# МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

# Федеральное государственное бюджетное образовательное учреждение

# высшего образования

«КОВРОВСКАЯ ГОСУДАРСТВЕННАЯ ТЕХНОЛОГИЧЕСКАЯ АКАДЕМИЯ

имени в.а. дегтярева»

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

Тихомирова Сергея Михайловича

Студента

Автоматики и электроники

Факультет

09.03.01 Информатика и вычислительнаятехника

Направление подготовки

ТЕМА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ:

Разработка расчетного модуля в автоматизированной системе расчета припусков на обработку типовых деталей

Руководитель ВКР

*Зяблицев Виталий Васильевич*

Допустить выпускную квалификационную работу к защите  
в Государственной экзаменационной комиссии

Заведующий кафедрой

*Котов В.В.*

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2017 г.

Общей задачей является разработка программного обеспечения для автоматизированного расчёта размерных цепей при обработке типовых деталей. В данной работе рассматривается разработка расчетного модуля в автоматизированной системе расчета припусков на обработку типовых деталей.

Перед началом работы были поставлены следующие задачи:

1. Анализ традиционного процесса решения задачи. Сбор исходных данных, необходимых для расчета.
2. Разработка алгоритма расчета.
3. Разработка программного обеспечения для реализации расчета припусков на обработку детали.
4. Апробация программного обеспечения.

В выпускной квалификационной работе реализована система, предназначенная для расчета припусков на обработку типовых деталей.

Объём пояснительной записки –125 страниц; Объём иллюстраций – 15 и таблиц – 4,

Объём графической части – 14 листов; Объем списка литературы – 24 источника.

The common goal is the development of software for the automated calculation of dimensional chains in the processing of typical parts. In this paper, we consider the development of a calculation module in an automated system for calculating allowances for the processing of typical parts.

Before the beginning of work the following tasks were set:

1. Analysis of the traditional process of solving the problem. Collection of input data required for calculation.

2. Development of the calculation algorithm.

3. Development of software for the implementation of the calculation of allowances for the processing of parts.

4. Approbation of software.

In the final qualification work, a system was implemented to calculate allowances for the processing of typical parts.

The volume of the explanatory note is 125 pages; The volume of illustrations - 15 and tables - 4,

The volume of the graphic part is 14 sheets; The volume of the list of references is 24 sources.

**Содержание**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Введение | | | | 5 |
| 1. | Анализ традиционного процесса решения задачи. | | | 7 |
|  | 1.1 | Место задачи в структуре технологического проектирования. | | 8 |
|  | 1.2 | Методика расчёта размерных цепей. | | 10 |
|  | 1.3 | Декомпозиция задачи. | | 25 |
|  | 1.4 | Алгоритмизация. | | 31 |
|  |  | 1.4.1 | Расчет пространственного отклонения. | 31 |
|  |  | 1.4.2 | Расчет погрешности установки. | 32 |
|  |  | 1.4.3 | Расчет минимального припуска. | 32 |
|  |  | 1.4.4 | Расчет номинального припуска. | 33 |
|  |  | 1.4.5 | Расчет размеров после каждой операции. | 34 |
|  | 1.5 | Обоснование и выбор средств реализации задачи. | | 35 |
|  |  | 1.5.1 | Описание информационной среды предприятия. | 35 |
|  |  | 1.5.2 | Обоснование выбора языка программирования. | 36 |
|  |  | 1.5.3 | Обоснование выбора СУБД | 38 |
| 2. | Особенности реализации задачи. | | | 40 |
|  | 2.1 | Хранение выбранных параметров. | | 41 |
|  | 2.2 | Особенности подготовки параметров к расчету. | | 48 |
|  | 2.3 | Особенности расчета. | | 51 |
|  |  | 2.3.1 | Расчет точности | 51 |
|  |  | 2.3.2 | Расчет пространственного отклонения. | 51 |
|  |  | 2.3.3 | Расчет погрешности установки. | 52 |
|  |  | 2.3.4 | Расчет припуска. | 52 |
|  |  | 2.3.5 | Расчет размеров после каждой операции. | 53 |
|  |  | 2.3.6 | Возвращение результатов. | 53 |
|  | 2.4 | Особенности организации технического обеспечения. | | 56 |
| 3. | Описание программы. | | | 58 |
|  | 3.1 | Описание выполнения программы. | | 59 |
|  |  | 3.1.1 | Руководство пользователя. | 59 |
|  |  | 3.1.2 | Руководство администратора. | 62 |
|  | 3.2 | Тестирование. | | 67 |
| Заключение. | | | | 72 |
| Библиографический список. | | | | 73 |
| Приложения. | | | |  |
|  |  | Приложение 1. | |  |
|  |  | Приложение 2. | |  |

**Введение**

Технологическая размерная цепь − последовательность размеров, получаемых в процессе изготовления детали, для каждой обрабатываемой поверхности.

Припуск –слой материала, удаляемый с поверхности заготовки в процессе механической отработки для достижения требуемого качества и точности обрабатываемой поверхности.

На практике применяют два метода определения припусков: расчетно-аналитический и опытно-статистический (табличный) метод. Опытно –статистические величины припусков обычно завышены, т.к. они не учитывают конкретные особенности производства. Назначенные табличные припуски могут быть откорректированы технологом в зависимости от его опыта и квалификации.

Использование расчетно-аналитического метода затруднено из-за большого количества факторов, которые должны быть учтены. Задача автоматизации расчета припусков является актуальной для машиностроительного производства, что подтверждается заказом на разработку соответствующего программного обеспечения от одного из машиностроительных предприятий г. Коврова.

Решение поставленной задачи включает в себя:

1.​ Разработку алгоритма расчета.

2.​ Разработку структуры хранения справочных данных, используемых в расчете. Предполагается возможность корректировки и дополнения данных, введение корректирующих коэффициентов.

3.​ Разработку структуры хранения результатов расчета. Предполагается возможность поиска данных по различным критериям, передача данных для проведения повторного расчета на основе образца.

4.​ Разработку программного обеспечения для реализации расчета припусков на обработку детали, обеспечивающего удобный интерфейс пользователя.

Согласно разработанному алгоритму для расчёта размерной цепи понадобятся входные данные, которые можно разделить на 3 группы:

1.​ Исходные данные для каждой обрабатываемой поверхности: длина, диаметр, параметры обработки (тип припуска, тип обрабатываемой поверхности).

2.​ Выбор заготовки: вид заготовки, точность, шероховатость, глубина дефектного слоя, допускаемое смещение по поверхности разъёма штампа/литейной формы.

3.​ Выбор последовательности операций. Для каждой операции задаются: вид операции, используемое приспособление, шероховатость.

Для каждого перехода вычисляется диаметр с учётом допусков в зависимости от параметров обработки.

Представляемая программа реализует разработанный алгоритм, предоставляет удобный интерфейс для ввода данных и просмотра полученных результатов. Результаты расчета можно сохранить в текстовом файле и в базе данных.

Изм

Лист

№ докум.

Подп.

Лист

Дата

Лит.

Листов

Разраб.

Провер.

Тихомиров

Зяблицев

КГТА  
И-113

АНАЛИЗ ТРАДИЦИОННОГО ПРОЦЕССА РЕШЕНИЯ ЗАДАЧИ

Реценз.

Н.контр.

Утв.

Котов В.В.

ВКР.КГТА.090301.12 ПЗ

У

Инв. № подп.

Подп. и дата

Взам. инв. №

Инв. № дубл.

Подп. и дата

Перв. примен.

Справ. №

1. АНАЛИЗ ТРАДИЦИОННОГО ПРОЦЕССА РЕШЕНИЯ ЗАДАЧИ

* 1. **Место задачи в структуре технологического проектирования**

Технологические процессы разрабатывают согласно ГОСТ 14.301-83 в определенной последовательности: анализ исходных данных, выбор действующего типового, группового технологического процесса или аналога единичного технологического процесса, выбор исходной заготовки и метода ее изготовления, выбор технологических баз, со­ставление технологического маршрута, разра­ботка технологических операций, нормирование технологического процесса, оформление технологического процесса.

Результатом технологического проектирования является комп­лект технологических документов, оформленных согласно ЕСТД [7].

При этом в записке кроме комплекта документов на технологический процесс механической обработки детали (и в случае необходимости технологического процесса и схемы сборки) должны присутствовать следующие разделы:

* Анализ технологичности конструкции детали
* Определение типа производства
* Выбор типового, группового или действующего единичного техпроцесса
* Анализ базового технологического процесса
* Выбор и проектирование прогрессивной заготовки
* Определение последовательности обработки поверхностей детали
* Разработка технологического процесса механической обработки
* Выбор оборудования
* Выбор приспособлений
* Выбор режущего и вспомогательного инструмента
* Выбор мерительного инструмента
* Выбор смазывающе-охлаждающих технологических сред
* Формирование операций
* Выбор технологических баз
* Размерный анализ технологических операций
* **Расчёт межоперационных припусков и размеров**
* Разработка операционных эскизов
* Расчёт режимов резания.
* Техническое нормирование операций механической обработки
* Проектирование технологических наладок
* Проектирование операций технического контроля
* Оформление технологического процесса.

Нашей задачей является автоматизация научно-обоснованного расчёта межоперационных припусков и исполнительных размеров, что приведёт к повышению эффективности подготовки производства.

* 1. **Методика расчёта размерных цепей**

В своей разработке мы использовали методику расчета припусков, разработанную на кафедре технологии машиностроения.

Размерная цепь представляет собой совокупность технологических размеров в принятой последовательности выполнения технологических переходов и операций.

*Ввести исходные данные*

* Ввести длину детали (из чертежа) – *l, м*
* Ввести заданный размер для диаметра – *d,* не диаметр - *L* (принимается из чертежа детали), мм.
* Ввести допуск на заданный размер *Td* или *TL* (принимается из чертежа детали), мм.
* Ввести шероховатость обработанной поверхности (принимается из чертежа детали и переводится в параметр Rz, мкм).
* Ввести значение заданной на чертеже глубины упрочнённого слоя *hy* (например при поверхностной закалке, деформационном упрочнении, азотировании, цементации и т.д.), по умолчанию *hy=0.*
* Выбрать тип размера: вал, отверстие (для отверстия задать глубину отверстия *lо)*.
* Выбрать тип обработанной поверхности: цилиндрическая, плоская, комбинированная, фасонная (фаска).

# Выбрать тип припуска: двусторонний, односторонний.

* Выбрать вид заготовки: прокат, поковка (ввести из чертежа поковки допускаемое смещение по поверхности разъёма - *ρсм*, мм), отливка (ввести из чертежа отливки допускаемое смещение по поверхности разъёма - *ρсм*, мм).
* Задать из техпроцесса изготовления детали последовательность операций обработки с указанием шероховатости обработанной поверхности (с указанием приспособления только при смене установки, иначе приспособление не изменяется либо его нет в списке и погрешность установки e = 0) : точение (черновое, получистовое, чистовое, тонкое), сверление и рассверливание, зенкерование (черновое, чистовое), развертывание (черновое, чистовое, тонкое), протягивание (черновое, чистовое), шлифование (черновое, чистовое, тонкое), фрезерование (черновое, чистовое, тонкое), электроэрозионная обработка (черновая, чистовая, отделочная), притирка, суперфиниширование, хонингование, полирование.

Для принятой последовательности операций записать последовательность переходов обработки исходя из заданной шероховатости.

Например:

Точение, шлифование Rа 0,32

Из таблицы определить метод получения требуемой шероховатости и записать последовательность переходов обработки в обратном направлении (шлифование чистовое - шлифование черновое - точение чистовое - точение черновое).

Точение, шлифование, доводка Rа 0, 1

(доводка - шлифование чистовое - шлифование черновое - точение чистовое - точение черновое).

*Присвоить операции переходу порядковый номер.*

Например: 0-заготовка, 1- черновое точение, 2-чистовое точение, 3-черновое шлифование и 4-чистовое шлифование.

*Определить для каждого перехода составляющие припуска: Rzi, hi, Ti.*

Из базы данных: табл.1.1, табл. 1.2, табл.1.3, табл.1.4.

*Определить составляющую припуска ρi для каждого перехода.*

Подпрограмма 1.

*Определить составляющую припуска εi для каждого перехода.*

Подпрограмма 2.

*Определить zi min для каждого перехода.*

*zi min = Rzi-1 +hi-1 +ρi-1+ εi,*

при последовательной обработке противолежащих поверхностей (для плоских поверхностей);

*,*

при обработке внутренних и наружных поверхностей вращения.

*Определить zi nom* *для каждого перехода.*

*2zi nom =2zi min + Tdi-1 или 2zi nom =2zi min + TLi-1*

для двустороннего припуска;

*zi nom =zi min +TLi-1,*

для одностороннего припуска.

*Определить межоперационный размер* *для каждого перехода.*

Для размера тип вал:

*di-1=di +2zi nom,* или *Li-1=Li +2zi nom,*

для двустороннего припуска;

*Li-1=Li+zi nom;*

для одностороннего припуска.

Для размера тип отверстие:

*di-1=di -2zi nom,* или *Li-1=Li -2zi nom,*

для двустороннего припуска;

*Li-1=Li-zi nom;*

для одностороннего припуска.

*Вывести полученные значения Rzi, hi,ρi, εi,* *Tdi, TLi, zi nom, di Li* для каждого перехода (операции) в виде таблицы.

*Построить размерную цепь* на основе полученных значений *Tdi, TLi, zi nom, di Li*

Если заготовка подвергается поверхностному упрочнению, которое для рассматриваемой поверхности должно быть удалено при обработке, то толщина упрочненного слоя прибавляется к припуску на чистовую обработку, которая обычно выполняется после черновой обработки на втором переходе.

Если hy>0, то z2min= z2min+hy

Подпрограмма 1.

*Определение составляющей припуска ρi для каждого перехода.*

*Определяем,ρ0 в зависимости от вида заготовки.*

Прокат

*ρ0 =Δk l,*

*Δk* - удельная кривизна, см. табл.1.8, *l –* длина детали.

Поковка



*ρсм* – допускаемое смещение по поверхности разъёма штампа (задаётся на чертеже поковки),

*ρk -* допускаемое коробление; *ρk = Δk l;* Δ*k* -удельная кривизна, см. табл. 1.8,

*l –* длина детали.

Литьё по выплавляемым моделям, литьё в кокиль, литьё под давлением



*ρсм* – допускаемое смещение по поверхности разъёма литейной формы (задаётся на чертеже отливки); *ρk -* допускаемое коробление; *ρk = Δk l;* Δ*k* -удельная кривизна, см. табл. 1.8; *l –* длина детали.

Сверление (вместо *ρо*использовать *ρу*)

*ρу* - увод сверла; *ρу = Δk lо;* Δ*k* -удельный увод, см. табл. 1.9; *lо –* глубина отверстия.

*Определяем,ρi для каждого перехода*

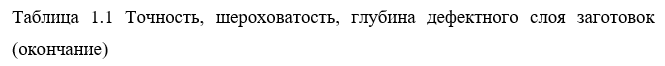
*ρi = ρ0 ky*

*ky* – коэффициент уточнения, см. табл.1.5.

*Определение составляющей припуска εi для каждого переход, табл. 1.6*

Таблица 1.1 Точность, шероховатость, глубина дефектного слоя заготовок

|  |  |  |  |
| --- | --- | --- | --- |
| Способ получения заготовки | Точность, IT | Шероховатость, Rz,  мм | Глубина дефектного слоя *h*, мм |
| Литьё в песчано-глинистые формы | 16 | 0,300 | 0,300 |
| Литьё в оболочковые формы | 15 | 0,040 | 0,160 |
| Литьё по выплавляемым моделям | 12 | 0,032 | 0,100 |
| Литьё в кокиль | 15 | 0,200 | 0,200 |
| Литьё под давлением в металлические формы | 12 | 0,040 | 0,100 |
| Центробежное литьё | 15 | 0,200 | 0,200 |
| Горячая объёмная штамповка на молотах, прессах и гкм (поковка) | 17 | 0,160 | 0,200 |
| Холодное выдавливание (поковка) | 14 | 0,40 | 0,040 |



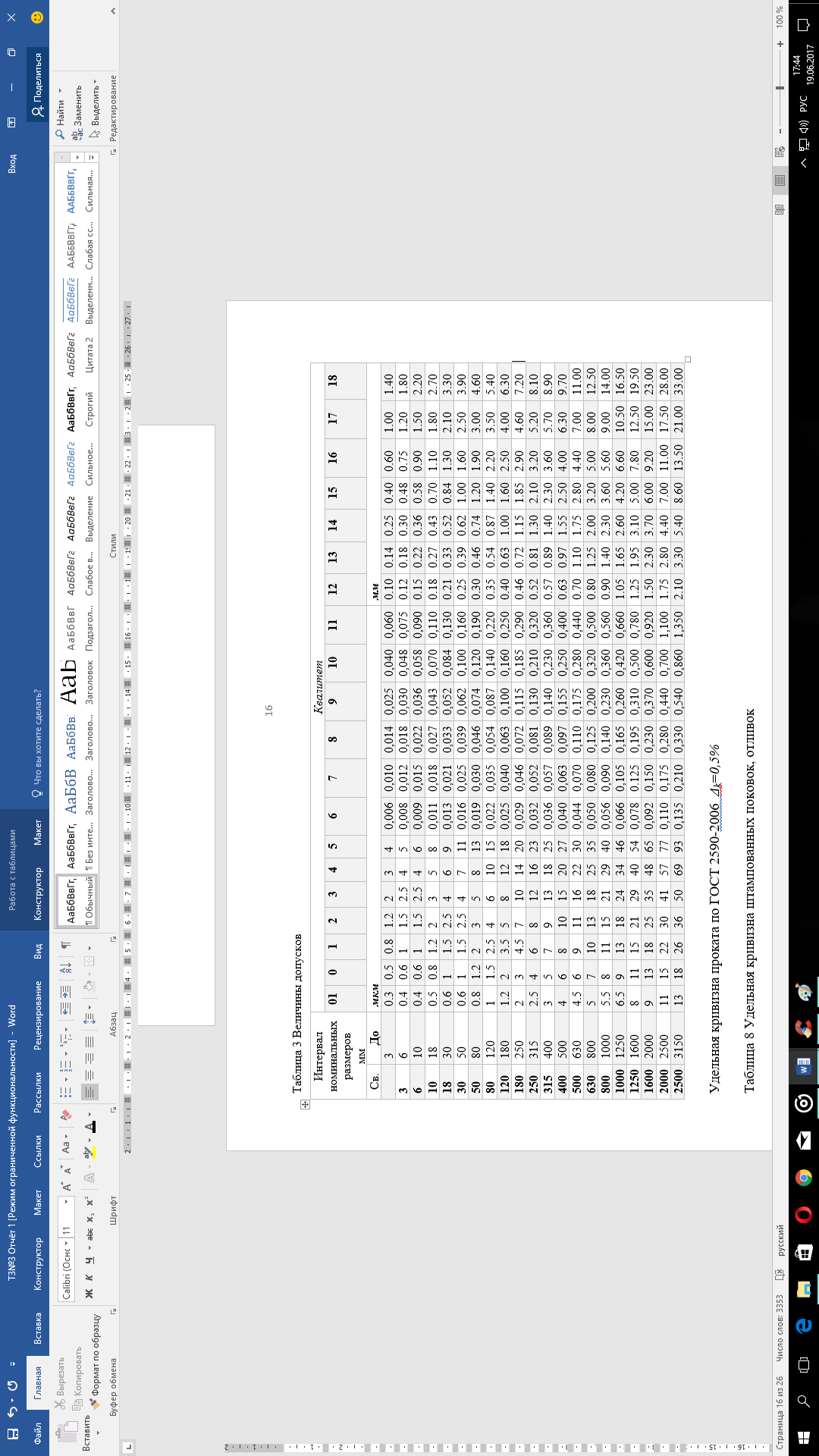
|  |  |  |  |
| --- | --- | --- | --- |
| Способ получения заготовки | Точность, IT | Шероховатость,  Rz, мм | Глубина дефектного слоя *h*, мм |
| прокат Обычная точность, В1 | 16 | 0,2 | 0,3 |
| прокат Повышенная точность, Б1 | 15 | 0,125 | 0,15 |
| прокат Высокая точность, А1 | 14 | 0,075 | 0,1 |

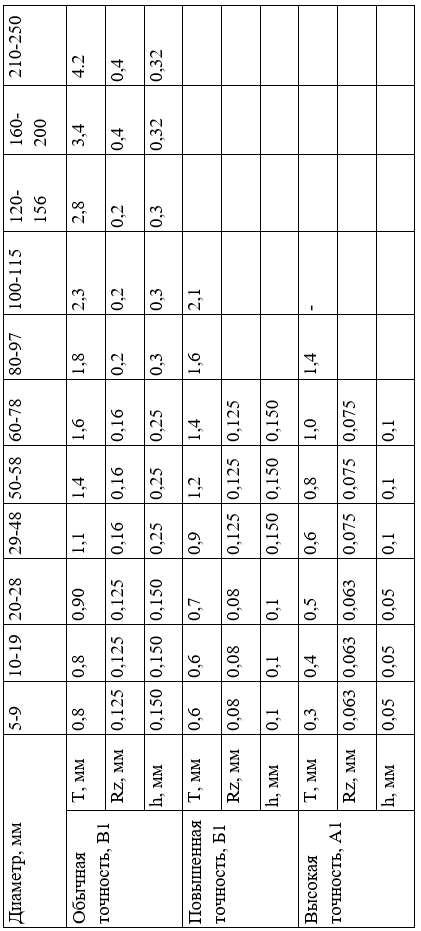
Таблица 1.2 Точность, шероховатость, глубина дефектного слоя поверхностей после различных методов обработки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Метод обработки | Достигаемая точность, IT | Достигаемая шероховатость, *Rz*, мм | Глубина дефектного слоя, *h*, мм | *ky* |
| Точение и растачивание: |  |  |  |  |
| черновое | 14 | 0,040 | 0,060 | 0,06 |
| получистовое | 12 | 0,020 | 0,040 | 0,05 |
| чистовое | 10 | 0,010 | 0,030 | 0,04 |
| тонкое | 7 | 0,005 | 0,010 | 0 |
| Сверление | 10 | 0,024 | 0,040 | 1 |
| Зенкерование: |  |  |  | 0 |
| черновое | 12 | 0,030 | 0,040 | 0 |
| чистовое | 8 | 0,010 | 0,015 | 0 |
| Развертывание: |  |  |  | 0 |
| черновое | 10 | 0,010 | 0,015 | 0 |
| чистовое | 7 | 0,0024 | 0,010 | 0 |
| тонкое | 5 | 0,001 | 0,005 | 0 |
| Протягивание: |  |  |  | 0 |
| черновое | 10 | 0,005 | 0,015 | 0 |
| чистовое | 7 | 0,002 | 0,010 | 0 |
| Шлифование: |  |  |  |  |
| черновое | 8 | 0,010 | 0,020 | 0,03 |
| чистовое | 7 | 0,005 | 0,010 | 0,02 |
| тонкое | 6 | 0,001 | 0,005 | 0 |
| Фрезерование: |  |  |  |  |
| черновое | 14 | 0,040 | 0,040 | 0,06 |
| чистовое | 10 | 0,020 | 0,020 | 0,04 |
| тонкое | 6 | 0,005 | 0,010 | 0 |
| Строгание: |  |  |  |  |
| черновое | 14 | 0,040 | 0,050 | 0,06 |
| чистовое | 11 | 0,020 | 0,025 | 0,04 |
| тонкое | 8 | 0,010 | 0,010 | 0 |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Метод обработки | Достигаемая точность, IT | Достигаемая шероховатость, *Rz*, мм | Глубина дефектного слоя, *h*, мм | *ky* |
| Притирка | 4 | 0,0005 | 0,004 | 0 |
| Суперфиниширование | 5 | 0,0005 | 0,004 | 0 |
| Хонингование | 5 | 0,0005 | 0,004 | 0 |
| Раскатывание | 6 | 0,0008 | 0 | 0 |
| Калибрование | 5 | 0,0005 | 0 | 0 |
| Алмазное выглаживание | 5 | 0,0005 | 0 | 0 |
| Полирование | 0 | 0,00032 | 0 | 0 |
| Электроэрозионная обработка: |  |  |  |  |
| черновая | 14 | 0,040 | 0,080 | 0 |
| чистовая | 10 | 0,020 | 0,040 | 0 |
| отделочная | 8 | 0,005 | 0,020 | 0 |

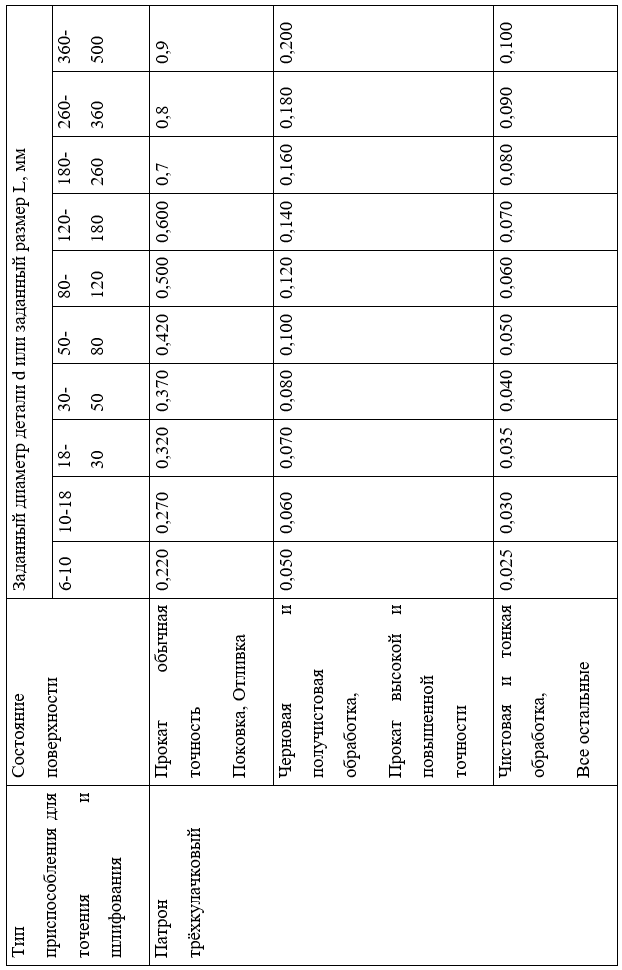


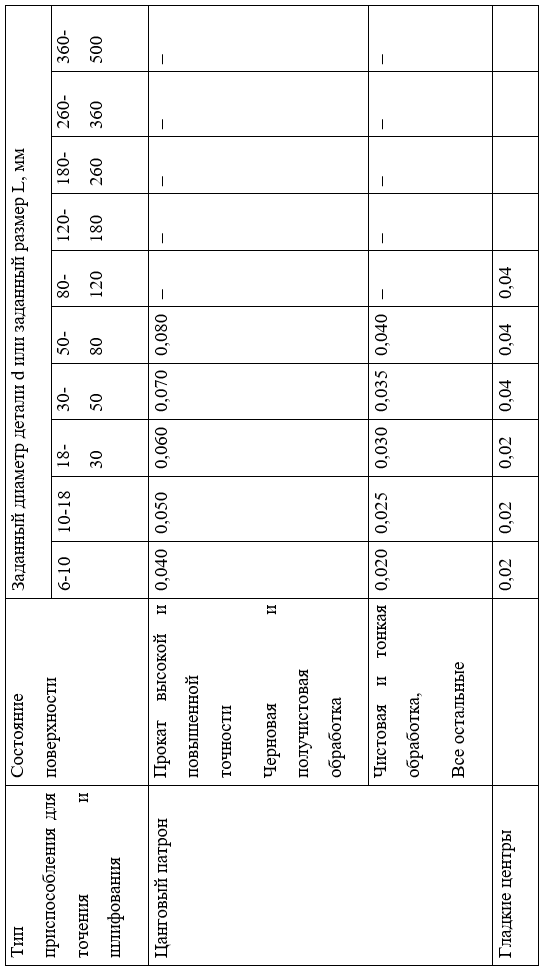


Удельная кривизна проката по ГОСТ 2590-2006 *Δk=0,5%*

Таблица 1.5 Коэффициент уточнения, для остальных *ky=0*

|  |  |
| --- | --- |
| Технологический переход | *ky* |
| Точение черновое | 0,06 |
| Точение чистовое | 0,04 |
| Шлифование черновое | 0,03 |
| Шлифование чистовое | 0,02 |





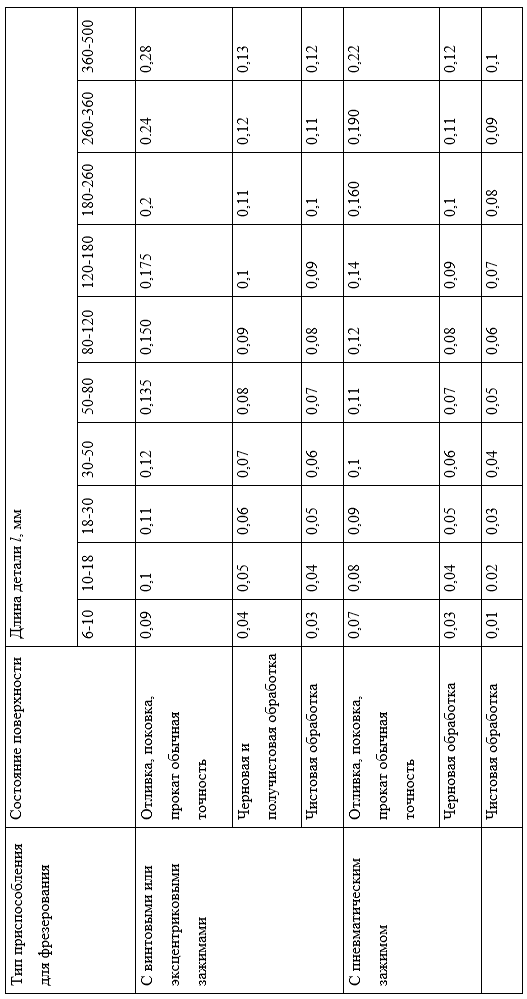


Таблица 1.7 Допускаемое смещение по поверхности разъёма штампа *ρсм,*, мм..

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Масса поковки, кг | До 0,5 | 0,5-1,0 | 1,0-1,8 | 1,8-3,2 | 3,2-5,6 | 5,6-10 | 10-20 |  |
| *ρсм, мм,* | 0,3 | 0,4 | 0,5 | 0,6 | 0,7 | 0,8 | 1,0 |  |

Таблица 1.8 Удельная кривизна штампованных поковок,

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Расчётный размер, мм | До 25 | 25-50 | 50-80 | 80-120 | 120-180 | 180-260 |
| *Δk,мм/мм,* | 0,004 | 0,003 | 0,002 | 0,0018 | 0,0016 | 0,0014 |

Таблица 1.9 Удельный увод оси отверстия при сверлении.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Расчётный диаметр отверстия, мм | 3-6 | 6,1-10 | 10.1-18 | 18,1-30 | 30,1-50 |
| *Δk,мм/мм,* | 0,0021 | 0,0017 | 0,0013 | 0,0009 | 0,0007 |

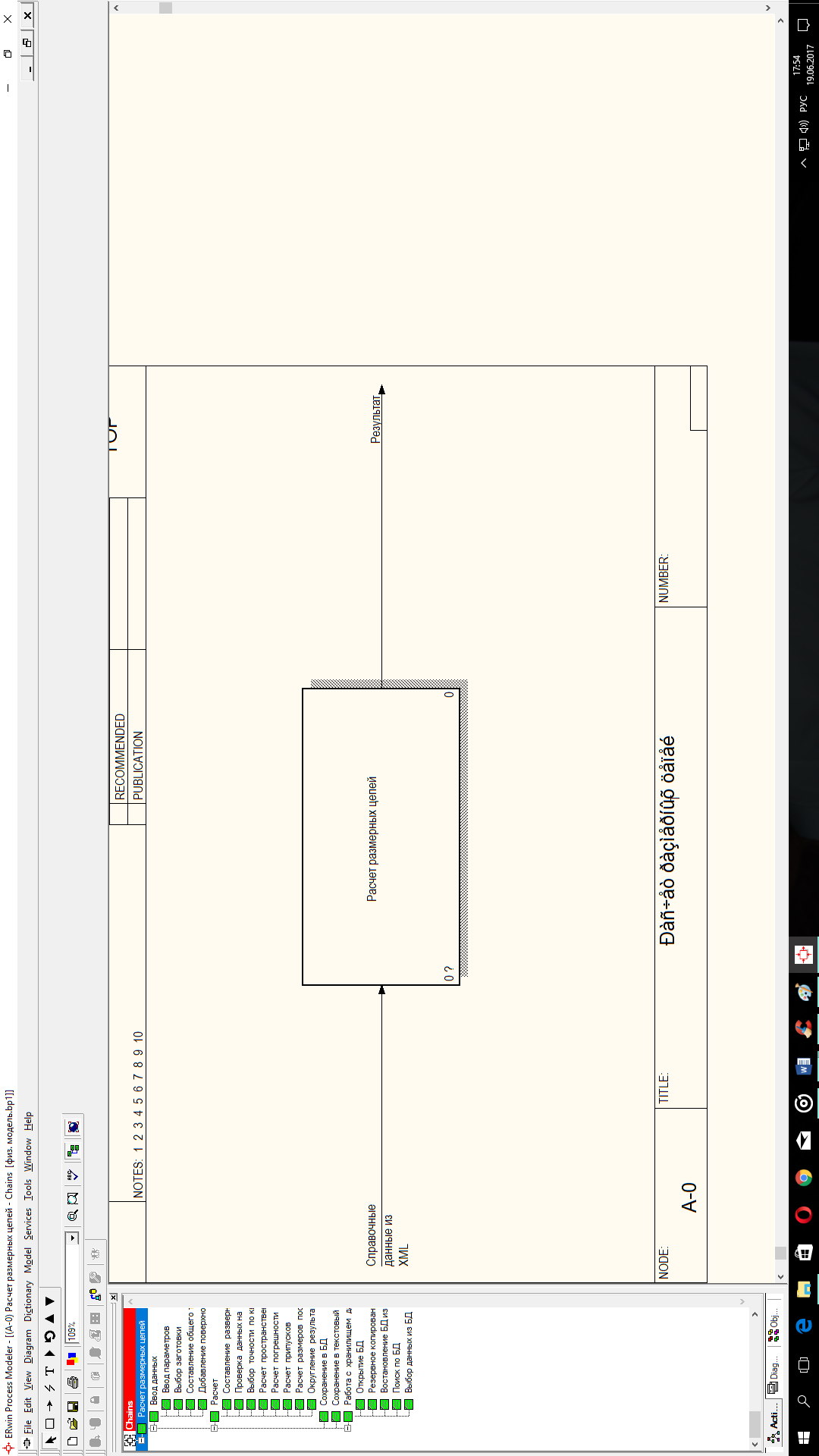
* 1. **Декомпозиция задачи**

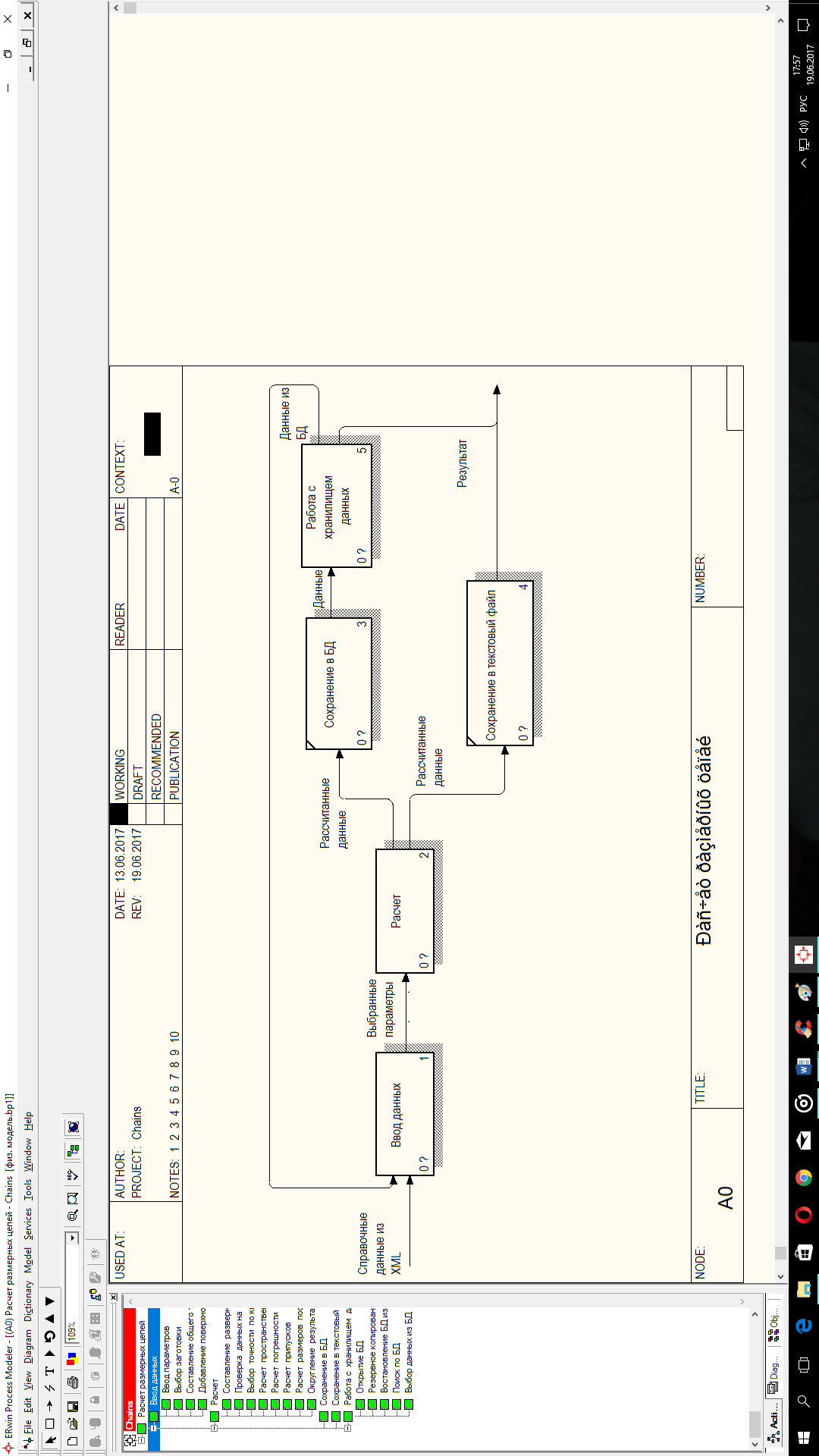
Общей задачей работы является разработка автоматизированной системы расчета припусков на обработку типовых деталей, которая включает в себя следующие подзадачи:

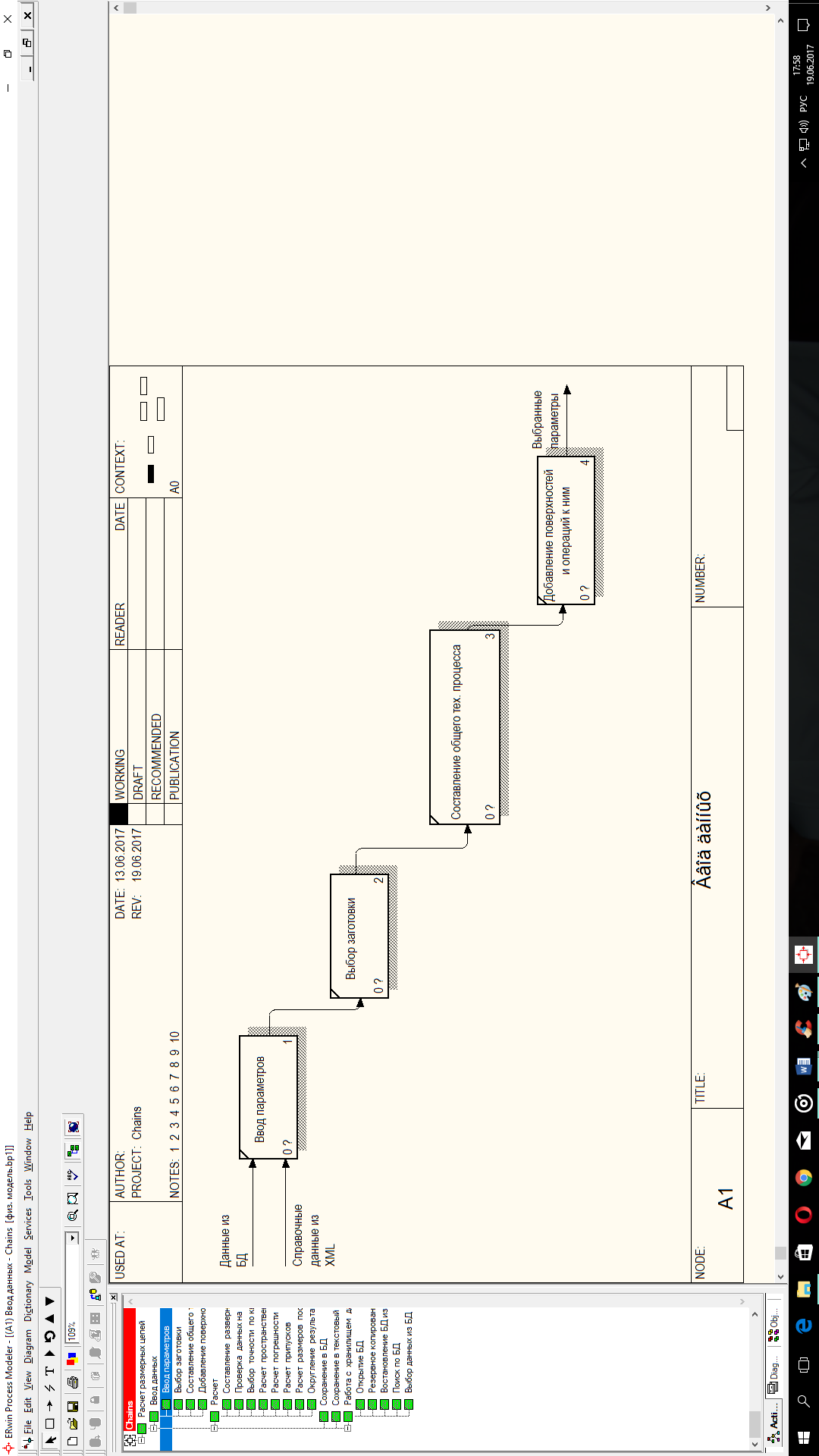
1. Анализ традиционного процесса решения задачи. Сбор исходных данных, необходимых для расчета.
2. Разработку алгоритма расчета.
3. Разработку структуры хранения справочных данных, используемых в расчете. Предполагается возможность корректировки и дополнения данных, введение корректирующих коэффициентов.
4. Разработку структуры хранения результатов расчета. Предполагается возможность поиска данных по различным критериям, передача данных для проведения повторного расчета на основе образца.
5. Разработку программного обеспечения для реализации расчета припусков на обработку детали, обеспечивающего удобный интерфейс пользователя.

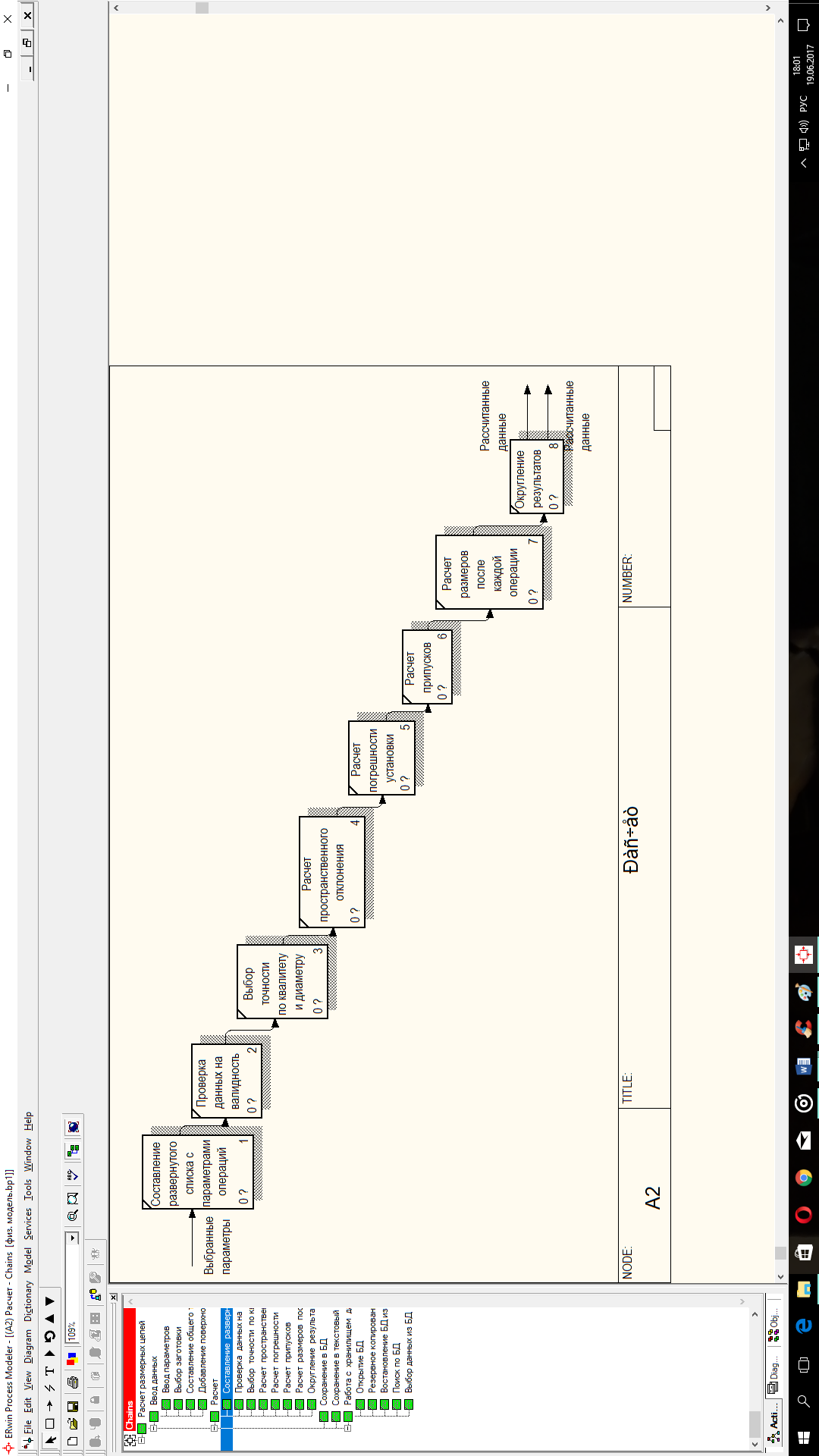
Проанализировав поставленные задачи была составлена следующая функциональная диаграмма.

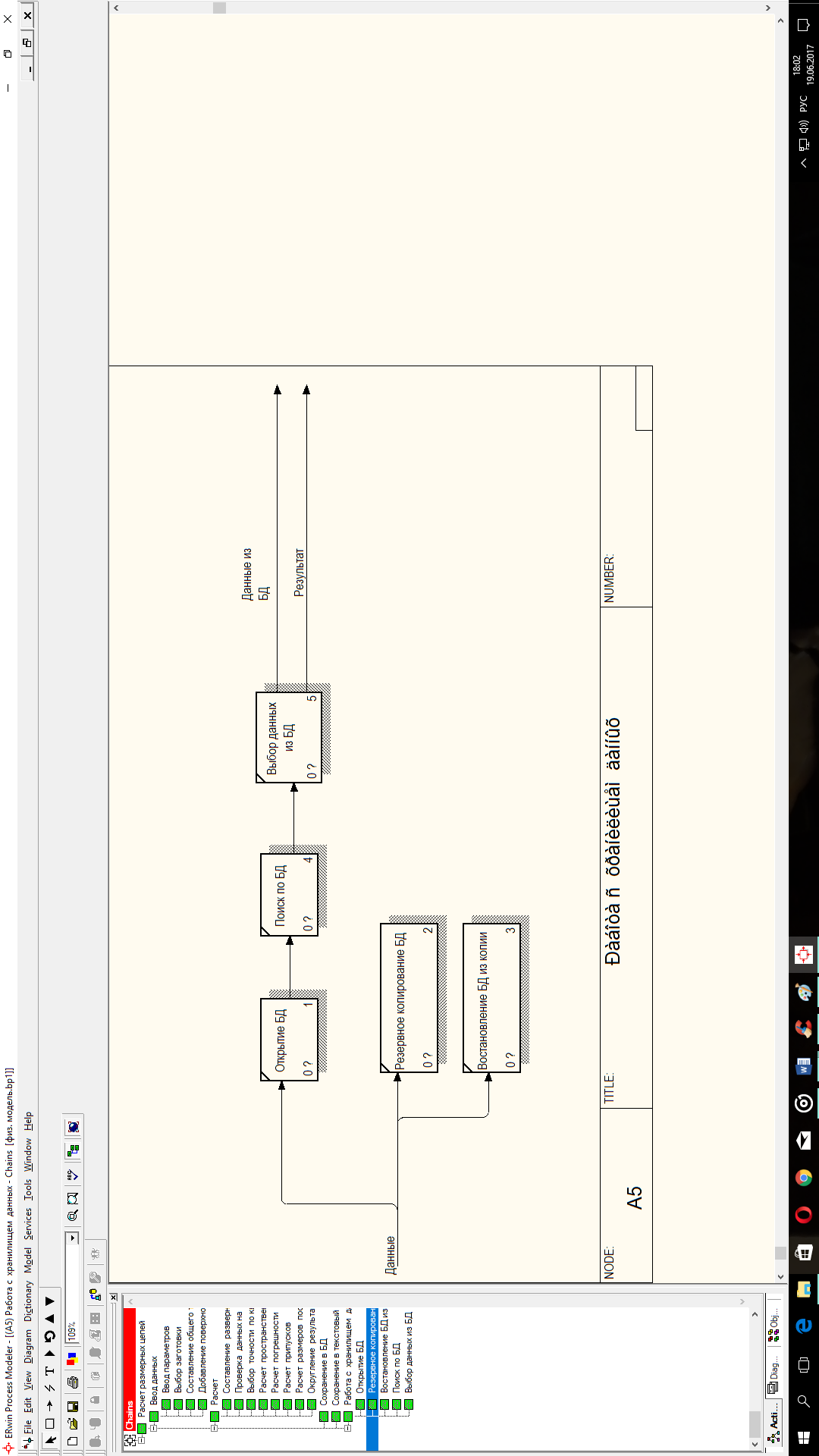
Схема декомпозиции по методологии IDEF0 приведена на рисунках 1.1 – 1.5.











* 1. **Алгоритмизация.**

В соответствии с диаграммой рис. 1.4, расчет содержит несколько функций, которые описаны ниже.

* + 1. **Расчет пространственного отклонения.**

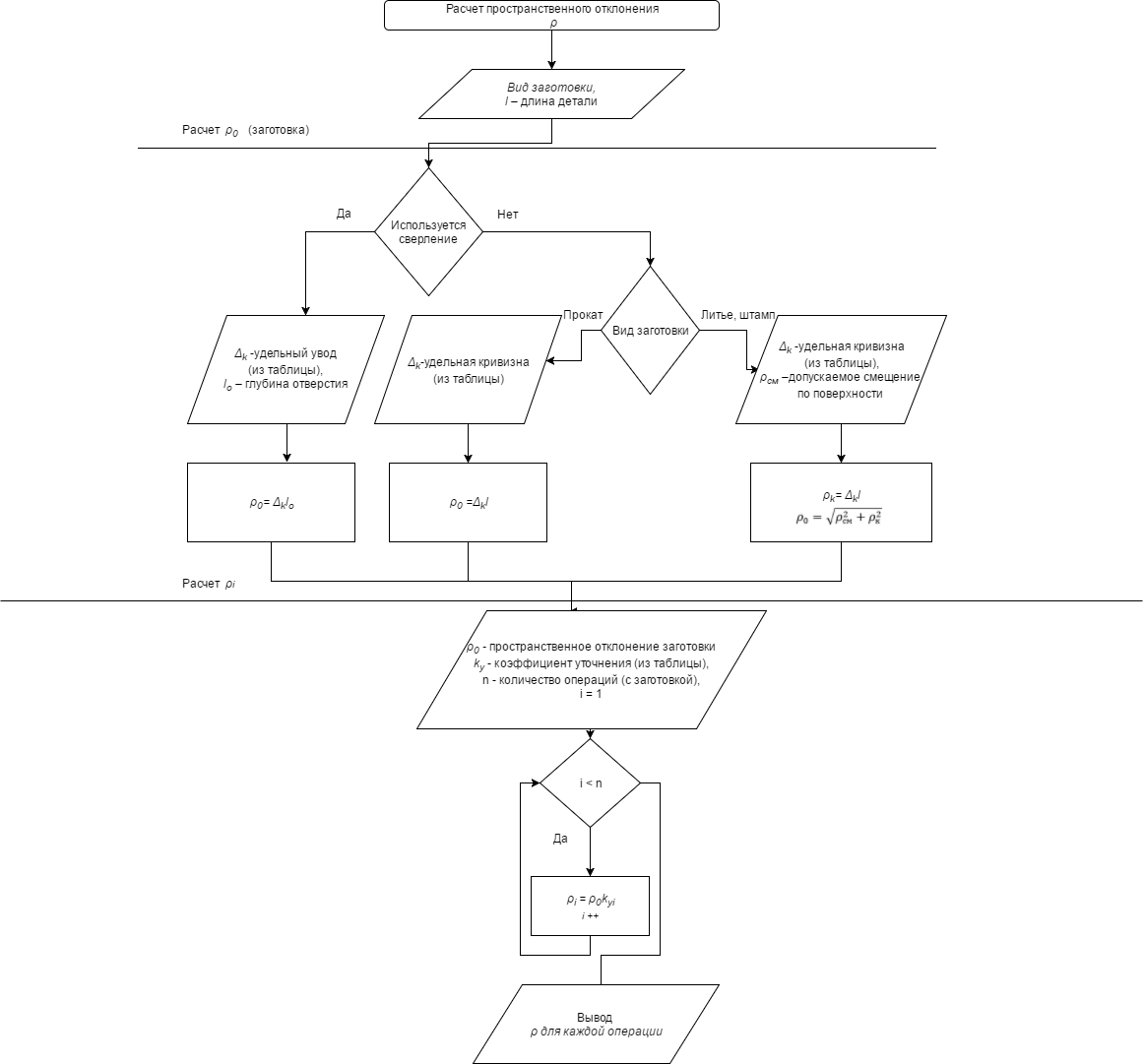
****

Рис.1.6. Расчет пространственного отклонения.

* + 1. **Расчет погрешности установки.**

Погрешность установки выбирается из таблицы, в зависимости от выбранного приспособления, выбранного размера и операции. Если приспособление не изменяется, то погрешность установки = 0.

* + 1. **Расчет минимального припуска.**

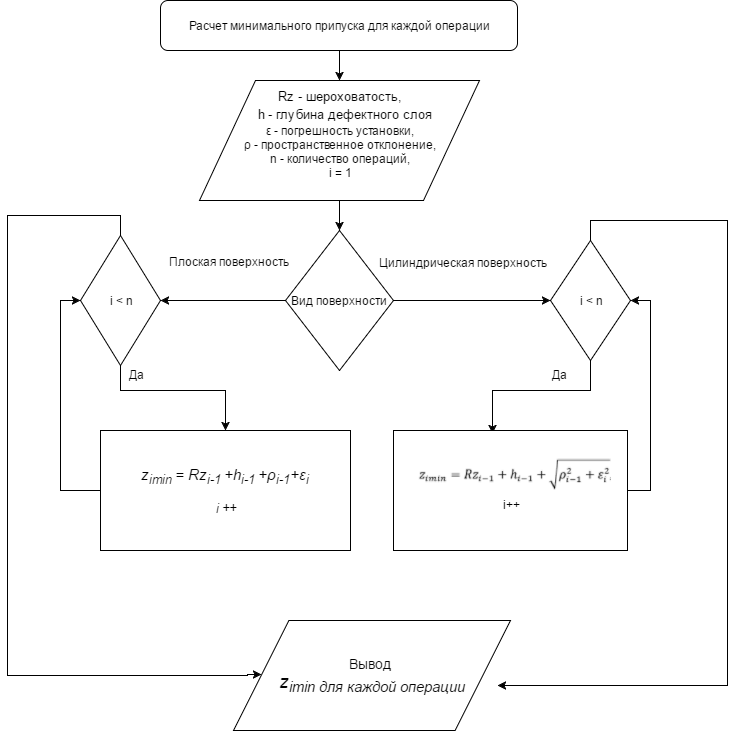


Рис.1.7. Расчет минимального припуска.

* + 1. **Расчет номинального припуска.**

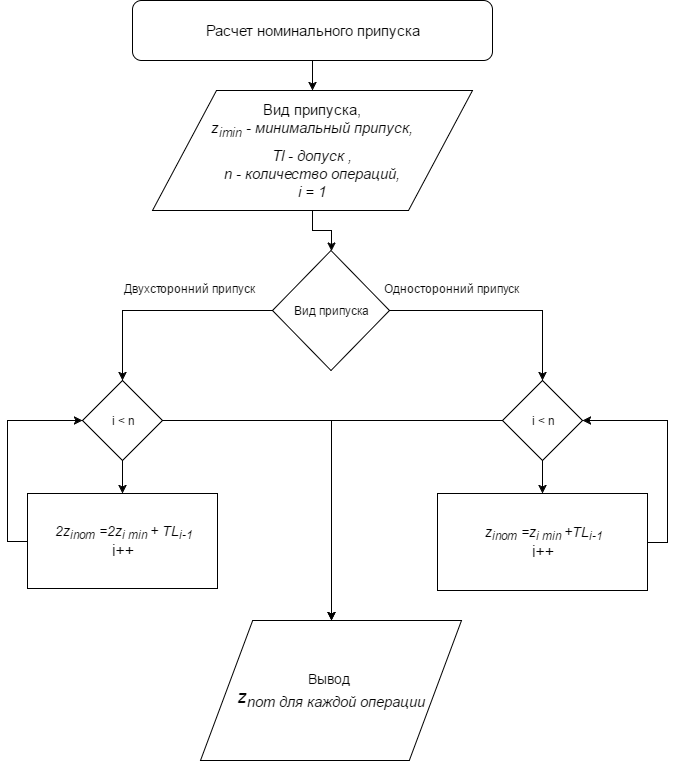
****

Рис.1.8. Расчет номинального припуска.

* + 1. **Расчет размеров после каждой операции.**

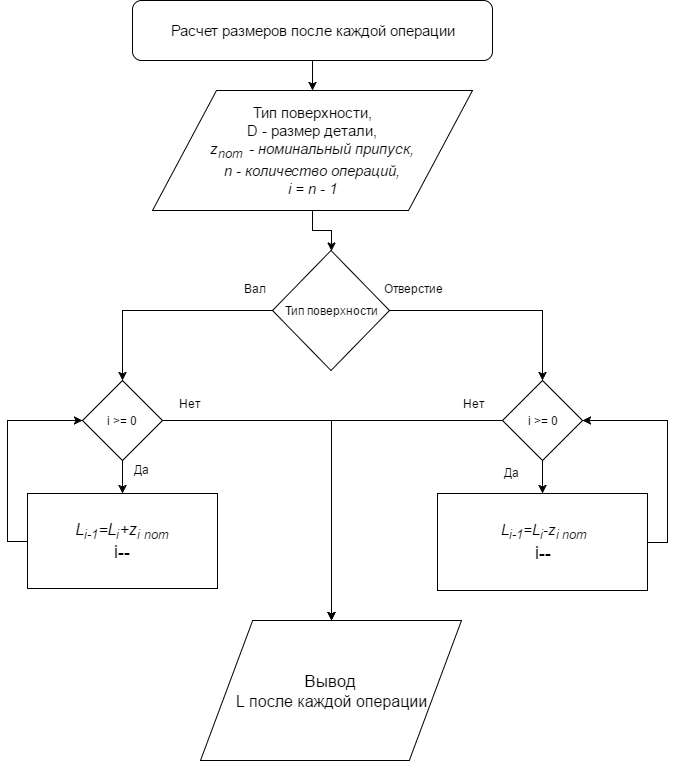
****

Рис.1.9. Расчет размеров после каждой операции.

* 1. **Обоснование и выбор средств реализации задачи.**
     1. **Описание информационной среды предприятия.**

На предприятии в главном технологическом отделе для автоматизации технологической подготовки производства используется программный продукт Techcard (разработчик: Intermech).

Возможности TECHCARD:

* Расцеховочные маршруты и материальное нормирование
* Технологические процессы
* Операционные эскизы
* Трудовое нормирование
* Технологические расчеты
* Технологические документы
* Единое информационное пространство
* Связь с внешними системами

После составления технического процесса происходит формирование управляющей программы. Управляющая программа может быть написано, как непосредственно кодом, так и с использованием программного обеспечения, который позволяет автоматизировать этот процесса. Для данных целей на предприятии используется программный продукт Esprit (разработчик: **корпорация** DP Technology**).**

Высокопроизводительная система ESPRIT предлагает мощные средства для любого станка с ЧПУ. Функциональность ESPRIT включает программирование фрезерной обработки от 2-х до 5-ти осей, токарной обработки от 2-х до 22-х осей, электроэрозию от 2-х до 5-ти осей, многозадачных токарно-фрезерных станков с синхронизацией, станков с осью B. ESPRIT предлагает прямой CAD - CAM интерфейс для импорта моделей из различных источников без необходимости правки или перестроения геометрии. ESPRIT позволяет обрабатывать любую комбинацию твердых тел, поверхностей, каркасной геометрии или STL, обеспечивая производственной гибкостью.

* + 1. **Обоснование выбора языка программирования.**

Несмотря на то, что синтаксис и особенности реализации унаследован языком программирования C# от «прародителей» (С++, Java), возможности этого языка программирования не ограничиваются ими.

К принципиально важным решениям, которые были реализованы, можно отнести следующие:

- компонентно-ориентированный подход к программированию;

- свойства как средство инкапсуляции данных);

- обработка событий (имеются расширения, в том числе в части обработки исключений, в частности, оператор try);

- унифицированная система типизации (соответствует идеологии Microsoft .NET в целом);

- делегаты (delegate – развитие указателя на функцию в языках C и C++);

- индексаторы (indexer – операторы индекса для обращения к элементам класса-контейнера);

- перегруженные операторы;

- оператор foreach (обработка всех элементов классов-коллекций);

- механизмы boxing и unboxing для преобразования типов;

- атрибуты (средство оперирования метаданными в COM-модели);

- прямоугольные массивы (набор элементов с доступом по номеру индекса и одинаковым количеством столбцов и строк).

На данный момент, С# успешно конкурирует с Java и С++ по популярности. Рассмотрим сходства этих языков.

Для начала перечислим сходства языков программирования C# и Java. Оба языка объектно-ориентированные и предполагают единственность наследования. Так же особенностями, которые делают схожими языки программирования C# и Java, являются механизмы интерфейсов, обработки исключительных ситуаций, нитей (threads). Оба языка имеют строгую типизацию и динамическую загрузку кода при выполнении программы.

От языка программирования C++, языком C# унаследованы механизмы: «перегруженные» операторы, небезопасные арифметические операции с плавающей точкой и множество других особенностей синтаксиса.

Исходя из особенностей языка программирования C#, сформулируем основные преимущества данного языка.

- Язык программирования C# претендует на подлинную объектную ориентированность (всякая языковая сущность претендует на то, чтобы быть объектом);

- Компонентно-ориентированный подход к программированию, способствующий меньшей машинно-архитектурной зависимости результирующего программного кода, гибкости, переносимости и легкости повторного использования (фрагментов) программ;

- Ориентация на безопасность кода (в сравнении с С и С++);

- Унифицированная система типизации;

- Расширенная поддержка событийно-ориентированного программирования.

Ввиду объектно-ориентированного дизайна, C# является хорошим выбором для быстрого конструирования различных компонентов - от высокоуровневой бизнес логики до системных приложений, использующих низкоуровневый код. Также следует отметить, что C# является и Web ориентированным - используя простые встроенные конструкции языка компоненты могут быть преобразованы в Web сервисы. Дополнительными возможностями языка C# - использование Web технологий, таких как: XML (Extensible Markup Language) и SOAP (Simple Object Access Protocol). Среда разработки Web сервисов позволяет программисту смотреть Web приложения, как на родные C# объекты, что дает возможность разработчикам соотнести имеющиеся Web сервисы с их познаниями в объектно-ориентированном программировании.

* + 1. **Обоснование выбора СУБД.**

Перед началом работы мы произвели сравнение СУБД, таких как Access, Mssql, Oracle и Mysql.

### Microsoft Office Access

### Эта база данных идет как часть MS Office. Как и SQLite, читать базу может несколько клиентов, писать — только один. Закрытый код и не совсем понятное соглашение. Получается, что пользоваться базой можно только в том случае если есть лицензия на приложение Office. С другой стороны, пользоваться ACCESS без установленного Office можно с помощью ODBC.

1. **Microsoft SQL Server (**разработчики: Sybase, Ashton-Tate, Microsoft)

Это серверная база данных от Microsoft. Для написания скриптов используется Transact-SQL. В полной версии поддерживается куб (OLAP) и присутствуют функции для сбора статистики и добычи данных (data mining). Среди редакций есть бесплатная, экспресс версия — [MS SQL Express](http://webarty.net/databases/sravnenie-sql-baz-dannyh#SQL_Server_Express_Edition).

1 поддерживаемый процессор (но может быть установлен на любой сервер),

1Гб адресуемой памяти,  
 4 Гб максимальный размер базы (10 Гб для версии SQL Server 2008 R2),  
Через интерфейс SQL Server Management Studio 2005 нет возможности экспорта/импорта данных (в версии 2008 эта возможность присутствует)

### Oracle (Oracle Corporation)

Это одна из самых древних, серверных баз данных. Об этом намекают и PL-SQL и обязательные sequence-ы, которые нужно создавать в ручную к инкрементальным полям. Лицензия стоит больших денег, хотя, как и у MSSQL есть Express edition — Oracle XE.

Размер базы данных до 4 Гбайт и поддержка только одного процессора и 1 Гбайт оперативной памяти.

### MySQL (разработчики MySQL AB (до 2008), Sun Microsystems (2008-2010), Oracle (с 2010)).

MySQL объединяет в себе несколько движков хранения данных (все они имеют свои достоинства и свои недостатки). Этот пункт, вместе с простотой использования сыграл важную роль в ее популяризации. Все хостинг компании предлагают именно эту базу данных в качестве основной.

Нам нужен был внедренный локальный сервер с возможностью переноса базы на другой сервер. Так как не требовалось никаких пожеланий подключения и установки на общезаводском сервере, то выбор производился из локальных серверов.

Для выбора СУБД мы сформулировали следующие критерии:

1. Продукт должен быть бесплатным
2. Задачу можно реализовать достаточно легко
3. Не требуется никаких дополнительных расходов
4. Мы владеем знаниями и опытом работы с ней.

Исходя из данных критериев нами была выбрана СУБД MSSQ

Изм

Лист

№ докум.

Подп.

Лист

Дата

Лит.

Листов

Разраб.

Провер.

Тихомиров

Зяблицев

КГТА  
И-113

ОСОБЕННОСТИ РЕАЛИЗАЦИИ ЗАДАЧИ

Реценз.

Н.контр.

Утв.

Котов В.В.

ВКР.КГТА.090301.12 ПЗ

У

Инв. № подп.

Подп. и дата

Взам. инв. №

Инв. № дубл.

Подп. и дата

Перв. примен.

Справ. №

1. ОСОБЕННОСТИ РЕАЛИЗАЦИИ ЗАДАЧИ

Для реализации хранения и расчета данных использовался язык программирования C#, среда разработки Visual Studio. Для хранения выбранных параметров разработано 7 классов. Для расчета разработано 5 классов. Ниже представлены разработанные классы.

* 1. **Хранение выбранных параметров.**

В разработанной программе общим хранилищем я является класс Part, в нем хранятся параметры детали, параметры заготовки, общий технологический процесс, параметры операций каждой поверхности, также данный класс содержит в себе методы для расчета, удобного ввода и вывода, валидации и вывода данных в основную форму.

**Структура класса Part.**

В данной программе требуется рассчитать припуски одной детали,

Поэтому пропадает необходимость создавать объект класса Part, это в свою очередь, позволяет сделать свойства и методы класса статическими.

Таблица 2.1. Структура класса Part.

|  |  |
| --- | --- |
| **Свойства класса** | |
| Объект класса ParametersOfPart | Содержит данные выбранных параметров детали и методы удобного взаимодействия с ними |
| Объект класса  ParametersWorkpiece | Содержит данные выбранных параметров заготовки и методы удобного взаимодействия с ними |
| Список класса  Surface | Элемент списка с индексом 0 (surfaces[0]) содержит данные о выбранном технологическом процессе и методы удобного взаимодействия с ним |
| Остальные элементы списка содержат данные о параметрах поверхности, операциях, а также методы для удобного ввода/вывода, расчета припусков поверхности и другие |

Продолжение таблицы 2.1.

|  |  |
| --- | --- |
| **Основные публичные методы** | |
| **Get/Set**  **методы** | |
| getSurfaceOnIndex  (int indexOfSurface) | Выводит поверхность с заданным индексом |
| setNameSurfaceOnIndex  (int indexOfSurface, string nameSurface) | Задает имя поверхности с заданным индексом |
| getCountSurfaces() | Выводит количество поверхностей |
| **Методы выполняющие какое-либо действие** | |
| deleteOperation  (int indexOfSurface,  int indexOfOperation) | Удаляет заданную операцию из заданной поверхности |
| deleteSurface  (int indexOfSurface) | Удаляет заданную поверхность |
| copyTechnologicalProcessInSurface  (int indexSurface) | Копирует операции из технологического процесса в заданную поверхность |
| addOperationInSurface  (Operation operation,  int indexSurface,  int indexOfOperation) | Добавляет операцию в заданную поверхность в элемент списка с заданным индексом операции |

Окончание таблицы 2.1.

|  |  |
| --- | --- |
| **Методы взаимодействия с формой** | |
| insertNameOfWorkpieceInTreeViewAndTextBox  (MainForm form) | Вставляет имя заготовки в текстовое поле и в дерево выбранных операций |
| insertListOfOperationsInTreeView  (MainForm form,  int indexOfSurface = 0) | Вставляет список операций в  дерево выбранных операций  (вызывает метод insertNameOfWorkpieceInTreeViewAndTextBox) |

**Структура класса ParametersWorkpiece**

Данный класс хранит параметры заготовки, методы для удобного ввода и вывода, взаимодействия с главной формой.

Таблица 2.2. Структура класса ParametersWorkpiece.

|  |  |
| --- | --- |
| **Свойства класса** | |
| string nameOfWorkpiece | Имя заготовки |
| int idWorkpiece | Индекс заготовки |
| double surfaceRoughnessRz | Шероховатость поверхности (Rz) |
| Interval recommendedIntervalRz | Рекомендуемый интервал шероховатости |
| int kvalitet | Квалитет |
| double thicknessOfDefectiveCoating | Глубина дефектного слоя |

Окончание таблицы 2.2.

|  |  |
| --- | --- |
| double validOffsetSurface | Допустимое смещение поверхности |
| int idWorkpieceInTable | Номер заготовки в таблице с параметрами заготовки |
| ResultsOfCalculation resultsOfCalculation | Результаты расчета |
| **Основные публичные методы** | |
| Конструктор | Аргументами конструктора являются все свойства, за исключением результатов расчета,  (Допустимое смещение поверхности по умолчанию = 0) |
| Get и Set методы | Позволяют вывести и ввести каждое из свойств класса |
| insertNameOfWorkpieceInTreeVie-wAndTextBox  (MainForm form) | Вставляет имя заготовки в текстовое поле и в дерево выбранных операций |

**Структура класса Surface.**

Данный класс содержит данные о параметрах поверхности, операциях, а также методы для удобного ввода/вывода, расчета припусков поверхности и другие. Surface в классе Part хранится в виде списка (преимуществом списка перед массивом в c# является возможность удаления из любого места и добавления элемента в любое место списка, при этом функционал списка автоматически изменит индексы элементов).

Таблица 2.3. Структура класса Surface.

|  |  |
| --- | --- |
| **Свойства класса** | |
| List<Operation> operations | Список операций |
| List<ParametersOperation> parametersOperation | Развернутый список операций |
| ParametersOfSurface parametersOfSurface | Параметры поверхности |
| string nameSurface | Имя поверхности |
| **Основные публичные методы** | |
| Get и Set методы | Позволяют вывести и ввести каждое из свойств класса |
| calculationOFSurface() | Расчет припусков для каждой операции поверхности |
| formationOfLongListWithParametersOfOperation() | На основе списка операций формируется развернутый список |
| insertListOfOperationsInTreeView  (MainForm form) | Вставляет список операций в дерево выбранных операций |
| deleteOperation(int indexOfOperation) | Удаление операции с заданным индексом |

**Структура класса Operation.**

При выборе операций пользователь выбирает нераскрытый список операций (List<Operation>), во время расчета список раскрывается (List<ParametersOperation>).

Пример нераскрытого списка: Точение, Шлифование.

Пример раскрытого списка: Точение черновое, Точение чистовое, Шлифование черновое, Шлифование чистовое.

Класс Operation хранит параметры нераскрытой операции и содержит методы для удобного ввода и вывода параметров операций.

Таблица 2.4. Структура класса Operation.

|  |  |
| --- | --- |
| **Свойства класса** | |
| string typeOfMachining | Тип операции |
| int idOperation | Индекс операции |
| string typeOfInstrument | Выбранное приспособление для установки |
| int idOnTechnologicalProcess | Индекс для связи с технологическим процессом |
| **Основные публичные методы** | |
| Конструктор | Конструктор перегружен 2-мя методами.  Или входными аргументами являются: тип операции, индекс операции, индекс для связи с технологическим процессом(по умолчанию равен -1).  Или входным аргументом является класс Operation(для клонирования объекта). |
| Get и Set методы | Позволяют вывести и ввести каждое из свойств класса |

**Структура класса ParametersOperation.**

При выборе операций пользователь выбирает нераскрытый список операций (List<Operation>), во время расчета список раскрывается (List<ParametersOperation>).

Пример нераскрытого списка: Точение, Шлифование.

Пример раскрытого списка: Точение черновое, Точение чистовое, Шлифование черновое, Шлифование чистовое.

Класс ParametersOperation хранит параметры раскрытой операции и содержит методы для удобного ввода и вывода параметров операций.

Таблица 2.5. Структура класса ParametersOperation.

|  |  |
| --- | --- |
| **Свойства класса** | |
| string typeOfMachining | Тип операции |
| string precisionOfMachining | Точность операции |
| double surfaceRoughnessRz | Шероховатость поверхности (Rz) |
| Interval recommendedIntervalRz | Рекомендуемый интервал шероховатости |
| int idOperation | Индекс операции |
| int kvalitet | Квалитет |
| double thicknessOfDefectiveCoating | Глубина дефектного слоя |
| double coefficientOfRefinement | Коэффициент уточнения |
| string typeOfInstrument | Выбранное приспособление для установки |

Окончание таблицы 2.5.

|  |  |
| --- | --- |
| **Свойства класса** | |
| ResultsOfCalculation resultsOfCalculation | Результаты расчета |
| Основные публичные методы | |
| Конструктор | Конструктор состоит из трех перегруженных методов:   1. Аргументами являются все свойства класса, за исключением результатов. 2. Аргументами являются все свойства класса, за исключением результатов в строковом виде. 3. Аргументом является объект класса ParametersOperation (для клонирования объекта) |
| Get и Set методы | Позволяют вывести и ввести каждое из свойств класса |

* 1. **Особенности подготовки параметров к расчету.**

После ввода всех параметров и нажатия кнопки расчета

вызывается метод Part.calculationOfSurfaces().

Данный метод в каждой поверхности вызывает аналогичный метод.

После в каждой поверхности создается развернутый список операций.

За подготовку параметров к расчету отвечает метод в Surface getParametersOperationsForCalculation().

Данный метод помещает параметры всех операций поверхности (включая заготовку) в класс ParametersOperationsForCalculation.

**Структура класса ParametersOperationsForCalculation**

Данный класс хранит параметры всех операций поверхности и заготовки в массивах (элемент с индексом 0 – параметры заготовки) и содержит методы для удобного ввода и вывода параметров операций.

Таблица 2.6. Структура класса ParametersOperationsForCalculation.

|  |  |
| --- | --- |
| **Свойства класса** | |
| double[] surfaceRoughnessRz | Массив с шероховатостью (Rz) |
| double[] kvalitets | Массив с квалитетами |
| double[] thicknessOfDefectiveCoating | Массив с глубиной дефектного слоя |
| double[] coefficientOfRefinement | Массив с коэффициентом уточнения |
| int[] idOperation | Массив с индексом заготовки/операции |
| string[] typeOfInstrument | Выбранное приспособление для установки |
| double validOffsetSurfacePcm | Допустимое смещение на поверхность размера штампа или литейной формы |

Окончание таблицы 2.6.

|  |  |
| --- | --- |
| **Основные публичные методы** | |
| Конструктор | Аргументами являются все свойства класса |
| Get и Set методы | Позволяют вывести и ввести каждое из свойств класса |

Далее создается экземпляр класса CalculationOfSurface.

**Структура класса CalculationOfSurface**

Данный класс содержит методы для расчета поверхности.

Таблица 2.7. Структура класса CalculationOfSurface.

|  |  |
| --- | --- |
| **Свойства класса** | |
| ParametersOfPart parametersOfPart | Параметры детали |
| ParametersOperationsForCalculation parametersForCalculation | Параметры для расчета |
| double[] accuracies | Точность |
| double[] spatialDeviationP | Пространственное отклонение |
| double[] deviationOfInstallationE | Погрешность установки |
| double[] nominalAllowanceZnom | Припуск |

Окончание таблицы 2.7.

|  |  |
| --- | --- |
| **Свойства класса** | |
| double[] sizeOfWorkprieceAfterOperation | Размер после операции |
| **Основные публичные методы** | |
| Конструктор | Аргументами являются экземпляр класса ParametersOfSurface и экземпляр класса ParametersOperationsForCalculation |

* 1. **Особенности расчета.**

После создания экземпляра класса CalculationOfSurface вызывается главный метод класса CalculationOfSurface.calculation(). Данный метод проверяет данные и рассчитывает поочередно каждое из свойств класса.

* + 1. **Расчет точности.**

За расчет точности отвечает метод kvalitetToAccuracy(). Данный метод с помощью TableOfTolerancesAndFits производит выборку точности для каждой операции по квалитету и диаметру и записывает её в массив accuracies

* + 1. **Расчет пространственного отклонения.**

За расчет пространственного отклонения отвечает метод CalculationOfSpatialDeviation(). Данный метод создает экземпляр класса SpatialDeviation и вызывает метод Calculation().

Аргументами конструктора данного класса являются: экземпляры класса ParametersOfSurface и класса ParametersOperationsForCalculation.

Из данных аргументов конструктор берет количество операций, длину детали, диаметр, индекс, коэффициент уточнения, допустимое смещение поверхности, глубину отверстия при сверлении.

Метод Calculation() рассчитывает отклонение для заготовки, исходя из установленных условий, затем рассчитывает отклонение каждой операции и выводит массив отклонений. Массив записывается в свойство spatialDeviationP.

* + 1. **Расчет погрешности установки.**

За расчет пространственного отклонения отвечает метод CalculationOfDeviationOfInstallation(). Данный метод создает экземпляр класса DeviationOfInstallation и вызывает метод Calculation().

Аргументами конструктора данного класса являются: экземпляры класса ParametersOfSurface и класса ParametersOperationsForCalculation.

Из данных аргументов конструктор берет количество операций, длину детали, диаметр, индекс, используемое приспособление.

Метод Calculation() производит выборку погрешности установки из InaccuracyOfPositioningPart на основе размера, индекса и используемого инструмента и возвращает массив погрешностей установки. Массив записывается в свойство deviationOfInstallationE.

* + 1. **Расчет припуска.**

За расчет припуска отвечает метод CalculationOfAllowance (). Данный метод создает экземпляр класса Allowance и вызывает метод Calculation().

Аргументами конструктора данного класса являются: экземпляры класса ParametersOfSurface, класса ParametersOperationsForCalculation, массив spatialDeviationP, массив deviationOfInstallationE, массив accuracies.

Из данных аргументов конструктор берет количество операций, шероховатость поверхности, пространственное отклонение, погрешность установки, точность, тип обрабатываемой поверхности, тип припуска.

Метод Calculation() рассчитывает минимальный припуск в зависимости от выбранных условий, далее на использует его для расчета номинального припуска в зависимости от условий и возвращает массив с номинальным припуском. Массив записывается в свойство nominalAllowanceZnom.

* + 1. **Расчет размеров после каждой операции.**

За расчет размеров отвечает метод CalculationOfSizeOfWorkprieceAfterOperation (). Данный метод создает экземпляр класса SizeOfWorkprieceAfterOperation и вызывает метод Calculation().

Аргументами конструктора данного класса являются: экземпляры класса ParametersOfSurface, класса ParametersOperationsForCalculation, массив nominalAllowanceZnom.

Из данных аргументов конструктор берет количество операций, диаметр, тип поверхности, номинальный припуск.

Метод Calculation() рассчитывает размер после каждой операции в зависимости от выбранных условий и возвращает массив с размерами. Массив записывается в свойство sizeOfWorkprieceAfterOperation.

* + 1. **Возвращение результатов.**

**Перед выводом результатов все расчеты округляются при помощи метода** rounding(). Далее формируется массив экземпляров класса ResultsOfCalculation и выводится в Surface, где распределяются по операциям.

**Структура класса ResultsOfCalculation**

Данный класс хранит результаты расчета и содержит методы для удобного вывода результатов.

Таблица 2.8. Структура класса ResultsOfCalculation.

|  |  |
| --- | --- |
| **Свойства класса** | |
| double accuracies | Точность |

Окончание таблицы 2.8.

|  |  |
| --- | --- |
| **Свойства класса** | |
| double spatialDeviationP | Пространственное отклонение |
| double deviationOfInstallationE | Погрешность установки |
| double nominalAllowanceZnom | Номинальный припуск |
| double sizeOfWorkprieceAfterOperation | Размер после операции |
| **Основные публичные** **методы** | |
| **Конструктор** | Конструктор принимает на вход аргументы, которые соответствуют свойствам класса |
| Get и Set методы | Позволяют вывести и ввести каждое из свойств класса |

Расчет можно представить в такой последовательности:

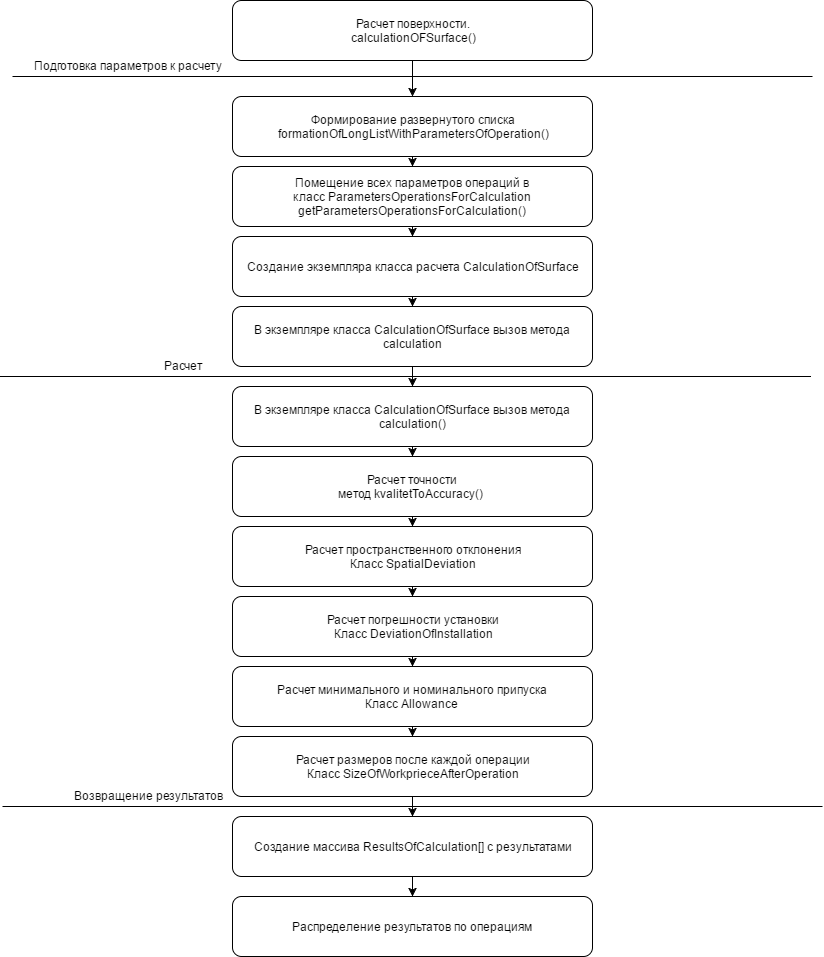
****

Рис.2.1. Схема расчета.

* 1. **Особенности организации технического обеспечения.**

Для реализации данной разработки мы использовали Microsoft SQL Server и Microsoft .NET Framework 4.0.

Табл. 2.9 Системные требования Microsoft SQL Server

|  |  |
| --- | --- |
| Компонент | Требование |
| Поддерживаемая операционная система | Windows 7; Windows 7 Service Pack 1; Windows 8; Windows 8.1; Windows Server 2008 R2; Windows Server 2008 R2 SP1; Windows Server 2012; Windows Server 2012 R2 |
| Быстродействие процессора | Минимальные:   * процессор x86: 1,0 ГГц * процессор x64: 1,4 ГГц   Рекомендуемые: 2,0 ГГц и выше |
| Тип процессора | * процессор x64: AMD Opteron, AMD Athlon 64, Intel Xeon с поддержкой Intel EM64T, Intel Pentium IV с поддержкой EM64T. * процессор x86: Процессор, совместимый с Pentium III или выше |
| Память | Минимальные:  Экспресс-выпуски: 512 МБ  Все другие выпуски: 1 ГБ  Рекомендуемые:  Экспресс-выпуски: 1 ГБ  Все другие выпуски: Для обеспечения оптимальной производительности требуется не менее 4 ГБ с последующим увеличением по мере роста размера базы данных. |

Табл. 2.9 Системные требования Microsoft SQL Server (окончание)

|  |  |
| --- | --- |
| Компонент | Требование |
| Объем жесткого диска | 4,2 ГБ дискового пространства. |
| Ограничения | Microsoft SQL Server Express поддерживает 1 физический процессор, 1 ГБ памяти и 10 ГБ дискового пространства. |

Табл. 2.10 Системные требования .NET Framework 4.0.

|  |  |  |
| --- | --- | --- |
| Требования к оборудованию | Версия 4 полная | Версия 4 клиент |
| Процессор |  |  |
| Минимум | 1 ГГц | 1 ГГц |
| Рекомендовано | 1 ГГц | 1 ГГц |
| ОЗУ |  |  |
| Минимум | 512 МБ | 512 МБ |
| Рекомендовано | 512 МБ | 512 МБ |
| Место на диске (минимум) |  |  |
| 32-разрядный | 850 МБ | 600 МБ |
| 64-разрядный | 2 ГБ | 1,5 ГБ |

Изм

Лист

№ докум.

Подп.

Лист

Дата

Лит.

Листов

Разраб.

Провер.

Тихомиров

Зяблицев

КГТА  
И-113

Описание программы

Реценз.

Н.контр.

Утв.

Котов В.В.

ВКР.КГТА.09.03.01.10 ПЗ

У

Инв. № подп.

Подп. и дата

Взам. инв. №

Инв. № дубл.

Подп. и дата

Перв. примен.

Справ. №

1. Описание программы
   1. **Описание выполнения программы.**

Программа для расчёта размерной технологической цепи состоит из форм главного окна, выбора заготовки, открытия из БД сохранения в БД и формы результатов.

Главное окно имеет в меню, в котором имеется возможность открытия или работы с БД. Работа с БД включается в себя выбор подключения, возможность сохранения и восстановления баз данных.

Основные формы программы изображены на рисунках 3.1- 3.7.

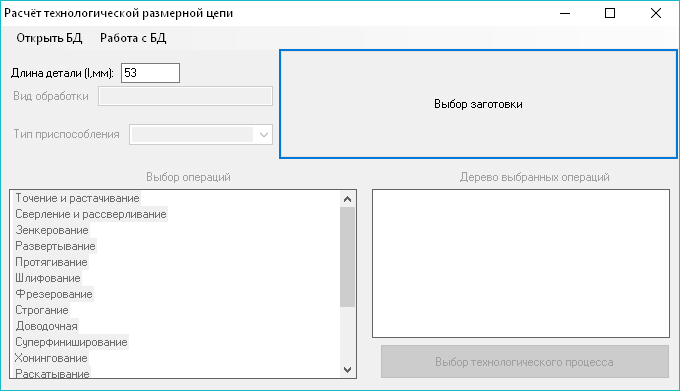


Рис. 3.1 Главное окно программы.

Работа в программе:

Пользователь выбирает способ работы с программой: ручной ввод данных или загрузка из БД:

- При загрузке из БД пользователь устанавливает соединение с БД нажатием кнопки выбор подключения подменю работа с БД, вводит имя сервера, выбирает файл БД. После загрузки БД пользователь нажимает кнопку открыть БД, в открывшейся форме выбирает нужную деталь и нажимает выбор записи. В данном окне имеется возможность поиска по наименованию или шифру детали.

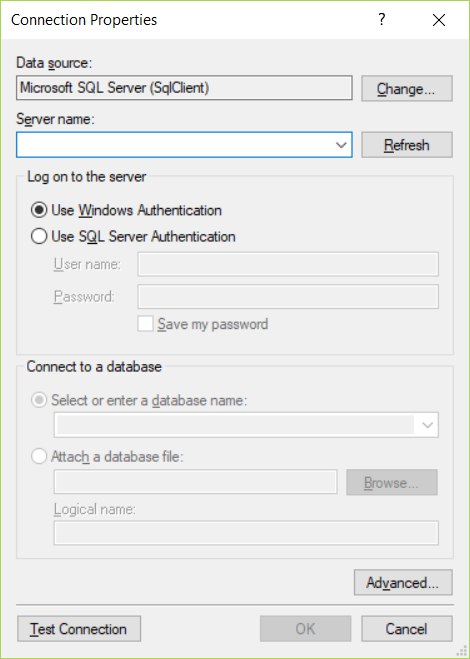


Рис. 3.2 Окно подключения к БД.

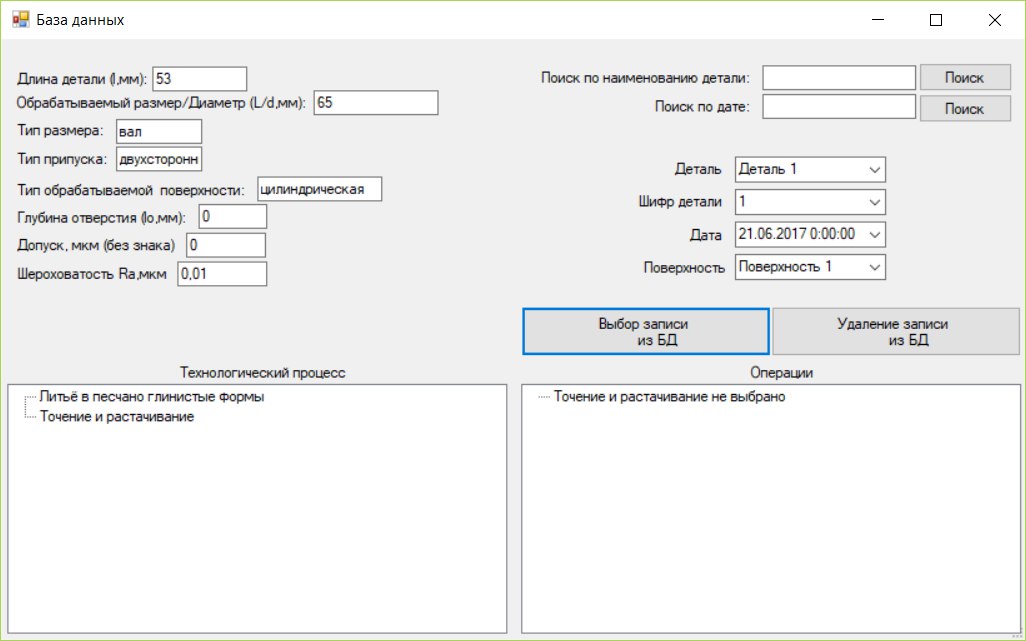


Рис. 3.3 Окно работы с БД.

- При ручном режиме работы пользователь вводит длину детали, нажимает кнопку Выбора заготовки для вызова окна выбора заготовки. (рис.3.3)

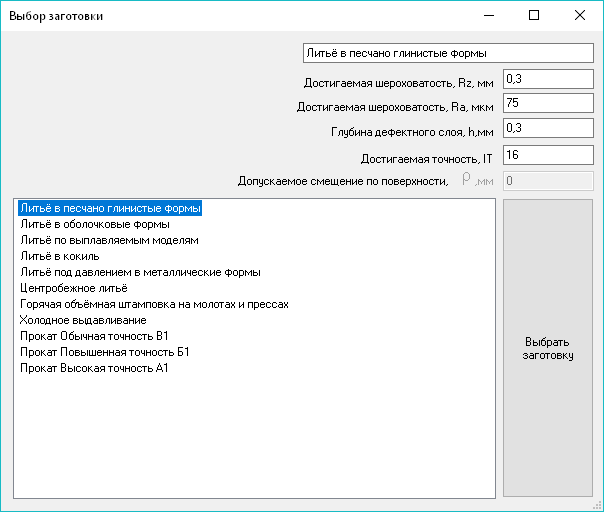


Рис. 3.4 Окно выбора заготовки.

В дереве пользователь выбирает нужную заготовку в дереве, нажимает кнопку добавить заготовку. В главном окне программы активируется возможность добавления из дерева операций технологического процесса. После выбора технологического процесса нажимаем соответствующую клавишу, вид программы изменяется, добавляется возможность работы с поверхностями (рис. 3.4)

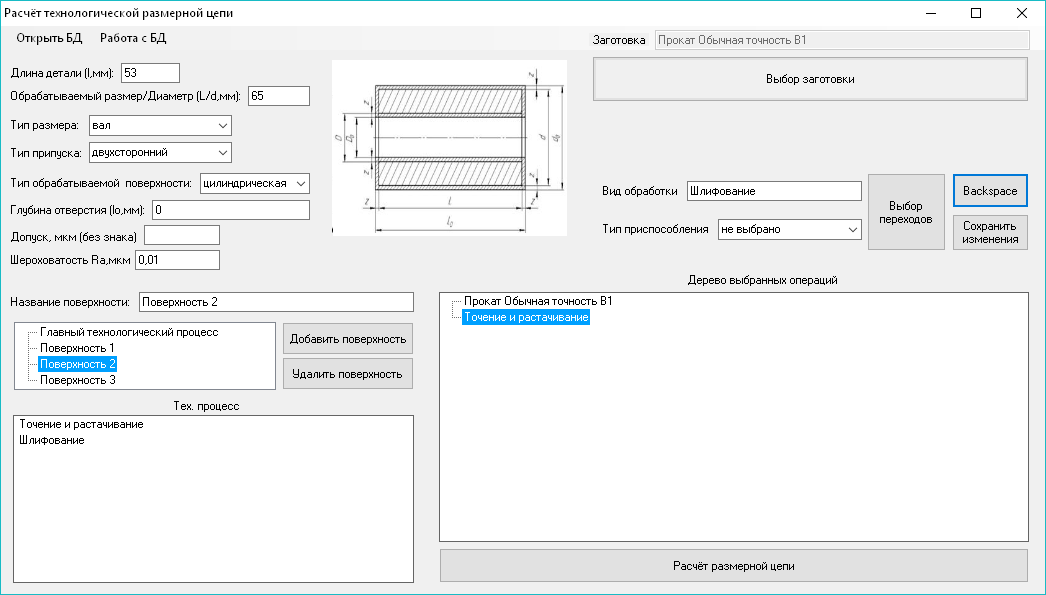


Рис. 3.5 Главное окно после выбора технологического процесса.

Дальнейшие шаги одинаковы для ручного ввода данных и после выбора из БД:

В программе имеется возможность добавления, удаления и работы с операциями поверхностей.

После ввода всех исходных данных пользователем нажимается кнопка расчёта размерной цепи, появляется форма результата с возможностью сохранения в текстовый файл или в БД. (рис.3.5,3.6)

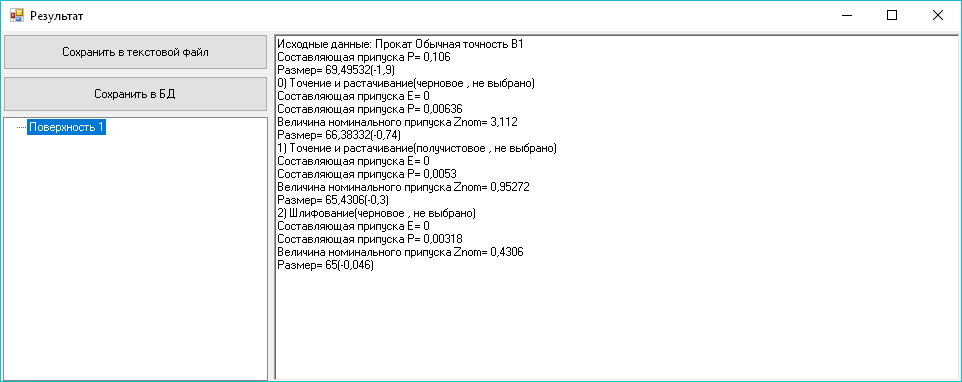


Рис. 3.6 Окно вывода результата.

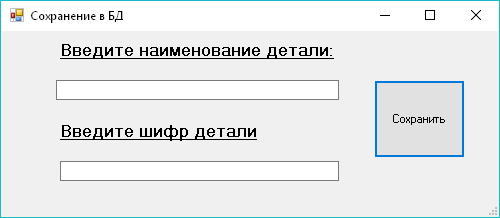


Рис. 3.7 Окно сохранения в БД.

На главной форме присутствует возможность резервного копирования БД и восстановления из резервной копии

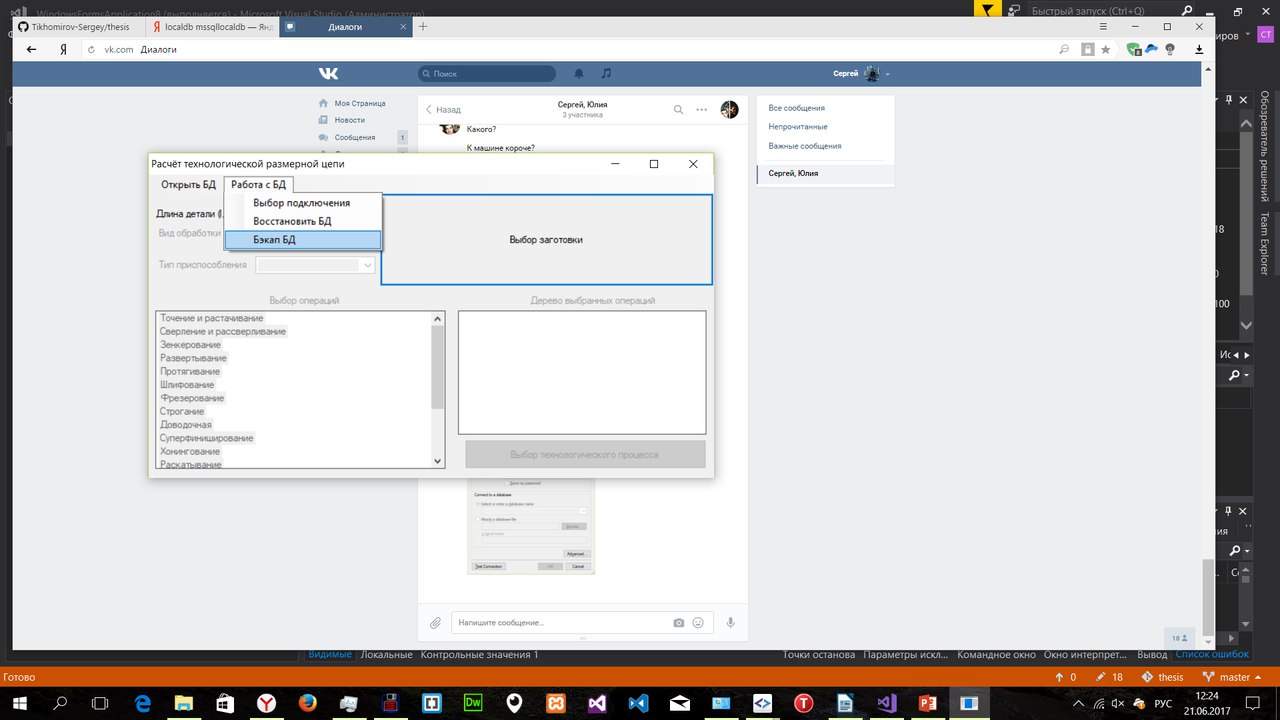


Рис. 3.8 Резервное копирование БД.

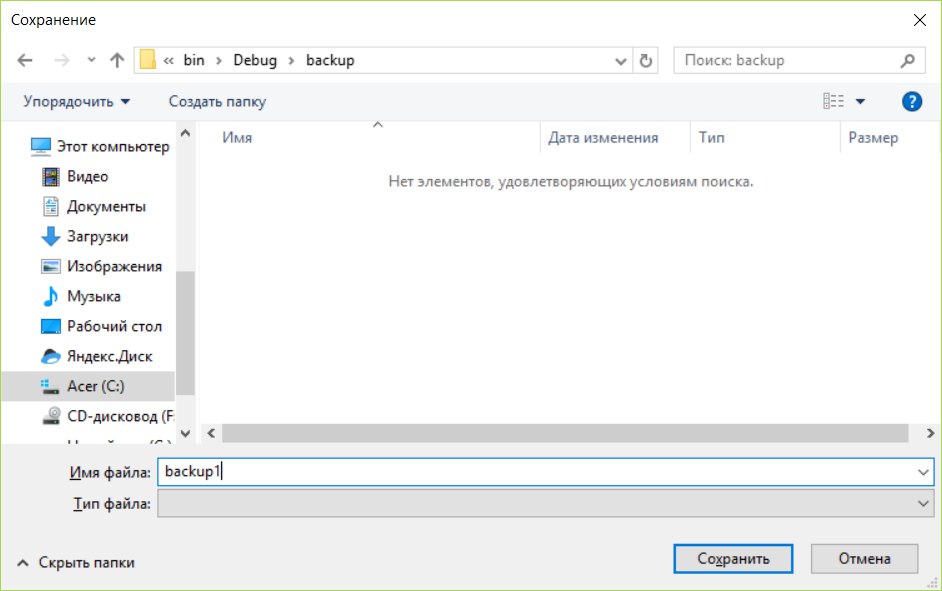


Рис. 3.9 Диалоговое окно сохранения.

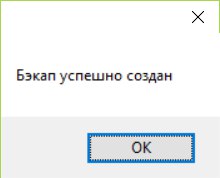


Рис. 3.10 Сообщение об успешном сохранении.

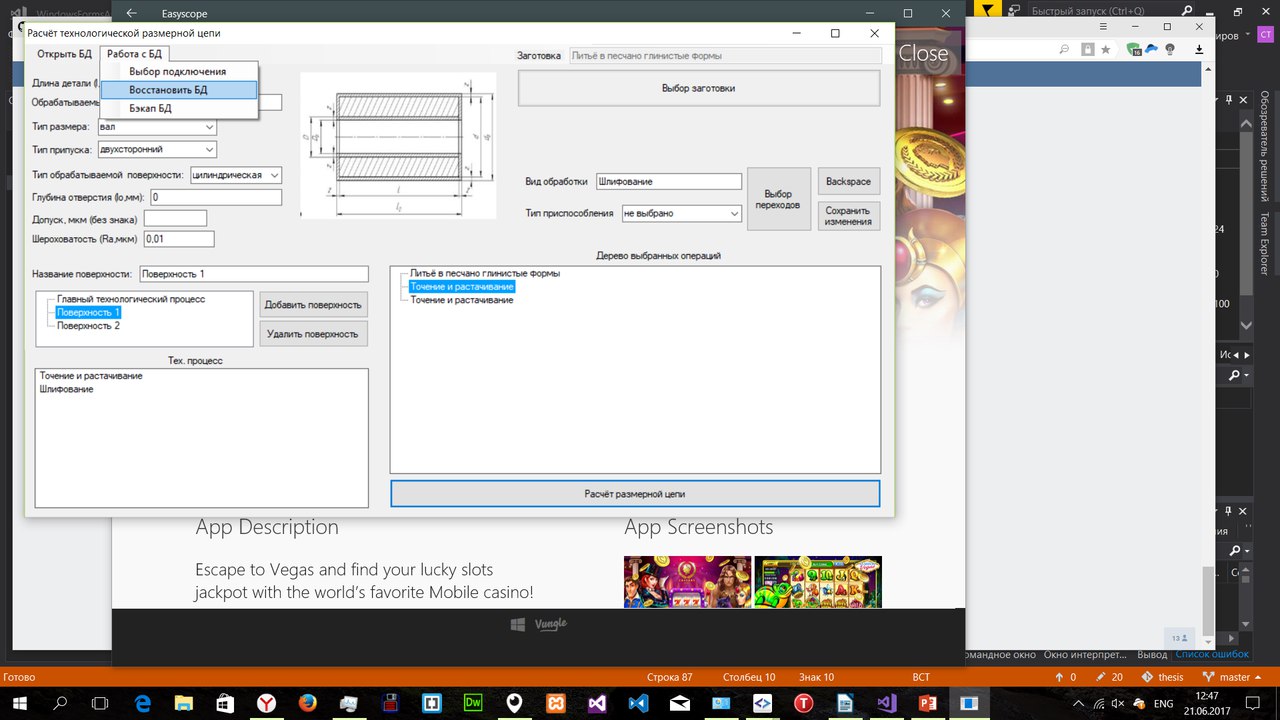


Рис. 3.11 Восстановление резервной копии БД.

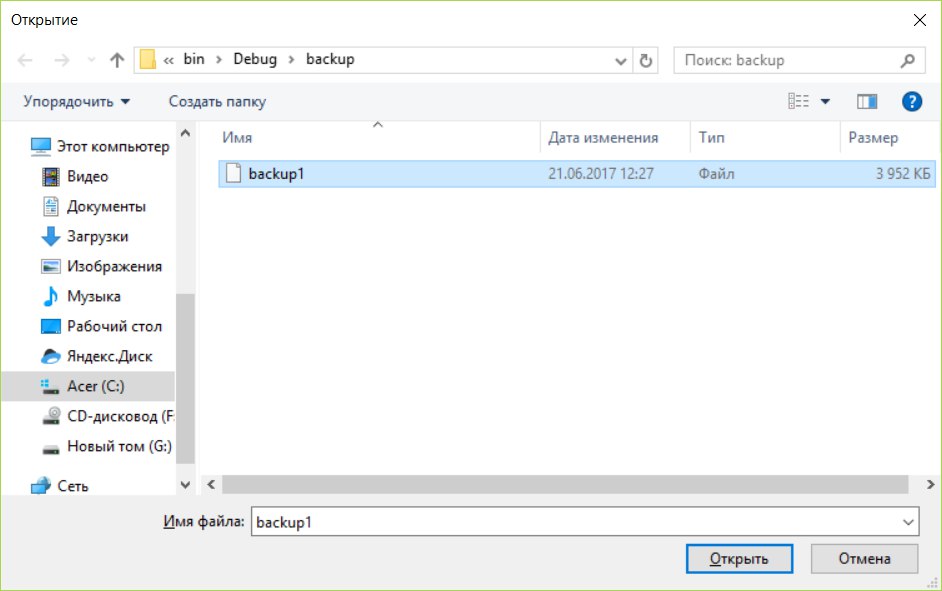


Рис. 3.12 Диалоговое окно открытия резервной копии БД.

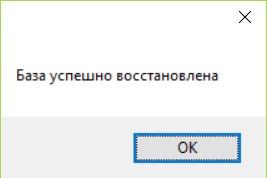


Рис. 3.13 Сообщение об успешном восстановлении.

* + 1. **Руководство администратора.**

Для того чтобы программу можно было использовать, на компьютере должен быть установлен .NET Framework 4.0 и Microsoft SQL Server. Это может быть либо серверная версия СУБД, либо локальная(LocalDB). Далее установка программы заключается в ее копировании.

* 1. **Тестирование**

**Рассмотрим примеры работы в программе:**

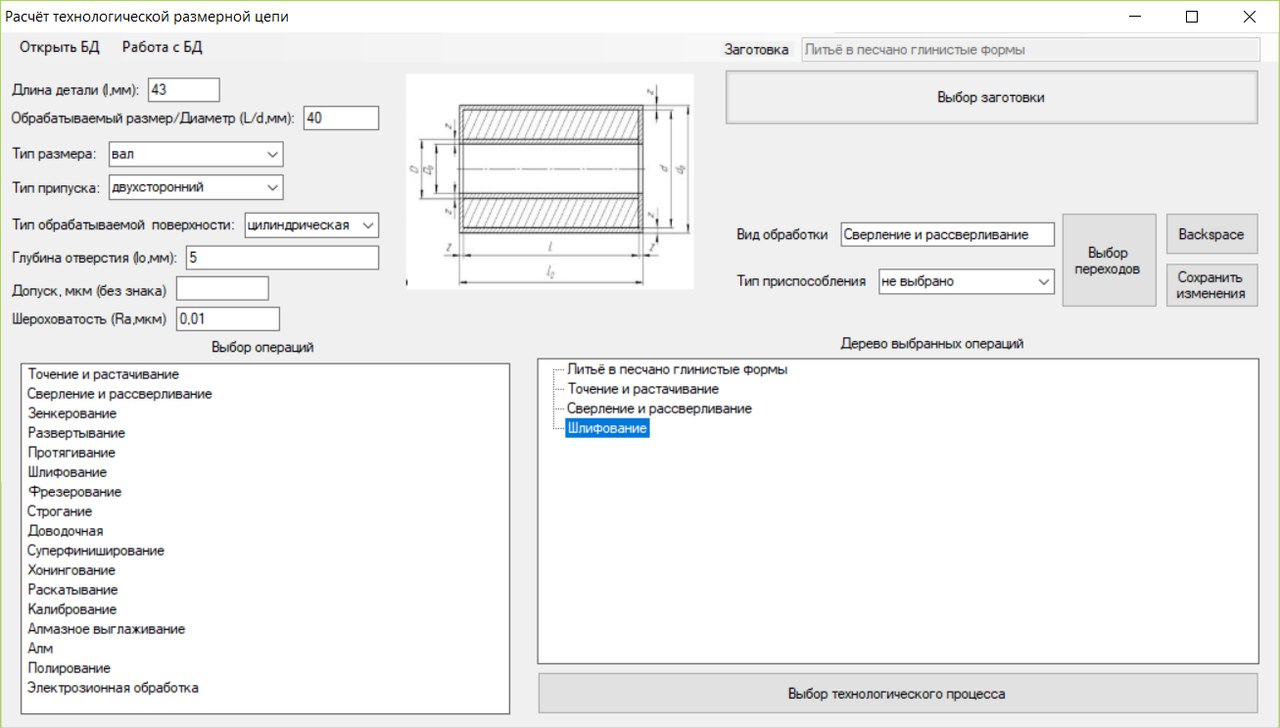


Рис. 3.14 Выбор операций технологического процесса.

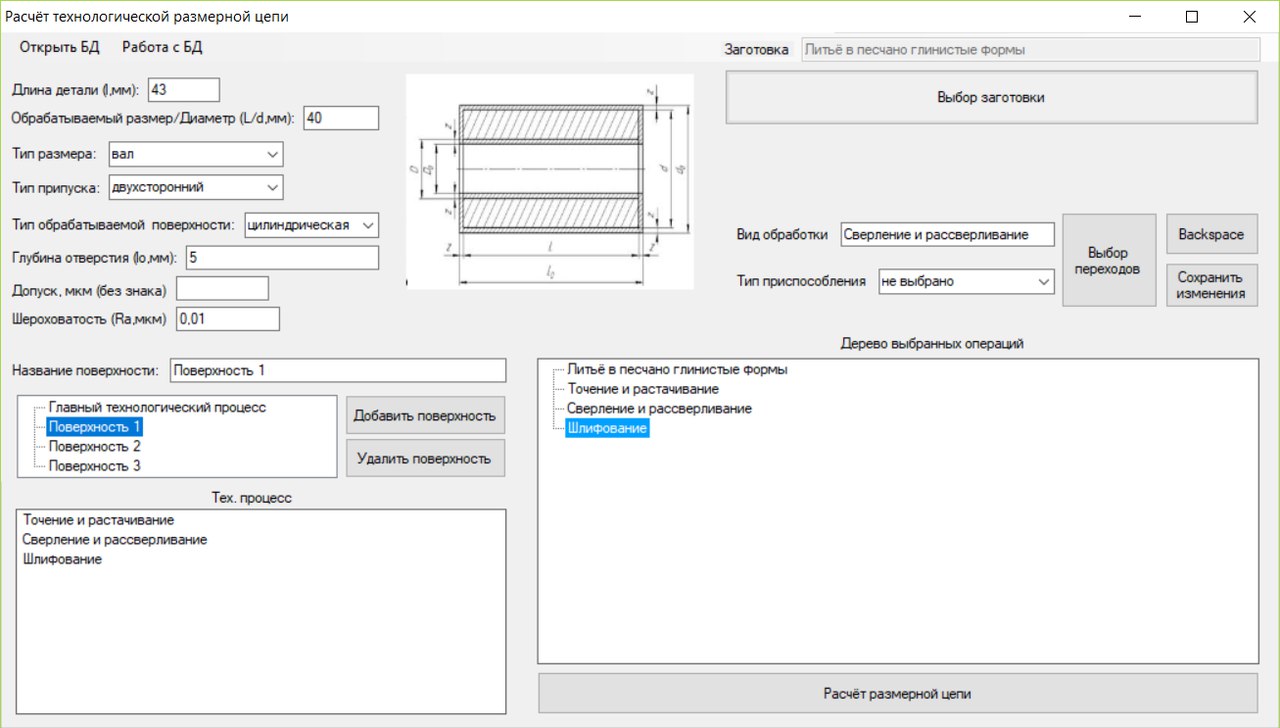


Рис. 3.15 Выбор операций первой поверхности.

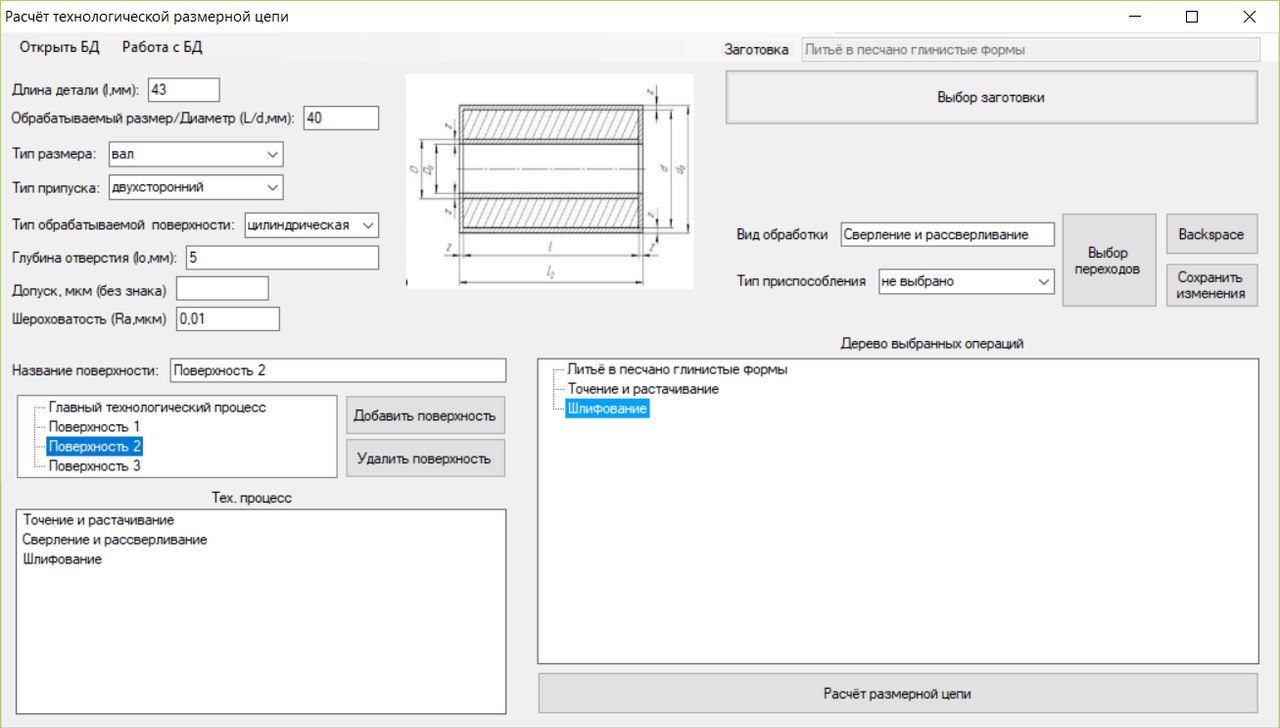


Рис. 3.16 Выбор операций второй поверхности.

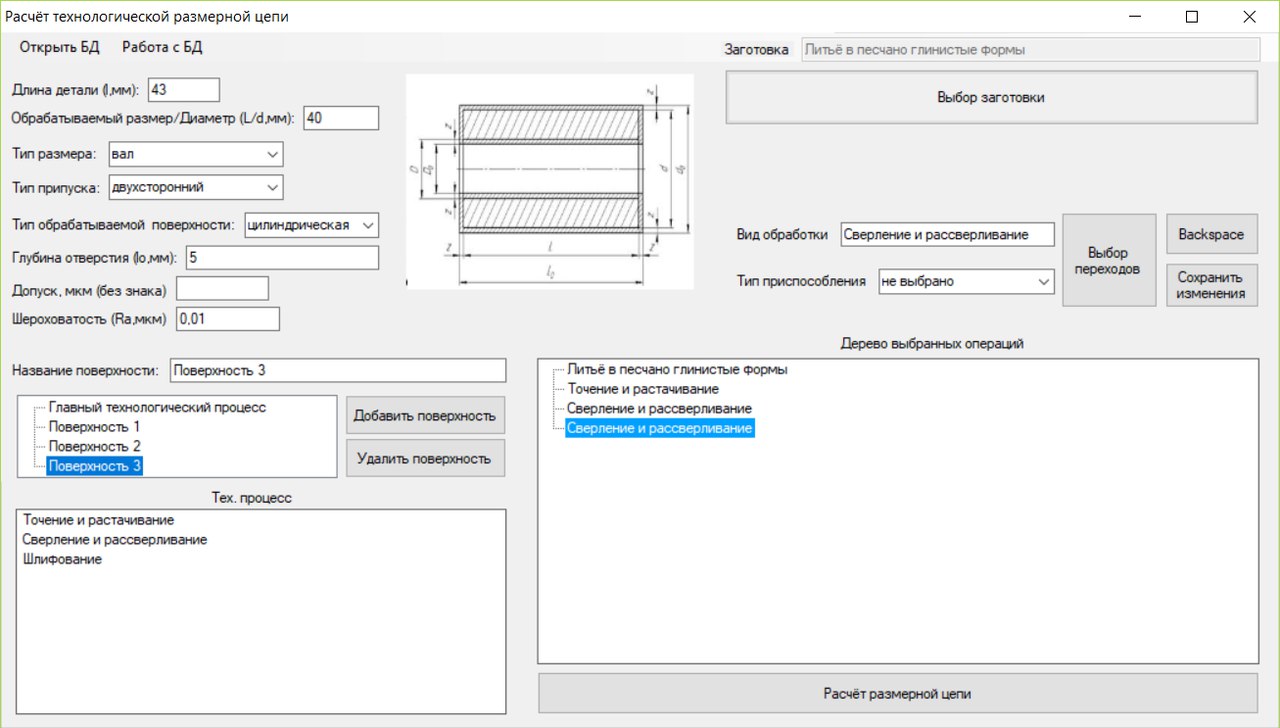


Рис. 3.17 Выбор операций третей поверхности.

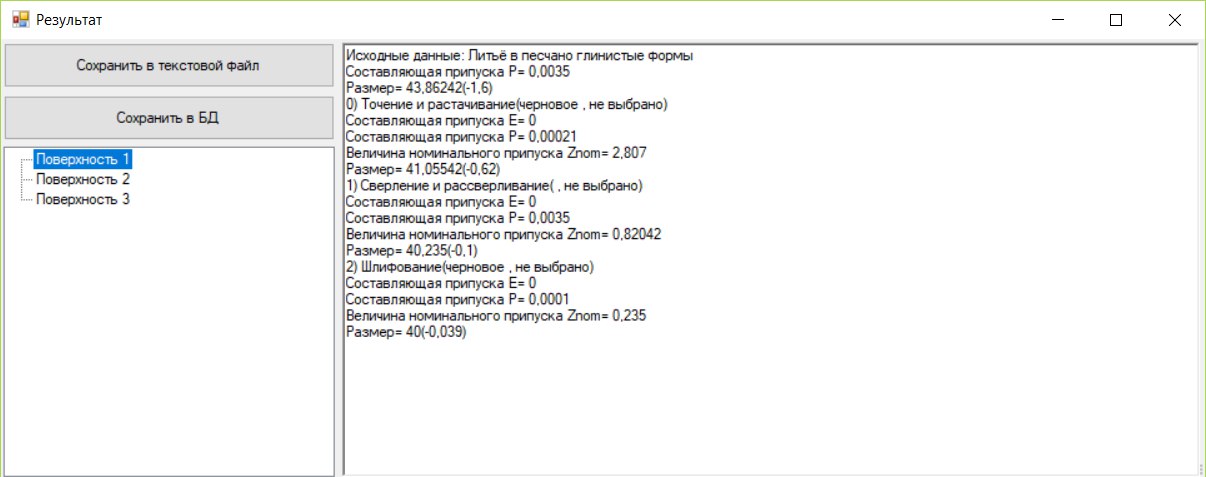


Рис. 3.18 Вывод результата расчёта первой поверхности.

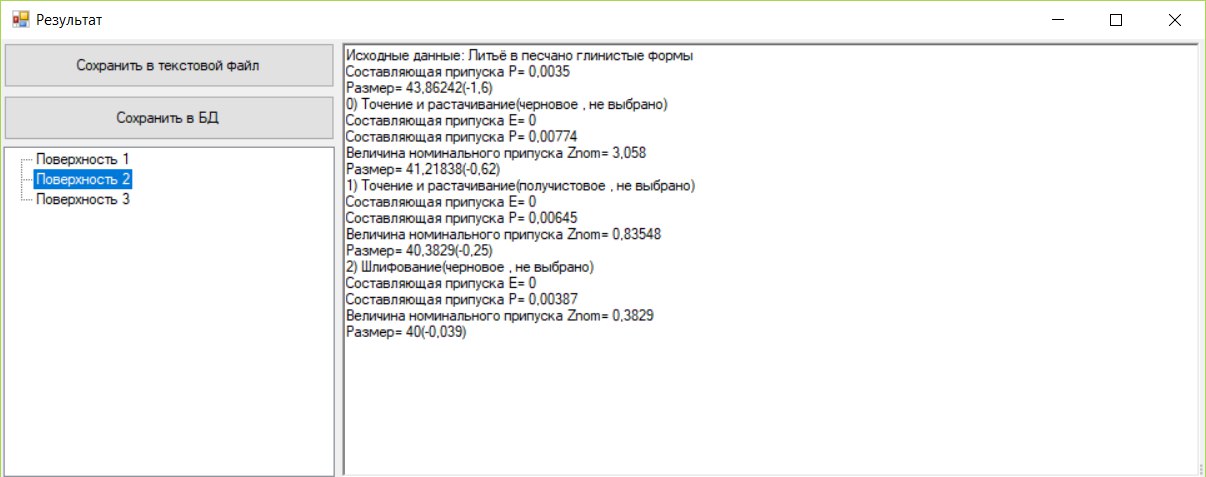


Рис. 3.19 Вывод результата расчёта второй поверхности.

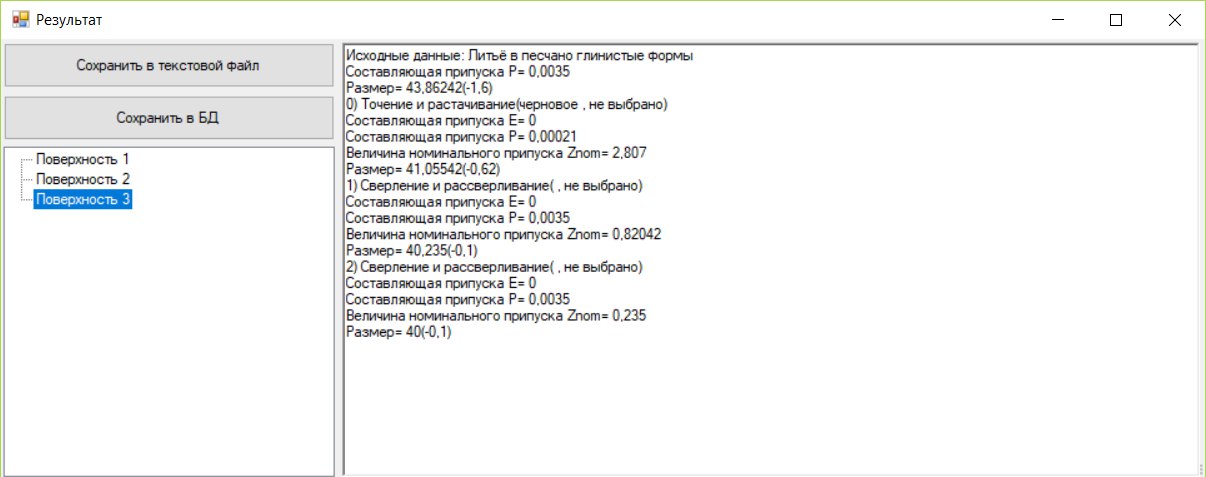


Рис. 3.20 Вывод результата расчёта третей поверхности.

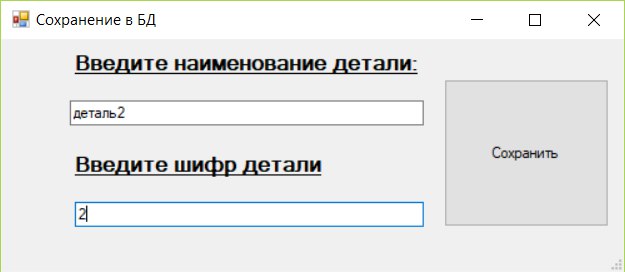


Рис. 3.21 Форма сохранения расчёта в БД.

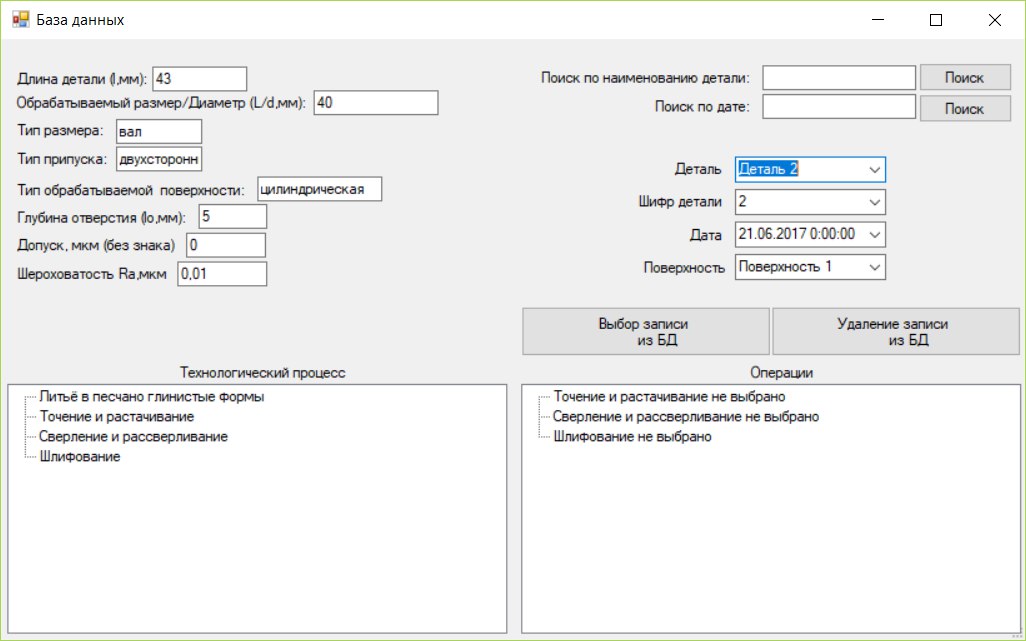


Рис. 3.22 Форма работы с БД.

**Заключение.**

В ходе работы поставленные задачи были выполнены.

1. Проведен анализ традиционного процесса решения задачи. Произведен сбор исходных данных, необходимых для расчета.
2. Разработан алгоритм расчета.
3. Разработан программное обеспечение для реализации расчета припусков на обработку детали.
4. Проведена апробация программного обеспечения.

Цель поставленной работы была достигнута.

**Библиографический список.**

1. ГОСТ 26645-85. Отливки из металлов и сплавов. Допуски размеров, массы и припуски на механическую обработку [Текст]. – М.: Изд-во стандартов, 1985.
2. ГОСТ 7505-89. Поковки стальные штампованные. Допуски, припуски и кузнечные напуски [Текст]. – М.: Изд-во стандартов, 1989.
3. ГОСТ 2591-2006. Прокат сортовой стальной горячекатаный квадратный. Сортамент [Текст]. – М.: Изд-во стандартов, 2006.
4. ГОСТ 2590-2006. Прокат сортовой стальной горячекатаный круглый. Сортамент [Текст]. – М.: Изд-во стандартов, 2006.
5. ГОСТ 103-2006. Прокат сортовой стальной горячекатаный полосовой. Сортамент [Текст]. – М.: Изд-во стандартов, 2006.
6. ГОСТ 2879-2006. Прокат сортовой стальной горячекатаный шестигранный. Сортамент [Текст]. – М.: Изд-во стандартов, 2006.
7. Корсаков В. С. Основы конструирования приспособлений в машиностроении. М.: Машиностроение, 1971, 286 с.
8. Справочник технолога – машиностроителя [Текст]. В 2 т./ под.ред. А.М. Дальского, А.Г. Суслова, А.Г. Косиловой, Р.К. Мещерякова – 5-еизд. – М.: Машиностроение, 2001. – 944 с.
9. Радкевич Я.А., Расчет припусков и межпереходных размеров в машиностроении: Учеб. пособие для машиностроительных спец. вузов/ Я.А. Радкевич, В.А. Тимирязев, А.Г. Схиртладзе, М.С. Островский; под ред. В.А. Тимирязева – М.: Высшая школа, 2004. – 272 с.
10. Объектно-ориентированный анализ и проектирование с примерами приложений [Текст] / Грэди Буч, Роберт А. Максимчук, Майкл У. Энгл и др. – М.: Вильямс, 2010. – 720 с.
11. Мейер Б., Объектно-ориентированное конструирование программных систем / Б. Мейер. – М.: Русская Редакция, 2005. – 1204 с.
12. Сайт компании Microsoft [Электронный ресурс]. – Режим доступа: [http://msdn.microsoft.com/ru-ru/](http://msdn.microsoft.com/ru-ru/library/66x5fx1b.aspx) , доступ: свободный.
13. Сайт компании ИНТЕРМЕХ [Электронный ресурс]. – Режим доступа: http://www.intermech.ru, доступ: свободный.
14. Сайт компании DP Technology [Электронный ресурс]. – Режим доступа: https://www.espritcam.ru, доступ: свободный.