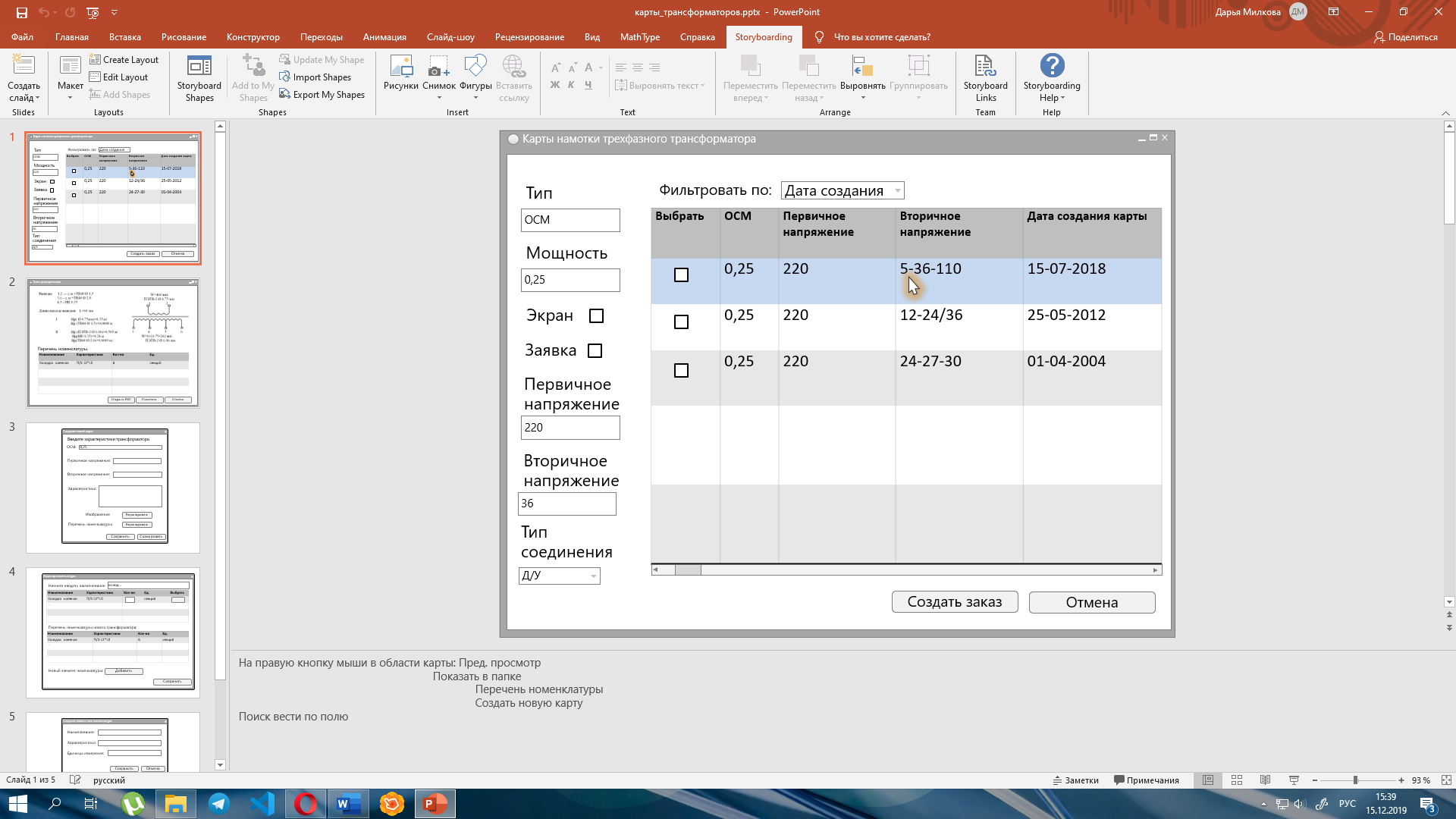


Рисунок 1 – Прототип интерфейса окна поиска карты

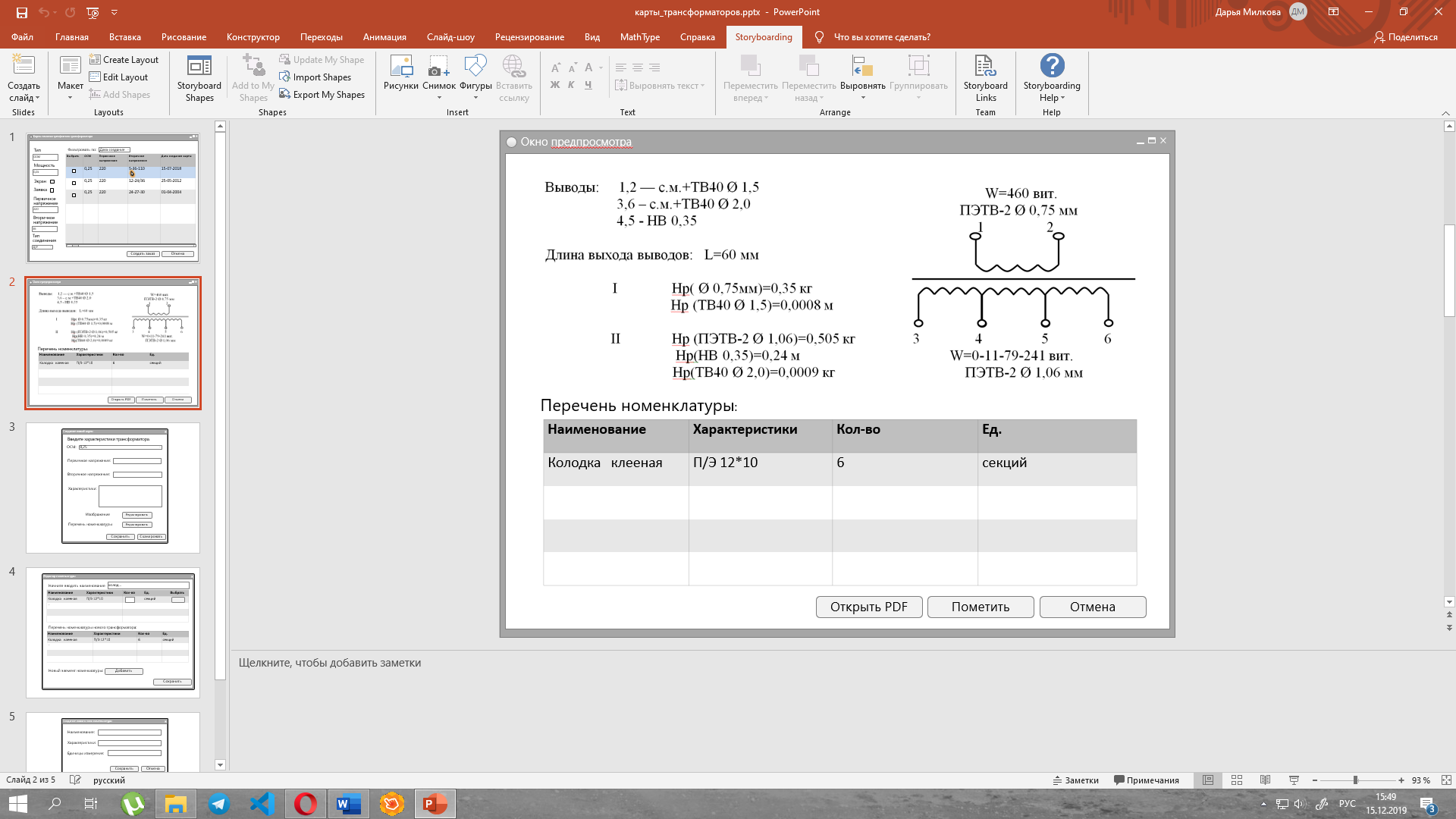


Рисунок 2 – Прототип окна просмотра карты

На рисунке 2 представлен прототип окна просмотра карты трансформатора. Предусмотрен вывод информации о количестве используемой номенклатуры. В правом верхнем углу окна программы располагается схема трансформатора.

По нажатию на кнопку «Открыть PDF» открывается оригинал карты, хранящийся на сервере предприятия. Карту можно пометить как требующую изменений, на удаление или при помещении в заказ. При нажатии на кнопку «Отмена» работа с этим окном прекращается.

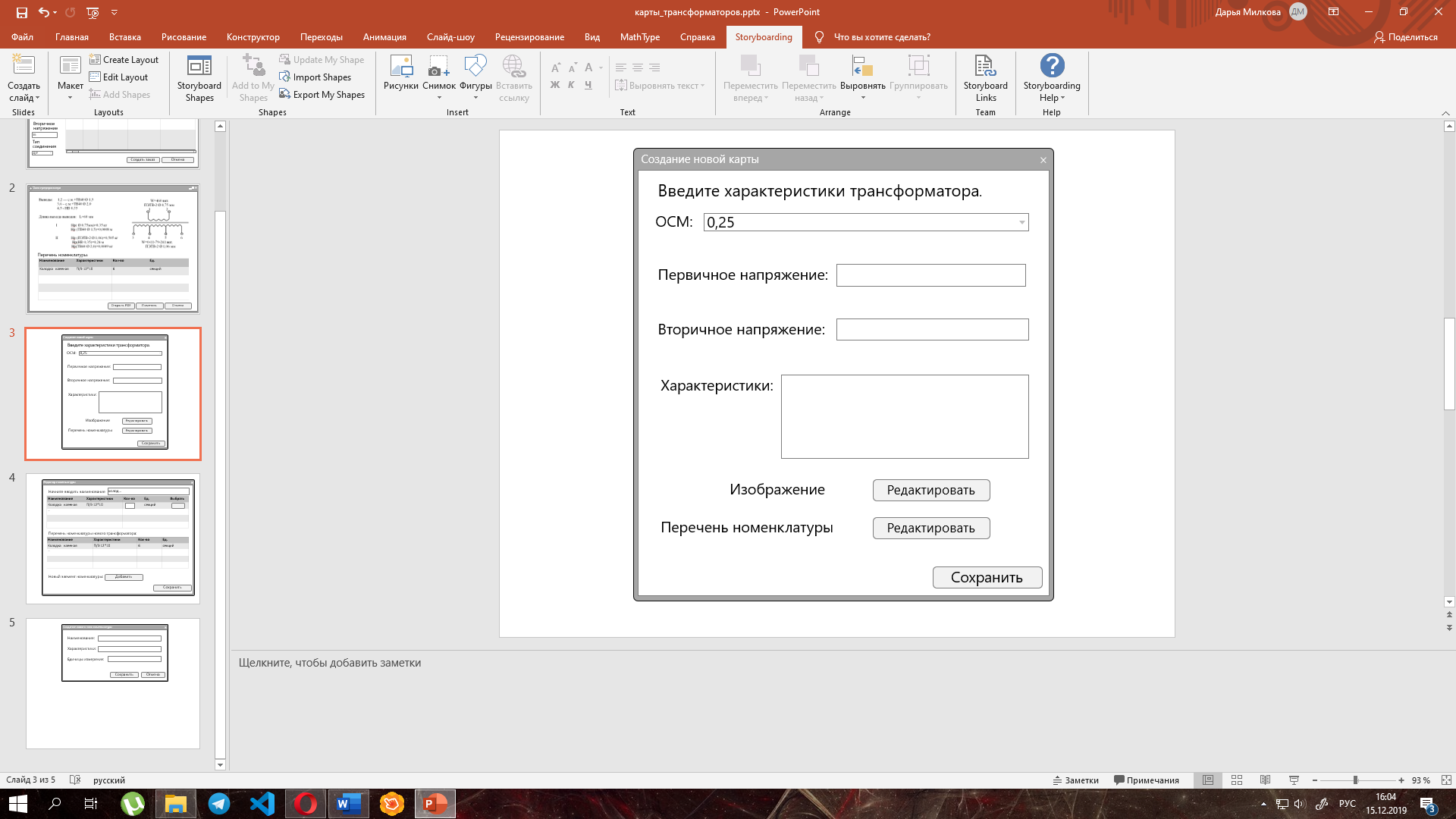


Рисунок 3 – Прототип окна создания новой карты

На рисунке 3 приведен прототип интерфейса окна создания новой карты намотки трансформатора. Когда невозможно подобрать трансформатор под потребности заказчика из существующей базы, необходимо разработать новую карту и поместить ее в систему. На форме представлены текстовые поля для заполнения характеристиками новой карты, а также две кнопки для привязки схемы трансформатора и списка требующейся номенклатуры. По нажатию на кнопку «Сохранить» изменения записываются в базу данных.

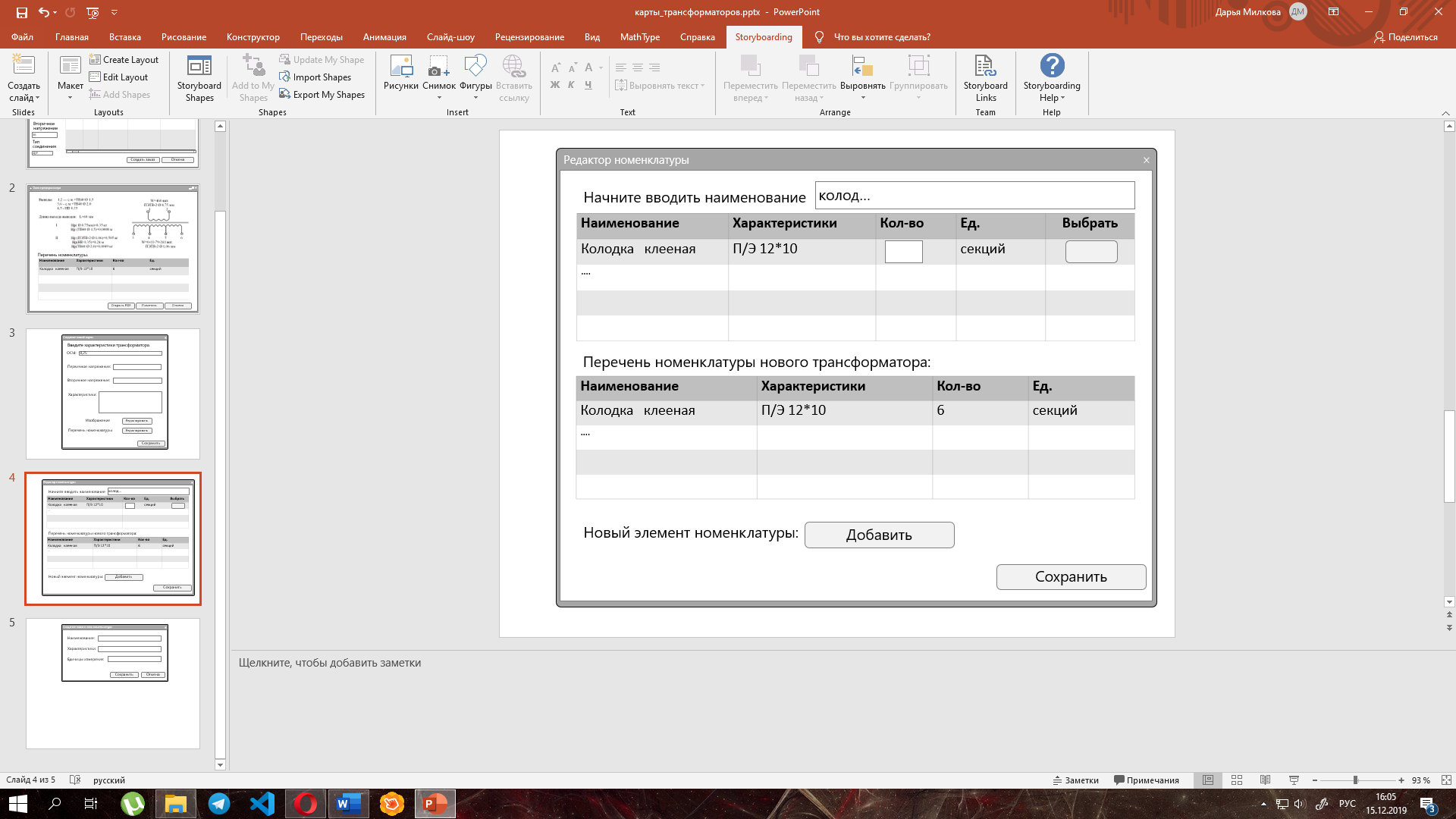


Рисунок 4 – Прототип окна технологического отдела

На рисунке 4 приведен прототип окна редактирования списка номенклатуры, требующейся для производства конкретного трансформатора. Сверху предусмотрено текстовое поле для поиска номенклатурной единицы в базе по параметрам. Результаты поиска отражаются в таблице ниже. Из этой таблицы можно выбрать нужную единицу, нажав на кнопку «Выбрать» в соответствующей строке. По нажатию на эту кнопку номенклатура переместится во вторую таблицу, что будет говорить, что для производства этого трансформатора необходимо именно эта номенклатурная единица в определенном количестве[6]. Если в производственный процесс вводится номенклатурная единица с новыми, ранее неизвестными параметрами, ее нужно ввести в систему с помощью кнопки «Добавить». По нажатию на эту кнопку отрывается окно добавления новой единицы номенклатуры, представленное на рисунке 5.

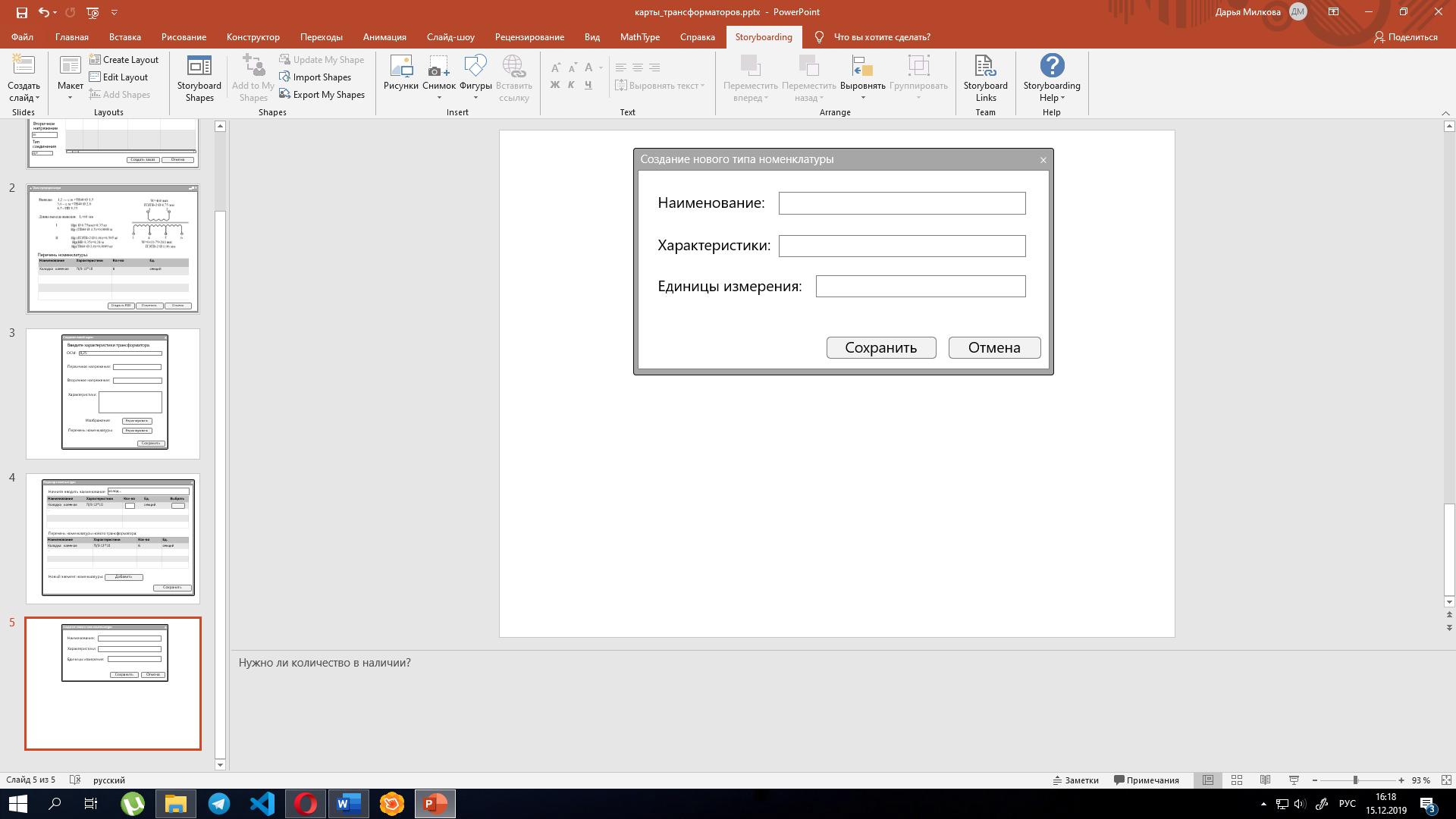


Рисунок 5 – Прототип окна новой единицы номенклатуры

Как можно заметить, это простое окно состоит из полей для ввода параметров и двух кнопок, отвечающих за запись информации в базу данных и за возможность выйти без сохранения изменений. Таким образом, здесь были приведены основные окна пользовательского интерфейса программы.

# 3.3 Разработка архитектуры приложения

Для реализации системы был выбран высокоуровневый язык программирования C#. Он является объектно-ориентированным и в этом плане много перенял у Java и С++. Например, 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. И C# продолжает активно развиваться, и с каждой новой версией появляется все больше интересных функциональностей, как, например, лямбды, динамическое связывание, асинхронные методы и т.д.[3]

Для работы с базой данных была выбрана СУБД PostgreSQL. Это программное обеспечение поставляется на условиях открытой лицензии и предоставляет требующиеся функции. Она хорошо задокументирована, имеет широкое комьюнити, а также является одним из самых распространенных решений для организации работы с реляционными базами данных за счет своей доступности и эффективности[1].

Ключевым моментом реализации является построение физической схемы базы данных, основанной на выбранной СУБД, логической модели данных и требованиях к архитектуре и быстродействию системы.

Приложение является законченным, то есть, не допускает ошибок времени исполнения, проходит все обозначенные в интерфейсе сценарии. Именно требование законченности приложения должно определять адекватный функционал на этапе проектирования системы[3].

# Проектно-технологическая часть

# 4.1 Проектирование начального и тестового наполнения базы данных

Начальное наполнение базы данных было получено из системы распознавания карт намотки трансформатора. Тестирование производилось на тестовом наборе данных.

В приложении 2 представлено тестовое наполнение таблицы карт намоток трансформаторов. Всего для тестирования были выбраны 5 карт, основанных на предоставленных предприятием аналогов. Физического смысла эти карты не имеют и были созданы в целях тестирования[4].

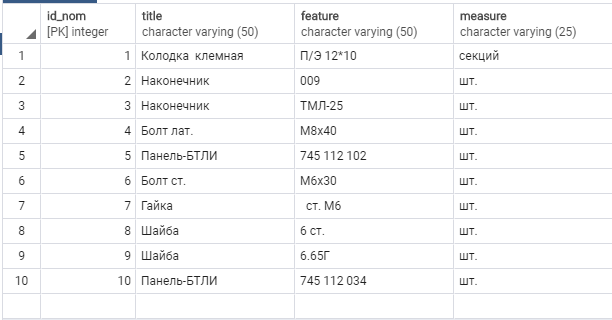


Рисунок 6 – Наполнение таблицы номенклатуры

Перечень номенклатуры, требующейся для создания данных 5 трансформаторов представлен на рисунке 6. На рисунке 7 представлено тестовое наполнение таблицы, связующей сущности карты намотки и номенклатурные единицы.

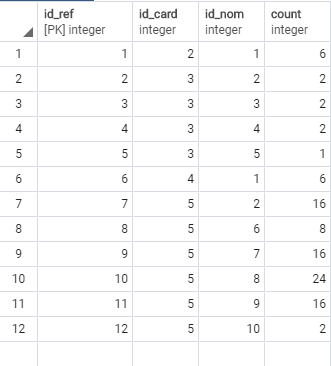


Рисунок 7 – Реализация связи между номенклатурой и картами

# 4.2 Технологические решения, поддерживающие эксплуатационный цикл программы

При проектировании системы было использовано архитектурное решение MVC совместно с инструментом построения интерфейсов JavaFX. Model-View-Controller (MVC, «Модель-Представление-Контроллер», «Модель-Вид-Контроллер») — схема разделения данных приложения, [пользовательского интерфейса](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81) и управляющей логики на три отдельных компонента: модель, представление и контроллер — таким образом, что модификация каждого компонента может осуществляться независимо[6].

*Модель* (*Model*) предоставляет данные и реагирует на команды контроллера, изменяя своё состояние.

*Представление* (*View*) отвечает за отображение данных модели пользователю, реагируя на изменения модели.

*Контроллер* (*Controller*) интерпретирует действия пользователя, оповещая модель о необходимости изменений.

Основная цель применения этой концепции состоит в отделении [бизнес-логики](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%BB%D0%BE%D0%B3%D0%B8%D0%BA%D0%B0) (модели) от её визуализации (представления, вида). За счёт такого разделения повышается возможность [повторного использования кода](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B2%D1%82%D0%BE%D1%80%D0%BD%D0%BE%D0%B5_%D0%B8%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%BA%D0%BE%D0%B4%D0%B0). Наиболее полезно применение данной концепции в тех случаях, когда пользователь должен видеть те же самые данные одновременно в различных контекстах и/или с различных точек зрения. В частности, выполняются следующие задачи[12].

К одной модели можно присоединить несколько видов, при этом не затрагивая реализацию модели. Например, некоторые данные могут быть одновременно представлены в виде [электронной таблицы](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%82%D0%B0%D0%B1%D0%BB%D0%B8%D1%86%D0%B0), [гистограммы](https://ru.wikipedia.org/wiki/%D0%93%D0%B8%D1%81%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0) и [круговой диаграммы](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D1%83%D0%B3%D0%BE%D0%B2%D0%B0%D1%8F_%D0%B4%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0).

Не затрагивая реализацию видов, можно изменить реакции на действия пользователя (нажатие мышью на кнопке, ввод данных) — для этого достаточно использовать другой контроллер.

Ряд разработчиков специализируется только в одной из областей: либо разрабатывают графический [интерфейс](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F), либо разрабатывают [бизнес-логику](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%BB%D0%BE%D0%B3%D0%B8%D0%BA%D0%B0). Поэтому возможно добиться того, что программисты, занимающиеся разработкой [бизнес-логики](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%BB%D0%BE%D0%B3%D0%B8%D0%BA%D0%B0) (модели), вообще не будут осведомлены о том, какое представление будет использоваться.

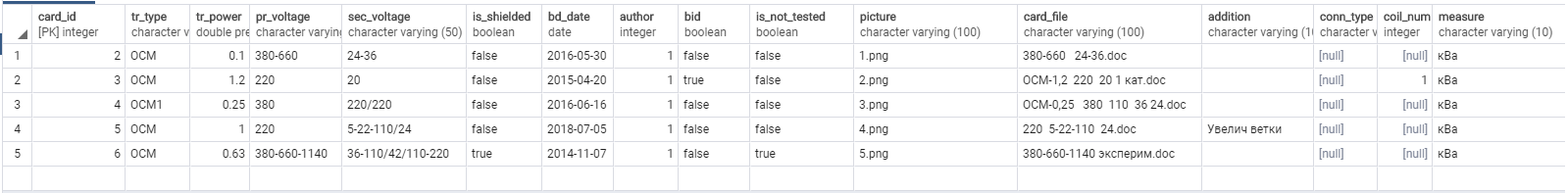
# Список литературы

1. Абрамов Г.В. Проектирование информационных систем : учебное пособие / Г.В. Абрамов, И.Е. Медведкова, Л.А. Коробова. - Воронеж : Воронежский государственный университет инженерных технологий, 2012,172 с. URL://biblioclub.ru/index.php?page=book&id=141626
2. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. - Приемы объектно-ориентированного проектирования. Паттерны проектирования. — СПб: Питер, 2001. — 368 с.: ил. (Серия «Библиотека программиста») ISBN 5-272-00355-1
3. Золотов С.Ю. Проектирование информационных систем: учебное пособие/ ЗолотовС.Ю.— Томск: Томский государственный университет систем управления и радиоэлектроники, Эль Контент, 2013.— 88 c.— URL: http://www.iprbookshop.ru/13965
4. Назаров С.В. - Архитектура и проектирование программных систем: моногра­фия /С.В. Назаров. — 2-е изд., перераб. и доп. — М : ИНФРА-М, 2016. — 374 с. — (Научная мысль). — [www.dx.doi.org/](http://www.dx.doi.org/) 10.12737/18292.ISBN 978-5-16-011753-9 (print)
5. Рак, И.П. Основы разработки информационных систем : учебное пособие / И.П. Рак, А.В. Платёнкин, А.В. Терехов ; Издательство ФГБОУ ВПО «ТГТУ», 2017. – 99 с. : ил. – Режим доступа: по подписке. – URL: <http://biblioclub.ru/index.php?page=book&id=499041>. – Библиогр.: с. 85. – ISBN 978-5-8265-1727-7. – Текст : электронный.
6. Рудаков А. В. 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»; 2014. — 192 с. ISBN 978-5-4468-0465-8
7. Рудинский И.Д. Технология проектирования автоматизированных систем обработки информации и управления. — М. : Горячая линия-Телеком, 2011. — 304 с. — URL: http://e.lanbook.com/book/5191
8. Рыбальченко, М.В. Архитектура информационных систем : учебное пособие / М.В. Рыбальченко ; Министерство образования и науки Российской Федерации, Южный федеральный университет. – Таганрог : Издательство Южного федерального университета, 2015. – Ч. 1. – 92 с. – Режим доступа: по подписке. – URL: <http://biblioclub.ru/index.php?page=book&id=462011> . – Библиогр. в кн. – ISBN 978-5-9275-1765-7. – Текст : электронный.
9. Сидоров В.Н., Сломинская Е.Н., Полникова Т.В., Макарова О.Ю. Оформление графической части выпускной квалификационной работы. Учебное пособие. М.: МГТУ им. Н.Э. Баумана, 2016.
10. Сырецкий Г.А. Проектирование автоматизированных систем. Часть 1: учебное пособие/Сырецкий Г.А.— Новосибирск: Новосибирский государственный технический университет, 2014.— 156 c.— URL: http://www.iprbookshop.ru/47714
11. Тихомиров П.М. - Расчет трансформаторов: Учеб, пособие для ву­зов.— 5-е изд., перераб. и доп. — М.: Энергоатом- издат, 1986. — 528 с.: ил.
12. Форд Н., Хайгард М., де Ора Б. - 97 этюдов для архитекторов программных систем. - Пер. с англ. - СПб. Символ-Плюс, 2010. - 224 с., ил. ISBN 978-5-93286-176-9

|  |
| --- |
| ***ОСМ– 0,25 220/110/36/22/5***    W=460 вит.  ПЭТВ-2 Ø 0,75 мм  1 2    3 4 5 6 7 8 9 10  W=241 вит. W=79 вит. W=48 вит. W=11 вит.  ПЭТВ-2 Ø 0,8 мм        Выводы: – : 1,2- с.м.+ТВ40 Ø1,5 мм  3-10- с.м.+ТВ40 Ø 1,5 мм      Длина выхода выводов: L=60 мм    I Нр(ПЭТВ-2 Ø 0,75мм)=0,35 кг  Нр(ТВ40 Ø 1,5)=0,0008 кг    II Нр (ПЭТВ-2 Ø 0,8 мм)=0,5 кг  Нр(ТВ40 Ø 1,5)=0,0032 кг  Колодка клееная П/Э 12\*10- 10 секций |

# Приложение 1 – Карта трансформатора

# Приложение 2 – Тестовое наполнение таблицы карт

****