**Техническое задание на разработку программного обеспечения для станка моторо-тракторного завода**

**1. Введение  
1.1 Краткое описание проекта**

Целью данного проекта является разработка программного обеспечения (ПО) для управления моторотракторным станком, которое позволит пользователям автоматизировать производственные процессы и повысить эффективность работы станка.  
  
**1.2 Задачи проекта**  
• Обеспечить управление всеми функциями станка, включая движение по осям, управление скоростью, подачей и т.д.  
• Позволить создавать и редактировать программы обработки на станке  
• Интегрировать станок с другими производственными системами и устройствами  
• Предоставить пользователям удобный и интуитивно понятный интерфейс  
  
**2. Требования к продукту  
2.1 Функциональные требования**  
• Управление всеми функциями станка  
• Создание и редактирование программ обработки  
• Интеграция с другими производственными системами и устройствами  
• Диагностика и устранение неисправностей  
• Отображение состояния станка в режиме реального времени  
• Поддержка различных форматов файлов обработки  
**2.2 Нефункциональные требования**  
• Удобный и интуитивно понятный интерфейс  
• Высокая производительность и надежность  
• Масштабируемость для поддержки будущих расширений  
• Соответствие отраслевым стандартам и требованиям безопасности  
  
**3. Описание проекта**  
**3.1 Интерфейсы**

ПО будет иметь графический пользовательский интерфейс (GUI), который позволит пользователям взаимодействовать с различными функциями станка. GUI должен быть разработан в соответствии с принципами эргономики и обеспечивать удобство и простоту использования.  
  
**3.2 Архитектура**

ПО будет основано на модульной архитектуре, которая позволит легко добавлять новые функции и расширять функциональность в будущем. Архитектура должна обеспечивать высокую производительность, надежность и масштабируемость.  
  
**3.3 Бизнес-логика**

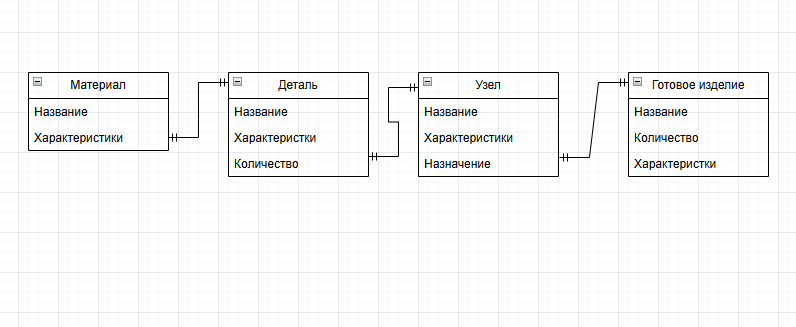
Бизнес-логика ПО должна реализовывать все необходимые функции управления станком, включая обработку программ обработки, управление осями, диагностику и устранение неисправностей. Бизнес-логика должна быть разработана с использованием современных принципов программирования и должна обеспечивать высокую производительность и надежность.  
  
**3.4 График работы**

Разработка ПО будет осуществляться в следующие этапы:  
  
• Этап 1 (2 недели): Анализ требований и проектирование  
• Этап 2 (4 недели): Разработка и тестирование  
• Этап 3 (2 недели): Интеграция и тестирование системы  
• Этап 4 (2 недели): Документирование и обучение  
  
**3.5 Бюджет и ресурсы для реализации продукта**

Бюджет на разработку ПО составляет 100 000 рублей. Для реализации проекта потребуется команда из 2 разработчиков и 1 тестировщика.  
  
**3.6 Сдача и приемка**

ПО будет сдано заказчику после успешного прохождения всех этапов разработки и тестирования. Приемка ПО будет осуществляться комиссией, назначенной заказчиком. Комиссия проверит соответствие ПО требованиям технического задания и его работоспособность.  
  
**3.7 Технические спецификации**  
• Операционная система: Windows 10 или выше  
• Язык программирования: C++ или Python  
• База данных: MySQL или PostgreSQL  
• Требования к оборудованию: ПК с процессором Intel Core i5 или выше, оперативной памятью 8 ГБ или выше и жестким диском на 250 ГБ или выше  
  
**3.8 Стандарты**

ПО должно соответствовать следующим стандартам:  
• ГОСТ Р ИСО/МЭК 12207-2011 "Информационные технологии. Системы жизненного цикла программного обеспечения. Процессы"  
• ГОСТ Р ИСО/МЭК 29119-1-2013 "Информационные технологии. Инженерия программного обеспечения. Требования"  
• ГОСТ Р ИСО/МЭК 15288-2015 "Информационные технологии. Системы жизненного цикла систем. Процессы"

**ER-диаграмма**:

**Языки программирования:**

• C++: Высокопроизводительный и надежный язык, который широко используется в разработке программного обеспечения для промышленного оборудования, включая станки с ЧПУ.

• Python: Удобный для изучения и использования язык, который поддерживает различные библиотеки для обработки данных, визуализации и машинного обучения.

**Технологии:**

• Qt: Кроссплатформенный фреймворк для разработки графических интерфейсов (GUI), который позволяет создавать удобные и интуитивно понятные интерфейсы для пользователей.

• OpenCV: Библиотека для обработки изображений и машинного зрения, которая может использоваться для автоматизации и контроля качества на станках с ЧПУ.

• gRPC: Фреймворк для создания и вызова удаленных процедурных вызовов (RPC), который позволяет легко интегрировать программное обеспечение станка с другими системами и устройствами в производственной сети.

• Базы данных: MySQL или PostgreSQL для хранения и управления данными о станке и обрабатываемых деталях.

**Причины выбора:**

• C++: C++ обеспечивает высокую производительность и надежность, что необходимо для управления станком в режиме реального времени. Он также имеет богатую экосистему библиотек для поддержки различных функций станка.

• Python: Python прост в изучении и использовании, что позволяет разработчикам быстро создавать и прототипировать программное обеспечение. Его обширные библиотеки для обработки данных и машинного обучения могут быть использованы для реализации интеллектуальных функций на станке.

• Qt: Qt предоставляет мощные инструменты для разработки современных и удобных графических интерфейсов, что делает программное обеспечение станка удобным и интуитивно понятным для операторов.

• OpenCV: OpenCV обеспечивает широкий спектр функций обработки изображений, которые могут быть использованы для автоматической проверки качества обработанных деталей и контроля состояния станка.

• gRPC: gRPC позволяет легко интегрировать программное обеспечение станка с другими производственными системами и устройствами, обеспечивая обмен данными и управление в режиме реального времени.

• Базы данных: MySQL или PostgreSQL предоставляют надежное и масштабируемое решение для хранения и управления данными о станке, обрабатываемых деталях и производственных процессах.