1. Аналитический раздел.

1.1. Анализ технологий автоматизированной обработки деталей

1.1.1. Общие сведения о станках с ЧПУ

Числовое программное управление (ЧПУ) — компьютеризованная система управления, управляющая приводами технологического оборудования, включая станочную оснастку. На рисунке приведено пример ЧПУ раскройного станка (Рисунок 1).

[](http://15.sls.name/wp-content/uploads/2015/01/Безымянный.png)

Рисунок 1. ЧПУ Раскройного станка

**Оборудование с ЧПУ может быть представлено:**

* станочным парком, например станками (станки, оборудованные числовым программным управлением, называются станками с ЧПУ ):
* для обработки металлов (например, фрезерные или токарные), дерева, пластмасс,
* для резки листовых заготовок,
* для обработки давлением и т.д.
* приводами асинхронных электродвигателей, использующих векторное управление;
* характерной системой управления современными промышленными роботами.

Несколько станков с ЧПУ могут объединиться в гибкую автоматизированную производственную систему (ГПС), которая в свою очередь может быть дополнена гибким автоматизированным участком (ГАУ) и войти в состав автоматической линии (производства масштаба участка либо цеха), ГАП.

 Под управлением станком принято понимать совокупность воздействий на его механизмы, обеспечивающие выполнения технологического цикла обработки, а под системой управления — устройство или совокупность устройств, реализующих эти воздействия.

Числовое программное управление ( ЧПУ) — это управление, при котором программу задают в виде записанного на каком-то носителе массива информации. Управляющая информация для систем ЧПУ является дискретной и ее обработка в процессе управления осуществляется цифровыми методами. Управление технологическими циклами практически везде осуществляется с помощью программируемых логических контроллеров, реализуемых на основе принципов цифровых электронных вычислительных устройств.

Системы ЧПУ практически вытесняют другие типы систем управления.

По технологическому назначению и функциональным возможностям системы ЧПУ подразделяют на четыре группы:

* позиционные, в которых задают только координаты конечных точек положения исполнительных органов после выполнения ими определенных элементов рабочего цикла;
* контурные или непрерывные, управляющие движением исполнительного органа по заданной криволинейной траектории;
* универсальные (комбинированные), в которых осуществляется программирование как перемещений при позиционировании , так и движения исполнительных органов по траектории, а также смены инструментов и загрузки
* выгрузки заготовок;
* многоконтурные системы, обеспечивающие одновременное или последовательное управление функционированием ряда узлов и механизмов станка.  
  Примером применения систем ЧПУ первой группы являются сверлильные, расточные и координатно — расточные станки. Примером второй группы служат системы ЧПУ различных токарных, фрезерных и круглошлифовальных станков. К третьей группе относятся системы ЧПУ различных многоцелевых токарных и сверлильно-фрезерно-расточных станков.

К четвертой группе относятся безцентровые круглошлифовальные станки, в которых от систем ЧПУ управляют различными механизмами: правки, подачи бабок и т.д. Существуют позиционные, контурные, комбинированные и многоконтурные циклы управления.

По способу подготовки и ввода управляющей программы различают так называемые оперативные системы ЧПУ (в этом случае управляющую программу готовят и редактируют непосредственно на станке, в процессе обработки первой детали из партии или ее имитации обработки) и системы, для которых управляющая программа готовится независимо от места обработки детали. Причем независимая подготовка управляющей программы может выполняться или с помощью средств вычислительной техники, входящих в состав систем ЧПУ данного станка, или вне ее (вручную или с помощью системы автоматизации программирования).

Программируемые контроллеры — это устройства управления електроавтоматичного станка. Большинство программируемых контроллеров имеют модельную конструкцию, в состав которой входят источник питания, процессорный блок и программируемая память, а также различные модули ввода/вывода. Для создания и отладки программ работы станка применяют программирующие аппараты. Принцип работы контроллера: считываются необходимые вводы/выводы и полученные данные анализируются в процессорном блоке. При этом решаются логические задачи и результат вычисления передается на соответствующий логический или физический вывод для подачи соответствующего механизма станка.(1)

1.1**.2 Способы** загрузки исходных данных в ЧПУ

Современные [станки с ЧПУ](https://infofrezer.ru/catalog) обладают высокой производительностью, обеспечивают отличное качество обработки и способны легко перестраиваться на обработку изделий другой партии. Достижение этих качеств ранее – на станках с ручным управлением и полуавтоматах – было практически невозможно. И только система числового программного управления (сокращённо ЧПУ) позволила внедрить обработку изделий в точном соответствии с заложенной программой. Достоинством такого способа управления является полное исключение роли оператора станка из процесса обработки. Станок «сам», в автоматическом режиме – согласно заложенной в память ЧПУ программе – выполняет технологические переходы обработки заготовки. Оператору остаётся лишь снимать/закреплять заготовки на рабочем столе, устанавливать и закреплять режущий инструмент, соответствующий данному этапу обработки (чистовому, черновому переходу и т.п.), и осуществлять общее наблюдение за оборудованием.

Однако, современное оборудование с ЧПУ не исключает полностью «человеческий фактор» из процесса обработки. Оно лишь переносит его во времени если в процессе обработки участие человека не требуется, то его роль во время подготовки управляющей программы и её загрузки на станок является определяющей для обеспечения высокого качества обработки.

Сущность работы системы ЧПУ заключается в преобразовании кодов управляющей программы (содержащей эскиз деталей и построенную на его базе траекторию движения фрезы) в сигналы для исполнительных элементов станка – электродвигателей инструментального портала и шпинделя. Таким образом, режущему инструменту сообщается необходимый по техпроцессу маршрут движения фрезы возле заготовки, а также режимы обработки (подача инструмента и частота вращения шпинделя).

Как и обычный персональный компьютер, система ЧПУ содержит микропроцессор, обрабатывающий команды и оперативную память – для хранения текущих данных. Именно в память системы ЧПУ требуется предварительная загрузка управляющей программы для фактической возможности фрезерного станка осуществлять обработку заготовок.

Как правило, современные фрезерные станки с ЧПУ рассчитаны на работу «в связке» с персональным компьютером. С его помощью можно осуществлять ручное управление перемещением шпинделя, а также загружать файлы управляющих программ и запускать обработку по ним в автоматическом режиме.

В некоторых случаях подключение ПК к фрезерному станку нежелательно (к примеру, при обработке каменных заготовок или графита образующаяся мелкая пыль может привести к выходу ПК из строя). Для таких ситуаций фрезерный станок комплектуется специальным пультом – DSP-контроллером. С его помощью доступны те же функции, что при использовании ПК.

И персональный компьютер, и [DSP-контроллер](https://infofrezer.ru/articles/opcii-frezernyh-stankov/dsp-kontroller-frezernogo-stanka-s-chpu) способны работать с внешними флешь-накопителями для загрузки файлов управляющих программ. Однако при работе «напрямую» с «флешки» порой возникают серьёзные затруднения.

Для сбора данных, требующихся для проведения структурного анализа системы является технология **IDEF3.** В отличие от большинства технологий моделирования бизнес-процессов, **IDEF3** не имеет жестких синтаксических или семантиче­ских ограничений, делающих неудобным описание неполных или нецелостных систем. Кроме того, автор модели (системный аналитик) избавлен от необходимости смешивать свои собственные предпо­ложения о функционировании системы с экспертными утвержде­ниями в целях заполнения пробелов в описании предметной области.

**IDEF3** также может быть использован как метод проектирования бизнес-процессов. **IDEF3-моделирование** органично дополняет тра­диционное моделирование с использованием стандарта [методологии IDEF0](http://itteach.ru/bpwin/metodologiya-idef0). В на­стоящее время оно получает все большее распространение как вполне жизнеспособный путь построения моделей проектируемых систем для дальнейшего анализа имитационными методами. Имитационное тестирование часто используют для оценки эксплуатационных ка­честв разрабатываемой системы. Более подробно методы имитацион­ного анализа будут рассмотрены ниже.

Основой модели IDEF3 служит так называемый сценарий биз­нес-процесса, который выделяет последовательность действий или подпроцессов анализируемой системы. Поскольку сценарий опреде­ляет назначение и границы модели, довольно важным является под­бор подходящего наименования для обозначения действий.

Сценарий для большинства моделей должен быть документиро­ван. Обычно это название набора должностных обязанностей челове­ка, являющегося источником информации о моделируемом процессе.

Также важным для системного аналитика является понимание це­ли моделирования — набора вопросов, ответами на которые будет служить модель, границ моделирования — какие части системы вой­дут, а какие не будут отображены в модели, и целевой аудитории — для кого разрабатывается модель.(2,3)

1.2 Совершенствование способов загрузки исходных данных в ЧПУ

Современные микропроцессорные устройства ЧПУ обеспечивают оперативное подготовку УП в режиме диалога с использованием графического дисплея. Гибкость и эффективность оперативного программирования обработки обеспечивается функциональными возможностями современных микропроцессорных устройств ЧПУ, основными из которых являются следующие: расширенный формат УП, с использованием всех основных функций кодов информации УП и дополнительных макрокоманд; краткое описание программируемого контура при помощи языков высокого уровня; возможность программирования в режиме “обучения” при обработке первой де-тали; развитая структура построения УП, содержащая основную программу, подпрограммы, программы коррекций и наладочных данных станка и инструментов; автоматическое формирование циклов движений, задаваемых в обобщенной виде при помощи параметров, значения которых присваивает оператор или они автоматически рассчитываются по определенным алгоритмам; задание технологических параметров обработки в естественной форме, удобной для оператора; автоматический расчет траектории инструмента, эквидистантной обрабатываемому контуру детали, с учетом радиальных и осевых размеров инструмента и его наладочных данных; развитые средства отладки и редактирования УП.

Управляющая программа для станков с современными микропроцессорными устройствам и ЧПУ имеет довольно сложную структуру. Необходимая для обработки детали информация может быть записана в основной программе (MPF), в подпрограммах (SPF), в дополнительных программах, в которых заданы значения коррекции инструментов, смещения нулевых точек станка и другие наладочные данные. С помощью основной программы организуется последовательность выполнения всей УП в соответствии с порядком записи ее кадров. В основной программе можно вызвать любые подпрограммы и указать число их последовательных исполнений, а также определить численные значения для этих подпрограмм. Обращение к программе коррекции осуществляется автоматически при вводе очередного кадра основной программы, номер которого соответствует номеру блока корректирующих записей. Вызов дополнительных данных, например, действительных размеров инструментов осуществляется как из основной программы, так и из подпрограмм заданием соответствующих команд. Каждая подпрограмма может содержать обращение к другим подпрограммам.

Современные микропроцессорные устройства ЧПУ с оперативной подготовкой УП имеют достаточно мощные средства автоматизированного программирования. Например, задание типовых геометрических образов для обрабатываемых конструктивных элементов детали сводится к определению типа стандартного цикла (проточка канавки для выхода резца при нарезании резьбы, нарезание резьбы резцом и т. д.) и его параметров. Макроопределения позволяют только на основе чертежных размеров без каких-либо дополнительных пересчетов описать фрагменты обрабатываемого контура. Например, в устройствах ЧПУ *Sinumerik* для токарных станков при программировании непосредственно по чертежу детали предусмотрено краткое описание фрагментов контура с использованием углов *ANG* пересечения прямых линий и одной из координат конечной точки, программировании обработки фаски задается указанием его длины в макроопределении *CHF*=

Управляющая программа (УП) – совокупность команд на языке программирования, соответствующая заданному алгоритму функционирования станка по обработке конкретной заготовки. УП содержит информацию о величинах и скоростях перемещения режущего инструмента относительно заготовки детали, указания об изменении частоты вращения шпинделя, смене инструмента, коррекции инструмента, подаче СОЖ и другие команды исполнительным органам станка. Эта информация записывается в кодах конкретного устройства ЧПУ в последовательности, соответствующей принятому техпроцессу обработки.

Кадр УП – составная часть УП, вводимая и обрабатываемая как единое целое и содержащее не менее одной команды.

Слово УП – составная часть кадра УП, содержащая данные о параметре процесса обработки заготовки или другие данные по выполнению управления.

Адрес УП – часть слова УП, определяющая назначение следующих за ним данных.

Для автоматизированной подготовки УП для станков с ЧПУ используются системы автоматизированного программирования (САП). САП для оборудования с ЧПУ комплекс технических, программных, языковых, информационных средств, осуществляющих преобразование данных чертежа детали и техпроцесса ее обработки в коды системы ЧПУ. В состав САП в

общем случае входят следующие части: технические средства, входной язык, процессор, промежуточный язык, постпроцессор.

Технические средства для построения САП представляют собою персональный компьютер необходимой конфигурации, специализированные программные устройства, в устройствах ЧПУ класса CNC технические средства устройства и САП составляют единое целое.

Входной язык САП – проблемно - ориентированный язык, предназначенный для описания исходных данных о детали и техпроцессе ее обработки на оборудовании с ЧПУ. На входном языке записываются исходные данные для автоматизированного программирования, например, в виде исходной программы(ИП).

**САП с формированием исходных данных на геометрическом вход-ном языке.** САП с формированием исходных данных на геометрическом входном языке обычно имеют графический редактор, в котором создается геометрическая модель (чертеж) обрабатываемой детали. Созданная модель импортируется в модуль разработки управляющих программ системы. В данном пособии САП подобного класса рассматривается на примере системы ADEM.

ADEM представляет собой комплексную систему автоматизированного проектирования. С её помощью можно решать целый ряд инженерных задач, таких как: проектирование изделий, подготовка конструкторской документации (чертежей, спецификаций, схем и т.д.), разработка управляющих программ для оборудования с ЧПУ, управление и хранение инженерных данных, разработка технологических процессов механической обработки, сборки, покрытия и т. д.

Подготовка УП осуществляется в модуле ADEM CAM. Исходными данными для автоматизированного программирования обработки на станках с ЧПУ в модуле ADEM CAM является последовательность технологических объектов. Каждый технологический объект, формируемый технологом-программистом в диалогово-интерактивном режиме, представляет собою информационно завершенную структуру, описывающий технологический переход (ТП) обработки конструктивного элемента (КЭ) детали. К отдельным технологическим объектам отнесены также определения (описания) положений точки начала обработки, безопасной позиции, плоскости холостых ходов, команд на выполнение технологического останова, поворота (шпинделя или детали) и т. п.

**САП с формированием исходных данных на проблемно-ориентированном входном языке**. В ряде САП исходные данные формируются на входном языке, представляющем проблемно-ориентированный язык для описания исходных данных о детали и техпроцессе ее обработки на станке с ЧПУ. На входном языке записывается исходная программа (ИП) в виде последовательности операторов. Набор операторов позволяет: определять геометрические объекты, параметры обработки; описывать траекторию движения инструмента, основные и вспомогательные функции станка и его системы управления.

В качестве примера рассмотрим САП МИКРОАПТ.

Система использует язык МИКРОАПТ для описания различных видов обработки на фрезерных, токарных, расточных, сверлильных и других станках. Допускается диалоговый режим общения с оператором. Язык имеет обобщенные технологические инструкции широкий набор средств для управления программой, организации подпрограмм и циклов. Это дает возможность параметрического задания размеров и технологических режимов, а также вычисления и переопределения параметров и геометрических эле-ментов в процессе работы программы.

Входной язык САП МИКРОАПТ использует приписные буквы русского и латинского алфавитов, арабские цифры и знаки – точка, запятая, двоеточие, плюс, минус, звездочка, черта дроби, равно, левая и права круглые скобки, точка с запятой, кавычки, пробел и некоторые другие знаки. При этом буквы латинского алфавита используются только для обозначения переменных и наименований элементарных функций за исключением буквы Z, используемой и для обозначения оси декартовой системы координат. Элементы входного языка содержат не менее одного символа.

Входной язык САП МИКРОАПТ предусматривает возможность определения трех видов геометрических элементов – точек, прямых и окружностей. Элемент каждого вида, за исключением окружности, обозначается начальной буквой наименования (Т-точка, П-прямая, К-окружность) и номером, которым может быть любое целое число от 0 до 99.

Геометрический элемент определяется оператором, состоящим из обозначения, обязательного разделителя – косой черты и собственно определения. Собственно определение геометрического элемента, в общем случае, состоит из четырех частей:

- опорных геометрических элементов;

- служебных слов;

- модификаторов выбора варианта;

- параметров (десятичные числа или произвольные арифметические выражения). (4,5 )

1.3 Постановка задачи

Задачей дипломной работы является разработка и внедрение программного обеспечения(ПО) для автоматизации систем загрузки станка с ЧПУ.

Требование к функциональным характеристикам:

* ввод входных данных,
* получить готовый файл с праматерями обработки для станка с ЧПУ.

Требования к информационным технологиям:

* малая стоимость,
* автономность в эксплуатации без специальных требований к условиям окружающей среды,
* высокая надежность работы
* "дружественность" операционной системы и прочего программного обеспечения, обусловливающая работу с ней пользователя без специальной профессиональной подготовки;

Требования к интерфейсам машинного и человеко-машинного взаимодействия:

* создание рабочего места: кресла, стола, или пульта управления, размещение приборов и органов управления (устройства ввода данных)
* [взаимодействие оператора](https://ru.wikipedia.org/wiki/%D0%A7%D0%B5%D0%BB%D0%BE%D0%B2%D0%B5%D0%BA%D0%BE-%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%BE%D0%B5_%D0%B2%D0%B7%D0%B0%D0%B8%D0%BC%D0%BE%D0%B4%D0%B5%D0%B9%D1%81%D1%82%D0%B2%D0%B8%D0%B5) со всеми [органами управления](https://ru.wikipedia.org/wiki/%D0%9E%D1%80%D0%B3%D0%B0%D0%BD_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F): их доступность и необходимые усилия, эффективность и скорость доступа

Требования к вычислительным средствам:

* надежность работы
* производительность

Требования к вычислительным средствам:

* надежность работы
* производительность

Требования к надежности

* Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением совокупности организационно-технических мероприятий

Требования к условиям эксплуатации:

* Условия эксплуатации ПО не должны быть ориентированы на специально обученный персонал.

Требования к информационной и программной совместимости:

* Пользовательский интерфейс должен быть интуитивно понятным
* Отчеты должны содержать лишь интересующую информацию
* Исходные коды программы должны быть реализованы на любом языке(PHP, JavaScript)
* Взаимодействие с СУБД и создание базы данных реализуется на языке SQL.

Требования к техническим показателям проектных решений:

* Основой для системы должна стать база данных, в которой будет храниться вся информация
* База данных должна включать в себя следующие таблицы: Детали, Циклы, Параметры

2. Разработка ИС формирования задания для станка с ЧПУ

2.1. Общая характеристика решения задачи автоматизации загрузки станка с ЧПУ (диаграмма деятельности)

Задачей дипломной работы автоматизация систем загрузки станка ЧПУ. Технология, выполненная в данной дипломной работе, уже существует в СУ Sinumerik 828D BASIC T, основное отличие созданного ПО и уже существующего решения это возможность получить входные данные для ЧПУ с любого устройства который имеет доступ в интернет. В целом созданное ПО дает возможность вводить входные данные полученные с чертежа и получить исходный файл, который в дальнейшем будет входными данными для ЧПУ, с любого устройства (ПК, ноутбук, планшет, смартфон) и местоположения. Тем самым оператор, имея в руках смартфон вводит все необходимые параметры, получает исходный файл, переносит его в USB носитель и предает их станку через USB порт.

Так же данное ПО позволяет без какого-либо риска обучить новых сотрудников(операторов). Где можно наглядно по чертежу добавлять циклы, задавать им параметры и получать исходный файл. Так же открыв этот файл в текстовом редакторе, есть возможность показать в каком формате принимает данные станок.

Основным языком программирования был выбран PHP. PHP скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов. Выбор данного языка обусловлен тем что он располагается на сервере в данном случае на внешнем хостинге, и доступ к нему можно получить с любой устройства имеющего возможность выхода в интернет.

Возможные ошибки при получении информации могут возникнуть если оператор введет неверные параметры, которые уже сам ЧПУ не примет.

Чертеж

Оператор

Получаем данные для введения в систему

По для ЧПУ

Загружаем данные

Выходные данные(файл)

Выгрузка данных

Загрузка данных в ЧПУ

Рисунок 2. Диаграмма деятельности

2.2. Проектные решения по информационному обеспечению решаемой задачи

Для решения задачи было принято сделать базу данных(БД) централизованной. БД будет располагаться на хостинге сервера.

Были созданы 4 таблицы (chpu\_for, details, save, parameter).

сhpu\_for – список циклов (Рисунок 4).

Включает в себя 4 столбца:

* id, идентификатор записи
* code, код цикла
* name, полное наименование цикла
* type, параметр определяющий возможность записи параметра через «;»

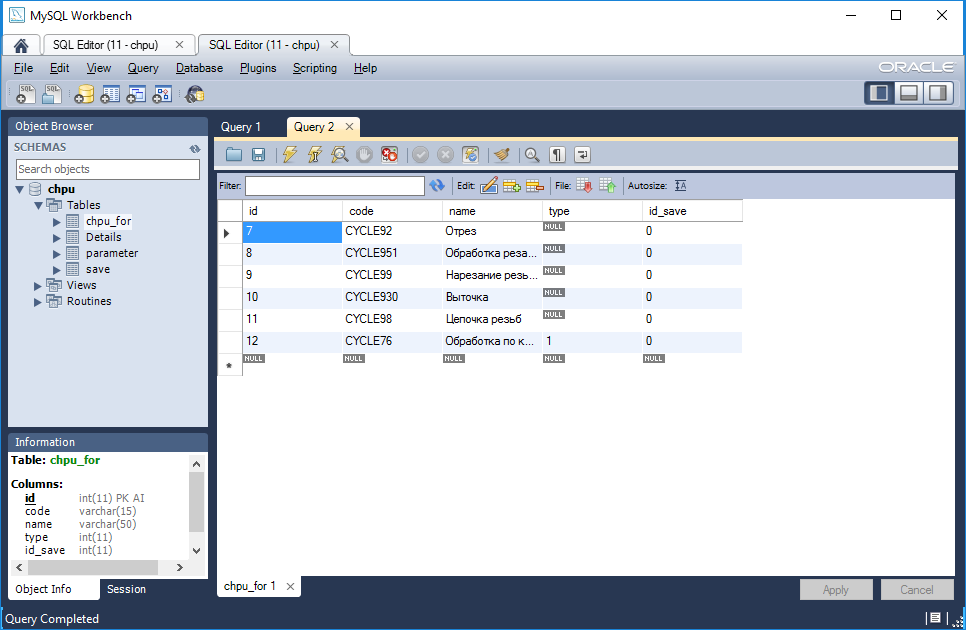


Рисунок 4. Список циклов

details – список деталей (Рисунок 5).

Включает в себя 3 столбца:

* id, идентификатор записи
* name, полное наименование детали
* img, название схемы для детали

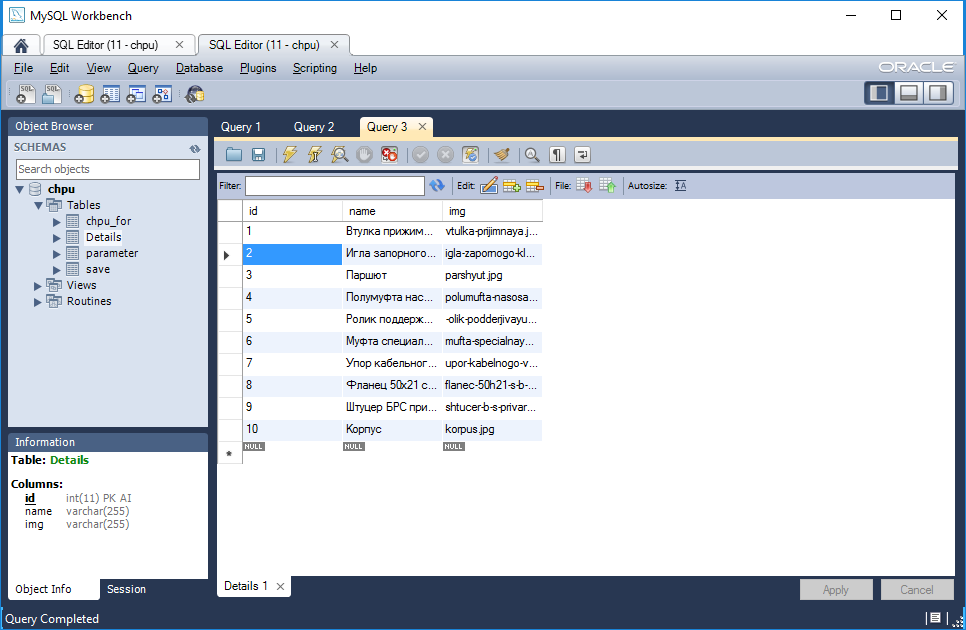


Рисунок 5. Список деталей

save – сохраненные параметры (Рисунок 6), параметры сохраняются в 2 видах. Первый понятный для человека, который отображается в самой программе. Второй сохраняться в виде машинного кода ЧПУ

Включает в себя 6 столбцов:

* id, идентификатор записи
* id\_for, идентификатор цикла
* text, сохраненный текст
* id\_details, идентификатор детали
* type, параметр определяющий к какому типу выводу он определен.
* id\_save, идентификатор сохраненного документа

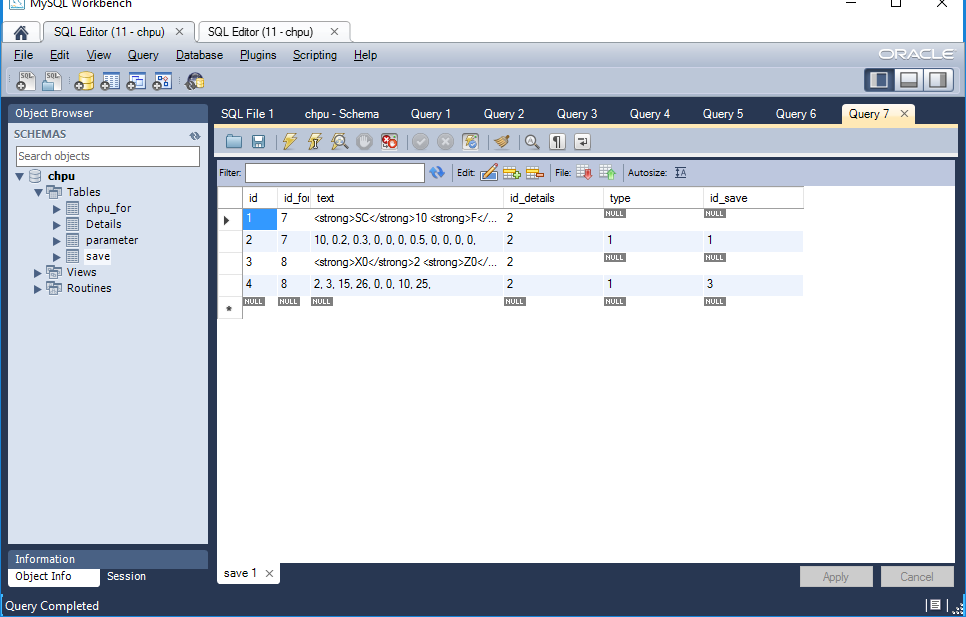


Рисунок 6. Сохраненные параметры

parameter – список параметров(Рисунок 7)

Включает в себя 5 столбцов:

* id, идентификатор записи
* code, код параметра
* name, полное наименование параметра
* id\_for, идентификатор цикла
* ext, единица измерения

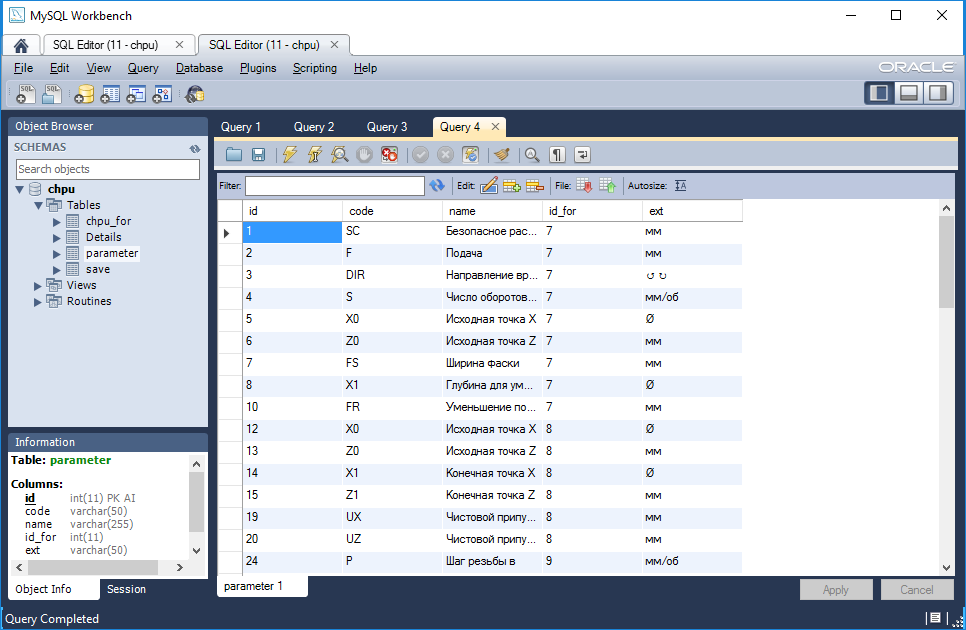


Рисунок 7. Список параметров

Выходные данные представляют с собой текстовый файл, с кодом (Рисунок 8).

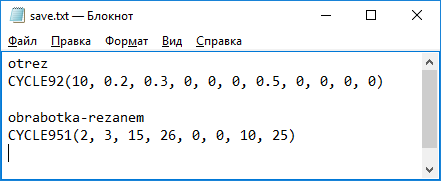


Рисунок 8. Выходные данные.

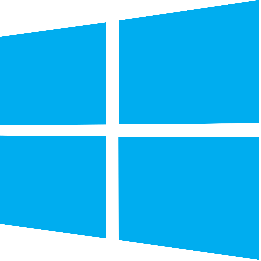
2.3. Проектные решения по программному обеспечению решаемой задачи

Программное обеспечение (ПО) включает совокупность программ, реализующих функции и задачи ИС и обеспечивающих устойчивую работу комплексов технических средств. В состав программного обеспечения входят общесистемные и специальные программы, а также инструктивно-методические материалы по применению средств программного обеспечения.

При выборе комплекса технических средств для разработки системы, одним из важнейших критерием является выбор операционной системы. Операционная система управляет техническими средствами компьютера, поддерживает запуск и выполнение тех или иных программ и приложений, обеспечивает защиту данных, выполняет различные сервисные функции. Каждая программа пользуется средствами, предоставляемыми операционной системой. Таким образом, выбор операционной системы очень важен, так как он определяет набор программ и формат исполняемых файлов, а также их взаимодействие с операционной системой.

На компьютерах, используемых в качестве рабочих мест пользователей, чаще всего применяются следующие операционные системы:

* Операционные системы семейства Windows от фирмы Microsoft (Windows XP/Vista/7/8/10),
* Операционные системы Linux/BSD семейства (UNIX подобные) от различных фирм - разработчиков (Red Hat, Debian, Novel, Mandrake soft, Gentoo, Slackware, IBM, Oracle, NetBSD, OpenBSD, FreeBSD)
* Операционные системы Mac OS от фирмы Apple
* Операционные системы Android семейства [UNIX-подобные ОС](https://ru.wikipedia.org/wiki/UNIX-%D0%BF%D0%BE%D0%B4%D0%BE%D0%B1%D0%BD%D0%B0%D1%8F_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0)



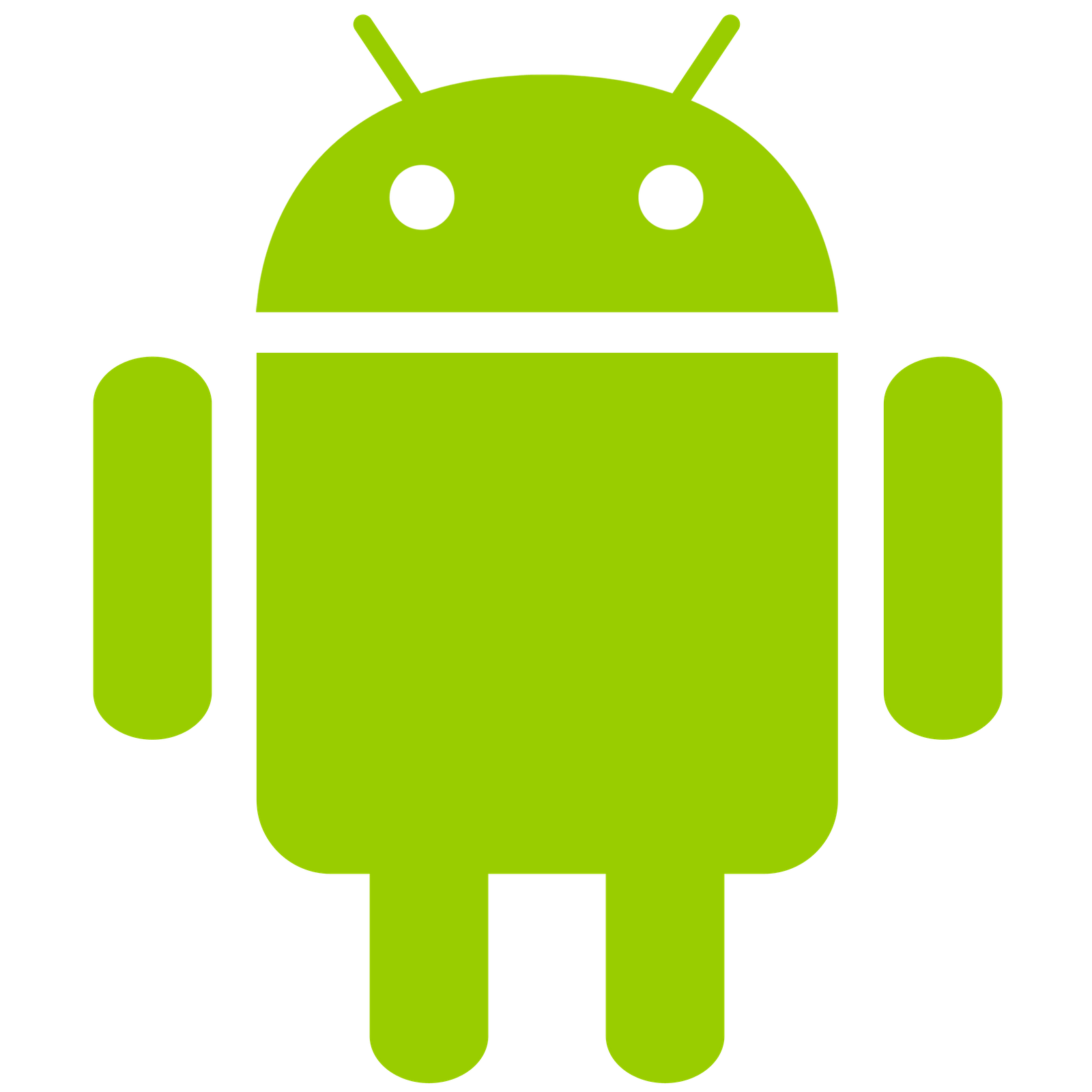


Рисунок 9. Эмблемы выше перечисленных Операционных систем(Windows/Linux/Apple/Android)

Для разработки программного приложения автоматизированной обработки выбор той или иной операционной системы не повлияет на функциональность системы по причине того, что при реализации алгоритмов программного приложения не требуется использования каких-либо специфических функций операционной системы.

Выбор системы управления баз данных (СУБД) представляет собой сложную многопараметрическую задачу и является одним из важных этапов при разработке приложений баз данных.

Основным принципом выбора СУБД следует считать определение программного продукта, в наибольшей мере соответствующего предъявляемым требованиям.

Таким образом, для проекта, рассматриваемого в данном дипломном проекте наиболее приемлема СУБД MySQL.

Кроме того, данная СУБД обладает следующими преимуществами:

* + многопоточность, поддержка нескольких одновременных запросов;
  + оптимизация связей с присоединением многих данных за один проход;
  + записи фиксированной и переменной длины;
  + ODBC драйвер;
  + гибкая система привилегий и паролей;
  + гибкая поддержка форматов чисел, строк переменной длины и меток времени;
  + интерфейс с языками C и Perl, PHP;
  + быстрая работа, масштабируемость;
  + совместимость с ANSI SQL;
  + бесплатна в большинстве случаев;
  + хорошая поддержка со стороны провайдеров услуг хостинга;
  + быстрая поддержка транзакций через механизм InnoDB.

Кроме СУБД, необходимы выбрать и язык веб-программирования.

В случае WEB-приложений речь идёт, конечно же, об ASP.NET и его основном в настоящее время конкуренте - РНР.



Рисунок 10. Эмблемы ASP.NET и PHP

Рассмотрим подробнее преимущества и недостатки обоих платформ.

Очевидные преимущества ASP.NET

Типизация. Языки программирования ASP.NET имеют строгую типизацию данных. Это безусловно выигрышный момент по сравнению с нетипизированным php: меньше будет логических ошибок, которые весьма трудно находить и исправлять. Некоторым утешением для сторонников php является возможность привести переменную к нужному типу - но увы, присвоение переменной, приведённой к целому типу, строкового значения не вызовет даже предупреждения со стороны интерпретатора.

Очевидные преимущества php

Доступность дистрибутивов. Дистрибутивы LAMP измеряются в десятках мегабайт (а не в DVD-дисках, как Windows/IIS/Visual studio/MS SQL Server) и доступны на сайтах разработчиков.

Open source (открытый код), представляющий возможность для аудита кода. Что особенно важно - аудитом кода занимаются программисты всего мира, способствуя устранению уязвимостей и прочих недостатков.

Поскольку обе технологии ограничены довольно простым протоколом HTTP и языком разметки HTML - их возможности примерно равны. Все задачи, решаемые в рамках одной технологии, столь же успешно решаются и в рамках другой. Можно спорить лишь о большем или меньшем удобстве предлагаемые технологиями решений - но мы воздержимся от споров о вкусах.

Таким образом, ни php, ни ASP.NET не дают технологического преимущества WEB-проекту. Различия проявляются в стоимости и трудоёмкости разработки и эксплуатации проекта. В этих показателях php значительно выгодней ASP.NET. А преимущества ASP.NET в области разработки и поддержки, провозглашаемые рекламой, в основном являются, увы, не более чем рекламой.

Кроссплатформенность. php портирован практически под все распространённые операционные системы, в то время как ASP.NET ориентирован на Windows, и то не всякую, а только 2000/XP/Vista.

Удачный набор функций. php предоставляет WEB-разработчику большое количество функций для решения типовых задач. Создатели php хорошо знают, какие задачи чаще всего решает разработчик WEB-приложений. В ASP.NET нет полезных функций, необходимых постоянно - при наличии огромного количества методов, которым и применения-то не придумать.

Поэтому для реализации проектируемой системы выберем язык программирования PHP.[6, 7].

3. Оценка технических решений

3.1. Описание интерфейсов программы.

Программа состоит из 4 частей:

1. Выбор Детали (Рисунок 11).
2. Выбор цикла (Рисунок 12).
3. Ввод данных. Для облегчения и во избежание каких-либо ошибок для ввода данных, для удобства ниже под всем параметрами появляется соответствующая схема детали. По которому можно сверять вводимые данные (Рисунок 13).
4. Выгрузка данных (Рисунок 14)

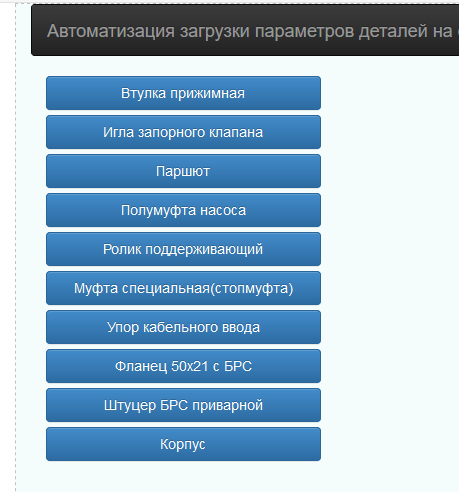


Рисунок 11. Интерфейс программы. Выбор детали

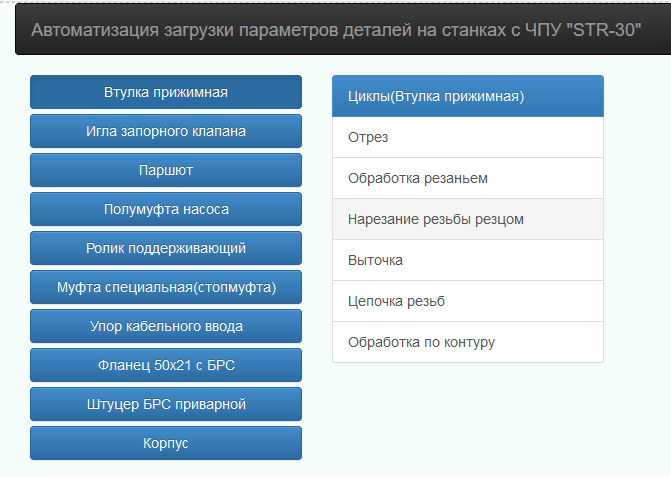


Рисунок 12. Интерфейс программы. Выбор цикла

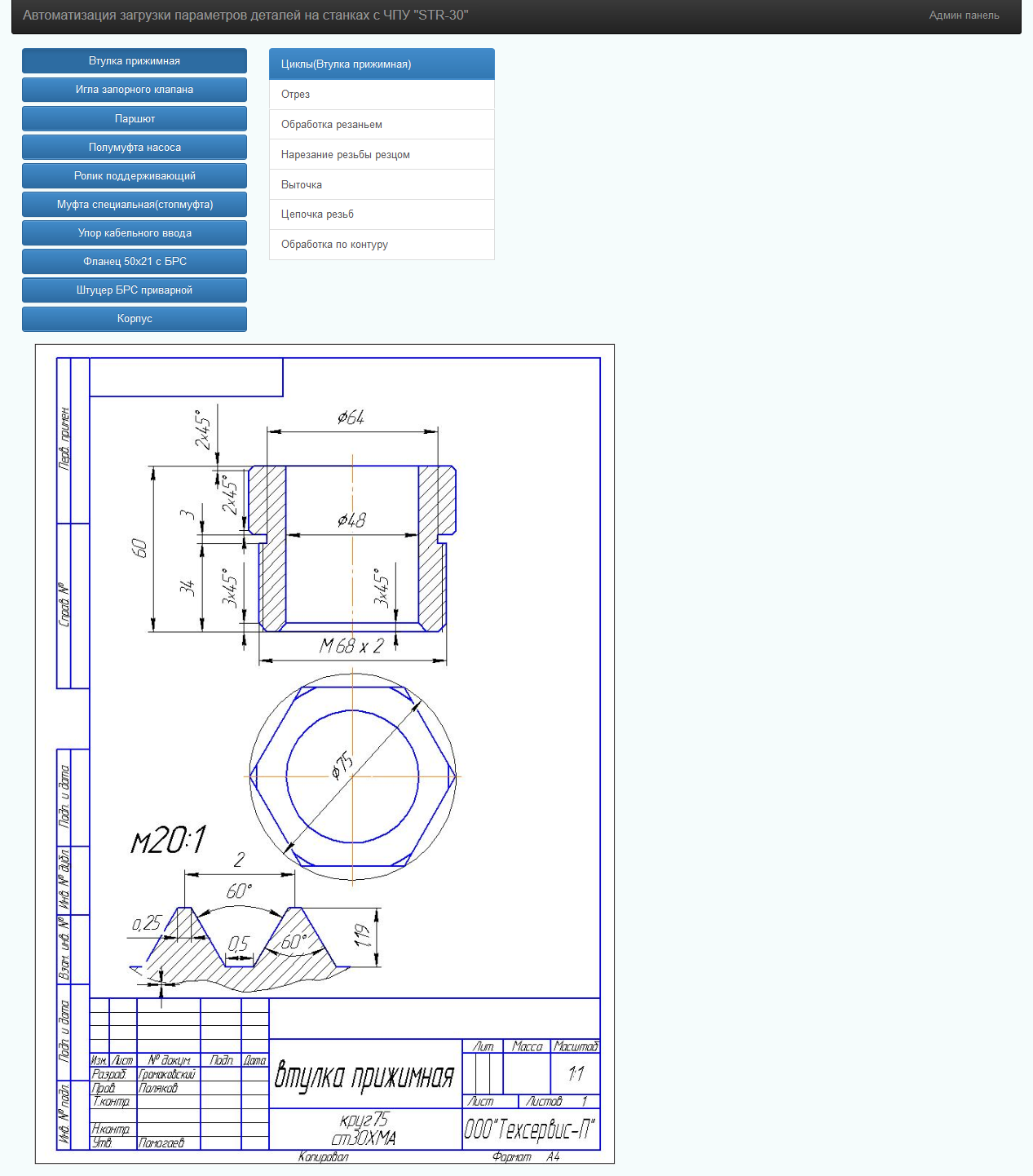


Рисунок 13. Интерфейс программы. Ввод данных

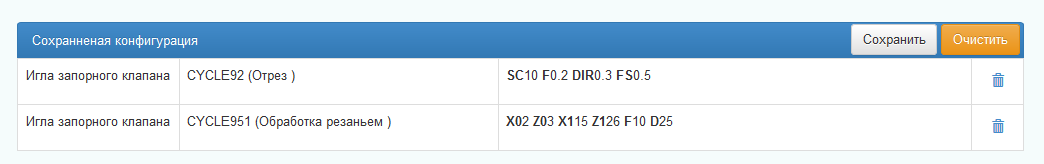


Рисунок 14. Интерфейс программы. Сохраненные данные

Так же на некоторых циклах необходимо вводить для одного параметра несколько данных, для этого необходимо вводить данные в поле через точку запятую «;». Это сделано что бы не добавлять в форму добавления еще дублирующих полей (Рисунок 15,16).

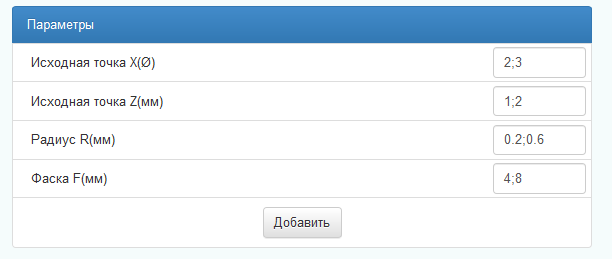


Рисунок 15. Вводим данные через точку запятую

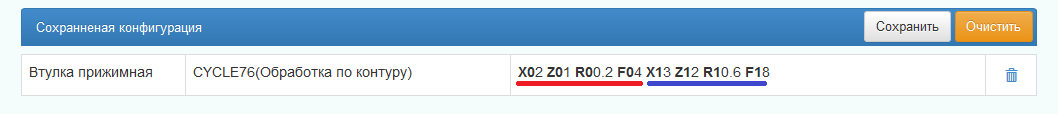


Рисунок 16. Результат сохранения данных после ввода через точку запятую

Все вышеперечисленные элементы и параметры можно изменить. Для этого была создана «Админ панель», страница где можно добавить/удалить детали, циклы, параметры к циклам. Если же кто то ошибся или какой-либо параметр был изменен, его можно изменить. Схемы для деталей можно добавить или удалить (Рисунок 17,18,19).

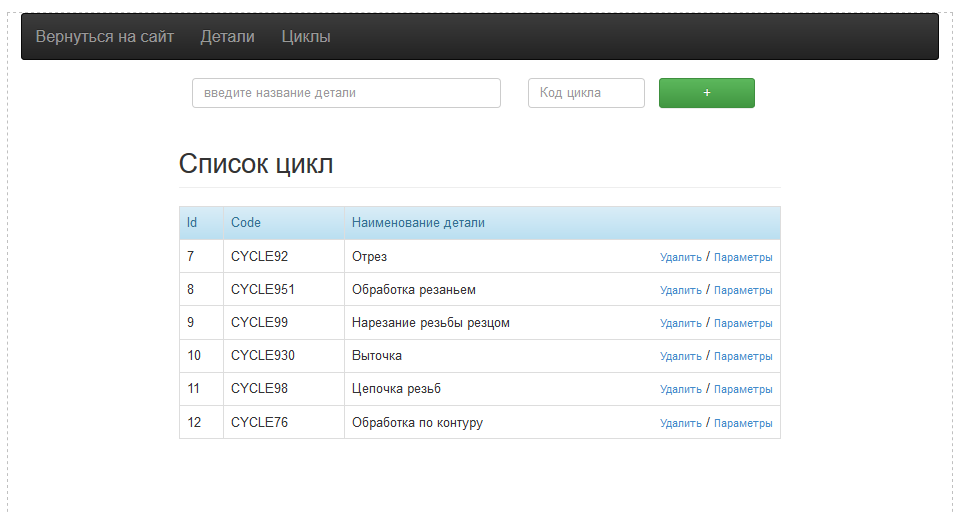


Рисунок 17. Админ панель. Циклы

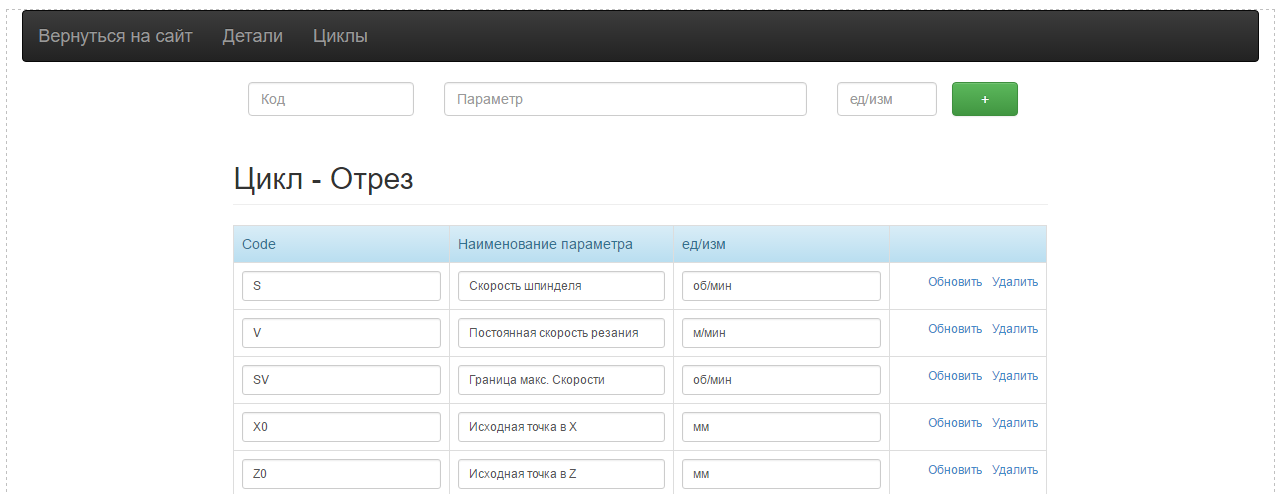


Рисунок 18. Админ панель. Параметры для выбранного цикла

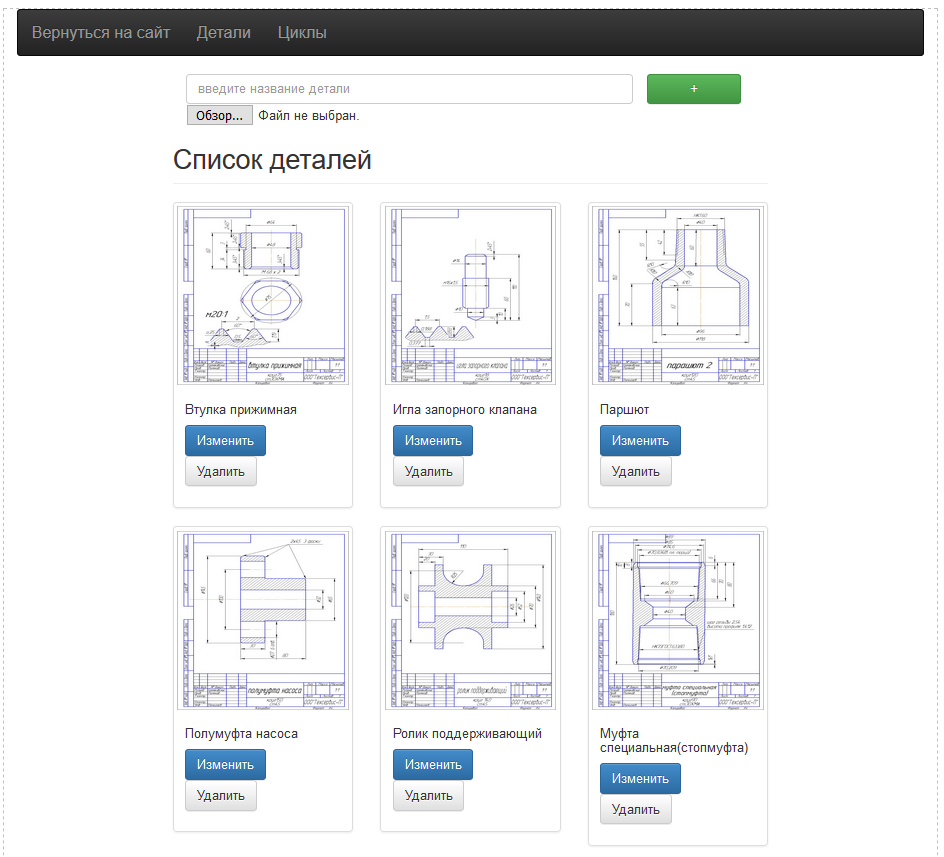


Рисунок 19. Админ панель. Детали

Дизайн программы адаптирован под мобильные устройства (Рисунок 20).

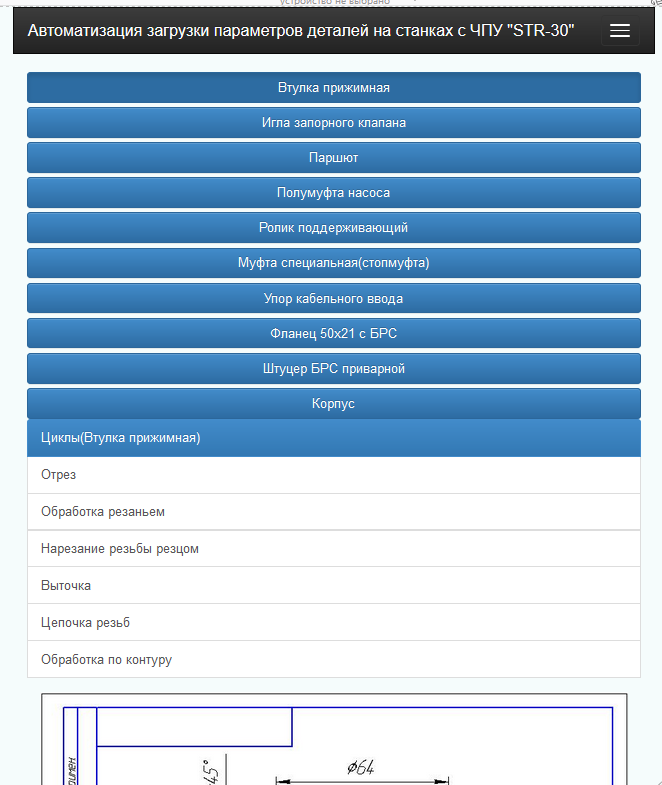


Рисунок 20. Адаптивный дизайн

3.2. Описание метода тестирования проектных решений.

Основная идея тестирования заключается в том, чтобы перечислить в некотором списке возможные ошибки или ситуации, в которых они могут появиться, а затем на основе этого списка написать тесты. Например, такая ситуация возникает при значении 0 на входе и выходе программы. Следовательно, можно построить тесты, для которых определенные входные данные имеют нулевые значения и для которых определенные выходные данные устанавливаются в 0. При переменном числе входов или выходов (например, число искомых входных записей при поиске в списке) ошибки возможны в ситуациях типа «никакой» и «один» (например, пустой список, список, содержащий только одну искомую запись). Другая идея состоит в том, чтобы определить тесты, связанные с предположениями, которые программист может сделать во время чтения спецификаций.

Во время тестирования программного обеспечения определились некоторые возможные ошибки. Основной частью этих ошибок оказалось то, что пользователь может ввести неверные данные. Например, ввести в поле параметров, буквенные символы и сохранить их. Решением было ограничить ввод в поле только цифры. Это делается двумя способами:

* Для поля ввода установить тип «number», но это работает только в браузерах которые поддерживают технологию HTML5.
* Для остальных типов браузеров эту проблему решили при помощи JavaScript кода.

Вторая проблема возникшая, это было ввод или изменении данных в базе данных через “Админ панель”. Пользователь мог ввести дублирующие данные, или во все не ввести ничего в поле данных. Для того что бы наша программа при добавления или изменении не дублировала раннее добавленные записи, в программа перед добавлением или изменением проверяет на наличие дублирующих строк после чего будет водиться ошибка. Ошибка будет выводиться и при добавлении пустых строк (Рисунок 21, 22).

Уникальными должны быть:

* Для деталей, наименование детали(в БД “name”)
* Для циклов, код и наименования цикла( в БД “code, name”)
* Для параметров, код и наименования параметра( в БД “code, name”)

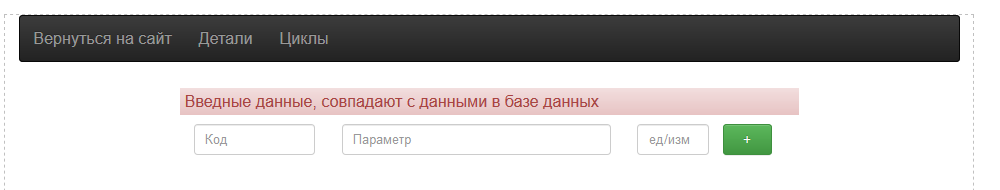


Рисунок 21. Ошибка при добавлении дублируюшей строки

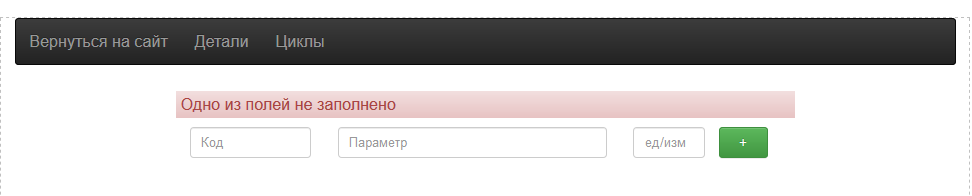


Рисунок 22. Ошибка при добавлении пустого поля

Все тестирования и проверка работоспособности проектируемой системы проводились на станке с ЧПУ Sinumerik 828D BASIC T. Проблем во время загрузки данных в ЧПУ не возникло. Все возможные проблемы и ошибки были исправлены во время ввода данных для загрузки в ЧПУ, ошибки и методы их решения были описаны выше. Все выходные данные с проектируемой системы ЧПУ принимал без каких-либо ошибок. Единственное ЧПУ сам преобразовывал входные данные в нужный ему формат, например в параметрах место «0» он выводил «0.000», так же и с другими параметрами.

Для проверка работоспособности был взят чертеж «Втулка прижимная», и провели 2 токарных цикла «выточка» и «обработка резанием» (Рисунок 23, 24, 25, 26, 27, 28).

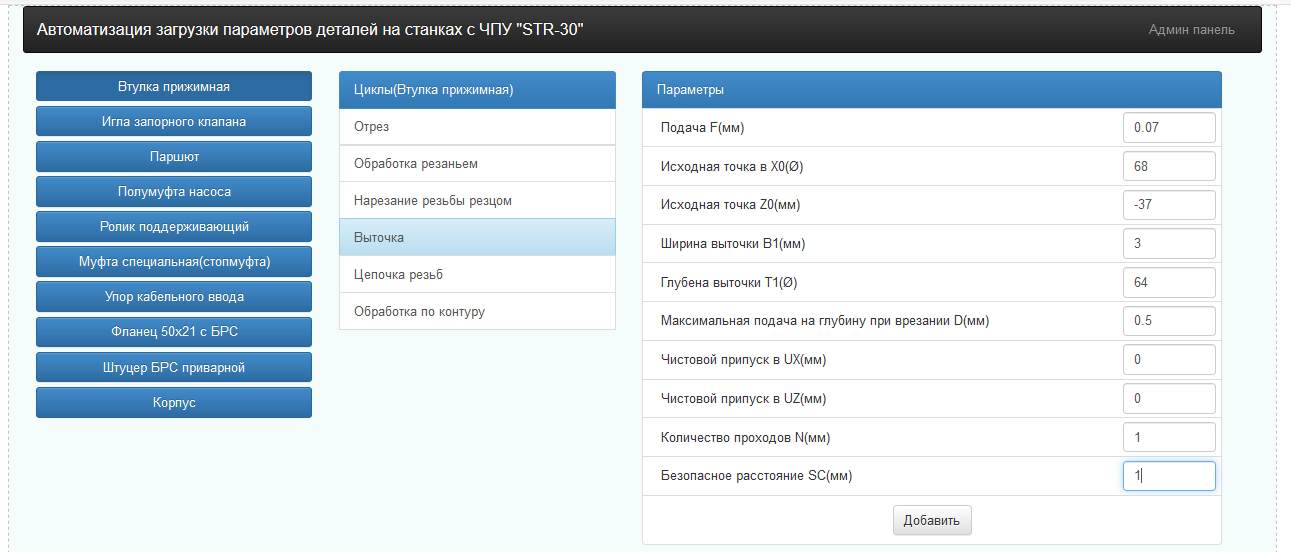


Рисунок 23. Ввод параметра для цикла «Выточка»

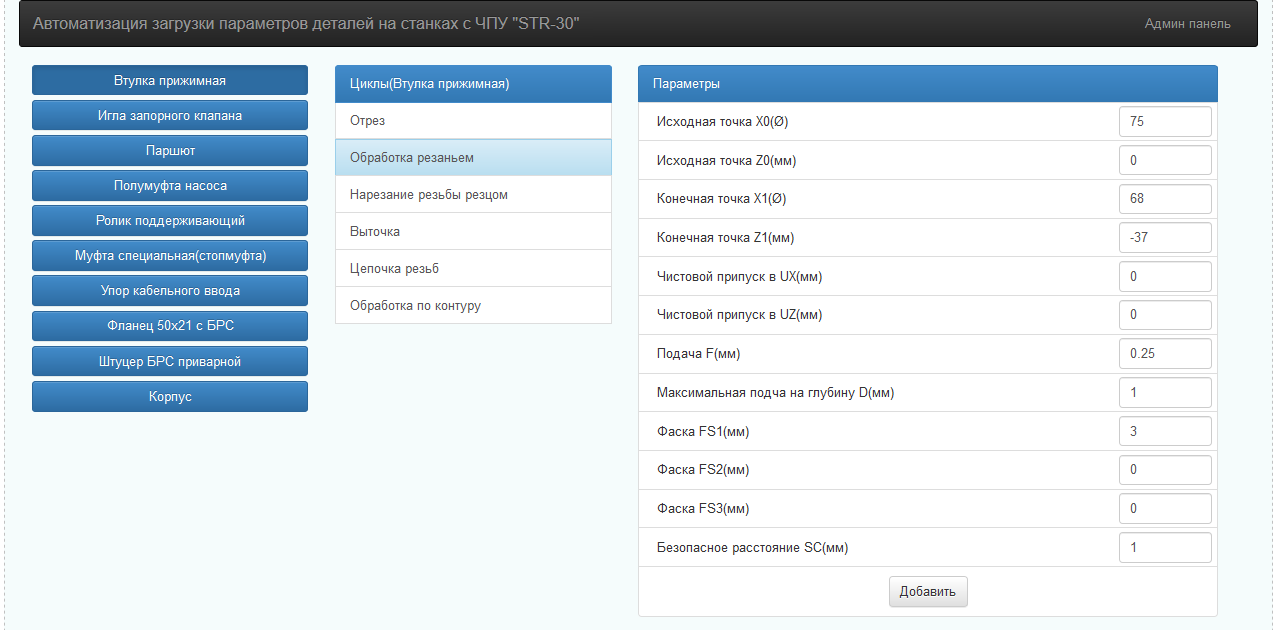


Рисунок 24. Ввод параметра для цикла «Обработка резанием»

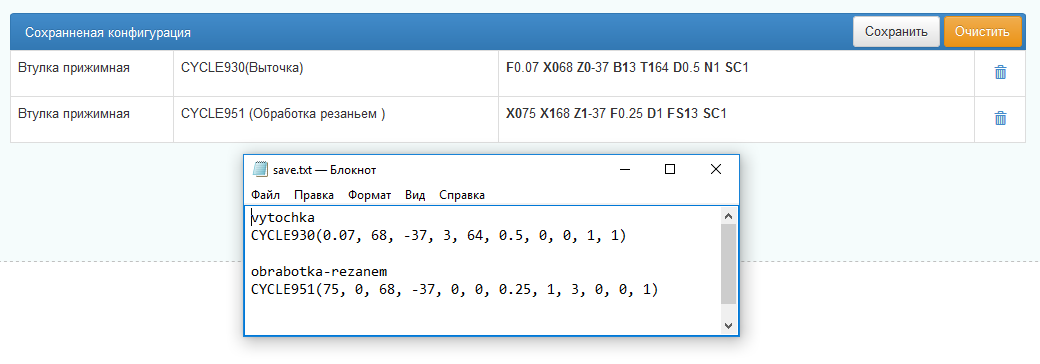


Рисунок 25. Выходные данные

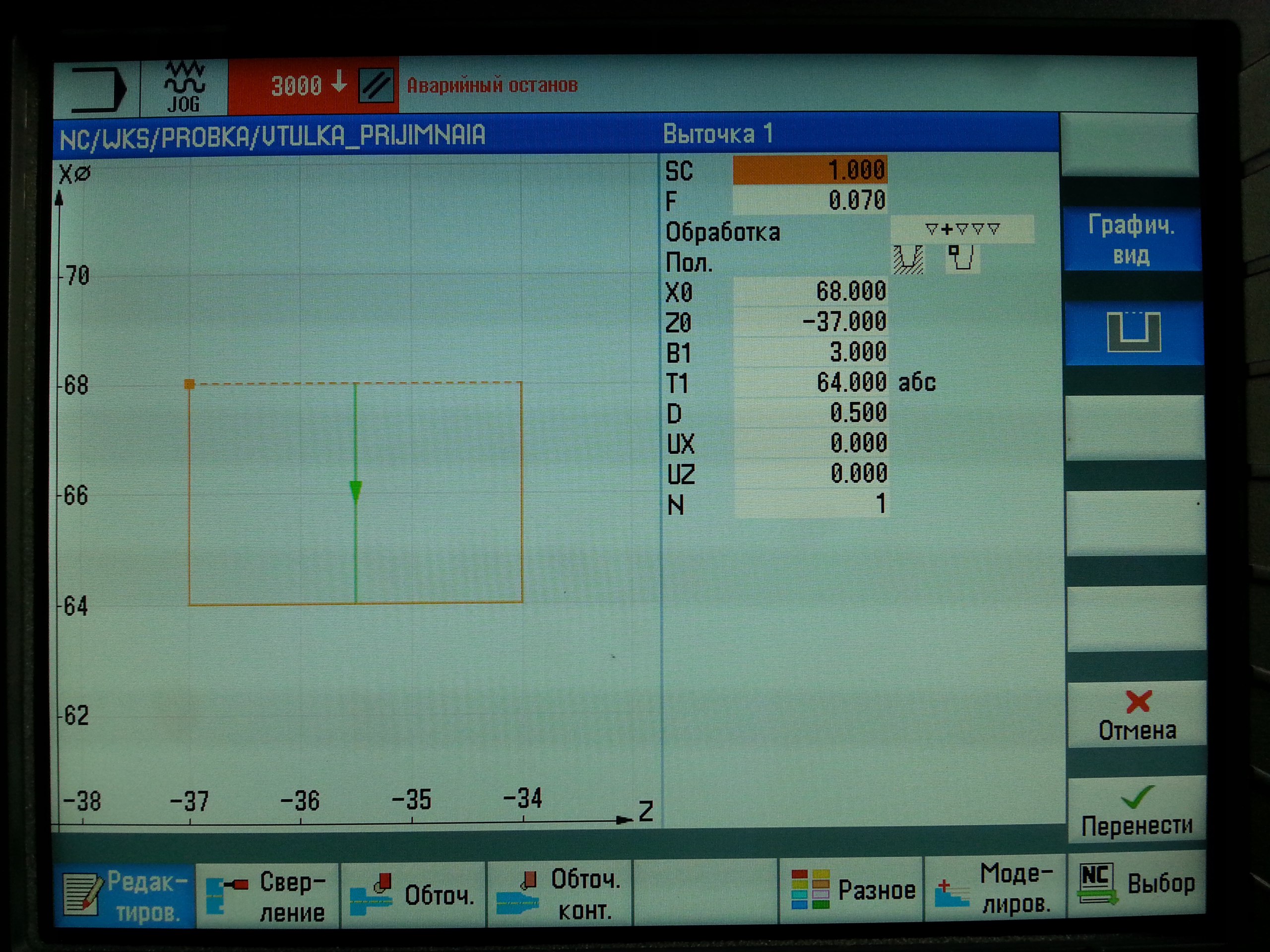


Рисунок 26. Отображение параметров после загрузки данных, циклы «Выточка»

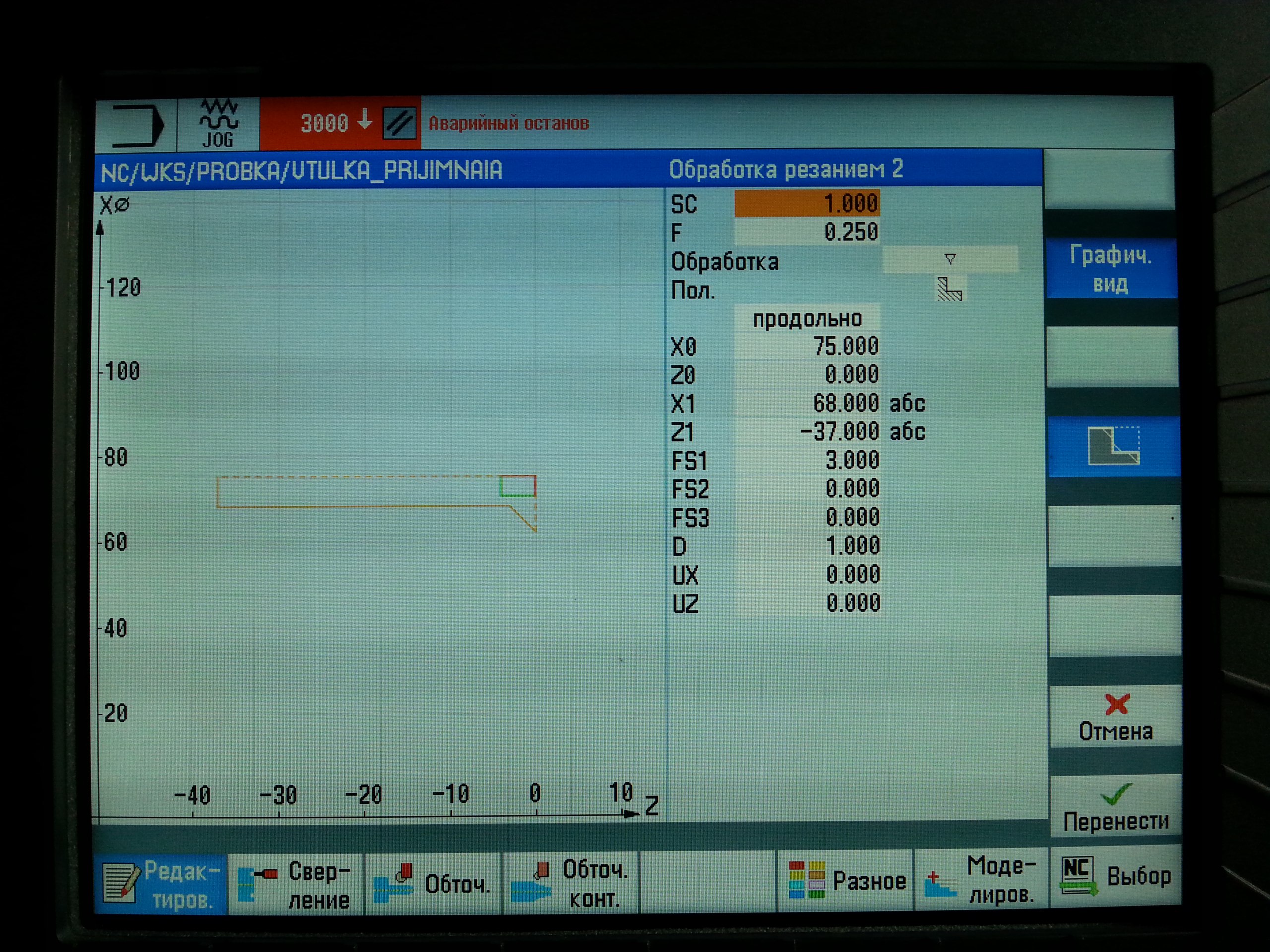


Рисунок 27. Отображение параметров после загрузки данных, циклы «Обработка резанием»



Рисунок 28. Результат работы, после запуска ЧПУ

Список литературы

1. http://pereosnastka.ru/articles/obshchie-svedeniya-o-stankakh-s-programmnym-upravleniem
2. http://itteach.ru/bpwin/metodologiya-idef3
3. https://infofrezer.ru/stati/kak-pravilno-ispolzovat-fleshki-pri-zagruzke-upravlyayushchikh-programm-dlya-frezernykh-stankov-s-chpu
4. http://www.zfkai.ru/sveden
5. Учебно-методическое пособие по дисциплине Обработка на станках с ЧПУ.
6. http://studbooks.net/2088636/informatika/obosnovanie\_proektnyh\_resheniy\_programmnomu\_obespecheniyu
7. https://ru.wikipedia.org