Министерство науки и высшего ОБРАЗОВАНИЯ РОССИЙСКОЙ федерации

федеральное государственное бюджетное образование учреждение

высшего образования

«Курганский государственный университет»

Кафедра «Программное обеспечение автоматизированных систем»

**Агрегатор онлайн-публикаций**

**ОТЧЕТ**

**по учебной практике**

Разработал студент гр. ИТ-0930222 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / С.Ю. Чернов /

подпись, дата

Направление 09.03.03 - «Прикладная информатика»

Руководитель, ассистент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Н.В. Сошников /

подпись, дата

Курган, 2024

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc167479555)

[1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 4](#_Toc167479556)

[1.1 Характеристика предметной области 4](#_Toc167479557)

[1.2 Спецификация требований к программному обеспечению 6](#_Toc167479558)

[2 ПРОЕКТИРОВАНИЕ СИСТЕМЫ 7](#_Toc167479559)

[2.1 MVVM - архитектура 7](#_Toc167479560)

[2.2 Диаграмма пакетов классов 8](#_Toc167479561)

[2.3 Диаграмма классов 9](#_Toc167479562)

[2.4 Структура таблицы для сохранения данных 18](#_Toc167479563)

[3 ОБОСНОВАНИЕ ВЫБОРА СРЕДСТВ РЕАЛИЗАЦИИ 21](#_Toc167479564)

[3.1 Выбор средств реализации 21](#_Toc167479565)

[ЗАКЛЮЧЕНИЕ 23](#_Toc167479566)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 24](#_Toc167479567)

# ВВЕДЕНИЕ

На любом производстве есть потребность в контроле и управлении запасами материалов, товаров, затрат и доходов, а также их анализе.

Данная работа посвящена разработки автоматизированной информационной системы (АИС) бухгалтерского и складского учета на производствах. Предприятия имеют свой локальный склад для хранения материалов и произведенных из них товаров. Помимо хранения, данные организации занимаются продажей произведенных потребительских или промышленных товаров, произведенных на станочном оборудовании. Примером такого производства может быть инженерная мастерская.

Внедрение автоматизированной информационной системы приведет к решению следующих проблем:

* Контроль запасов материалов и готовой продукции.
* Контроль затрат и доходов.
* Анализ и прогнозирование денежных потоков.

Целью работы является повышение удобства и структурированное руководство складской части производства.

Для достижения поставленной цели были поставлены следующие задачи:

1) Изучить предметную область АИС.

2) Определить требования к АИС.

3) Разработать архитектуру.

4) Спроектировать структуру хранения данных.

5) Обосновать выбор средств реализации.

6) Программно реализовать систему.

# 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1 Характеристика предметной области

Основное назначение АИС состоит в формировании купленных материалов и контроле уже имеющихся на складе, формировании на их основе товаров и себестоимости товаров с дальнейшим анализом затраченных и пришедших денежных средств.

На ***первом этапе*** идет утверждение материальной накладной - первичного документа по отпуску продукции для оформления операций по приему материалов. Она включает в себя номенклатуру - перечень групп материалов схожего или единого происхождения.

Затем материалы проверяются и отправляется на склад, где наступает ***второй этап***.

При приходе материалов они учитываются по нескольким критериям:

* название,
* тип (каким оборудованием будет обрабатываться),
* цена,
* основные свойства (цвет, количество в мере измерения),
* особые свойства (частные, узконаправленные характеристики).

Все материалы вносятся в хранилище данных. А затем сортируется на складском помещении по типу материала и его имени.

Склад — это помещение, содержащее ограниченное число товаров. Его вместимость определяется площадью и количеством груза находящимся в нем. Хранилище делиться на зоны размещения продукции, например, А - теплую и B - холодную. В них находятся пронумерованные стеллажи, каждый из которых отвечает за свою группу продуктов (по видам) в соответствии с товарным соседством. Сортировка проводится в зависимости от наименования (А-Я).

Затем наступает ***третий этап***, который заключается в переносе необходимых материалов.

При производстве товаров в цехе или мастерской необходимые материалы поставляются из основного склада в зону используемых материалов. В этой зоне расположены материалы, которые применялись хоть раз, например отрезан кусок листового материала.

После идет ***четвертый этап*** – изготовление товара. Из используемых материалов формируется товар по определенному техническому процессу. Материалы, которые не подлежат дальнейшему использованию идут на списание.

Затем идет ***пятый этап*** – внесение в хранилище данных информации о товаре:

* название,
* описание,
* составные материалы и их количество.

Из количества использованных материалов формируется себестоимость товара.

***Шестой этап*** – перенос готового изделия к месту хранения на складе, где он будет храниться до продажи.

После продажи товара в бухгалтерскую систему вносятся поступление, дата сделки и описание.

Из вышесказанного можно выделить роли пользователей системы и их функции:

1. Кладовщик работает принимает новые материалы и распределяет их на складе, при необходимости переносит нужные материалы в зону используемых материалов. Списывает не подлежащие дальнейшему использованию материалы.
2. Мастер изготавливает товар из материалов и отдает его кладовщику для хранения.
3. Специалист по закупкам и продажам, соответственно работает с закупками материалов и продажей товаров.

## 1.2 Спецификация требований к программному обеспечению

Основные функции автоматизированной системы:

1. Формирование материала;
2. Управление каталогом материалов, с возможностью перемещения материалов в зону используемых материалов;
3. Возможность списания используемого материала;
4. Формирование товара из используемых материалов;
5. Формирование товарной карты (представление информации о товаре);
6. Формирование себестоимости товара;
7. Управление каталогом товаров, с возможностью продажи;
8. Учет списаний и поступлений;
9. Анализ денежных потоков с предсказанием их дальнейшего движения.

# 2 ПРОЕКТИРОВАНИЕ СИСТЕМЫ

## 2.1 MVVM - архитектура

Для организации работы приложения использовалась MVVM-архитектура (Рисунок 2.1). MVVM — это паттерн архитектуры, который разделяет приложение на три ключевых компонента: Model, View и ViewModel.

* Model: это бизнес-логика и данные. Здесь обрабатываются данные, выполняются операции и происходит взаимодействие с источниками данных.
* View: это пользовательский интерфейс (UI) приложения. View ответственен за отображение данных и реагирование на взаимодействие пользователя.
* ViewModel: ViewModel является посредником между Model и View. Он содержит логику, связанную с UI, и управляет состоянием приложения. ViewModel предоставляет данные, которые View отображает, и обрабатывает действия пользователя.

Преимущества выбора данной архитектуры:

1. Разделение логики и представления. MVVM четко разделяет представление (View), бизнес-логику (ViewModel) и данные (Model). Это позволяет поддерживать код в чистоте и лучше организовать проект.
2. Уменьшение кода. Большая часть логики обработки данных выносится в ViewModel, что уменьшает объем кода, который нужно писать в коде представления.
3. Поддержка повторного использования компонентов. ViewModel можно легко переиспользовать в разных частях приложения или даже в разных проектах, поскольку они не зависят от конкретного представления.
4. Гибкость. Отделение ViewModel от View позволяет легко изменять или обновлять представление без изменения логики приложения.
5. Модульность. Каждый компонент (Model, ViewModel, View) разрабатывается отдельно, что делает их более модульными и менее взаимозависимыми [1].

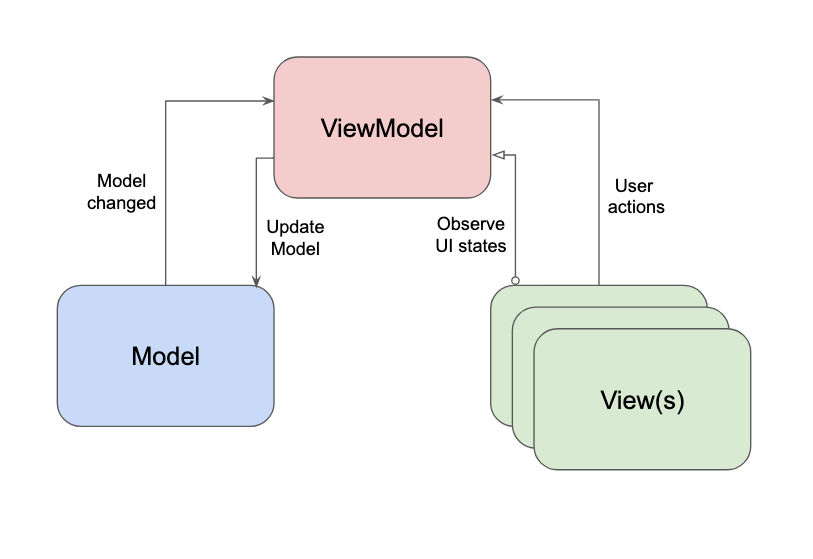


Рисунок 2.1 - Диаграмма архитектуры – MVVM

## 2.2 Диаграмма пакетов классов

На рисунке 2.2 представлена диаграмма пакетов:

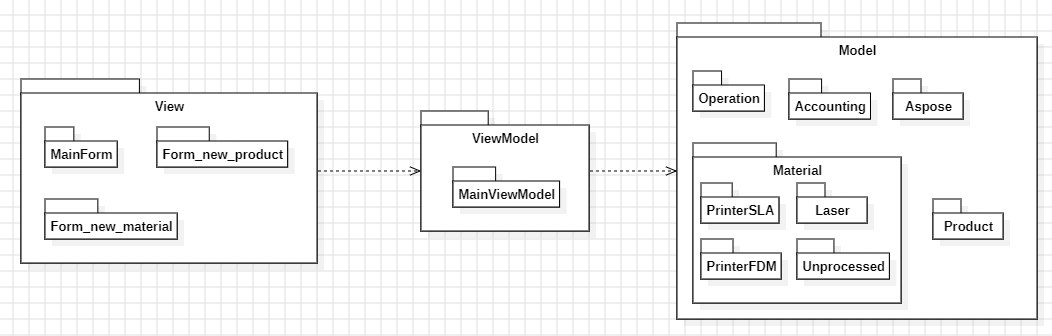


Рисунок 2.2 - Диаграмма пакетов

Диаграмма пакетов включает в себя:

1. Model – классы, содержащие данные и методы по их использованию;
2. ViewModel – классы связи Model с представлением View;
3. View – классы, предоставляющие формы для отображения данных пользователю.

## 2.3 Диаграмма классов

На рисунке 2.3 представлена диаграмма классов.

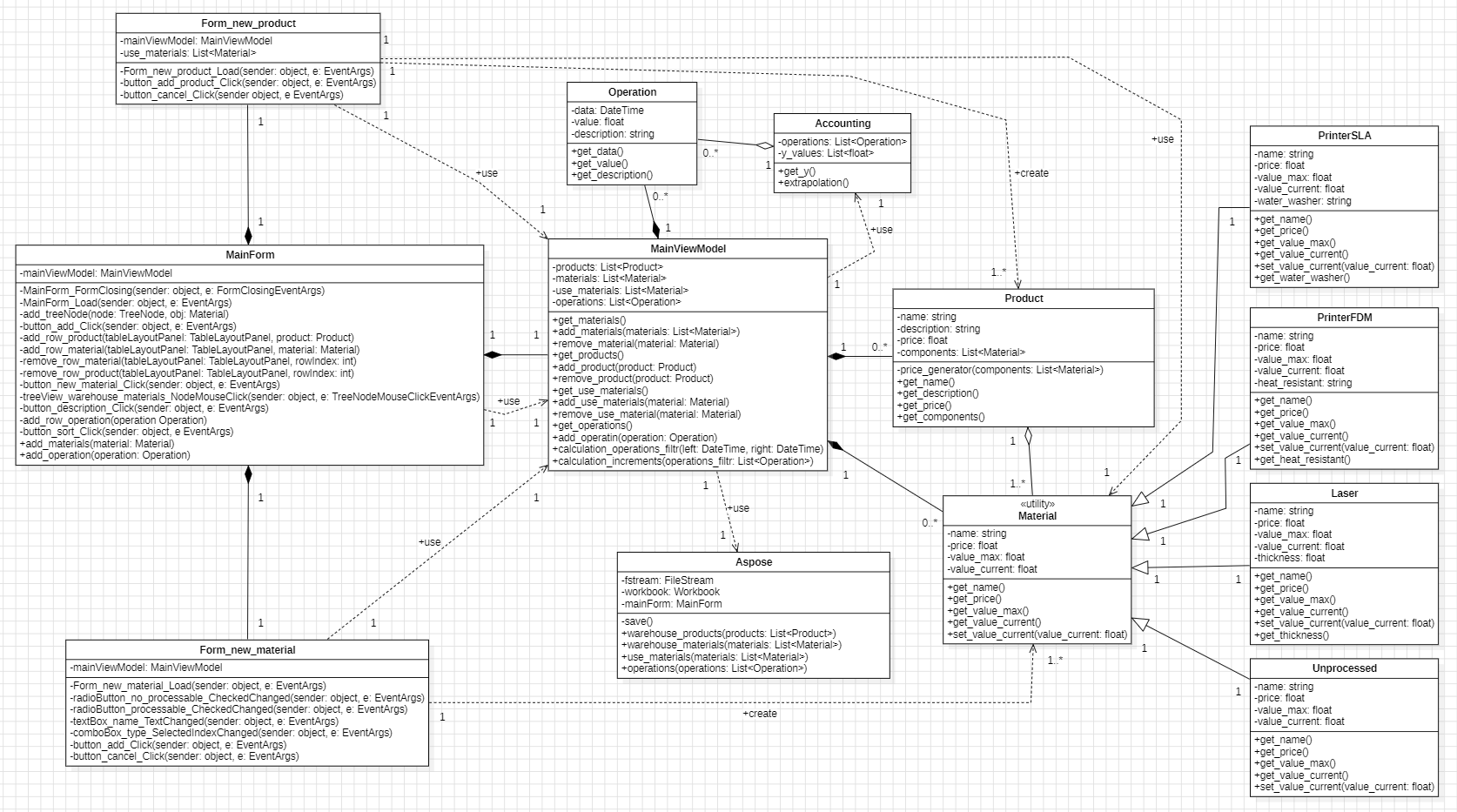


Рисунок 2.3 – Диаграмма классов

Класс MainForm – главная форма приложения, на которой отображаются основная информация о товарах и материалах, а также анализе денежных потоков.

Класс Form\_new\_material – форма, предназначенная для добавления новых материалов.

Класс Form\_new\_product – форма, предназначенная для создания новых товаров, на основе используемых материалов.

Класс MainViewModel – модель представления, к которой могут обращаться формы для предоставления данных. В таблице 2.1 представлены поля класса. В таблице 2.2 представлены методы класса.

Таблица 2.1 – поля класса MainViewModel

|  |  |
| --- | --- |
| Поле | Описание |
| products: List<Product> | Лист, хранящий все материалы, имеющиеся на складе. |
| Materials: List<Material> | Лист, хранящий все товары, имеющиеся на складе. |
| Use\_materials: List<Material> | Лист, хранящий все используемые материалы, имеющиеся на складе. |
| Operations: List<Operation> | Лист, хранящий историю всех операций поступлений и списаний. |

Таблица 2.2 – методы класса MainViewModel

|  |  |
| --- | --- |
| Метод | Описание |
| get\_materials() | Возвращает лист всех материалов, имеющихся на складе. |
| add\_materials(materials:List<Material>) | Добавляет в конец листа материалов новый материал. |
| remove\_material(material:Material) | Удаляет из листа материалов выбранный материал. |
| get\_products() | Возвращает лист всех товаров, имеющихся на складе. |
| add\_product(product:Product) | Добавляет в конец листа товаров новый товар. |
| remove\_product(product:Product) | Удаляет из листа товаров выбранный товар. |
| get\_use\_materials() | Возвращает лист всех используемых материалов, имеющихся на складе. |
| add\_use\_materials(material:Material) | Добавляет в конец листа используемых материалов новый материал. |
| remove\_use\_material(material:Material) | Удаляет из листа используемых материалов выбранный материал. |
| get\_operations() | Возвращает лист всех операций поступлений и списаний. |
| add\_operatin(operation:Operation) | Добавляет в конец листа всех операций поступлений и списаний новую операцию. |
| calculation\_operations\_filtr(left:DateTime, right:DateTime) | Возвращает лист операций поступлений и списаний, отсортированный по начальной и конечной дате. |
| calculation\_increments(operations\_filtr:  List<Operation>) | Возвращает лист выборки приращений, состоящий из операций поступлений и списаний, и предсказанного значения, записанного последним элементом листа. |

*Окончание таблицы 2.2*

Класс Aspose – класс, реализующий процесс сохранения объектов всех бизнесовых классов и их выгрузку при запуске приложения. В таблице 2.3 представлены поля класса. В таблице 2.4 представлены методы класса.

Таблица 2.3 – поля класса Aspose

|  |  |
| --- | --- |
| Поля | Описание |
| fstream: FileStream | Поток для чтения данных из таблицы. |
| workbook: Workbook | Рабочая книга из таблицы. |
| mainViewModel: MainViewModel | Ссылка на модель представления для доступа к данным. |

*Окончание таблицы 2.3*

Таблица 2.4 – методы класса Aspose

|  |  |
| --- | --- |
| Методы | Описание |
| save() | Сохраняет все листы с экземплярами классов в таблицу Excel. |
| warehouse\_products(products:List<Product>) | Считывает данные из таблицы Excel для построения экземпляров класса товаров, находящихся на складе. Возвращает лист с экземплярами класса. |
| warehouse\_materials(materials:List<Material>) | Считывает данные из таблицы Excel для построения экземпляров класса материалов, находящихся на складе. Возвращает лист с экземплярами класса. |
| use\_materials(materials:List<Material>) | Считывает данные из таблицы Excel для построения экземпляров класса используемых материалов, находящихся на складе. Возвращает лист с экземплярами класса. |
| operations(operations:List<Operation>) | Считывает данные из таблицы Excel для построения экземпляров класса операций поступлений и списаний. Возвращает лист с экземплярами класса. |

Класс Operation – класс, предназначенный для хранения информации о поступлении или списании. В таблице 2.5 представлены поля класса. В таблице 2.6 представлены методы класса.

Таблица 2.5 – поля класса Operation

|  |  |
| --- | --- |
| Поля | Описание |
| data: DateTime | Дата совершения операции. |
| value: float | Денежная сумма операции. |
| description: string | Описание совершенной операции. |

Таблица 2.6 – методы класса Operation

|  |  |
| --- | --- |
| Методы | Описание |
| get\_data() | Возвращает дату совершения операции. |
| get\_value() | Возвращает денежную сумму операции. |
| get\_description() | Возвращает описание совершенной операции. |

Класс Accounting – класс, предназначенный для получения выборки значений для построения графиков денежных потоков, вычисления будущего движения денежных средств. В таблице 2.7 представлены поля класса. В таблице 2.8 представлены методы класса.

Таблица 2.7 – поля класса Accounting

|  |  |
| --- | --- |
| Поля | Описание |
| operations: List<Operation> | Лист совершенных операций. |
| y\_values: List<float> | Лист значений выборки по оси ординат. |

Таблица 2.8 – методы класса Accounting

|  |  |
| --- | --- |
| Методы | Описание |
| get\_y() | Возвращает лист совершенных операций. |
| extrapolation() | Возвращает предсказанное значение будущего движения денежных средств, используя метод экстраполяции исходной выборки. |

Абстрактный класс Material – общий класс для всех видов материалов, хранящий основную информацию о них. В таблице 2.9 представлены поля класса. В таблице 2.10 представлены методы класса.

Таблица 2.9 – поля класса Material

|  |  |
| --- | --- |
| Поля | Описание |
| name: string | Название материала. |
| price: float | Цена материала. |
| value\_max: float | Исходное значение единиц измерения материала. |
| value\_current: float | Текущее значение единиц измерения материала. |

Таблица 2.10 – методы класса Material

|  |  |
| --- | --- |
| Методы | Описание |
| get\_name() | Возвращает название материала. |
| get\_price() | Возвращает цену материала. |
| get\_value\_max() | Возвращает исходное значение единиц измерения материала. |
| get\_value\_current() | Возвращает текущее значение единиц измерения материала. |
| set\_value\_current(value\_current:float) | Задает новое значение текущего значения единиц измерения материала. |

Класс Unprocessed – класс материалов, не подлежащих обработке. В таблице 2.11 представлены поля класса. В таблице 2.12 представлены методы класса.

Таблица 2.11 – поля класса Unprocessed

|  |  |
| --- | --- |
| Поля | Описание |
| name: string | Название материала. |
| price: float | Цена материала. |
| value\_max: float | Исходное значение единиц измерения материала. Во всех случаях задается как 1. |
| value\_current: float | Текущее значение единиц измерения материала. Во всех случаях задается как 1 или 0. |

*Окончание таблицы 2.11*

Таблица 2.12 – методы класса Unprocessed

|  |  |
| --- | --- |
| Методы | Описание |
| get\_name() | Возвращает название материала. |
| get\_price() | Возвращает цену материала. |
| get\_value\_max() | Возвращает исходное значение единиц измерения материала. |
| get\_value\_current() | Возвращает текущее значение единиц измерения материала. |
| set\_value\_current(value\_current:float) | Задает новое значение текущего значения единиц измерения материала. |

Класс Laser – класс материалов, обрабатываемых лазерным станком. В таблице 2.13 представлены поля класса. В таблице 2.14 представлены методы класса.

Таблица 2.13 – поля класса Laser

|  |  |
| --- | --- |
| Поля | Описание |
| name: string | Название материала. |
| price: float | Цена материала. |
| value\_max: float | Исходное значение единиц измерения материала в . |
| value\_current: float | Текущее значение единиц измерения материала в . |
| thickness: float | Ширина раскраиваемого материала в мм. |

*Окончание таблицы 2.13*

Таблица 2.14 – методы класса Laser

|  |  |
| --- | --- |
| Методы | Описание |
| get\_name() | Возвращает название материала. |
| get\_price() | Возвращает цену материала. |
| get\_value\_max() | Возвращает исходное значение единиц измерения материала. |
| get\_value\_current() | Возвращает текущее значение единиц измерения материала. |
| set\_value\_current(value\_current:float) | Задает новое значение текущего значения единиц измерения материала. |
| get\_thickness() | Возвращает ширину раскраиваемого материала. |

PrinterFDM – класс материалов, обрабатываемых филаментным 3-D принтером. В таблице 2.15 представлены поля класса. В таблице 2.16 представлены методы класса.

Таблица 2.15 – поля класса PrinterFDM

|  |  |
| --- | --- |
| Поля | Описание |
| name: string | Название материала. |
| price: float | Цена материала. |
| value\_max: float | Исходное значение единиц измерения материала в граммах. |
| value\_current: float | Текущее значение единиц измерения материала в граммах. |
| heat\_resistant: string | Является ли материал термостойким. |

Таблица 2.16 – методы класса PrinterFDM

|  |  |
| --- | --- |
| Методы | Описание |
| get\_name() | Возвращает название материала. |
| get\_price() | Возвращает цену материала. |
| get\_value\_max() | Возвращает исходное значение единиц измерения материала. |
| get\_value\_current() | Возвращает текущее значение единиц измерения материала. |
| set\_value\_current(value\_current:float) | Задает новое значение текущего значения единиц измерения материала. |
| get\_heat\_resistant() | Возвращает значение, обозначающее, является ли материал термостойким. |

PrinterSLA - класс материалов, обрабатываемых фотополимерным 3-D принтером. В таблице 2.17 представлены поля класса. В таблице 2.18 представлены методы класса.

Таблица 2.17 – поля класса PrinterSLA

|  |  |
| --- | --- |
| Поля | Описание |
| name: string | Название материала. |
| price: float | Цена материала. |
| value\_max: float | Исходное значение единиц измерения материала в мл. |
| value\_current: float | Текущее значение единиц измерения материала в мл. |
| water\_washer: string | Является ли материал растворимым в воде. |

Таблица 2.18 – методы класса PrinterSLA

|  |  |
| --- | --- |
| Методы | Описание |
| get\_name() | Возвращает название материала. |
| get\_price() | Возвращает цену материала. |
| get\_value\_max() | Возвращает исходное значение единиц измерения материала. |
| get\_value\_current() | Возвращает текущее значение единиц измерения материала. |
| set\_value\_current(value\_current:float) | Задает новое значение текущего значения единиц измерения материала. |
| get\_water\_washer() | Возвращает значение, обозначающее, является ли материал растворимым в воде. |

*Окончание таблицы 2.18*

## 2.4 Структура таблицы для сохранения данных

Лист «WarehouseProducts» предназначен для хранения информации о всех товарах на складе. Пример заполнения листа представлен на рисунке 2.4.



Рисунок 2.4 – Пример заполнения полей листа «WarehouseProducts»

В колонке «Товар» вносится название товара, в колонку «Описание» - описание товара.

Затем с новой строки содержатся все составные материалы товара. Знак «\*» в колонке «Товар» информирует, что в этой строке записан материал. Знак «\*\*» информирует о том, что это последний материал. В колонке «Материал» содержится название материала. В колонке «Тип» содержится название класса материала. В колонке «Цена» содержится цена материала. В колонке «Особенность» содержится информация, характеризующая особые свойства материала. В колонке «Мера измерения» содержится число исходных единиц измерения материала. В колонке «Текущее значение» содержится информация о текущем количестве единиц измерения материала.

Лист «WarehouseMaterials» предназначен для хранения информации о всех материалах на складе. Пример заполнения листа представлен на рисунке 2.5.

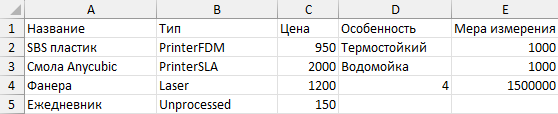


Рисунок 2.5 – Пример заполнения полей листа «WarehouseMaterials»

В колонке «Название» содержится название материала. В колонке «Тип» содержится название класса материала. В колонке «Цена» содержится цена материала. В колонке «Особенность» содержится информация, характеризующая особые свойства материала. В колонке «Мера измерения» содержится число исходных единиц измерения материала.

Лист «UseMaterials» предназначен для хранения информации о всех используемых материалах на складе. Пример заполнения листа представлен на рисунке 2.6.

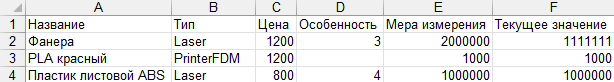


Рисунок 2.6 – Пример заполнения полей листа «UseMaterials»

В колонке «Название» содержится название материала. В колонке «Тип» содержится название класса материала. В колонке «Цена» содержится цена материала. В колонке «Особенность» содержится информация, характеризующая особые свойства материала. В колонке «Мера измерения» содержится число исходных единиц измерения материала. В колонке «Текущее значение» содержится информация о текущем количестве единиц измерения материала.

Лист «Accounting» предназначен для хранения информации о всех операциях поступлений и списаний. Пример заполнения листа представлен на рисунке 2.7.

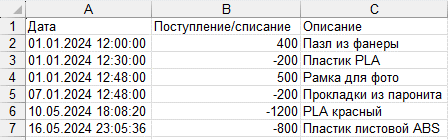


Рисунок 2.7 – Пример заполнения полей листа «Accounting»

В колонке «Дата» содержится дата совершенной операции. В колонке «Поступление/списание» содержится сумма операции. В колонке «Описание» содержится описание операции.

# 3 ОБОСНОВАНИЕ ВЫБОРА СРЕДСТВ РЕАЛИЗАЦИИ

## 3.1 Выбор средств реализации

В качестве средств реализации клиент-серверного приложения были выбраны следующие инструменты:

* Язык программирования С#, .NET Framework и Windows Forms.
* Microsoft Excel в качестве хранения и доступа к данным.

C# — это кроссплатформенный язык общего назначения, который делает разработчиков продуктивным при написании высокопроизводительного кода. С миллионами разработчиков C# является самым популярным языком .NET. C# имеет широкую поддержку в экосистеме и всех рабочих нагрузок .NET. На основе объектно-ориентированных принципов он включает множество функций из других парадигм, а не наименее функционального программирования. Низкоуровневые функции поддерживают сценарии высокой эффективности без написания небезопасного кода. Большая часть среды выполнения и библиотек .NET написана на C#, и прогресс в C# часто используется для всех разработчиков .NET.

Платформа .NET Framework — это технология, которая поддерживает создание и выполнение веб-служб и приложений Windows. Платформа .NET Framework состоит из общеязыковой среды выполнения (среды CLR) и библиотеки классов .NET Framework. Основой платформы .NET Framework является среда CLR. Среду выполнения можно считать агентом, который управляет кодом во время выполнения и предоставляет основные службы, такие как управление памятью, управление потоками и удаленное взаимодействие. При этом средой накладываются условия строгой типизации и другие виды проверки точности кода, обеспечивающие безопасность и надежность. Фактически основной задачей среды выполнения является управление кодом. Код, который обращается к среде выполнения, называют управляемым кодом, а код, который не обращается к среде выполнения, называют неуправляемым кодом. Библиотека классов является комплексной объектно-ориентированной коллекцией повторно используемых типов, которые применяются для разработки приложений — начиная с обычных приложений, запускаемых из командной строки, и приложений с графическим интерфейсом (GUI) и заканчивая приложениями, использующими последние технологические возможности ASP.NET, такие как веб-формы и веб-службы XML.

Windows Forms — это платформа пользовательского интерфейса для создания классических приложений Windows. Она обеспечивает один из самых эффективных способов создания классических приложений с помощью визуального конструктора в Visual Studio. Такие функции, как размещение визуальных элементов управления путем перетаскивания, упрощают создание классических приложений.

# ЗАКЛЮЧЕНИЕ

Во время прохождения практики были выполнены следующие задачи:

* 1. Проведен анализ предметной области.
  2. Разработана спецификация требований к системе.
  3. Создана архитектура системы.
  4. Спроектирована структура таблицы для сохранения данных.
  5. Описан выбор средств реализации.
  6. Реализованы классы программы.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Простая архитектура с использованием MVVM [Электронный ресурс] / Хабр : сайт. Электрон. текстовые дан. – Режим доступа:

<https://habr.com/ru/articles/776344/>, свободный - Загл. с экрана.

1. Краткий обзор языка C# [Электронный ресурс] / Microsoft : сайт. Электрон. текстовые дан. – Режим доступа:

<https://learn.microsoft.com/ru-ru/dotnet/csharp/tour-of-csharp/overview>, свободный - Загл. с экрана.

1. Общие сведения о платформе .NET [Электронный ресурс] / Microsoft : сайт. Электрон. текстовые дан. – Режим доступа:

<https://learn.microsoft.com/ru-ru/dotnet/framework/get-started/overview>, свободный - Загл. с экрана.

1. Общие сведения о платформе .NET [Электронный ресурс] / Microsoft : сайт. Электрон. текстовые дан. – Режим доступа:

<https://learn.microsoft.com/ru-ru/dotnet/desktop/winforms/overview/?view=netdesktop-8.0>, свободный - Загл. с экрана.