бюджетное профессиональное образовательное учреждение Вологодской области

«Череповецкий лесомеханический техникум им. В.П. Чкалова»

Специальность **09.02.07**«Информационные системы и программирование»

**ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

**ПП по ПМ.05 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОНЫХ СИСТЕМ**

Выполнил студент 3 курса группы ИС-31

Авдонина Елизавета Максимовна

подпись \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

место практики: ООО «Малленом Системс»

Период прохождения:

с «08» июня 2025 года

по «21» июня 2025 года

Руководитель практики от

предприятия, должность:

Южакова Н.В., специалист по кадрам

подпись\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

МПРуководитель практики от

техникума: Материкова А.А.

Оценка:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2025 года

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc200984466)

[1. ОБЩАЯ ХАРАКТЕРИСТИКА КОМПАНИИ 4](#_Toc200984467)

[1.1 Общая информация (сфера деятельности) 4](#_Toc200984468)

[1.2 Роль информационных систем в работе организации 4](#_Toc200984469)

[1.3 Основные используемые технологии 5](#_Toc200984470)

[2 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННЫХ СИСТЕМ 6](#_Toc200984471)

[2.1 Анализ требований и моделирование информационных систем 6](#_Toc200984472)

[2.2 Проектирование информационных систем 6](#_Toc200984473)

[2.3 Разработка информационных систем 7](#_Toc200984474)

[2.4 Тестирование информационных систем 7](#_Toc200984475)

[2.5 Внедрение, эксплуатация и сопровождение информационных систем 8](#_Toc200984476)

[3 ВЫПОЛНЯЕМЫЕ ЗАДАНИЯ 9](#_Toc200984477)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 20](#_Toc200984478)

[ПРИЛОЖЕНИЯ 21](#_Toc200984479)

# ВВЕДЕНИЕ

Производственная практика является важным компонентом образовательного процесса, который способствует качественной подготовке специалистов и их успешной интеграции в профессиональное сообщество.Она помогает студентам познакомиться с культурой работы в компании, включая понимание внутренней структуры, процедур и этики. Практика проходила в ООО «Малленом Системс» с 08.06.2025 по 21.06.2025.

Целью практики было освоение основного вида деятельности по направлению «Проектирование и разработка информационных систем» и формирование соответствующих общих и профессиональных компетенций.

В рамках выполнения поставленной цели, были сформулированы следующие задачи:

1. Провести анализ требований и моделирование информационных систем;
2. Выполнить проектирование информационных систем;
3. Произвести разработку информационных систем;
4. Провести тестирование информационных систем;
5. Осуществить внедрение, эксплуатацию и сопровождение информационных систем.

# ОБЩАЯ ХАРАКТЕРИСТИКА КОМПАНИИ

## Общая информация (сфера деятельности)

Малленом Системс - ведущая российская компания, специализирующаяся на разработке и внедрении систем компьютерного зрения, промышленной видеоаналитики и интеллектуальной обработки данных. Компания создает решения на базе технологий машинного зрения и искусственного интеллекта (машинное обучение, нейронные сети глубокого обучения).

Компания ориентирована на наукоемкие ИТ-проекты для повышения эффективности и безопасности промышленных предприятий в различных отраслях (транспорт, машиностроение, нефтегазовая, металлургическая, пищевая, фармацевтическая, алмазодобывающая, атомная и другие).

## Роль информационных систем в работе организации

Информационные системы играют ключевую роль в следующих аспектах работы Малленом Системс:

* Автоматизация: Автоматизация производственных процессов и промышленных предприятий.
* Видеоконтроль: Обеспечение видеоконтроля в различных областях, таких как транспорт и промышленность.
* Анализ данных: Интеллектуальная обработка данных для поиска закономерностей и прогнозирования.
* Безопасность: Повышение безопасности на промышленных предприятиях.
* Эффективность: Повышение эффективности технологических процессов и производства.
* Принятие решений: Предоставление информации для принятия обоснованных решений.
* Прослеживаемость: Обеспечение прослеживаемости продукции на различных уровнях (L1-L2).

## Основные используемые технологии

* Языки программирования: C++, Python (для машинного обучения и анализа данных), Java или C# (для разработки клиент-серверных приложений).
* Фреймворки: TensorFlow, PyTorch (для машинного обучения), OpenCV (для компьютерного зрения), .NET или Java frameworks (для разработки приложений).
* СУБД: PostgreSQL, MySQL, MS SQL Server (для хранения и управления данными).
* Системы контроля версий: Git (с использованием платформ, таких как GitHub, GitLab или Bitbucket).
* Инструменты проектирования: UML-диаграммы (для проектирования архитектуры систем), инструменты моделирования данных.
* Методологии разработки: Agile (Scrum, Kanban) - для гибкой и итеративной разработки, возможно, с элементами Waterfall для более крупных проектов.
* Системы управления задачами: Jira, Trello, Asana (для управления проектами и задачами).

Кроме того, в компании используются собственные решения на базе нейронных сетей и детерминированных алгоритмов анализа изображений, а также алгоритмы от Cognex и Hikrobot.

# ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННЫХ СИСТЕМ

Проектирование и разработка информационных систем — это комплексный процесс создания ИС, включающий в себя этапы анализа, проектирования, кодирования, тестирования и внедрения

## Анализ требований и моделирование информационных систем

Анализ требований и моделирование информационных систем — это начальная фаза, в которой выясняется, зачем вообще нужна система, что она должна делать, для кого и как. Главная цель - понять потребности заказчика и пользователей и зафиксировать их в виде четких и понятных требований. Затем эти требования преобразуются в модели, чтобы визуализировать систему и убедиться, что все поняли ее одинаково.

1. Сбор требований: Интервью с заказчиками, опросы пользователей, изучение существующих документов и систем.
2. Анализ требований: Выявление противоречий, неясностей, приоритезация требований.
3. Спецификация требований: Документ, в котором подробно описаны все требования к системе (функциональные и нефункциональные).
4. Моделирование: Создание диаграмм и схем, которые показывают структуру системы, ее поведение, взаимодействие с другими системами. Примеры: диаграммы вариантов использования (use case), диаграммы классов, диаграммы последовательностей.

## Проектирование информационных систем

На основе требований разрабатывается план создания системы. Определяется архитектура системы, структура базы данных, интерфейсы пользователя, логика работы отдельных модулей.

1. Архитектурное проектирование: Выбор архитектурного стиля (например, клиент-сервер, микросервисы), определение основных компонентов системы и их взаимодействия.
2. Проектирование базы данных: Разработка схемы базы данных, определение таблиц, полей, связей, индексов.
3. Проектирование интерфейса пользователя (UI/UX): Разработка внешнего вида, навигации, логики работы интерфейса, чтобы он был удобным и понятным для пользователей.
4. Проектирование модулей: Определение функциональности, входных и выходных данных для каждого модуля системы.

## Разработка информационных систем

Реализация проекта на практике. Написание кода, создание базы данных, разработка интерфейса, интеграция всех компонентов в единую систему.

1. Написание кода: Программирование модулей и компонентов системы на выбранных языках программирования.
2. Создание базы данных: Реализация спроектированной структуры базы данных.
3. Разработка интерфейса пользователя: Создание графического интерфейса, который соответствует спроектированным макетам.
4. Интеграция: Объединение всех разработанных компонентов в единую работающую систему.

## Тестирование информационных систем

Проверка того, что система работает правильно и соответствует требованиям. Выявление и исправление ошибок.

1. Разработка тестовых сценариев: Создание набора тестов, которые проверяют различные аспекты системы.
2. Модульное тестирование: Проверка отдельных модулей.
3. Интеграционное тестирование: Проверка взаимодействия между модулями.
4. Системное тестирование: Проверка системы в целом.
5. Приемочное тестирование: Проверка системы заказчиком перед внедрением.

## Внедрение, эксплуатация и сопровождение информационных систем

Ввод системы в эксплуатацию, обучение пользователей, обеспечение ее стабильной работы, исправление ошибок и внесение изменений.

1. Развертывание: Установка системы на серверах или в облаке.
2. Миграция данных: Перенос данных из старых систем в новую систему.
3. Обучение пользователей: Проведение тренингов и предоставление инструкций.
4. Техническая поддержка: Решение проблем и вопросов пользователей.
5. Сопровождение: Исправление ошибок, внесение изменений, добавление новых функций.

# ВЫПОЛНЯЕМЫЕ ЗАДАНИЯ

* 1. Первое задание

Сначала я провела анализ систем управления проектами, основанные на методологии kanban.

Обзор популярных систем управления проектами, основанных на Kanban, с их плюсами и минусами:

Trello:

* Плюсы: Простой в использовании, интуитивно понятный интерфейс, бесплатный тарифный план, интеграция с множеством других инструментов.
* Минусы: Ограниченные возможности для сложных проектов, нет встроенной поддержки для отслеживания времени.

Jira:

* Плюсы: Мощный, настраиваемый, хорошо подходит для крупных команд и сложных проектов, широкие возможности отслеживания и отчетности.
* Минусы: Более сложный в освоении, более дорогой, требует настройки.

Asana:

* Плюсы: Гибкий, подходит для различных типов проектов, хорошие возможности для совместной работы, интеграция с другими инструментами.
* Минусы: Может быть перегружен функциями для небольших проектов.

Microsoft Planner:

* Плюсы: Интегрирован с Microsoft 365, простой в использовании, подходит для небольших команд.
* Минусы: Ограниченные возможности по сравнению с другими системами.

ClickUp:

* Плюсы: Очень настраиваемый, множество функций, бесплатный тарифный план с большим количеством возможностей.
* Минусы: Может быть перегружен функциями, требуются усилия для настройки под конкретные нужды.

GitHub Projects/GitLab Issues:

* Плюсы: Тесно интегрированы с репозиториями кода, идеально подходят для разработки программного обеспечения, бесплатные для публичных проектов.
* Минусы: Ориентированы в основном на разработку, менее удобны для других типов проектов.

YouTrack:

* Плюсы: Разработан специально для команд разработчиков, мощные инструменты отслеживания и отчетности, интеграция с другими инструментами JetBrains.
* Минусы: Ориентирован в основном на разработку, может быть сложным в освоении.

Учитывая требования и задачи проекта (подсистема хранения данных для мониторинга износа режущих инструментов, сервис для обработки данных, удаление/добавление инструментов), я выбрала Asana, так как она обеспечивает высокую степень настраиваемости, необходимую для управления сложной системой с различными типами данных и взаимодействиями. Ее возможности отслеживания и отчетности будут полезны для мониторинга прогресса разработки и выявления проблем.

Далее я создала GIT репозиторий, в котором будут располагаться выполненные задания, отчёты и прочие документы текущей практики.

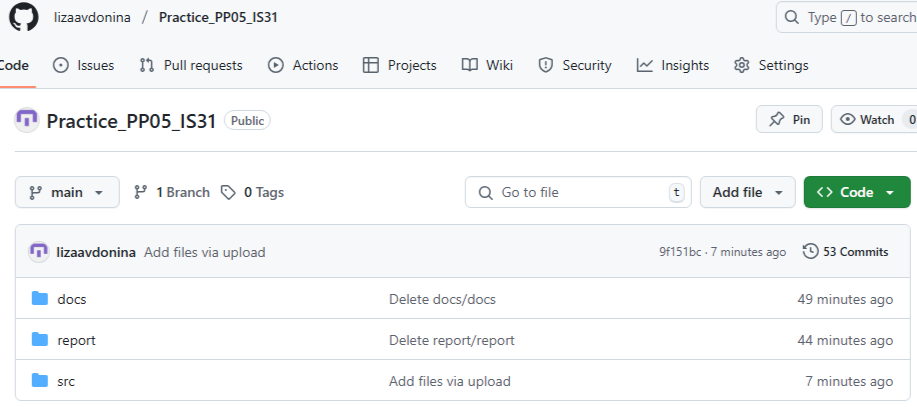


Рисунок 3.1 – Репозиторий «Practice\_PP05\_IS31»

* 1. Второе задание

Необходимо было определиться с направлением разработки. Я выбрала тему «Подсистема хранения данных для мониторинга износа режущих инструментов».

Для выбранной предметной области я составила задачи и ввела их на доску kanban (рисунок 3.2).

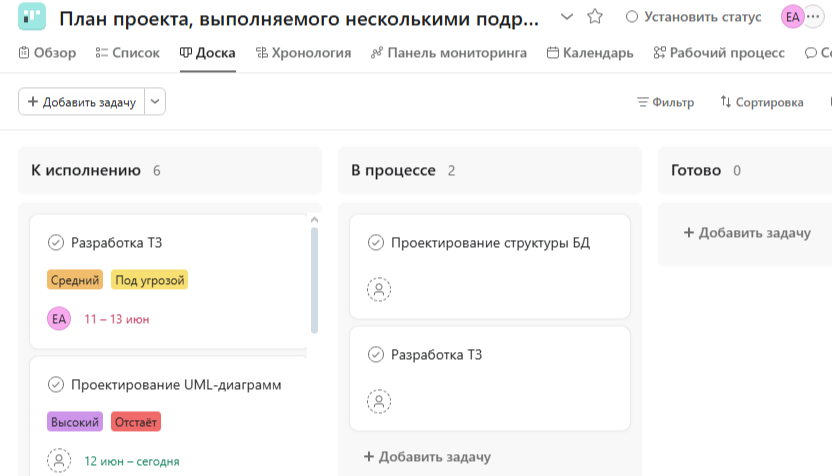


Рисунок 3.2 – Доска kanban

**Этапы и задачи:**

1. Backlog:

* Проектирование структуры БД (2 дня).
* Подготовка руководства пользователя и администратора (0,5 дня).
* Проектирование ER-диаграммы (0,5 дня).
* Проектирование UML-диаграмм (1-2 дня).
* Разработка ТЗ (2 дня).

1. To Do

* Разработка ТЗ.
* Проектирование UML-диаграмм
* Проектирование структуры БД.

Далее я написала техническое задание.

Смотреть Приложение А.

Также я написала руководство пользователя и администратора.

Смотреть Приложение Б.

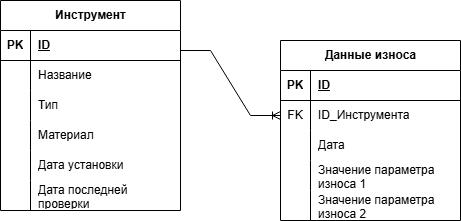


Рисунок 3.3 – ER-диаграмма

Общая структура:

Диаграмма представляет собой модель данных, описывающую инструменты и данные об их износе. Она состоит из двух сущностей: Инструмент (Tool) и ДанныеИзноса (WearData), связанных отношением "один ко многим".

Сущности и атрибуты:

Сущность "Инструмент" (Tool), содержит следующие атрибуты:

* ID: Уникальный идентификатор инструмента
* Название: Название инструмента.
* Тип: Тип инструмента (например, сверло, фреза, токарный резец).
* Материал: Материал, из которого изготовлен инструмент.
* Дата установки: Дата установки инструмента в оборудование или начала его использования.
* Дата последней проверки: Дата последней проверки состояния инструмента.

Сущность "ДанныеИзноса" (WearData), содержит следующие атрибуты:

* ID: Уникальный идентификатор записи об износе (вероятно, первичный ключ).
* ИнструментID: Идентификатор инструмента, к которому относится запись износа (внешний ключ, ссылающийся на ID в таблице Инструмент).
* Дата: Дата, когда были сняты данные об износе.
* Значение параметра износа 1: Численное значение первого параметра, характеризующего износ (например, глубина износа, ширина износа).
* Значение параметра износа 2: Численное значение второго параметра, характеризующего износ (например, изменение шероховатости поверхности).

Связь:

* Тип связи: "Один ко многим" (One-to-Many).
* Направление связи: Один инструмент может иметь много записей в таблице ДанныеИзноса. Каждая запись в ДанныеИзноса относится только к одному конкретному инструменту.
* Значение связи: Эта связь позволяет отслеживать историю износа каждого инструмента, регистрируя параметры износа через определенные промежутки времени.

В целом, эта ER-диаграмма моделирует систему, которая отслеживает информацию об инструментах и ​​собирает данные об их износе, что позволяет анализировать состояние инструментов, прогнозировать их срок службы и оптимизировать процессы обслуживания.

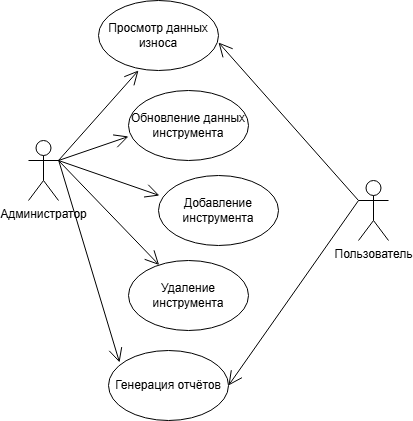


Рисунок 3.4 – диаграмма вариантов использования

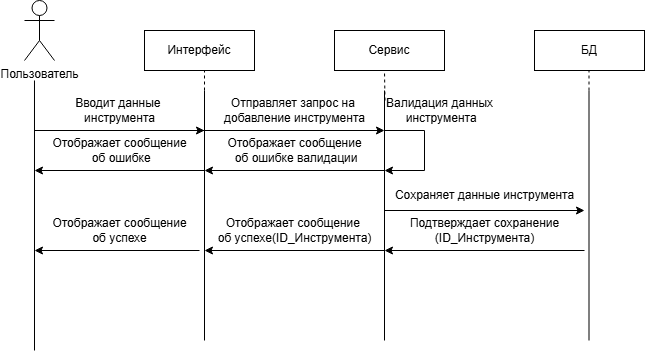


Рисунок 3.5 – диаграмма последовательностей



Рисунок 3.6 – диаграмма компонентов

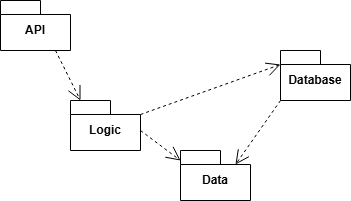


Рисунок 3.7 – диаграмма пакетов

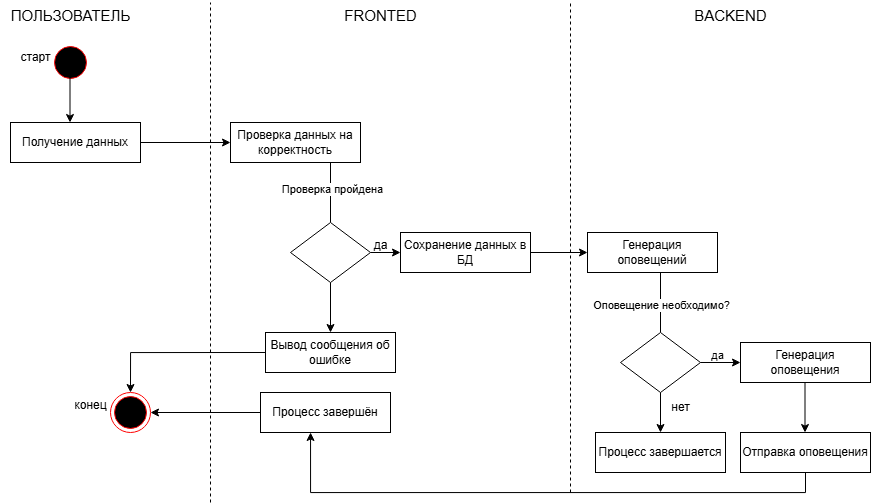


Рисунок 3.8 – диаграмма деятельности

* 1. Третье задание

В рамках разработки подсистемы хранения данных для мониторинга износа режущих инструментов было создано консольное приложение на языке C# в среде Visual Studio Code. Процесс разработки включал следующие этапы:

1. Подготовка среды разработки: Установлен .NET SDK, необходимый для компиляции и запуска C# кода. Также установлено расширение C# для Visual Studio Code, обеспечивающее поддержку языка C# в редакторе кода.
2. Создание проекта: Создана новая папка, предназначенная для хранения файлов проекта. Эта папка была открыта в Visual Studio Code. В терминале Visual Studio Code выполнена команда dotnet new console, которая сгенерировала базовую структуру консольного приложения C#, включая файл Program.cs.
3. Реализация логики приложения: Содержимое автоматически созданного файла Program.cs было заменено на разработанный код, реализующий функциональность подсистемы хранения данных. Этот код включает в себя:

Класс Tool: Представляет сущность режущего инструмента. Каждый объект класса Tool обладает свойствами ToolId (уникальный идентификатор), Type (тип инструмента, например, фреза или сверло) и WearLevel (уровень износа инструмента, выраженный в процентах).

Класс ToolDatabase: Отвечает за хранение и управление информацией об инструментах. Внутри класса используется список (List<Tool>) для хранения объектов Tool. Класс предоставляет методы для:

* AddTool(): Добавление нового инструмента в базу данных (в список инструментов).
* RemoveTool(): Удаление инструмента из базы данных по его уникальному идентификатору (ToolId).
* UpdateWearLevel(): Обновление уровня износа существующего инструмента на основе его ToolId.
* GetTool(): Получение информации об инструменте по его ToolId. Метод возвращает объект Tool или null, если инструмент с указанным ToolId не найден.
* ListAllTools(): Вывод в консоль информации о всех инструментах, хранящихся в базе данных.

Класс Program: Содержит метод Main(), который является точкой входа в приложение. В методе Main() выполняется следующий алгоритм:

* Создается экземпляр класса ToolDatabase.
* В базу данных добавляются несколько инструментов с использованием метода AddTool().
* Выводится список всех инструментов с использованием метода ListAllTools().
* Обновляется уровень износа одного из инструментов с использованием метода UpdateWearLevel().
* Получается информация об инструменте с использованием метода GetTool() и выводится в консоль.
* Удаляется один из инструментов из базы данных с использованием метода RemoveTool().
* Снова выводится список всех инструментов, чтобы продемонстрировать удаление.

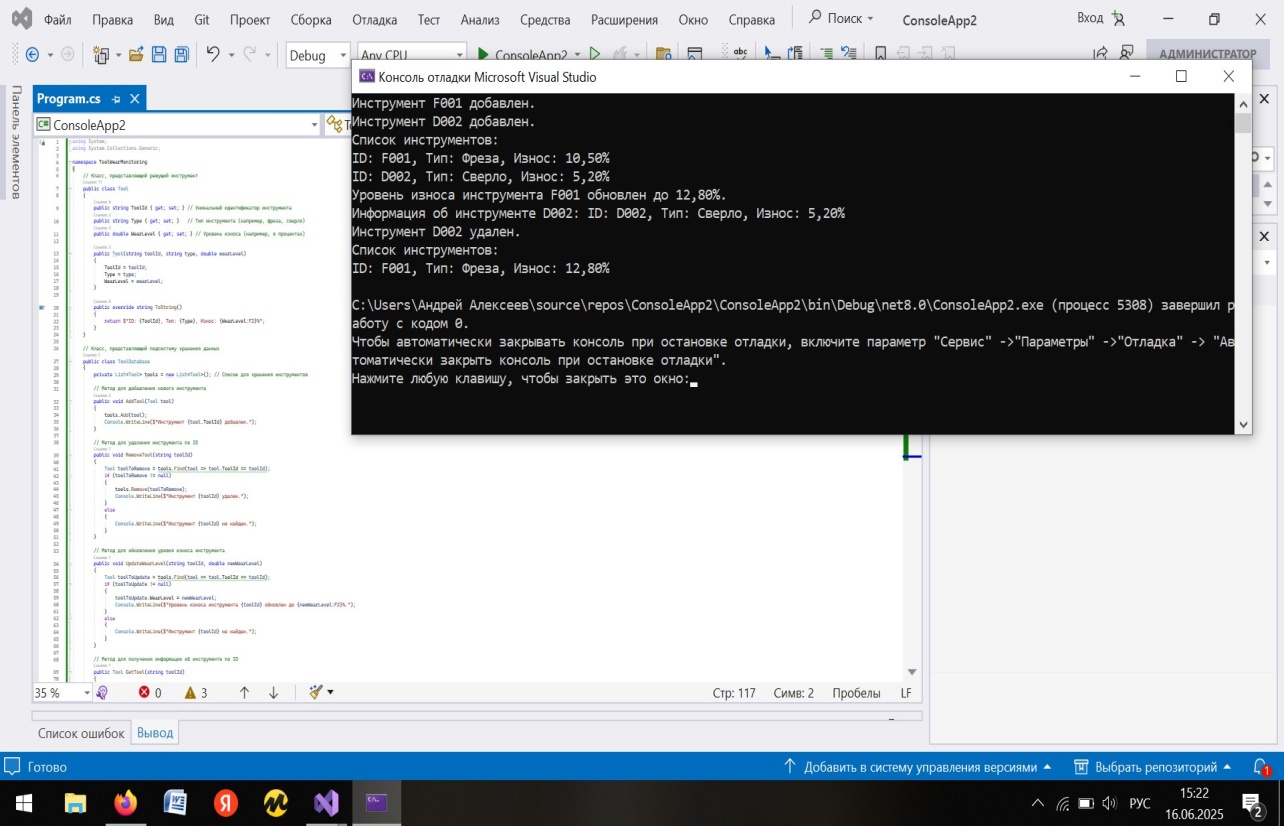


Рисунок 3.9 – Реализация

Смотреть Приложение В.

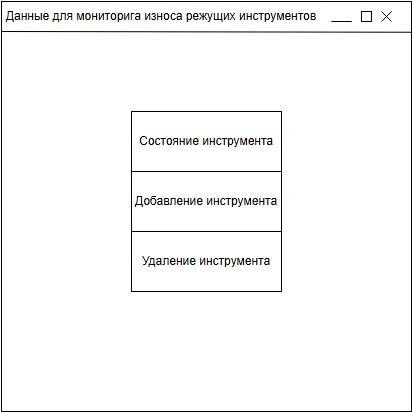


Рисунок 3.10 – Графический интерфейс

ЗАКЛЮЧЕНИЕ

На практике я освоила следующее:

1. Научилась проводить анализ требований и моделировать информационные системы;
2. Освоила выполнение проектирование информационных систем;
3. Произведение разработки информационных систем;
4. Проведение тестирования информационных систем;
5. Осуществление внедрения, эксплуатации и сопровождения информационных систем.

В процессе практики был изучен ключевой цикл проектирования и разработки информационных систем в компании, а также применены ранее приобретенные знания на практике при создании реальной информационной системы.

э

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

* 1. GitHub/репозиторий практики [Электронный ресурс] – режим доступа: https://github.com/cnciks/Practice\_PP05\_IS31/tree/main
  2. Компания "Малленом Системс" [Электронный ресурс] – режим доступа: https://www.mallenom.ru/
  3. Проектирование и разработка ИС [Электронный ресурс] – режим доступа: https://blog.iteam.ru/proektirovanie-informatsionnyh-sistem/
  4. Графический редактор диаграмм [Электронный ресурс] – режим доступа: https://app.diagrams.net/
  5. UML-диаграммы [Электронный ресурс] – режим доступа: https://practicum.yandex.ru/blog/uml-diagrammy/?

# ПРИЛОЖЕНИЯ

Приложение А

1. Общие сведения
   1. Полное наименование системы

Подсистема хранения данных для мониторинга износа режущих инструментов

* 1. Наименование Исполнителя и Заказчика

Заказчик: ООО «Малленом Системс».

Исполнитель: Авдонина Елизавета Максимовна.

* 1. Цель разработки

Создание надежной, масштабируемой и производительной подсистемы хранения данных для сбора, обработки и анализа информации о состоянии режущих инструментов, получаемой в процессе мониторинга их износа.

* 1. Задачи разработки
* Разработка схемы базы данных для хранения информации о режущих инструментах, параметрах их работы и данных мониторинга износа.
* Разработка API для интеграции с системами мониторинга износа режущих инструментов.
* Реализация механизмов для эффективного хранения и обработки больших объемов данных.
* Обеспечение безопасности и целостности хранимых данных.
* Реализация функциональности для анализа и визуализации данных об износе режущих инструментов.
  1. Плановые сроки начала и окончания работ по создания подсистемы

Начало работ: 08.06.2025

Окончание работ: 21.06.2025

1. Назначение и цели создания подсистемы
   1. Назначение подсистемы

Подсистема предназначена для сбора, хранения, обработки и анализа данных, поступающих от систем мониторинга износа режущих инструментов, с целью:

* Обеспечения возможности прогнозирования остаточного ресурса инструментов.
* Оптимизации режимов резания.
* Снижения затрат на замену инструментов.
* Повышения качества выпускаемой продукции.
  1. Цели создания подсистемы
     1. Основные цели создания подсистемы
* Создание единого централизованного хранилища данных об износе режущих инструментов.
* Обеспечение возможности оперативного доступа к данным мониторинга.
* Предоставление инструментов для анализа данных и выявления закономерностей.
  + 1. Целевая аудитория
* Инженеры-технологи.
* Операторы станков с ЧПУ.
* Специалисты по обслуживанию и ремонту оборудования.
* Руководители производственных подразделений.

1. Требования к подсистеме
   1. Требования к подсистеме в целом
      1. Требования к структуре и функционирование подсистемы

* Подсистема должна иметь модульную структуру, обеспечивающую возможность расширения функциональности.
* Подсистема должна обеспечивать масштабируемость для обработки растущих объемов данных.
* Подсистема должна обеспечивать высокую доступность и отказоустойчивость.
* Подсистема должна быть совместима с существующей IT-инфраструктурой Заказчика.
  + 1. Требования к персоналу

Для эксплуатации подсистемы требуется персонал, обладающий знаниями и навыками в области:

* Администрирования баз данных.
* Программирования (Python, SQL).
* Анализа данных.
* Информационной безопасности.
  + 1. Требования к сохранности информации
* Подсистема должна обеспечивать резервное копирование данных с заданной периодичностью.
* Подсистема должна обеспечивать восстановление данных в случае сбоев.
* Подсистема должна обеспечивать аудит действий пользователей.
  + 1. Требования к разграничению доступа
* Подсистема должна обеспечивать разграничение доступа к данным на основе ролей пользователей.
* Подсистема должна обеспечивать аутентификацию пользователей.
* Подсистема должна обеспечивать защиту от несанкционированного доступа.
  1. Требования к функциям, выполняемые подсистемой
     1. Основные требования
        1. Структура подсистемы
* Модуль сбора данных
* Модуль хранения данных
* Модуль обработки данных
* Модуль анализа данных
* Модуль визуализации данных
* Модуль управления пользователями
* Модуль интеграции
  + 1. Требования к функциональным возможностям
* Сбор данных: Поддержка различных протоколов сбора данных
* Хранение данных: Поддержка различных типов баз данных
* Обработка данных: Фильтрация, нормализация, агрегация данных.
* Анализ данных: Статистический анализ, машинное обучение
* Визуализация данных: Создание интерактивных графиков, диаграмм и отчетов.
* Интеграция: Предоставление API для интеграции с другими системами.
  1. Требования к видам обеспечения
     1. Требования к языкам программирования

Для разработки компонентов системы хранения данных следует использовать язык программирования C#.

* + 1. Требования к лингвистическому обеспечению

Интерфейс пользователя и документация должны быть доступны на русском и английском языках.

* + 1. Требования к программному обеспечению

Для работы подсистемы хранения данных необходимо использовать надежное и проверенное программное обеспечение, такое как Windows.

1. Состав и содержание работ по созданию подсистемы

* Анализ требований и проектирование архитектуры подсистемы.
* Разработка схемы базы данных.
* Разработка API для интеграции с системами мониторинга износа режущих инструментов.
* Реализация модулей подсистемы
* Тестирование подсистемы.
* Разработка документации.
* Внедрение и настройка подсистемы.
* Обучение персонала.

1. Порядок контроля и приёмки подсистемы
   1. Виды, состав, объём и методы испытаний

Функциональное тестирование: Проверка соответствия функциональности подсистемы требованиям ТЗ.

Интеграционное тестирование: Проверка корректности взаимодействия Подсистемы с другими системами.

Нагрузочное тестирование: Оценка производительности подсистемы при больших объемах данных.

Тестирование безопасности: Проверка защиты от несанкционированного доступа.

Приемочные испытания: Проводятся Заказчиком после завершения всех этапов разработки.

* 1. Общие требования к приёмке подсистемы
* Подсистема должна соответствовать требованиям, указанным в ТЗ.
* Подсистема должна успешно пройти все виды испытаний.
* Должна быть предоставлена полная документация на подсистему.
* Персонал заказчика должен быть обучен работе с подсистемой.

1. Требования к составу и содержанию работ

* Планирование работ.
* Разработка.
* Тестирование.
* Документирование.
* Согласование результатов с Заказчиком.

Приложение Б

1. Введение

Настоящее руководство пользователя и администратора предназначено для предоставления информации о настройке, управлении и использовании подсистемы хранения данных для мониторинга износа режущих инструментов. Подсистема состоит из сервиса, который обрабатывает данные о состоянии инструментов, предоставляет API для добавления, удаления и обновления информации об инструментах.

1. Целевая аудитория

* Пользователи: Операторы станков, инженеры-технологи, специалисты по техническому обслуживанию, использующие API подсистемы для получения данных об инструментах и внесения изменений.
* Администраторы: Системные администраторы, отвечающие за установку, настройку, обслуживание и мониторинг подсистемы.

1. Обзор системы

Подсистема хранения данных предназначена для централизованного хранения и управления информацией о режущих инструментах, используемых в производственном процессе. Подсистема позволяет отслеживать состояние инструментов, прогнозировать износ, планировать замену и оптимизировать использование инструментов.

1. Руководство пользователя
   1. Доступ к API:

* Подсистема предоставляет API для взаимодействия с другими компонентами системы мониторинга.
* Доступ к API осуществляется по протоколу HTTP/HTTPS.
* Для аутентификации и авторизации используется OAuth 2.0.
  1. Основные операции API:
* Получение информации об инструменте (GET /tools/{id})
* Создание нового инструмента (POST /tools)
* Обновление информации об инструменте (PUT /tools/{id})
* Удаление инструмента (DELETE /tools/{id})
* Поиск инструментов (GET /tools?type={type}&status={status})
  1. Обработка ошибок:

В случае возникновения ошибок API возвращает код ошибки HTTP и сообщение об ошибке в формате JSON.

1. Руководство администратора
   1. Установка подсистемы:

Требования к оборудованию:

* Указать минимальные и рекомендуемые требования к серверу.

Требования к программному обеспечению:

* Указать необходимые версии операционной системы, базы данных, веб-сервера.

Процесс установки:

* Подробно описать шаги по установке и настройке Подсистемы, включая установку необходимых пакетов, настройку базы данных и веб-сервера.
  1. Обслуживание и мониторинг

Основные параметры для мониторинга:

* CPU Usage.
* Memory Usage.
* Disk Space Usage.
* Network Traffic.

Резервное копирование:

* Необходимо регулярно создавать резервные копии базы данных и конфигурационных файлов.
* Рекомендуется использовать автоматическое резервное копирование с помощью cron.
  1. Устранение неполадок:

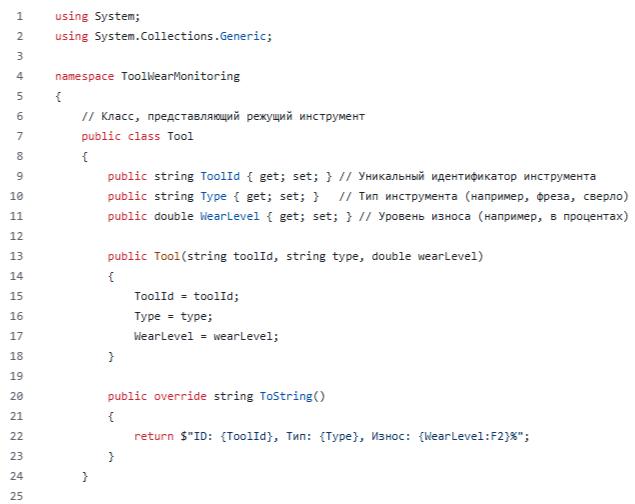
Ошибка подключения к базе данных:

* Проверьте правильность строки подключения к базе данных в файле конфигурации.
* Проверьте, запущен ли сервер базы данных.
* Проверьте, доступен ли сервер базы данных с сервера, на котором работает подсистема.
* Ошибка аутентификации:
* Проверьте правильность предоставленных учетных данных.
* Проверьте, правильно ли настроена аутентификация в файле конфигурации.

1. Безопасность

* Регулярно обновляйте программное обеспечение подсистемы и операционной системы.
* Используйте надежные пароли для доступа к подсистеме и базе данных.
* Настройте брандмауэр для защиты от несанкционированного доступа.
* Регулярно создавайте резервные копии данных.
* Обучите пользователей правилам безопасности.

Приложение В





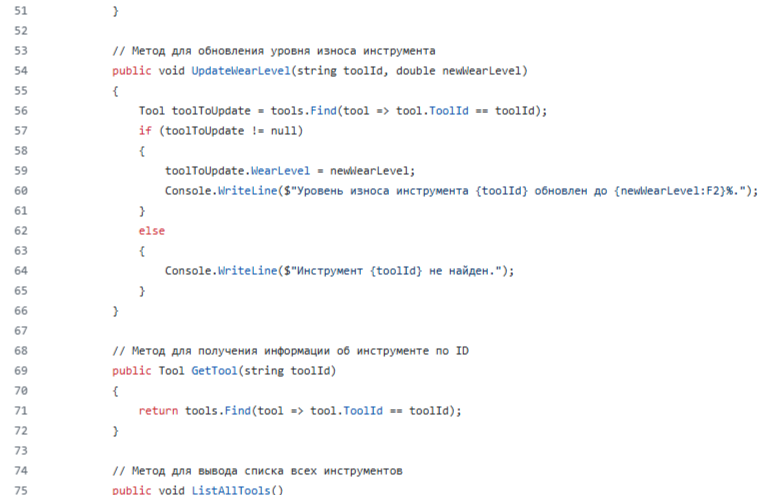






Рисунок А.1 – Код подсистемы