

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

ИНСТИТУТ МАШИНОСТРОЕНИЯ
Кафедра « Сварка, обработка материалов давлением и родственные процессы»

15.04.01 Машиностроение

(код и наименование направления подготовки, специальности)

Системы автоматизированного проектирования в машиностроении

(профиль)

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на тему «Разработка метода формализации баз данных и знаний процесса проектирования мелких и средних формообразующих штампов» _____

Студент	<u>В.А. Соколенко</u> (И.О. Фамилия)	_____	(личная подпись)
Научный руководи- тель	<u>А.В. Скрипачёв</u> (И.О. Фамилия)	_____	(личная подпись)
Консультанты	<u>П.Н. Шенбергер</u> (И.О. Фамилия)	_____	(личная подпись)
	_____	_____	(личная подпись)

Руководитель программы к.т.н., доцент Е.Н. Почекуев _____
(ученая степень, звание, И.О. Фамилия) (личная подпись)

« _____ » _____ 20 _____ Г.

Допустить к защите

Заведующий кафедрой д.т.н., профессор В.В. Ельцов _____

_____ (ученая степень, звание, И.О. Фамилия) (личная подпись)
« _____ » _____ 20 _____ Г.

Тольятти 2016

Содержание

Введение	3
1. Обзор существующих конструкций мелких и средних формообразующих штампов и характеристики оборудования	7
1.1. Анализ формы изделий и технологии листовой штамповки	7
1.2. Оборудование и его характеристики	11
1.3. Сравнение существующих конструкций формообразующих штампов	14
2. Формализация баз данных и знаний процесса проектирования мелких и средних формообразующих штампов	30
2.1. Создание таблицы вариантов технических решений	30
2.2. Создание И-ИЛИ морфологического дерева	34
3. Создание системы автоматизированной поддержки процесса проектирования штамповой оснастки	38
3.1 Интерфейс программы	38
3.2 Разработка кода программы	45
3.3 Реализация связи морфологического И-ИЛИ дерева и кода программы	58
Заключение	63
Список используемой литературы	64
Приложение А	67

Введение

Среди всех операций проектирования можно выделить широкий класс алгоритмических операций, для которых уже созданы или могут быть созданы формальные модели. К ним можно отнести все расчёты, выполненные по стандартам: расчёты деталей машин на прочность, надёжность, а также кинематический и динамический анализ. Однако алгоритмические операции и процедуры составляют лишь часть процесса проектирования. Кроме них в процессе проектирования применяются такие операции и процедуры отличающиеся от алгоритмических неопределённостью в постановке задачи, методе решения и в окончательном результате технических решений и выборе из них оптимального. Например, опытный конструктор владеет приемами и методами поиска вариантов технических решений, их сравнения и выбора наилучшего, анализа технических систем, синтеза, определения оптимальных параметров и др. Данные процедуры чаще выполняются человеком, но могут быть реализованы с помощью компьютерных средств.

В настоящее время задача систем автоматизированного проектирования (САПР) состоит в совершенствовании техники проектирования, что позволяет снизить время на проектирование, а также увеличить качество выполняемых работ.

В процессе проектирования штамповой оснастки возникают трудности формализации интеллектуальной деятельности на начальных этапах разработки конструкции мелких и средних формообразующих штампов. Сложности создания формализованных баз данных процесса проектирования мелких и средних штампов заключаются в применении инженером-конструктором таких операций и процедур, для которых постановка задачи, метод решения и результат задан неопределённо.

Таким образом, создание компьютерной программы для помощи инженеру-конструктору в проектировании мелких и средних формообразующих штампов является актуальной задачей процесса подготовки технической документации.

Целью работы является разработка метода информационной поддержки принятия проектных решений и программных приложений, предназначенных для проектирования мелких и средних формообразующих штампов позволяющих упростить процесс проектирования за счет сокращения времени и повышения качества процесса.

Для достижения поставленной в работе цели необходимо решить следующие **задачи**:

- 1) Проведение анализа существующих конструкций формообразующих штампов, определить область их применения и выделить основные признаки, отвечающие за их отличительные особенности.
- 2) С помощью методов морфологического анализа выявить признаки и связи между узлами и деталями штампов, а также провести поиск всех возможных вариантов конструкций формообразующего инструмента.
- 3) На основании собранных данных и проведённого анализа, разработать программное приложение позволяющее получить текстовое описание возможных вариантов конструкции штампов, с учетом введенных пользователем данных в окнах диалога.

Основные положения работы позволили создать программное приложение для помощи инженеру – конструктору и отработать методику создания формализованных баз данных и знаний.

Методы исследования: Для решения поставленных в работе задач применяются методы объектно-ориентированного программирования и морфологического анализа.

Достоверность результатов работы подтверждается корректным применением методов анализа и репрезентативным компьютерным моделированием.

Научная новизна. В диссертационной работе получены следующие новые результаты: предложен новый метод проектирования мелких и средних формообразующих штампов на основе методов морфологического анализа и объектно-ориентированного программирования.

Теоретическая и практическая ценность. На основе метода формализации баз данных для мелких и средних формообразующих штампов был предложен новый более простой алгоритм создания программных приложений на основе которых могут создаваться параметризованные базы данных и знаний математических моделей для САПР.

Результаты работы были использованы в процессе проектирования штампов для оснащения нового проекта компании «Renault-Nissan» на предприятии ОАО «ТЗТО».

Апробация результатов. Основные положения диссертационной работы докладывались и обсуждались на заседаниях кафедры обработки материалов давлением ТГУ, а также на конференциях «Дни науки ТГУ».

Публикации. По теме диссертационной работы опубликовано две печатные работы. Первая « Разработка метода формализации баз данных и знаний процесса проектирования последовательных разделительных штампов» опубликована в сборнике «Теплофизические и технологические

аспекты повышения эффективности машиностроительных производств», сборник составлен из трудов IV международной научно-технической конференции (Резниковские чтения) 2015 г. Вторая работа «Формализация баз данных и знаний процесса проектирования мелких и средних формообразующих штампов» в данный момент публикуется в сборнике научных работ кафедры «Сварка, обработка материалов давлением и родственные процессы» за 2016 г.

Объем и структура работы. Диссертация состоит из введения, трёх глав, заключения, списка использованной литературы и приложения. Общий объем диссертации – 91 страницы. Список литературы состоит из 30 наименований. Работа содержит 61 рисунок и 2 таблицы.

Глава 1. Обзор существующих конструкций мелких и средних формообразующих штампов

1.1. Анализ формы изделий и технологии листовой штамповки

Усовершенствование конструкции автомобиля приводит к необходимости изготовления большого количества разнообразных мелких и средних кузовных деталей сложной формы, которые получают различными формообразующими операциями. К таким деталям относятся, например, кронштейн крепления проводов в сборе (рисунок 1.1), кожух защитный рычага ручного привода тормоза (рисунок 1.2 – 1.3) и т.д.

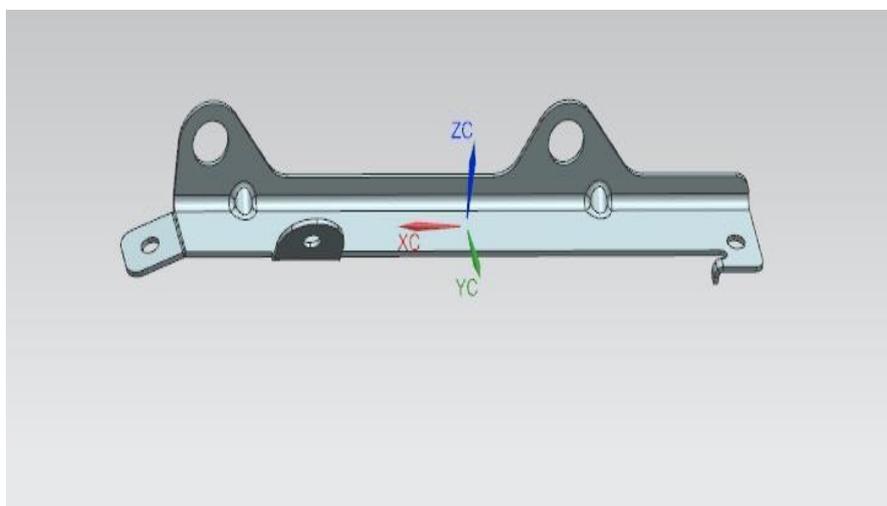


Рисунок 1.1 - Кронштейн крепления проводов в сборе

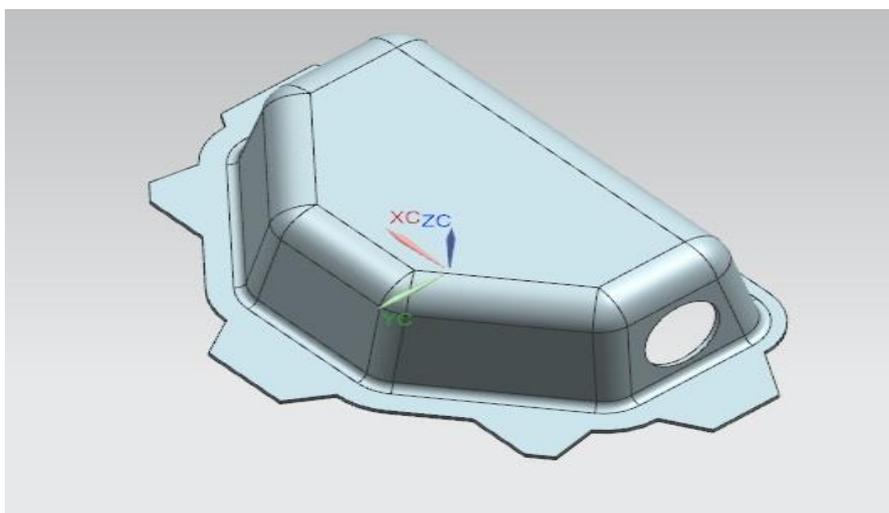


Рисунок 1.2 - Кожух защитный рычага ручного привода тормоза

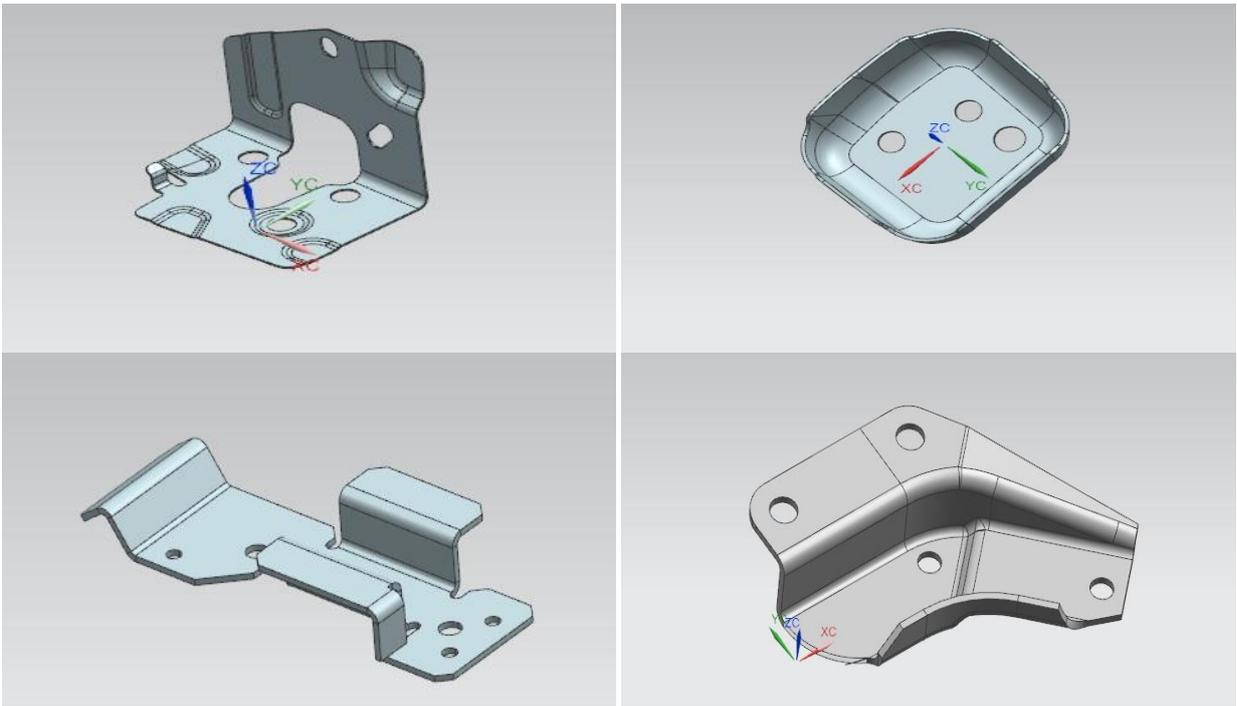


Рис.1.3 - Электронные модели мелких и средних деталей, получаемых методами холодной листовой штамповки

Требования к деталям их размерам и допускам предъявляют согласно чертежа изделия (рисунок 1.4)

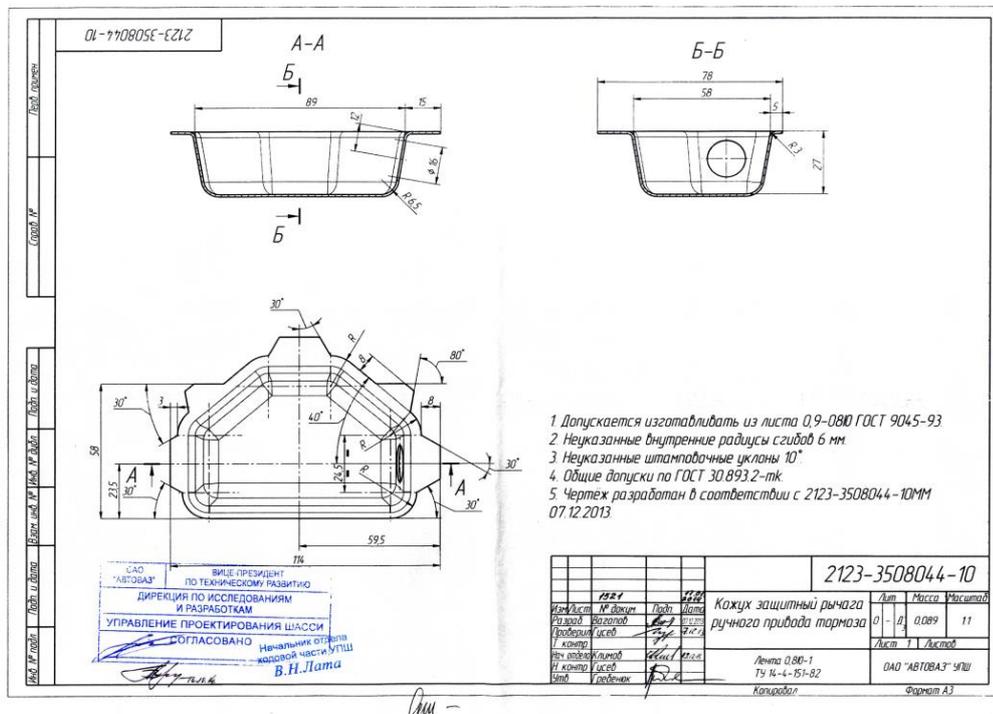


Рисунок 1.4 – Чертеж изделия

Технологические особенности данных деталей:

1. Габариты деталей не превышают 250 x 250 мм.
2. Толщина материала определяется в интервале от 0.5 до 4 мм.
3. Практически в каждой детали присутствуют отверстия, выполненные от 6 до 10 квалитета точности.
4. Заготовками являются карточки, получаемые на вырубных штампах из ленты.
5. Материалом служат низкоуглеродистые стали, такие как Ст 08пс, 08кп, 08ю, Ст 10, Ст 20, Ст 45, Ст 50, Ст 70, Ст 65г, Ст 07ГБЮ, а также их зарубежные аналоги со схожими механическими свойствами.

Формообразующие операции листовой штамповки могут быть совмещёнными, решение о совмещении операций делает инженер-технолог, исходя из симуляции процесса штамповки в САЕ программах. Примеры технологических процессов штамповки представлены на рисунках 1.5 - 1.6.

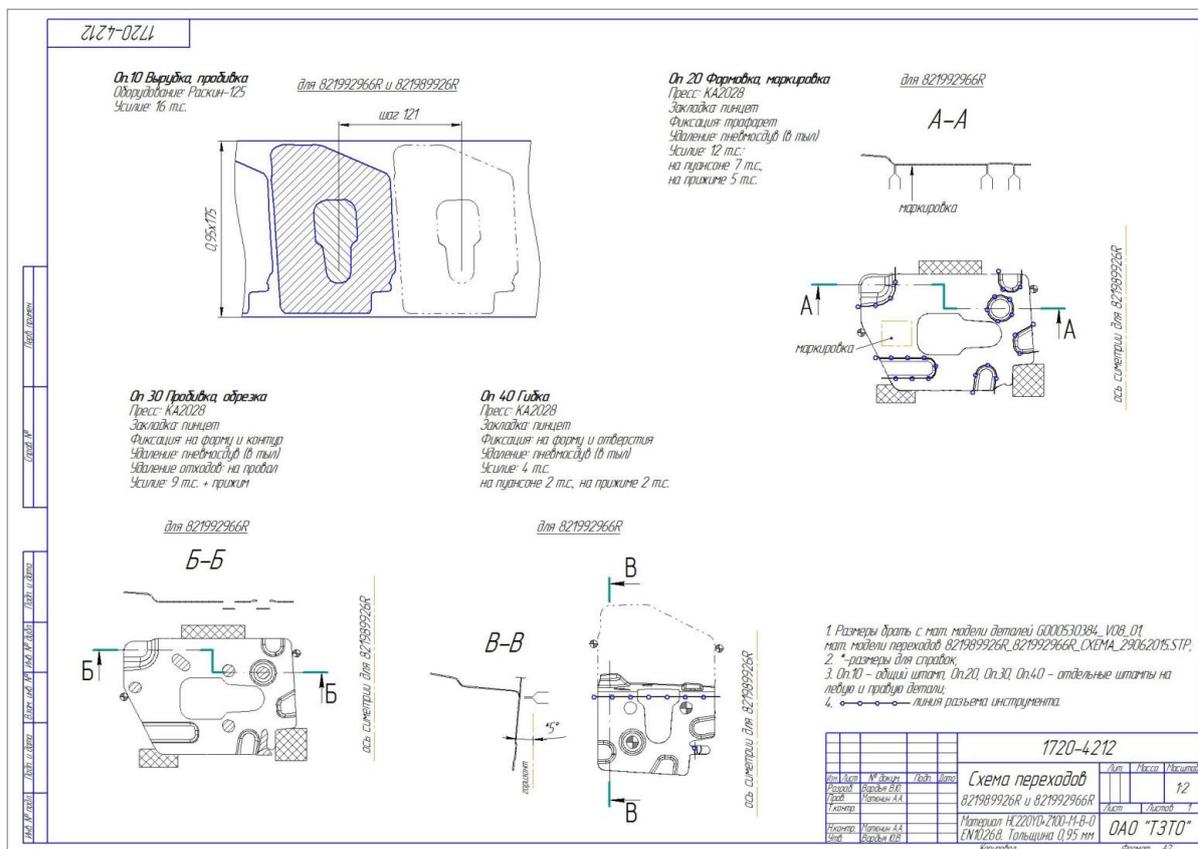
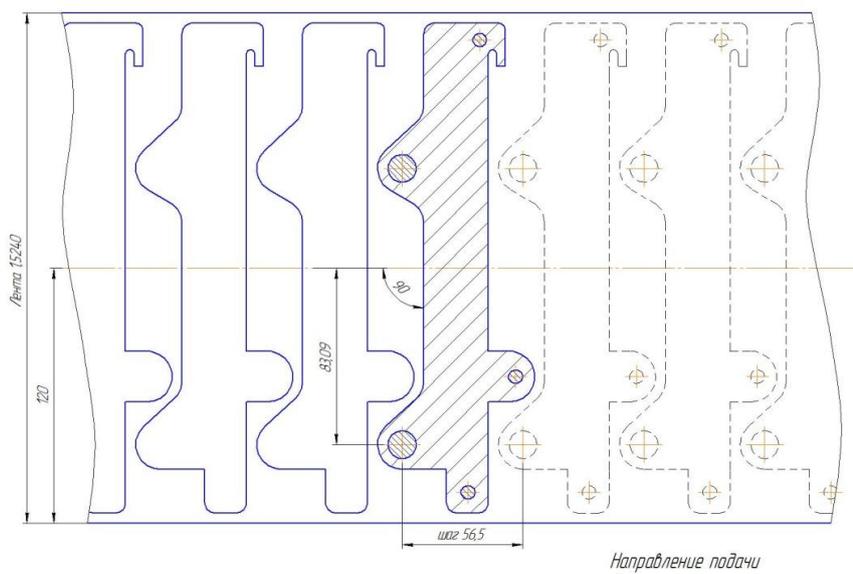


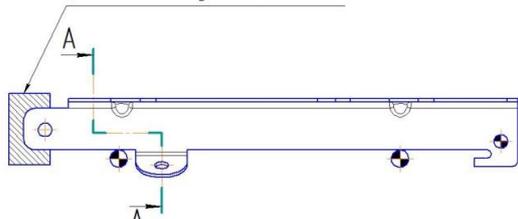
Рисунок 1.5 – Пример технологического процесса

Оп. 10 "Вырубка-пробивка"
 Pp=40,5 т.с.
 Оборудование: пресс-автомат Раскин-125 ус. 125 т.с.



Оп. 20 "1-я гибка, формовка"
 Pp=10,5 т.с.
 Оборудование: пресс КА-2028 ус. 63 т.с.
 Фиксация: на отверстие, на контур.
 Загрузка детали: пинцет.
 Удаление детали: пневмосдув.

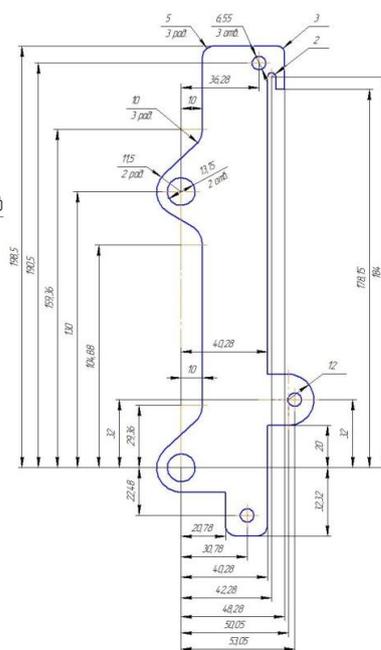
Фиксация на контур.
 Для однозначной укладки заготовки.



ФРОНТ

Выполнить подчеканку радиусов

A-A

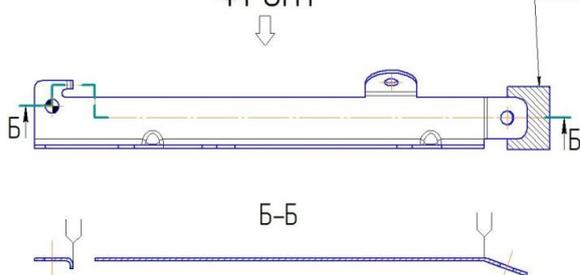


Контур вырубки

Оп. 30 "2-я гибка"
 Pp=0,5 т.с.
 Оборудование: пресс КА-2028 ус. 63 т.с.
 Фиксация: на отверстие, на контур.
 Загрузка детали: пинцет.
 Удаление детали: пневмосдув.

ФРОНТ

Фиксация на контур.
 Для однозначной укладки заготовки.



B-B

1. Материал: Сталь А 08Пс ВГ-2
Лента 15240
Шаг 56,5
2. Недостающие размеры брать с мат. моделей карты переходов.
3. Неуказанные радиусы вырубки 0,5 мм.

Рисунок 1.6 – Пример технологического процесса для детали кронштейн крепления проводов в сборе

1.2. Оборудование и его характеристики

Рассматриваемые детали получают методами традиционной штамповки на маломощном оборудовании, например на кривошипных прессах с Г-образной станиной К18014 – 1,60 МН (рисунок 1.7) и К2028 - 0,63 МН.(рисунок 1.8) В этих установках, имеющих механизмы кривошипно-ползунного вида, движение вращательного привода преобразуется в поступательное движение ползуна, благодаря чему функционирует пресс.

Кривошипный механический пресс может выполнять работу в условиях одиночных ходов. То есть это означает, что работа выполняется с выключением муфты, которое можно осуществлять после каждого выполненного полного хода, или же в режиме автомата, это означает, что муфта может быть включена все время. Наиболее важные характеристики прессы, а это, прежде всего ход ползуна, размеры стола, количество ходов ползуна в минуту в автоматическом режиме, номинальное усилие, в комплексе определяющие его возможности представлены в таблице 1.1.

Таблица 1.1 - Основные характеристики оборудования

Характеристики.	К18014 – 1,60 МН	К2028-0.63 МН
Число ходов ползуна в минуту.	50	90
Ход ползуна. мм.	220	120
Регулировка закрытой высоты. мм.	100	90
Размеры стола. мм.	670x1000	790x480
Усилие пневмоподушки. МН	0.125	0.6
Ход подушки. мм.	90	70
Рекомендованная закрытая высота. мм.	310	230
Управление.	Двухручный пульт, педаль.	Двухручный пульт, педаль.



Рисунок 1.7 - Пресс К18014 – 1,60 МН



Рисунок 1.8 - Пресс К2028 - 0,63 МН

Инструмент или как его еще называют рабочая часть, представляет собой штамп. Его неподвижная часть крепится непосредственно к столу, а подвижную часть прикрепляют к ползуну механического прессы. В условиях совершения одного оборота кривошипа шатунный механизм выполняет полноценный ход, в течение которого при условии движения самого ползуна вперед выполняется штамповка. Привод представляет из себя механизм, состоящий из маховика, электродвигателя, тормоза. Вращение маховика осуществляется с помощью электродвигателя. Под воздействием силы инерции данного маховика на самом кривошипном валу появляется крутящий момент.

1.3 Сравнительный анализ существующих конструкций средних и мелких формообразующих штампов

Формообразующие штампы для производства мелких и средних деталей могут значительно различаться в зависимости от требований предъявляемых к конструкции.

1) Анализ оборудования рассматриваемой группы штампов позволил установить две основные модели прессов, применяемых для мелкой и средней штамповки. К такому оборудованию относятся прессы К18014 - 160 т.с и К2028 – 63 т.с.

2) По виду осуществляемой операции анализируемые мелкие и средние формообразующие штампы представлены штамповой оснасткой, предназначенной для гибки, формовки, вытяжки, отбортовки.

3) Рабочие части штампа крепятся к верхним и нижним плитам штампа. Верхняя плита штампа крепится к ползуну и является подвижной, нижняя плита закрепляется на неподвижном столе прессы. Плиты изготавливают из различных сталей, таких как Ст 3, Ст 45, Ст 40Х. На рисунках 1.9 – 1.12 показаны электронные модели конструкций плит.

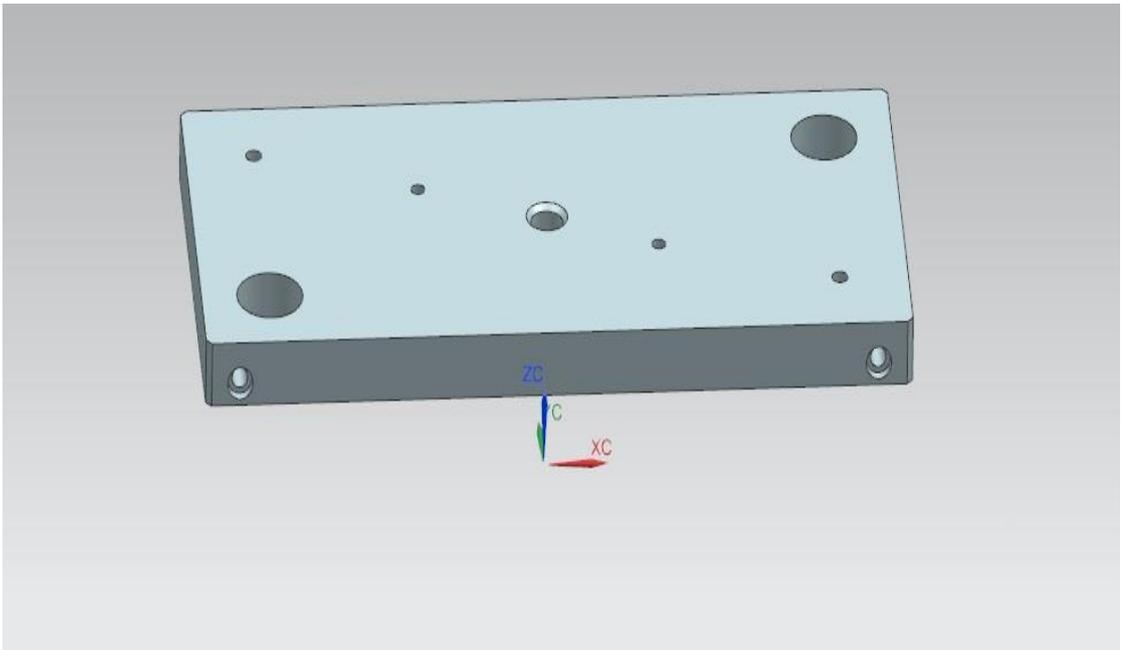


Рисунок 1.9 – Верхняя плита для прессы K2028 – 0.63 МН

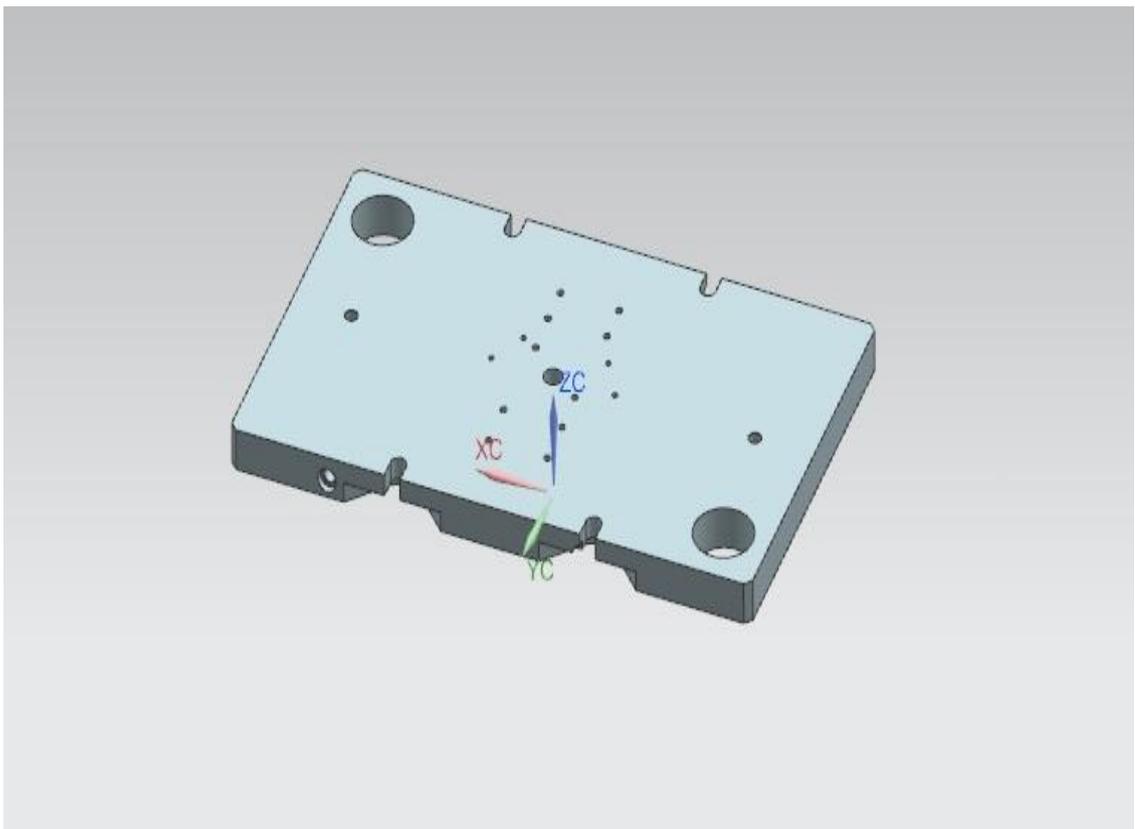


Рисунок 1.10 – Верхняя плита для прессы K18014 – 1.60 МН

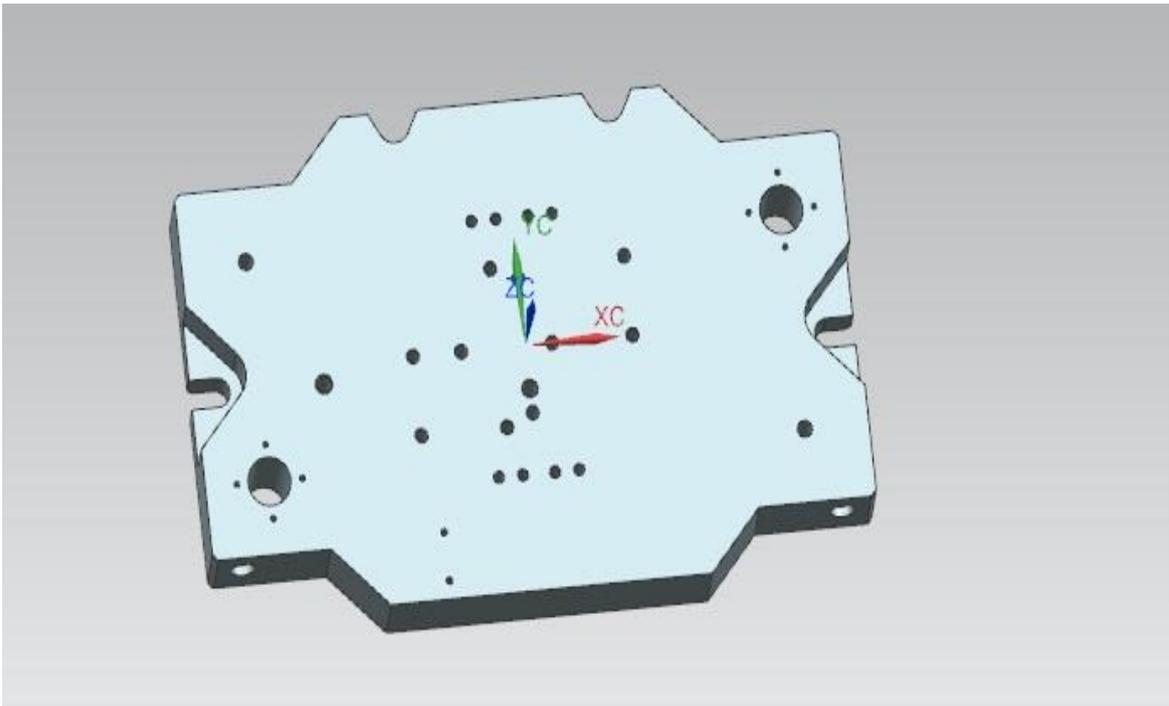


Рисунок 1.11 – Нижняя плита для пресса К2028 – 0.63 МН

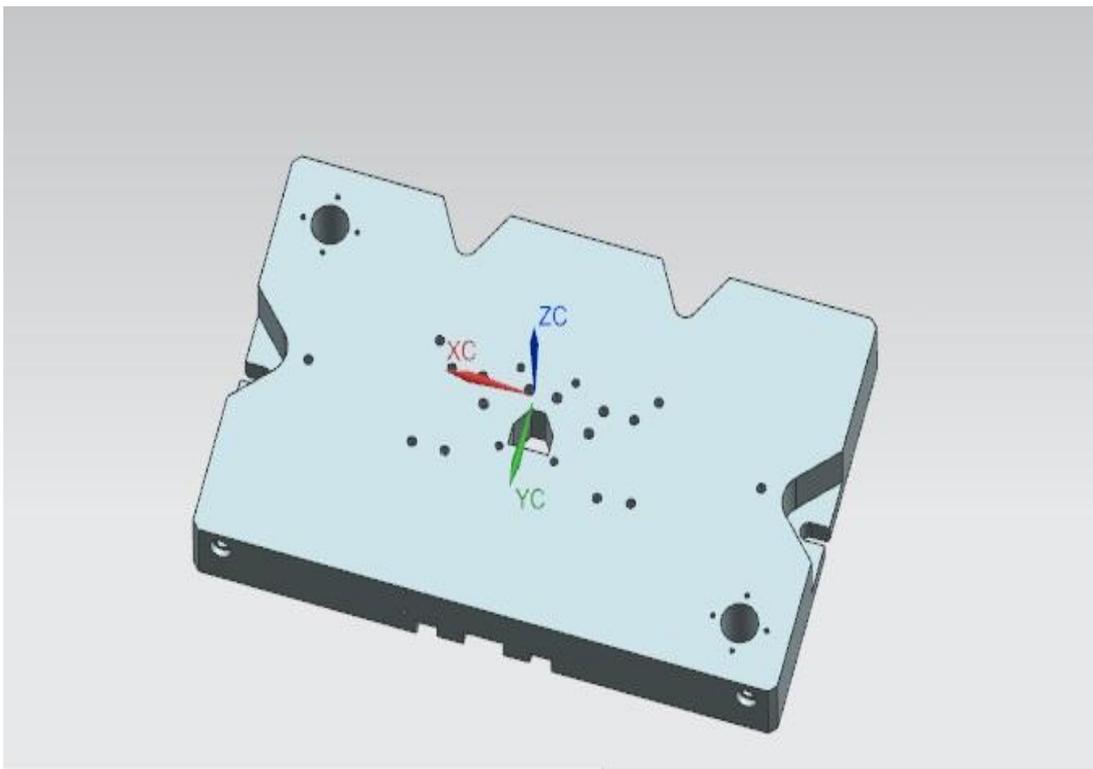


Рисунок 1.12 – Нижняя плита для пресса К18014 – 1.60 МН

Крепление верхних плит к ползуну пресса может осуществляться за счёт хвостовиков как показано на рисунке 1.13. Дополнительно в плите могут

быть выполнены освобождения для крепления плиты верха винтами за счет использования Т-образных пазов ползуна (рисунок 1.14).

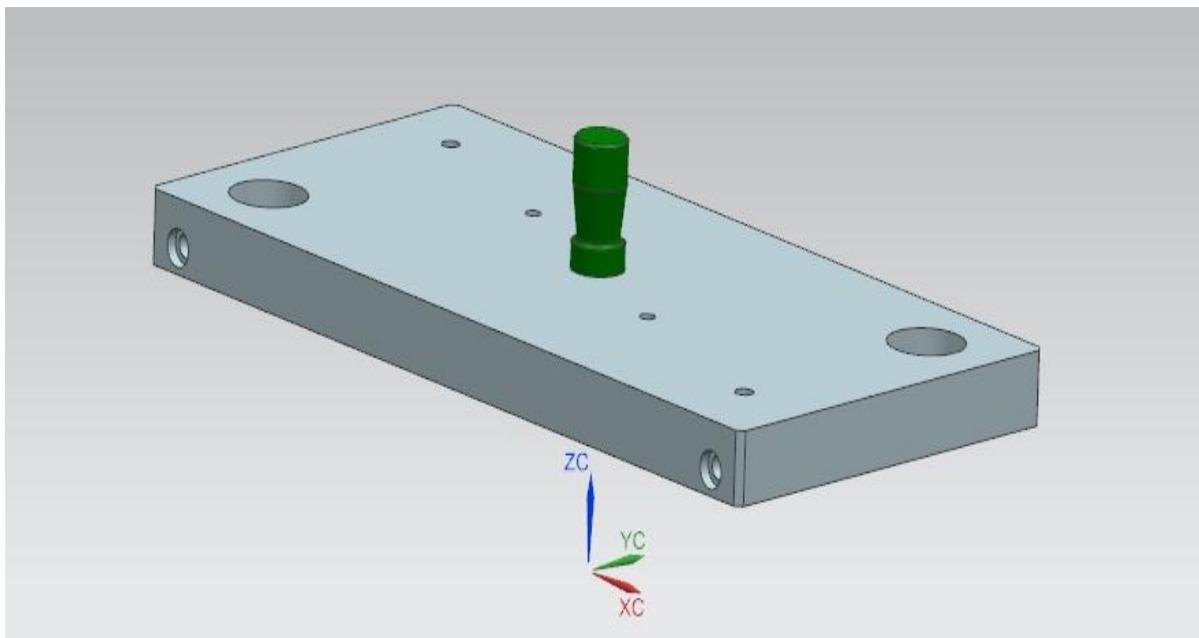


Рисунок 1.13 – Крепление плиты верха с помощью хвостовика

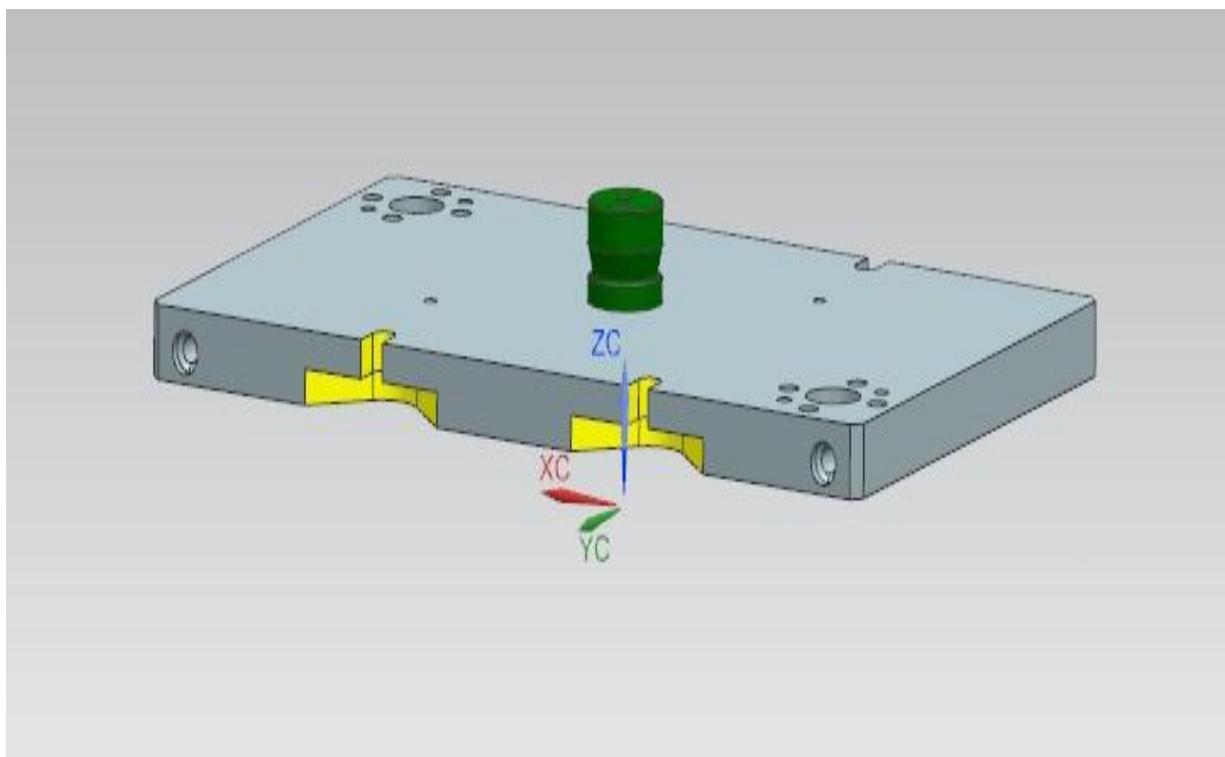


Рисунок 1.14 – Крепление плиты верха с помощью хвостовика и крепежных пазов

На нижних плитах выполняются только крепежные пазы для крепления к столу пресса (рисунок 1.15).

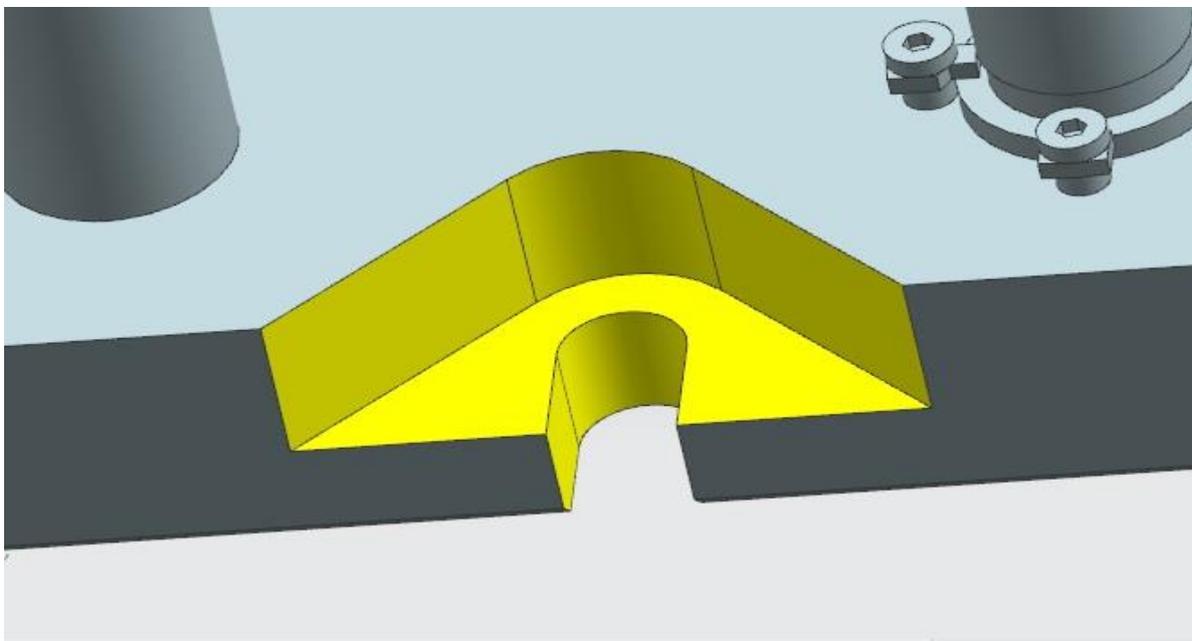


Рисунок 1.15 – Освобождение и крепежный паз плиты низа

Для быстрой установки штампа на пресс с тыла нижних плит выполняются освобождения (рисунок 1.16).

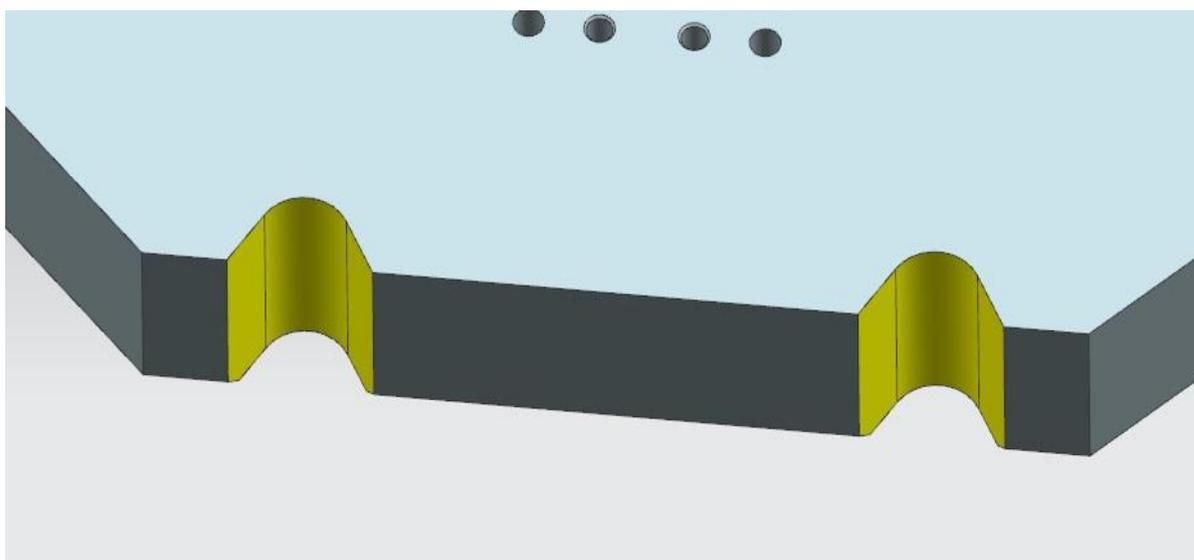


Рисунок 1.16 – Освобождения для быстрой установки штампа на пресс

3) В качестве деталей направляющих узлов в мелких и средних штампах используются колонки и втулки. Втулки используют ступенчатые запрессовываются в верхнюю плиту (рисунок 1.17) или сначала в держатель, а затем в плиту (рисунок 1.18). Колонки направляющие используют гладкие с проточкой. Колонки запрессовываются в нижнюю плиту (рисунок 1.19 - рисунок 1.20). Направляющие колонки и втулки изготавливают из низкоуглеродистой стали с цементацией и закалкой.

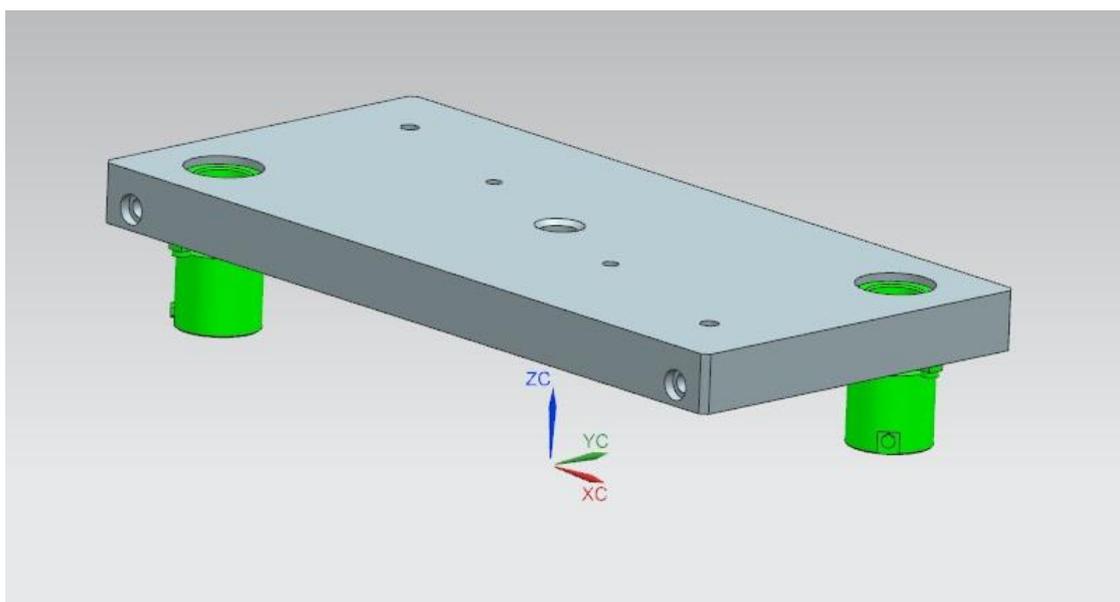


Рисунок 1.17 - Втулки запрессованные в плиту

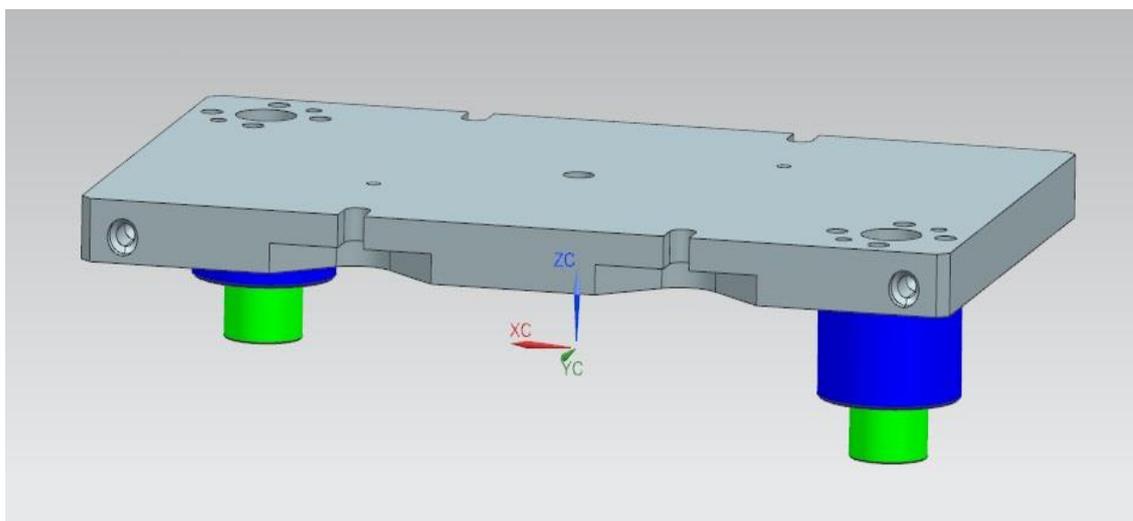


Рисунок 1.18 - Втулки запрессованные в держатель

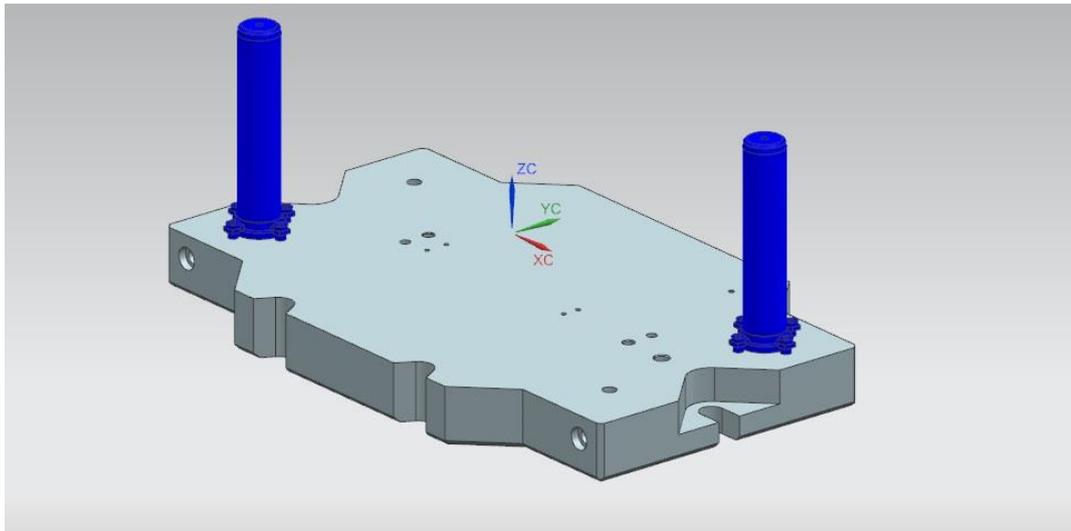


Рисунок 1.19 - Колонки запрессованные в нижнюю плиту (исполнение 1)

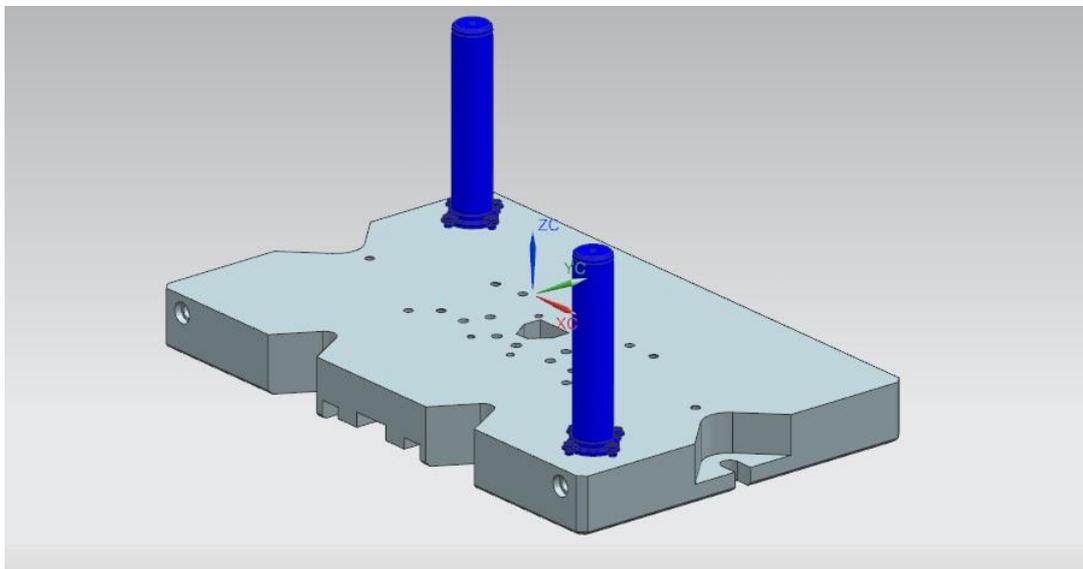


Рисунок 1.20 - Колонки запрессованные в нижнюю плиту (исполнение 2)

Как видно из рисунков направляющие узлы располагаются по диагонали в угловых зонах штамповых плит. Для рассматриваемой группы штампов является характерным и достаточным использование двух пар направляющих колонок и втулок.

4) Для транспортировки мелких и средних формообразующих штампов используют рым – болты, цапфы (рисунок 1.21) и транспортировочные накладки (рисунок 1.22). Транспортные элементы расположены на верхней и

нижней плитах штампа. Транспортные элементы изготавливают из конструкционных сталей. Выбор их типоразмеров зависит от массы штампа.

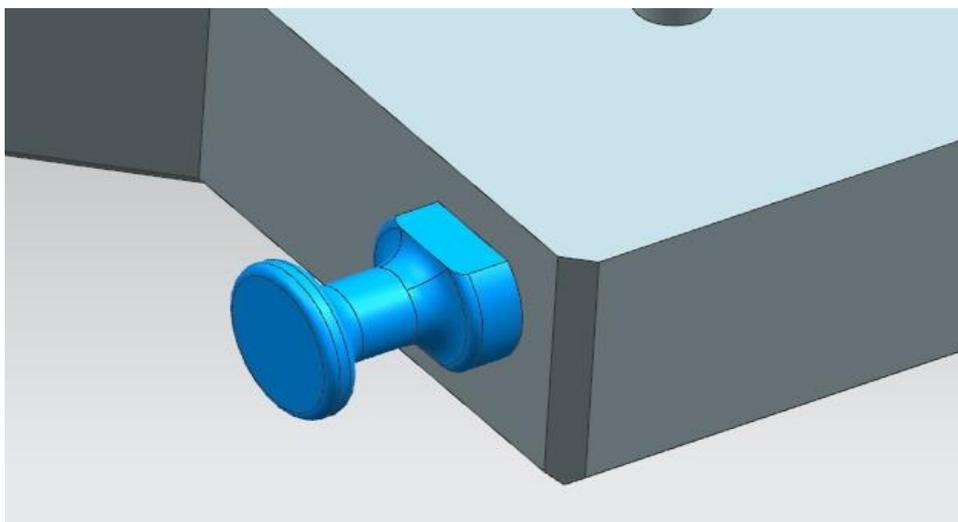


Рисунок. 1.21 - Цапфы транспортные

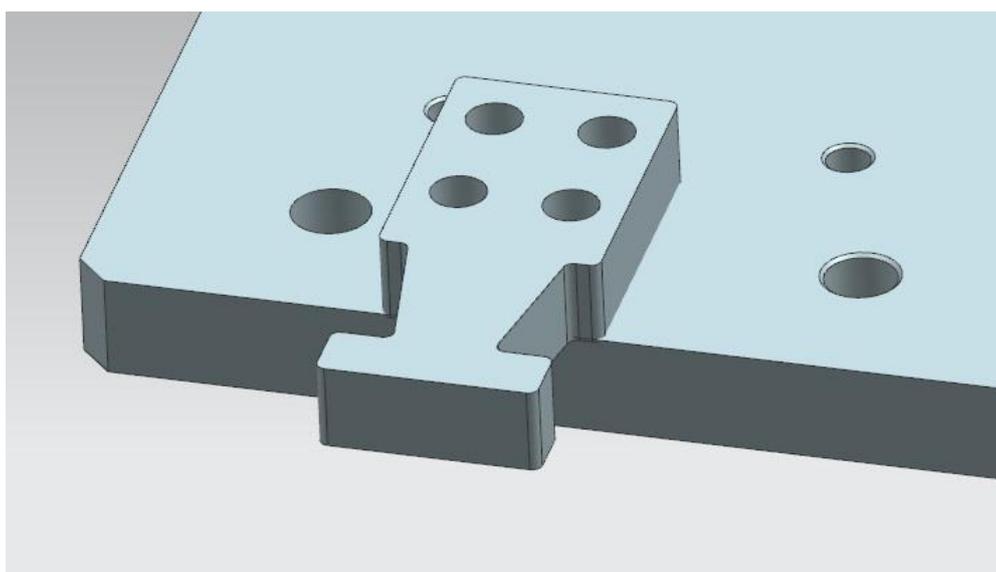


Рисунок. 1.22 - Накладки транспортировочные

5) Ограничители закрытой высоты предназначены для предотвращения удара рабочих частей. Цилиндрические ограничители рассматриваемой группы штампов расположены в углах плит (рисунок 1.23) или на направляющем узле в виде кольца на колонке (рисунок 1.24). Зазор между ограничителями при работе после наладки примерно равен 0.2 мм. Изготавливают ограничители из конструкционных сталей.

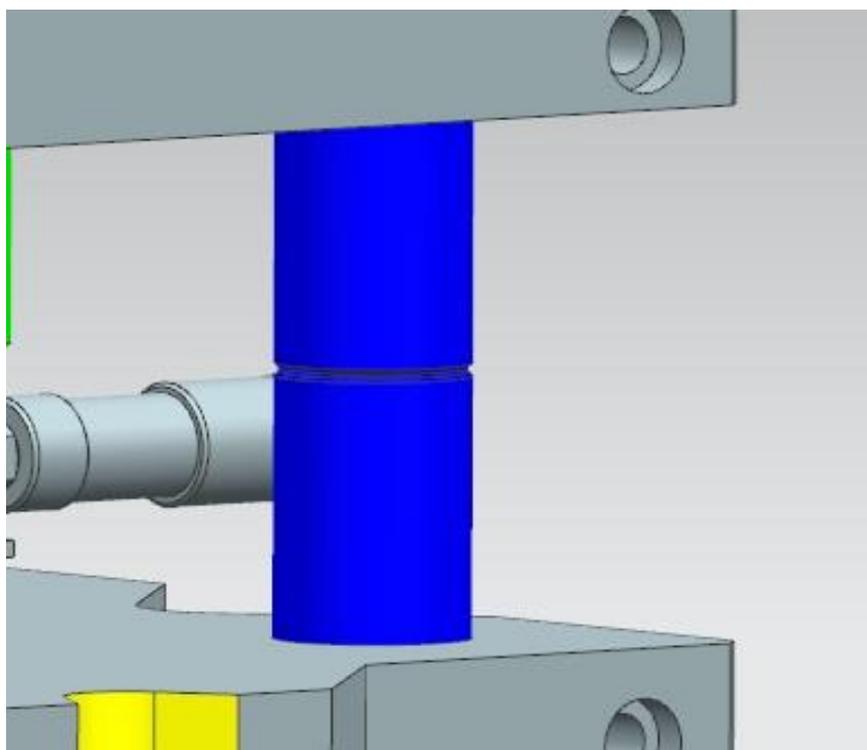


Рисунок. 1.23 - Цилиндрические ограничители

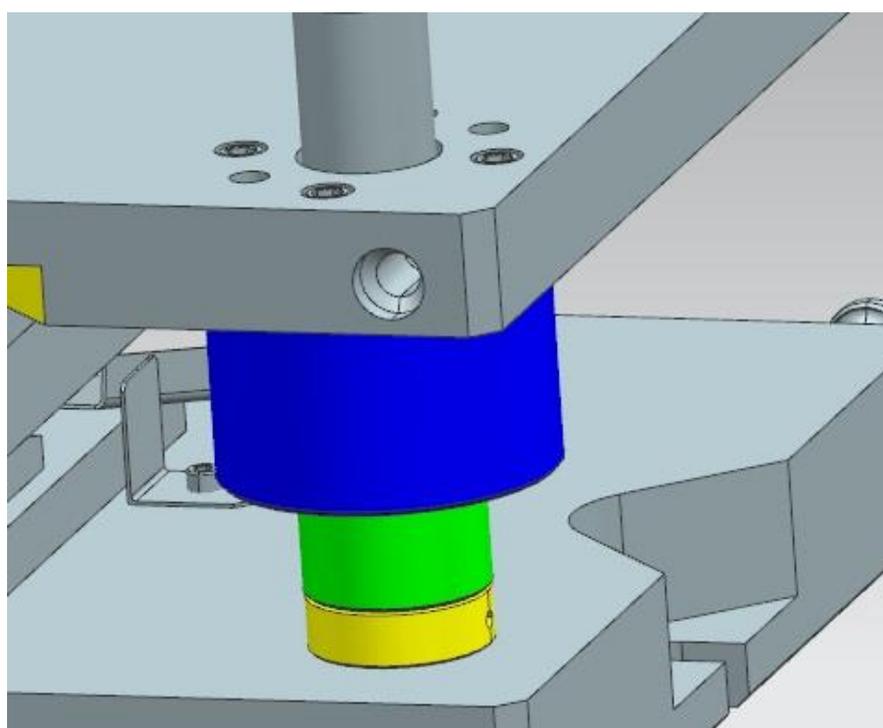


Рисунок. 1.24 - Ограничитель выполненный в виде кольца на колонке

б) Закладка заготовок в штамп может осуществляться как в ручную (пинцетом) так и за счёт автоматического пневмоукладчика (рисунок 1.25), работающего от пневмосистемы прессы.



Рисунок 1.25 – Пневмоукладчик

7) Удаление деталей производится вручную (пинцетом) или автоматически. При ручном удалении деталей на рабочих частях штампа может выполняться освобождения для удобства работы (рисунок 1.26). Автоматическое удаление осуществляется пневмосдувом (рисунок 1.27) и пневмоподбрасывателями (рисунок. 1.28).

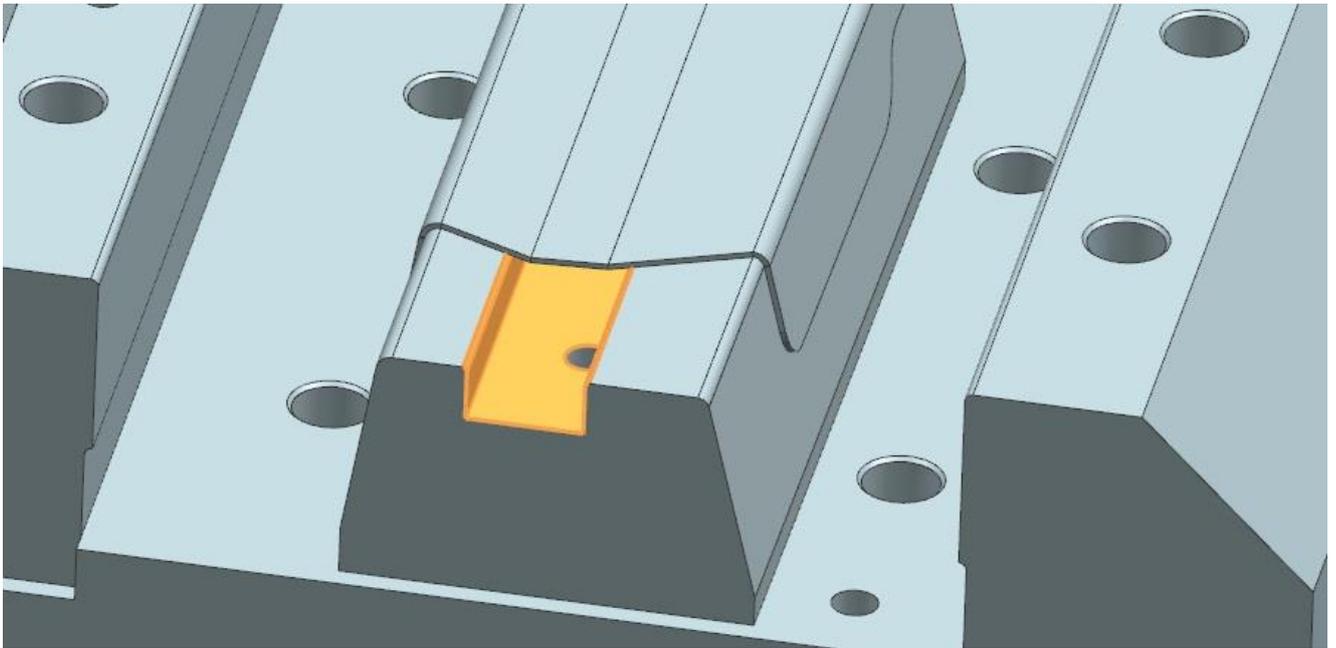


Рисунок 1.26 - Освобождение для удобства удаления

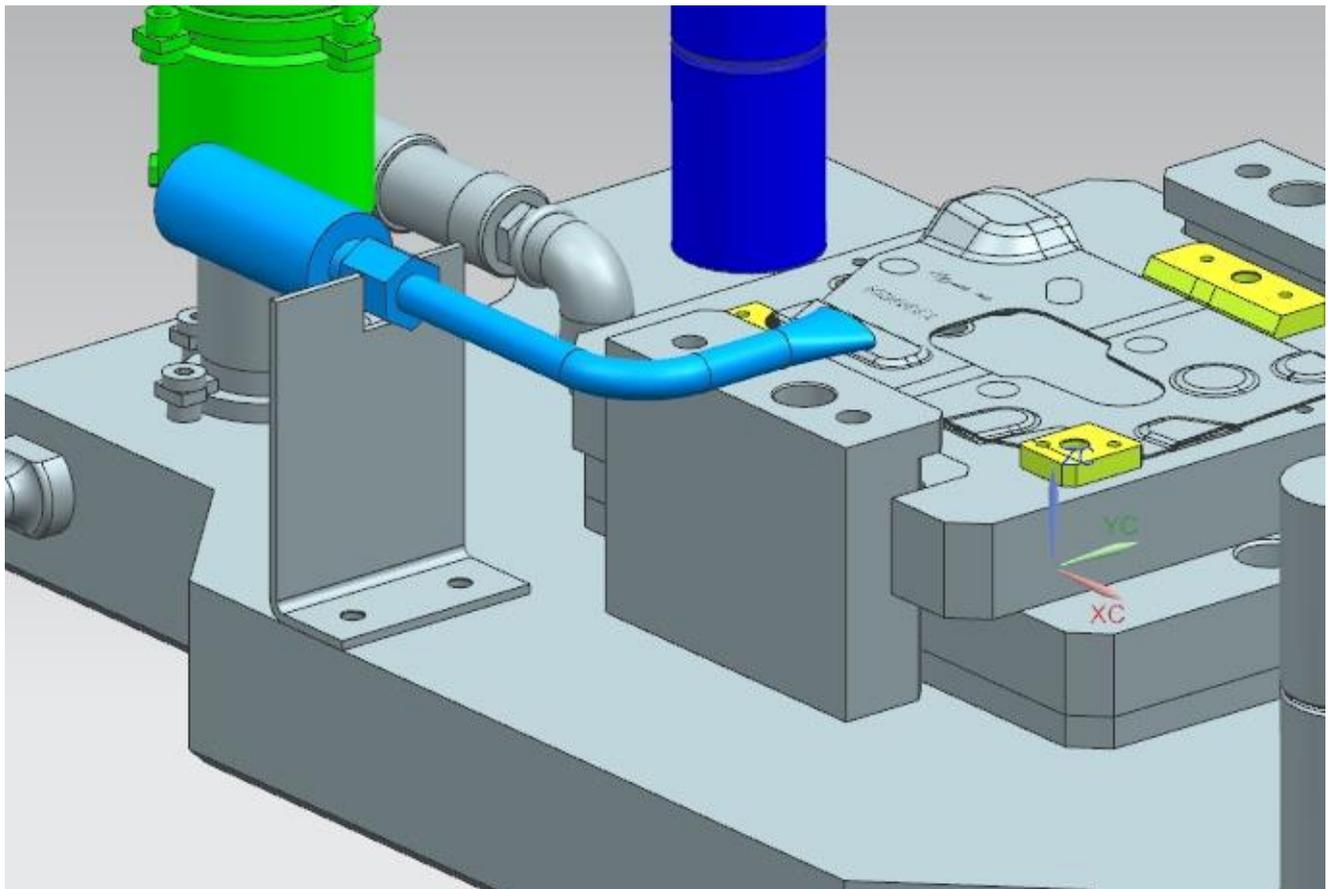


Рисунок 1.27 - Пневмосдуд

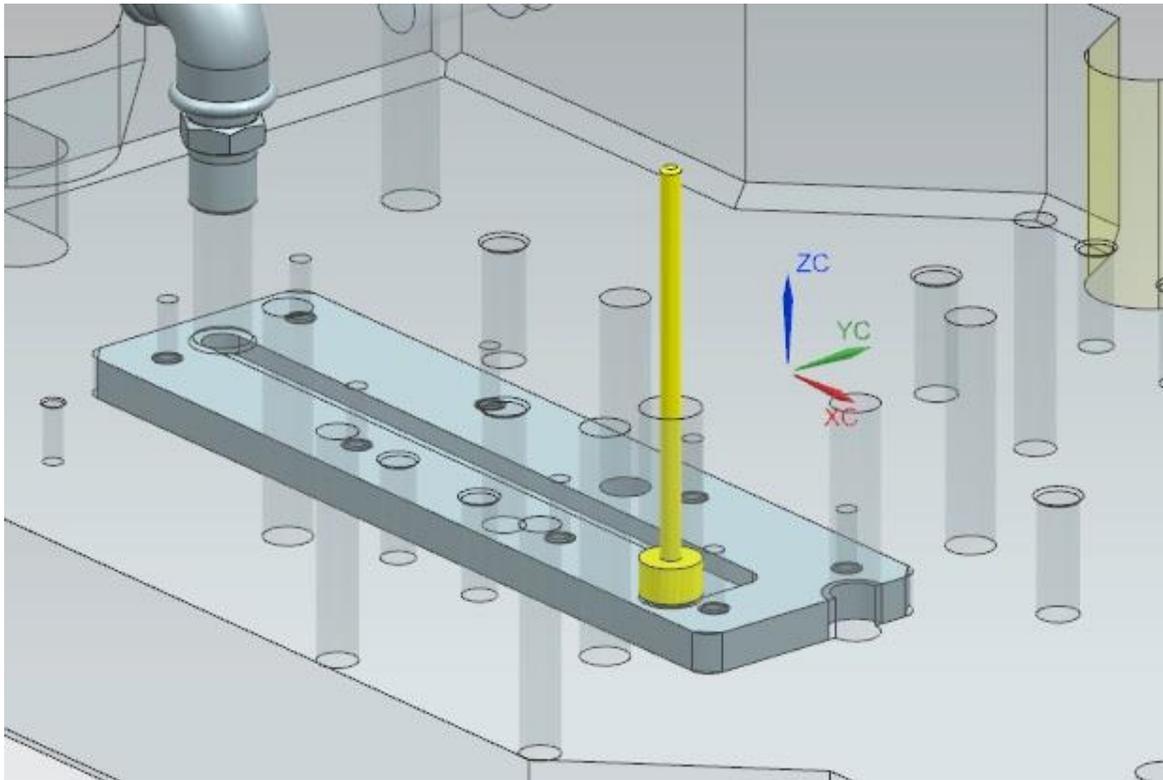


Рисунок 1.28 - Подбрасыватели

8) Фиксация деталей в штампах может осуществляться по отверстиям (рисунок 1.29), по контуру детали (рисунок 1.30), по форме детали (рисунок 1.31). Фиксаторы изготавливают из инструментальных легированных сталей с дополнительной термообработкой.

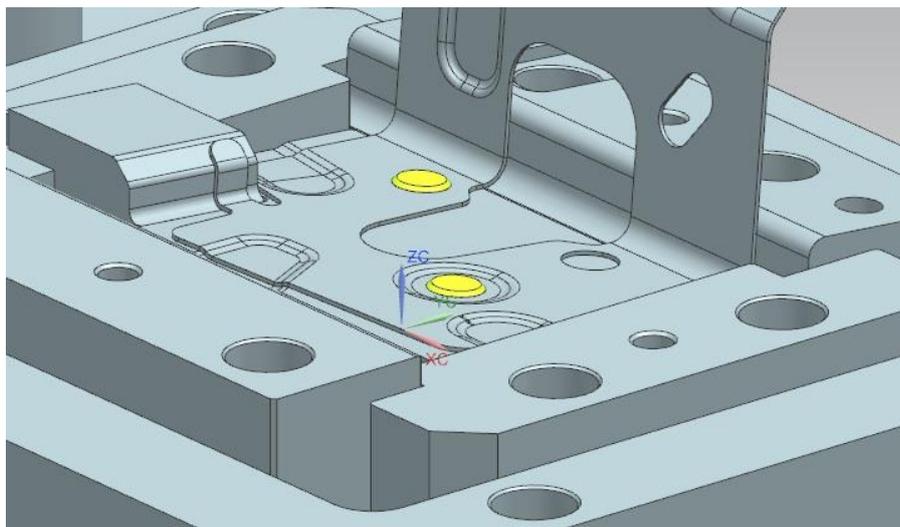


Рисунок 1.28 - Фиксация деталей по отверстиям

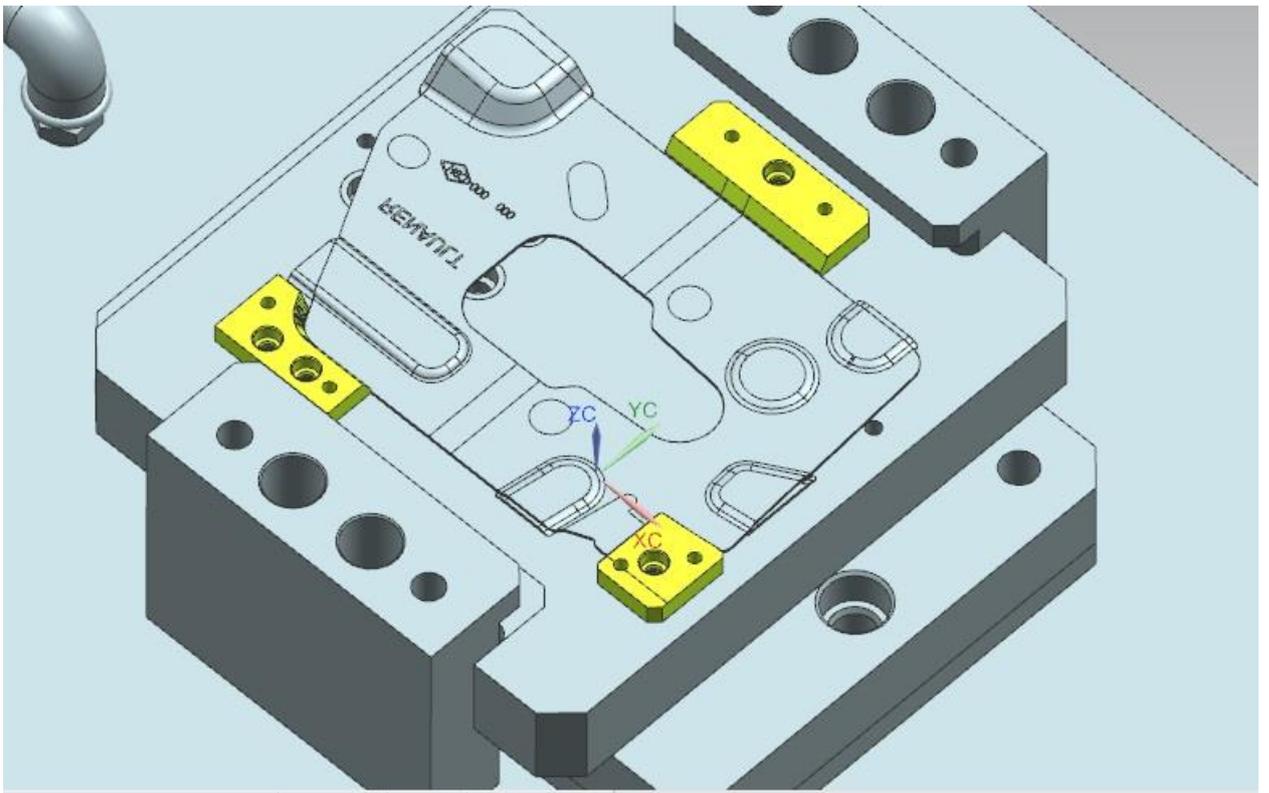


Рисунок 1.30 - Фиксация деталей по контуру

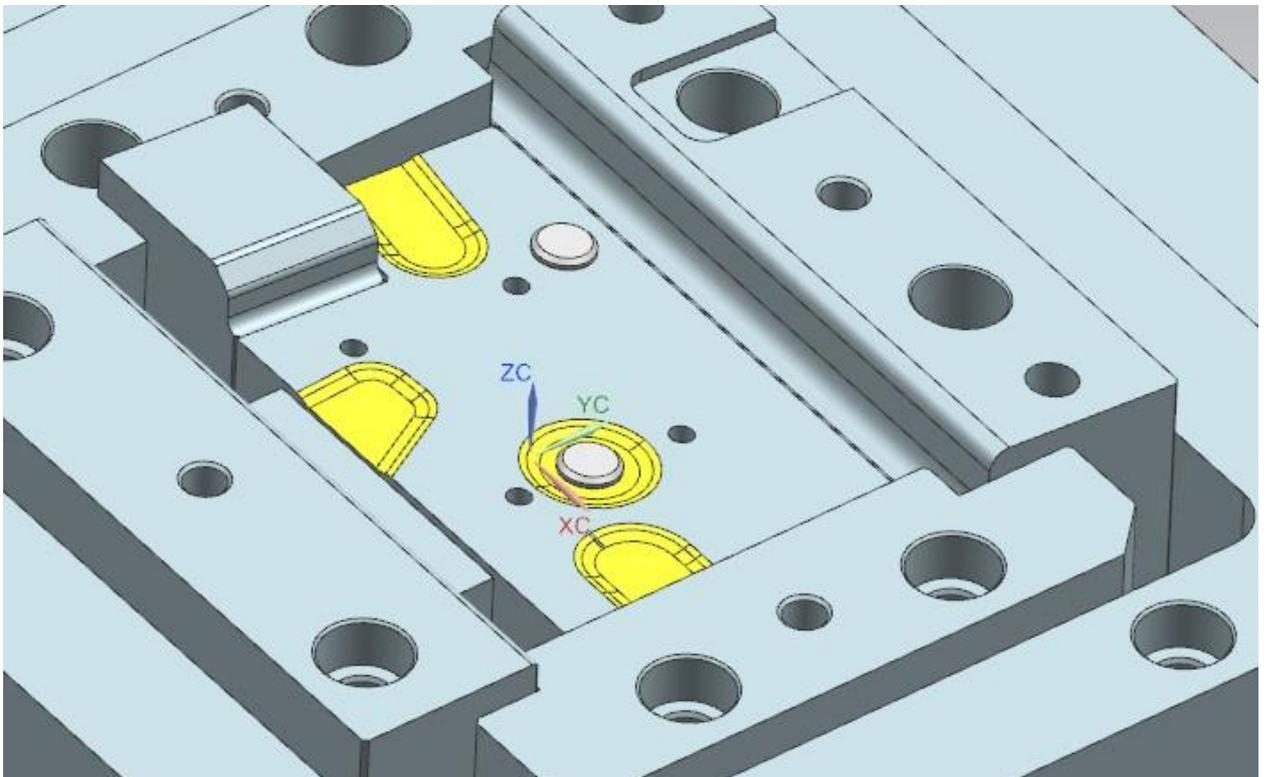


Рисунок 1.31 - Фиксация деталей по форме

9) При выполнении операции вытяжка в конструкцию штампа вводится прижим. Необходимое усилие прижима реализуется за счёт маркета прессы (рисунок 1.32). В таком случае прижим находится снизу от детали. При невозможности использования маркетной подушки прессы усилие прижима можно набрать при помощи пружин (витых, газовых, полиуретановых) (рисунок 1.33).

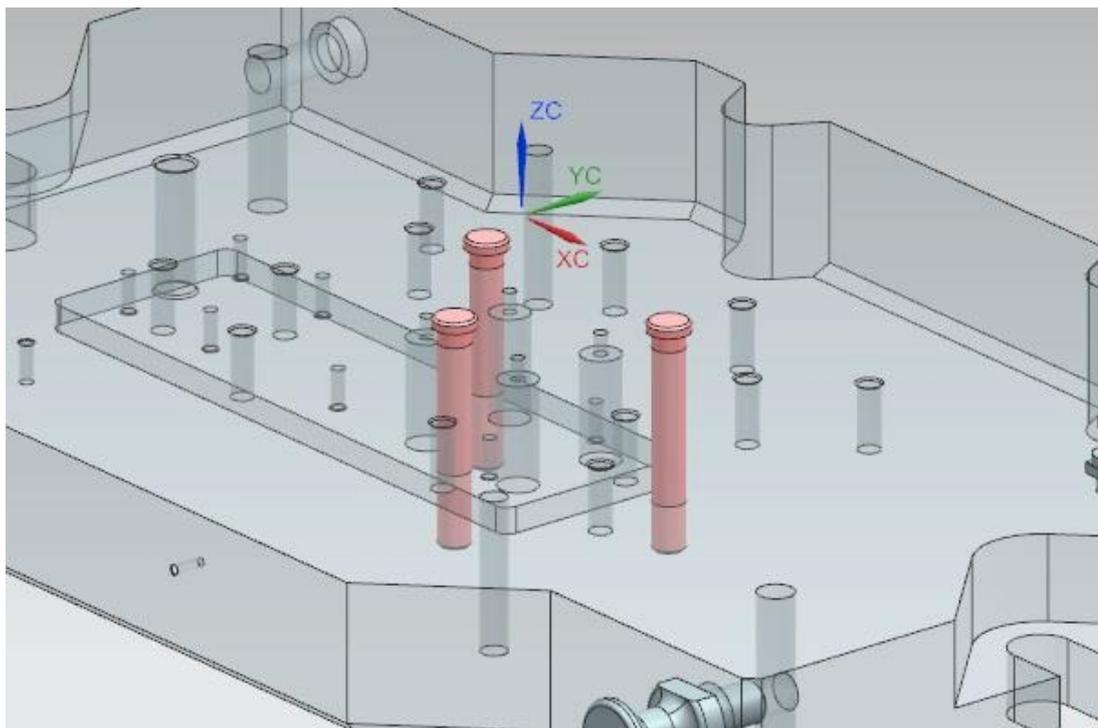


Рисунок 1.32 - Маркетные толкатели

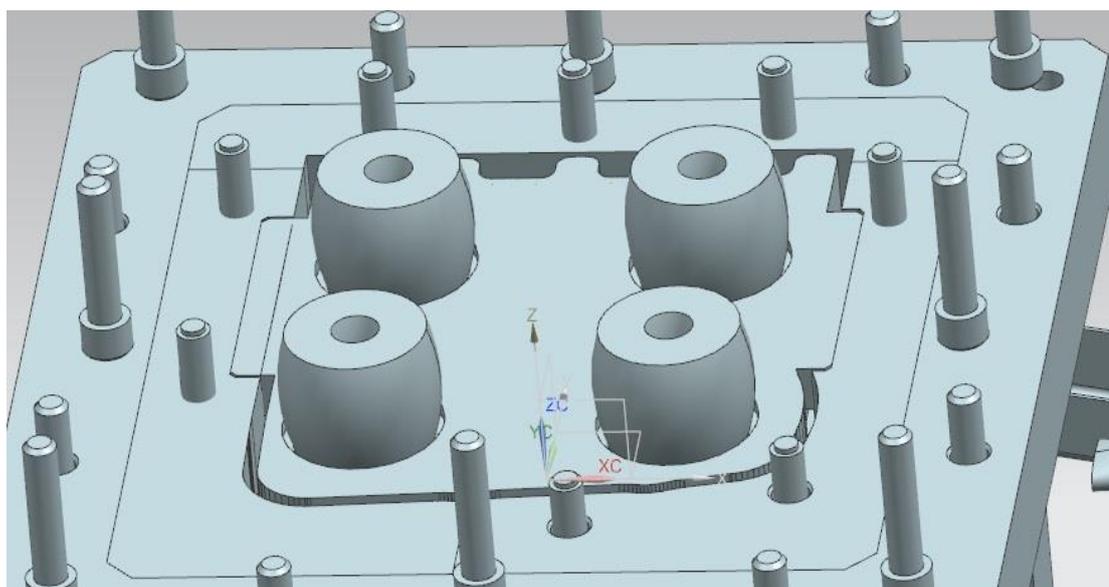


Рисунок 1.33 - Пружины полиуритановые

Направление прижима может задаваться за счёт колонок и втулок (рисунок 1.34), направляющих скоб (рисунок 1.35) или по секциям (рисунок 1.36).

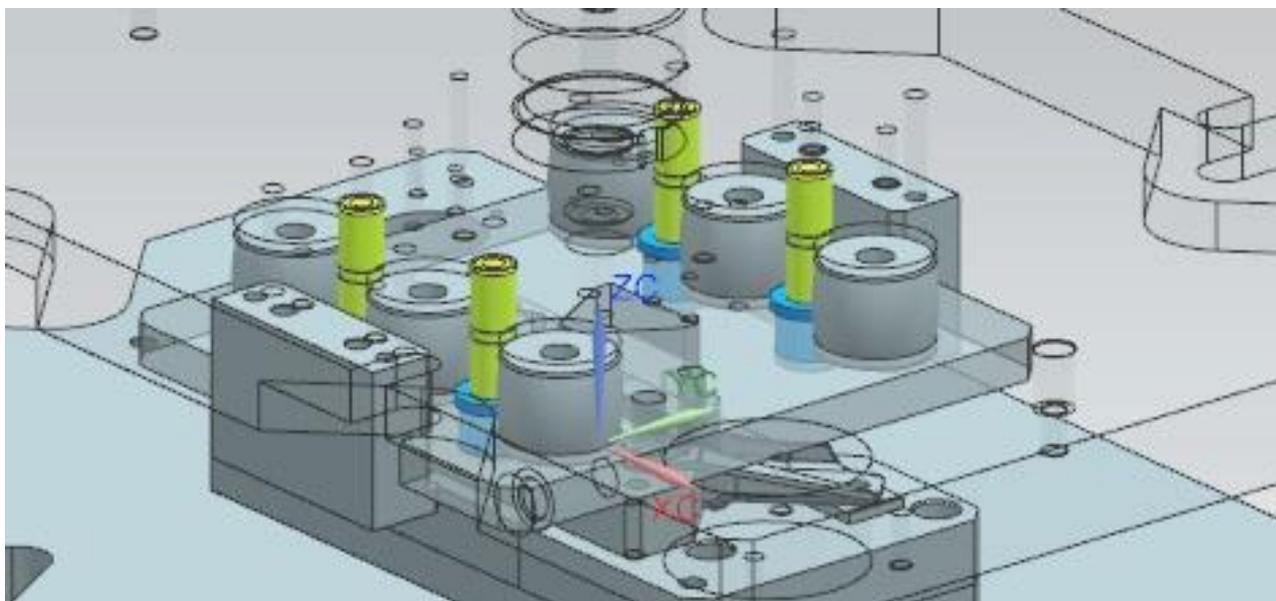


Рисунок 1.34 – Прижим, направляющийся по колонкам и втулкам

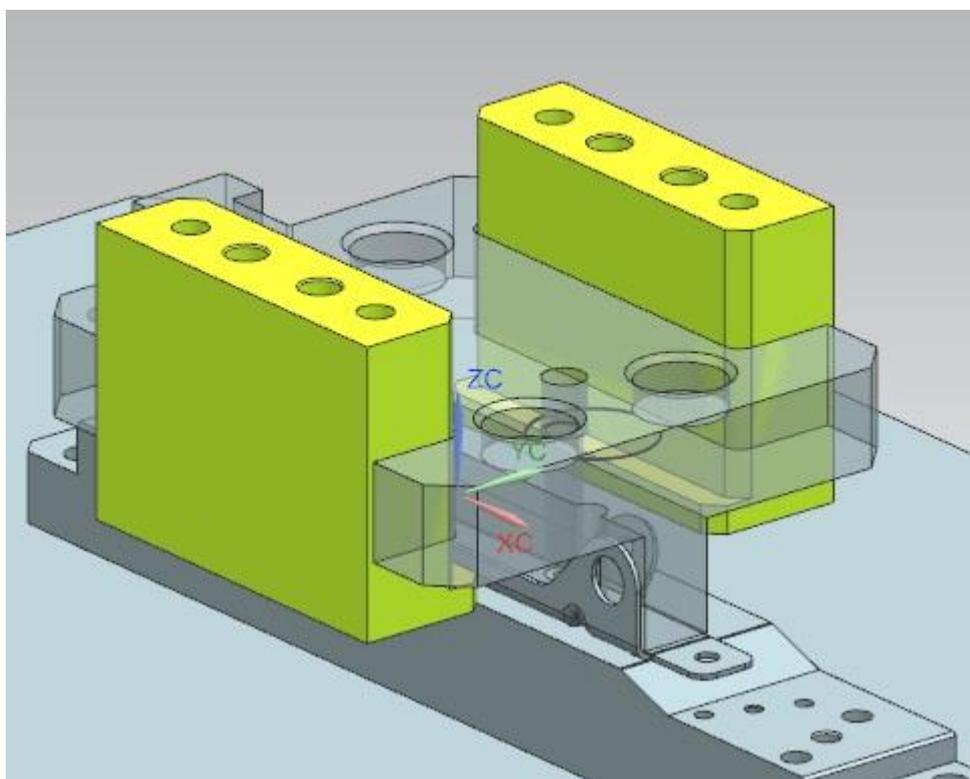


Рисунок 1.35 – Прижим, направляющийся по скобам

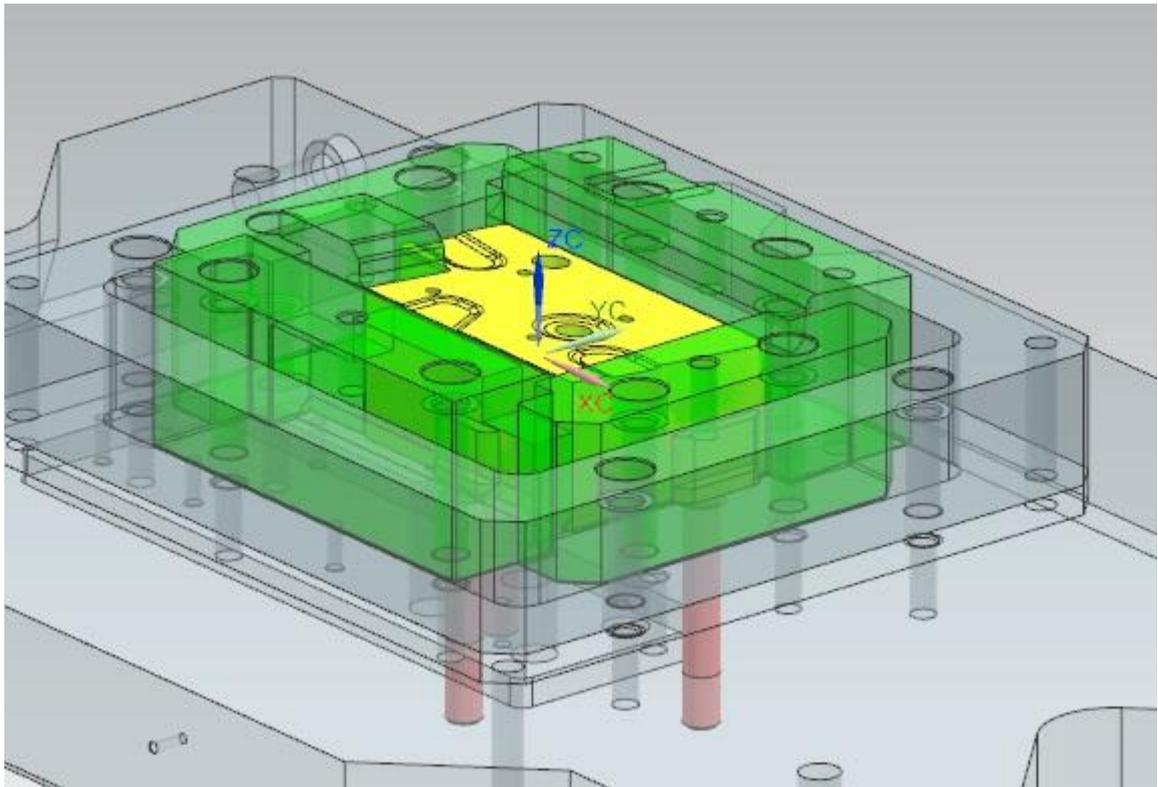


Рисунок 1.36 – Прижим, направляющийся по секциям

Анализ конструкций штампа позволяет выделить следующие основные признаки, отвечающие за отличительные особенности в конструкциях штампа:

- 1) По используемому оборудованию.
- 2) По виду операции.
- 3) По типу плиты штампа и креплению к прессу.
- 4) По расположению и конструкции направляющего узла.
- 5) По способам транспортировки штампов.
- 6) По видам ограничителей закрытой высоты.
- 7) По способам закладки заготовок в штамп.
- 8) По типу удаления деталей из штампа.

9) По способам фиксации деталей в штампе.

10) По варианту набора необходимого усилия и направления прижима.

Предложенные признаки определяют разнообразие в конструктивном исполнении штампов.

Таким образом, проведенный анализ конструкций мелких и средних формообразующих штампов позволил определить их конструктивные особенности, а также выделить основные признаки, отвечающие за отличительные особенности конструкций и по которым может быть осуществлена классификация рассматриваемой группы штамповой оснастки.

2. Формализация баз данных и знаний процесса проектирования мелких и средних формообразующих штампов

2.1. Создание таблицы вариантов технических решений

Для организации поиска технических решений конструкций мелких и средних формообразующих штампов представляется возможным использование принципов и методов морфологического анализа.

Целью метода морфологического анализа является систематическое использование всех воображаемых вариантов, выходящих из закономерностей строения (то есть морфологии) усовершенствованной системы.

В усовершенствованной технической системе выделяют несколько характерных для нее структурных или функциональных морфологических признаков. Каждый признак может характеризовать параметры или характеристики системы.

Алгоритм применения метода:

- 1) Точное формулирование задачи (проблемы), что подлежит решению.
- 2) Составление списка всех морфологических признаков, то есть всех возможных характеристик объекту, его параметров, от которых зависит решение проблемы и достижения основной цели.
- 3) Раскрытие возможных вариантов за каждым морфологическим признаком (характеристикой) путем составления матрицы. Очень важно, чтобы вплоть до данного этапа не возникал вопрос о практической осуществлённости и ценности того или другого варианта решения.
- 4) Определение функциональной ценности всех полученных вариантов решений. Должны быть рассмотрены все варианты решений, вытекающих

из структуры морфологической таблицы, и проведены сравнения поодиночке или несколькими наиболее важным для данной технической системы показателям.

5) Выбор наиболее рациональных решений.

Морфологический анализ основан на построении таблицы. По каждому выделенному морфологическому признаку составляют список его вариантов, альтернатив. Признаки с их альтернативами можно располагать в форме таблицы, названной морфологическим ящиком, который позволяет лучше представить себе поисковое поле. При создании морфологической таблицы перебираются всяческие объединения альтернативных вариантов решения задачи, которые при простом переборе могли быть упущены. В морфологической таблице перечисляются все основные элементы, составляющие объект и указывается, возможно, большее число известных вариантов реализации этих элементов. Комбинируя варианты реализации элементов объекта, можно получить самые неожиданные новые решения.

Морфологический подход связан со структурными взаимосвязями между объектами, явлениями и концепциями. Основной идеей морфологического анализа является упорядочение процесса выдвижения и рассмотрения различных вариантов решения задачи. Расчет строится на том, что в поле зрения могут попасть варианты, которые ранее не рассматривались. Принцип морфологического анализа легко реализуется с помощью компьютерных средств.

На основе проведенного исследования конструктивных особенностей мелких и средних формообразующих штампов создается таблица вариантов технических решений (таблица 2.1).. Каждый из вариантов технических решений представляет собой возможное сочетание способов выполнения частных функций.

Таблица 2.1 – Варианты технических решений

Усилие операции т.с МН	>0.63 МН				-	-		
	<0.63 МН		-	-				
Группы рабочего инструмента.	Инструмент для вытяжки		-		-			
	Инструмент для формовки			-				
	Инструмент для гибки					-		
	Инструмент для отбортовки							
Плиты штампа	Верхние плиты	Плита верхняя (вар.1)	-	-	-			
		Плита верхняя (вар.2)				-		
	Нижние плиты	Плита нижняя (вар.1)	-	-	-			
		Плита нижняя (вар.2)				-		
Узел направления	Колонки		-	-	-	-		
	Втулки		-	-	-	-		
	Варианты крепления узла	Колонки запрессованы в держателе		-	-			
		Втулки запрессованы в держателе				-	-	
		Колонки запрессованы в плиту						
		Втулки запрессованы в плиту		-	-		-	
	Расположение узла	Колонки в верхней плите		-	-		-	
		Колонки в нижней плите				-		
		Втулки в верхней плите				-		
		Втулки в нижней плите		-	-		-	
	Крепление штампа на прессе	Нижняя плита крепится винтами		-	-	-	-	
		Верхняя плита крепится хвостовиком		-	-		-	
Верхняя плита крепится винтами				-				
Транспортные элементы	Рым болты		-	-				
	Накладки транспортировочные				-			
	Транспортные отверстия					-		

Продолжение таблицы 2.1

Ограничители закрытой высоты	Цилиндрические ограничители		-	-			
	Ограничительное кольцо совместно с направляющим узлом				-	-	
Закладка заготовки.	Вручную		-	-		-	
	Автоматически				-		
Удаление деталей	Вручную					-	
	Пневмоздувом и подбрасывателями		-	-	-		
Фиксация	По контуру		-	-			
	На отверстия				-		
	По форме					-	
Прижимы	Расположение	Прижим сверху	-	-		-	
		Прижим снизу			-		
	Способ направления	По колонкам и втулкам	-	-		-	
		По скобам			-		
		По секциям				-	
	Усилие	За счет маркета			-		
		За счёт пружин	-	-		-	
Общая функция	X		X1	X2	X3	X4	Xn

В таблице представлены все варианты конструкций включая и невозможные. Знаком «-» обозначено наличие того или иного элемента в конструкции штампа.

Каждый из вариантов технических решений представляет собой возможное сочетание способов выполнения частных функций. Например, решение таблицы X1 позволяет получить конструкцию: пресс усилием больше 0.63 МН. Штамп для вытяжки. Верхняя и нижняя плита первого варианта исполнения. Используются колонки и втулки, втулки запрессованы в держателе и крепятся к нижней плите, колонки запрессованы в верхнюю плиту. Нижняя

плита крепится винтами а верхняя хвостовиком. Транспортировка штампа осуществляется за счет рым-болтов. Укладка заготовки в ручную а удаление пневмосдувом и подбрасывателями. Закрытая высота ограничивается цилиндрическими ограничителями. Фиксация детали осуществляется по контуру. Усилие прижима набирается за счёт пружин, направляется по колонкам и втулкам и располагается сверху.

2.2. Создание И-ИЛИ морфологического дерева

Множество связей между элементами конструкции и их признаками может быть представлено в виде графа в качестве И-ИЛИ морфологического дерева (рисунок 2.3). Морфологическое И-ИЛИ-дерево является одним из способов представления модели морфологического множества уровня идентификации.

Морфологическое дерево имеет два типа вершин: ИЛИ-вершины, представляющие собой классификационные признаки и означающие возможность выбора одного из ребер, которые соответствуют значениям классификационных признаков, и И-вершины, обычно представляющие агрегацию подсистем. ИЛИ-вершины обычно изображают черными, а И- изображают не закрашенными (белым цветом).

Морфологическое И-ИЛИ-дерево включает в себя все структурные идентификаторы объектов, рассматриваемого класса, которые получаются путем вырождения ИЛИ-вершин. Структурными элементами являются узлы, механизмы и детали штампа, а так же их признаки.

С помощью морфологических И/ИЛИ-деревьев для морфологических множеств относительно небольшой мощности (содержащих сравнительно малое число структурных решений) можно учитывать запрещенные сочетания значений классификационных признаков. Но для морфологических множеств большой мощности это становится проблематично. Для этих целей

лучше использовать специальные языки моделирования морфологического множества.

Морфологические деревья могут быть иерархическими, когда морфологическое дерево, представляющее структурные решения одного класса объектов, включает в себя в качестве агрегатов другие морфологические деревья, представляющие структурные решения объектов, принадлежащих другим классам. Кроме того, возможны рекурсивные морфологические деревья, у которых к одной из вершин дерева, подключается такое же морфологическое И/ИЛИ-дерево.

Дерево строится на основании ранее созданной таблицы вариантов технических решений. И/ИЛИ - дерево позволяет выявить ограничения и связи между элементами конструкции, что позволит отсеять невозможные варианты конструкций или противоречащие друг другу элементы. Для выявления связей дерево разбито на двадцать уровней в зависимости от признаков конструкций. Каждому признаку был присвоен номер который указывает номер уровня и номер варианта технического решения, например на рисунке 2.1 показан элемент четырнадцатого уровня и первого варианта конструкции. Ограничения заданы И/ИЛИ вершинами например на рисунке 2.3 показан элемент «181 Усилие прижима за счёт маркета» из которого выходит вершина И которая связывает его с элементом «192 Прижим снизу» из которого выходит вершина ИЛИ связывающая его с элементами «201 Направление прижима по колонкам и втулкам», «202 Направление прижима по скобам», «203 Направление прижима по секциям».

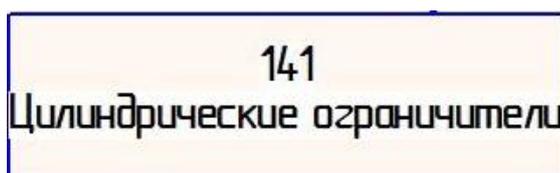
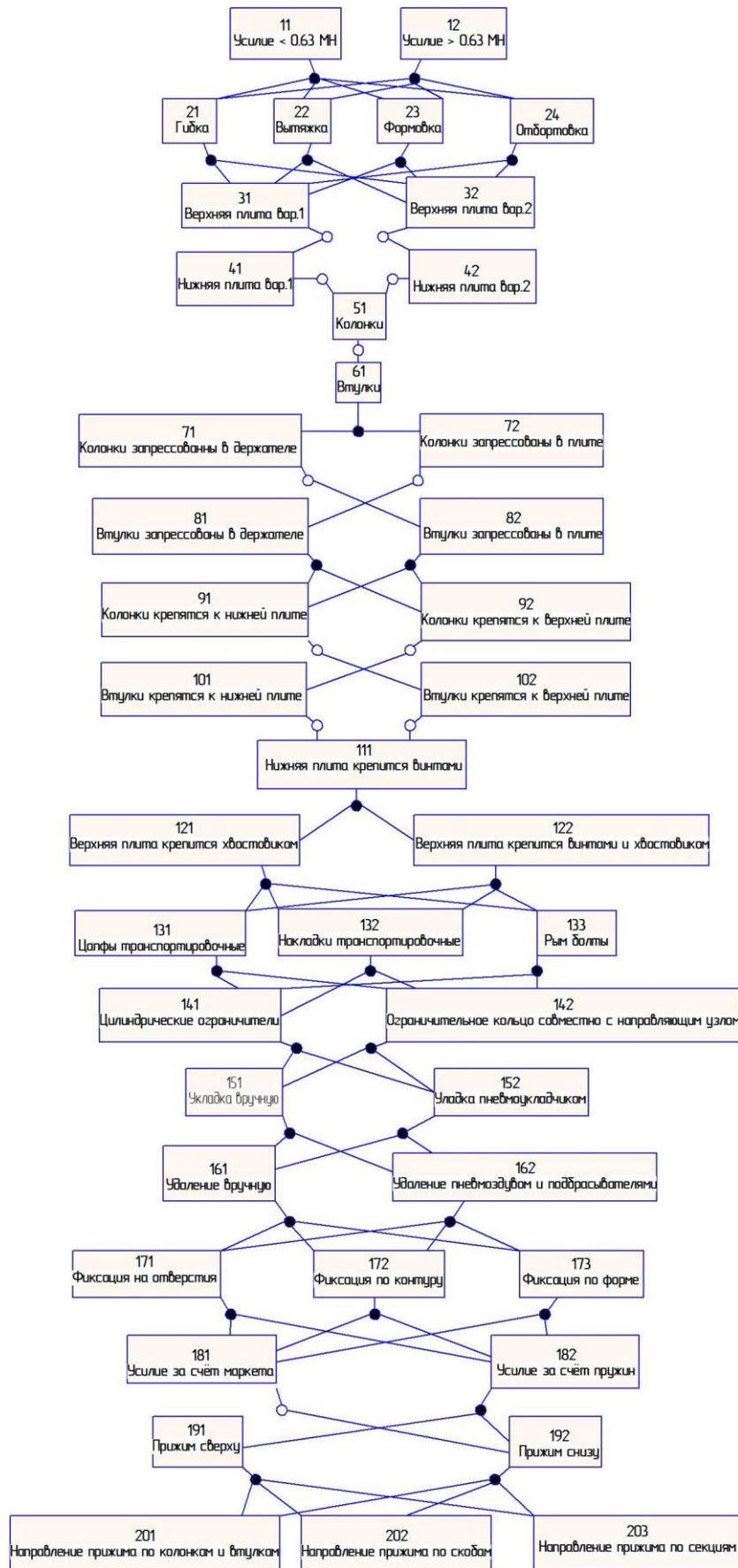


Рисунок 2.1 – Элемент конструкции штампа



И – ○ ИЛИ – ●

Рисунок 2.2 - И – ИЛИ дерево

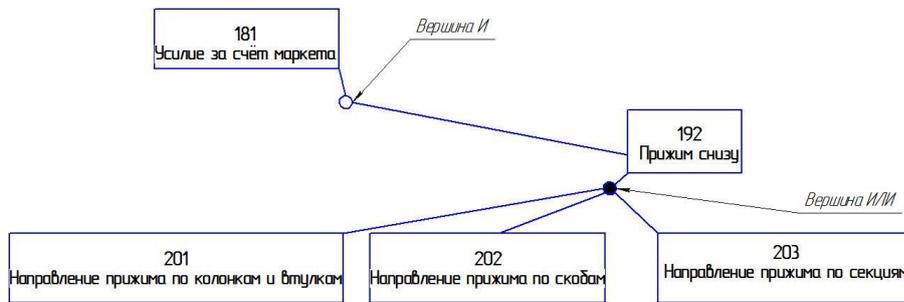


Рисунок 2.3 – I/II вершины

Формализация процедур выбора необходимой конструкции штампа позволяет перенести их в область автоматизированного проектирования. Элементы конструкции, их признаки и связи, выявленные по результатам проведённого морфологического анализа, могут быть перенесены в адаптированном виде в программный код.

3. Создание системы автоматизированной поддержки процесса проектирования штамповой оснастки

3.1 Разработка интерфейса программы

Программный код был разработан на языке программирования C# в программе SharpDeveloper. Разработанная программа позволяет получить текстовое описание конструкции штампа на основании введенных пользователем данных в окнах диалога.

Интерфейс программы состоит из семи последовательно открывающихся окон в которых пользователь вводит данные и выбирает необходимые конструктивные особенности. Дизайн выполнен с помощью инструментов Windows Forms.

Первое окно будет называться «Выбор операции и расчёт усилий» (рисунок 3.1)

Определение операций и расчёт усилий.

Выбор операции

Гибка Формовка

Вытяжка Отбортовка

Характеристики материала

Толщина мм. Предел прочности кгс/мм²

1 0,25

Параметры операции

Для гибки, отбортовки: Для вытяжки, формовки:

Длина линии периметра мм. Длина линиигиба мм.

0 0

Применить Сброс

Рисунок 3.1 – Дизайн первого диалогового окна

Выбор операции реализуется при нажатии кнопок RadioButton с названием операции (рисунок 3.2).

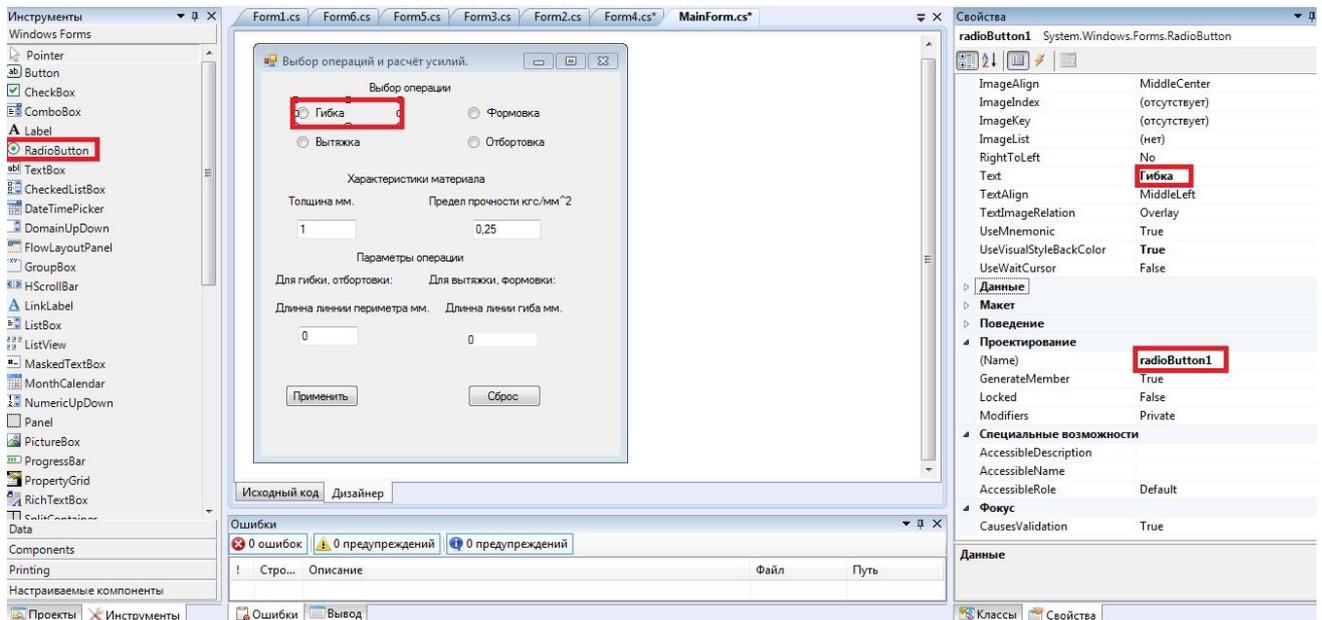


Рисунок 3.2 – Кнопки выбора операций

Ввод параметров осуществляется в окнах TextBox. Начальное значение задаётся в свойствах, например значение толщины материала по умолчанию равно 1 мм, предел текучести равен 0.25 кгс/мм² (рисунок 3.3).

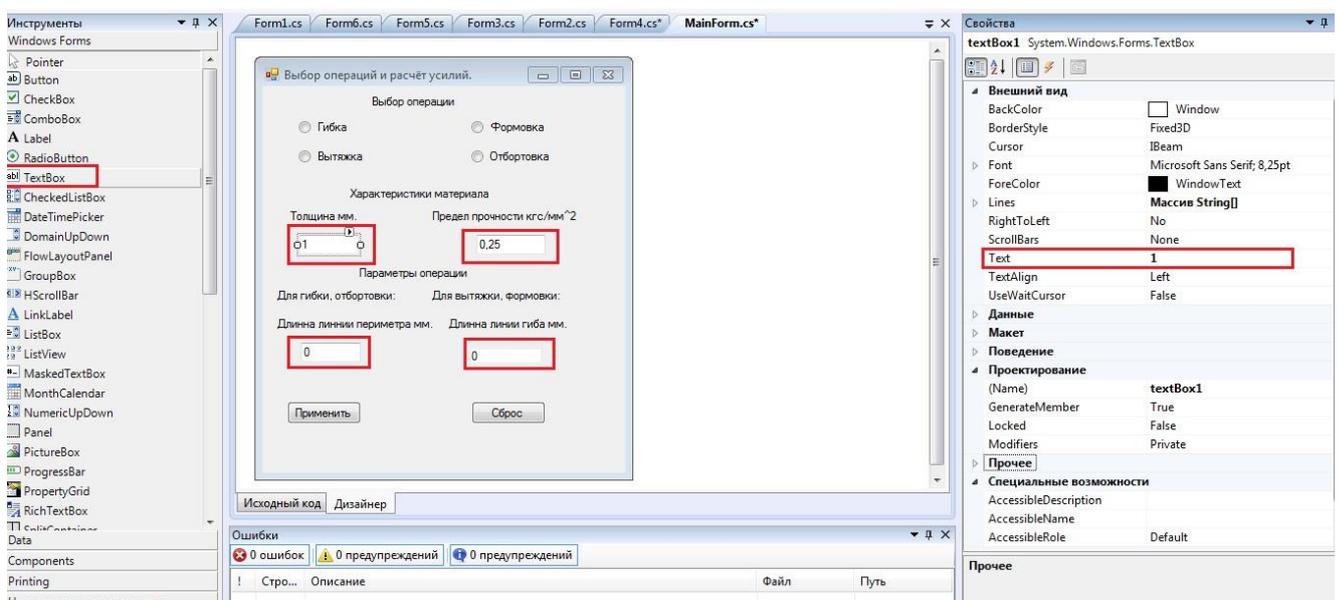


Рисунок 3.3 – Окна ввода параметров операций

Пояснительные надписи в окне выполнены с помощью функции Label (рисунок 3.4).

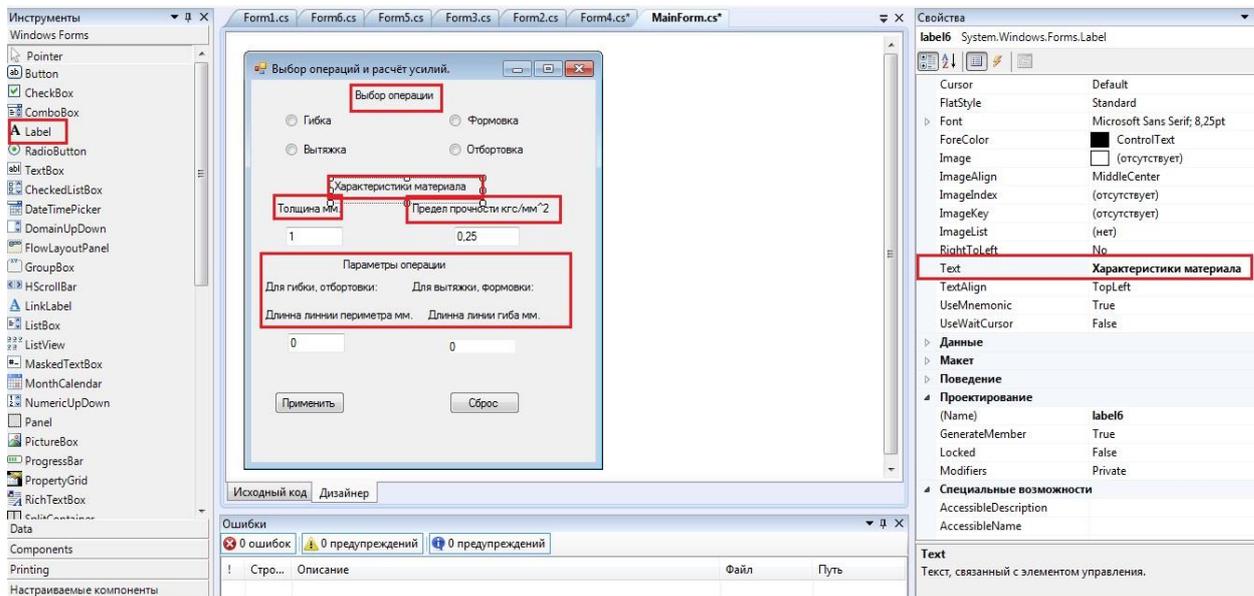


Рисунок 3.4 – Пояснительные надписи в окнах

В окне имеются две кнопки «Применить» которая сохраняет все данные введённые пользователем и «Сброс» которая очищает все окна и сбрасывает выбор операции. Кнопки выполнены при помощи инструмента Button (рисунок 3.5).

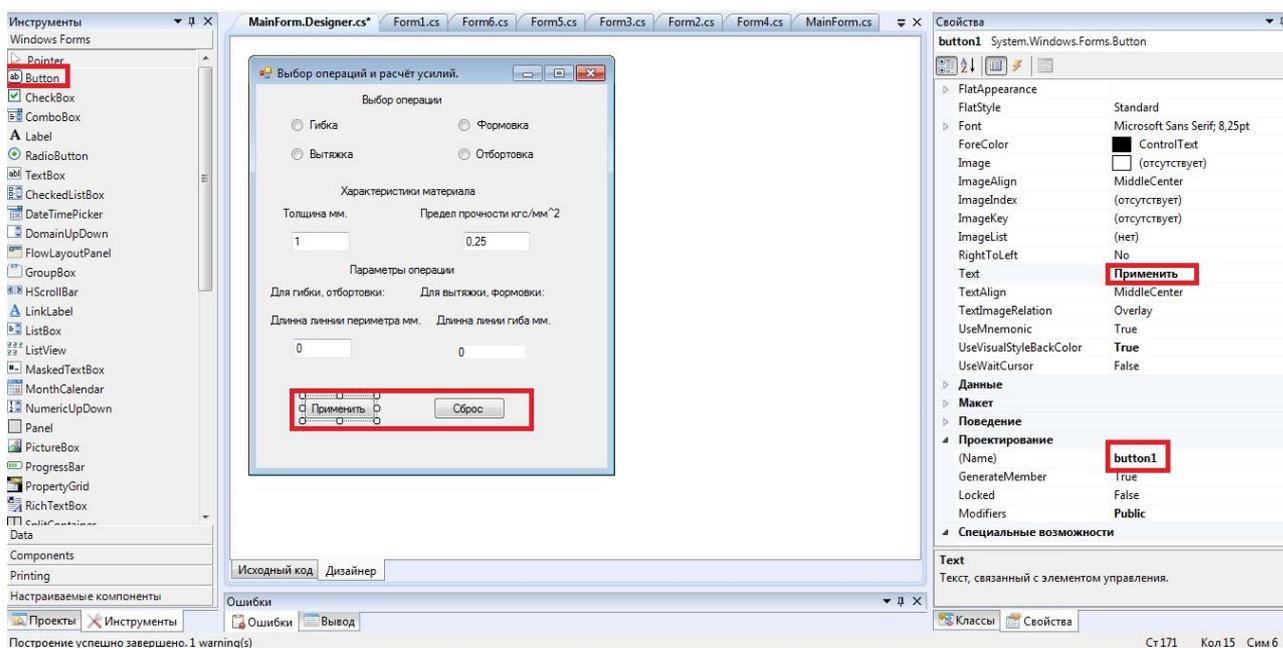


Рисунок 3.5 – Кнопки «Применить» и «Сброс»

Второе окно называется «Укладка деталей» (рисунок 3.6).

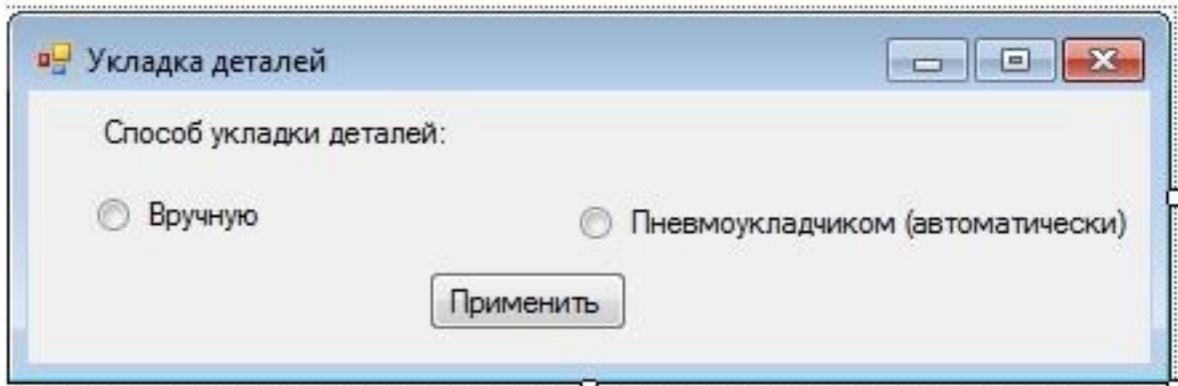


Рисунок 3.6 – Дизайн второго диалогового окна

В этом окне присутствуют две кнопки созданные с помощью инструмента RadioButton и одна кнопка инструментом Button, пояснительные надписи реализованы инструментом Label (рисунок 3.7).

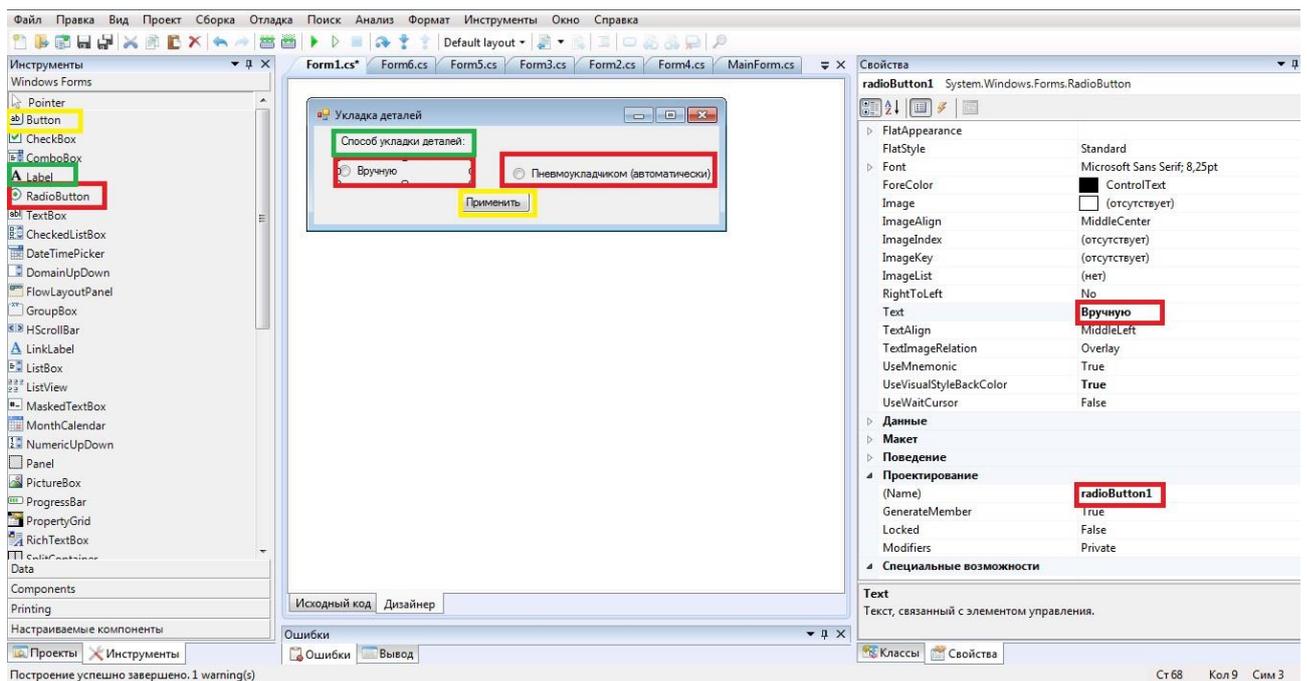


Рисунок 3.7 – Инструменты Button, Lable и RadioButton

Дизайн третьего, четвёртого, пятого и шестого окна схож по структуре с дизайном второго окна и созданы при помощи инструментов RadioButton, Button и Lable (рисунок 3.8 – рисунок 3.11).

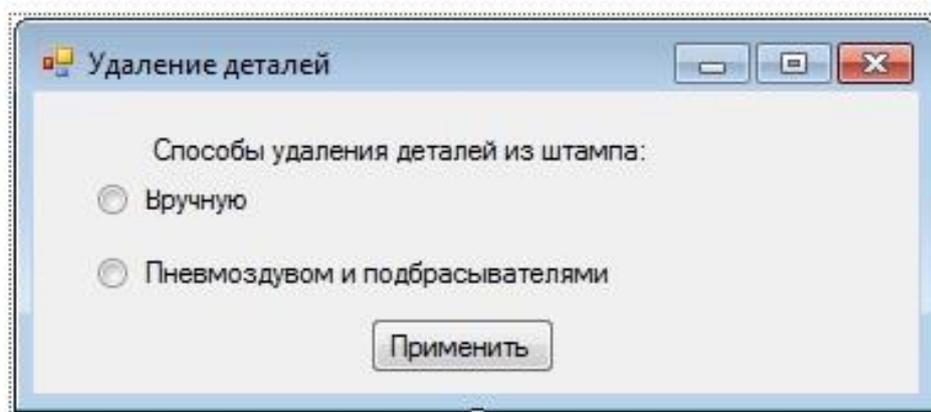


Рисунок 3.8 – Дизайн третьего окна «Удаление деталей»

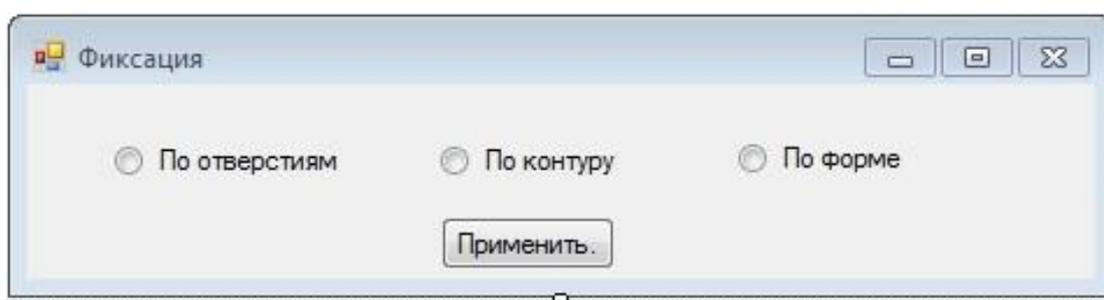


Рисунок 3.9 – Дизайн четвёртого окна «Фиксация»

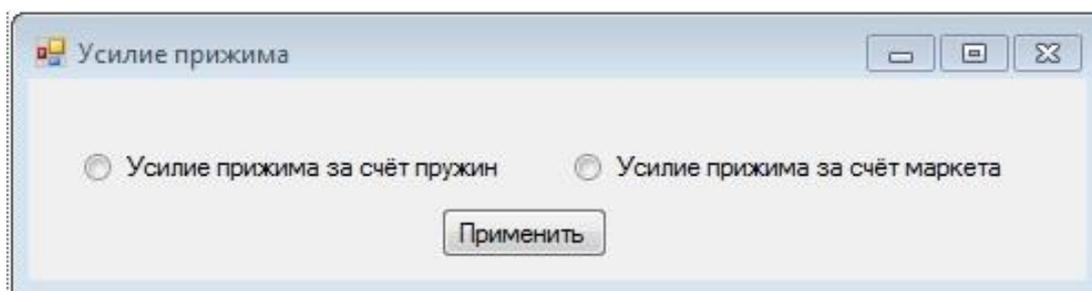


Рисунок 3.10 – Дизайн пятого окна «Усилие прижима»

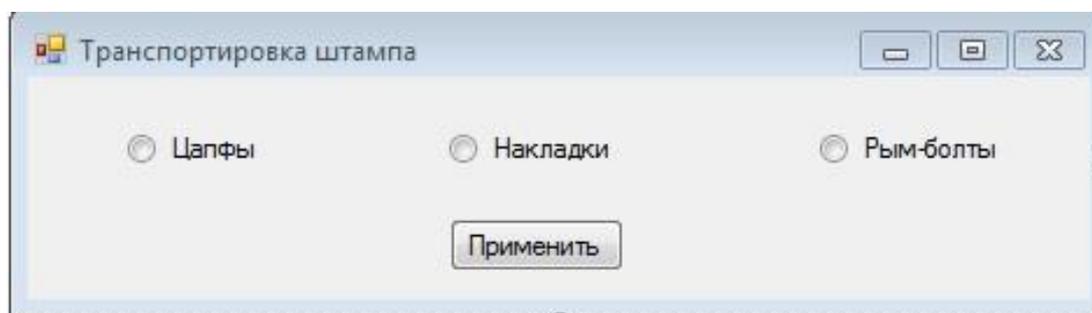


Рисунок 3.11 – Дизайн шестого окна «Транспортировка штампа»

Седьмое окно называется «Варианты конструкций штампов» оно является конечным и в нём выводится результат подбора конструкций (рисунок 3.12).

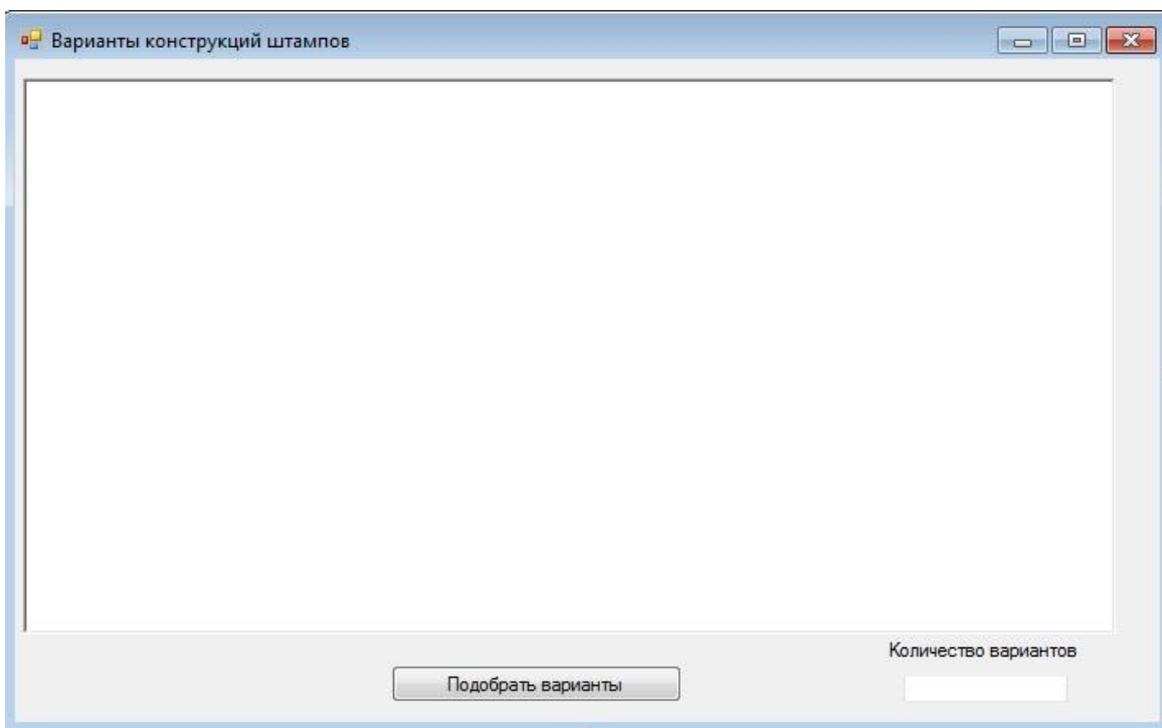


Рисунок 3.12 – Дизайн седьмого окна «Варианты конструкций штампов»

Текстовое описание конструкции штампа выводится в окно созданное с помощью инструмента RichTextBox (рисунок 3.13).

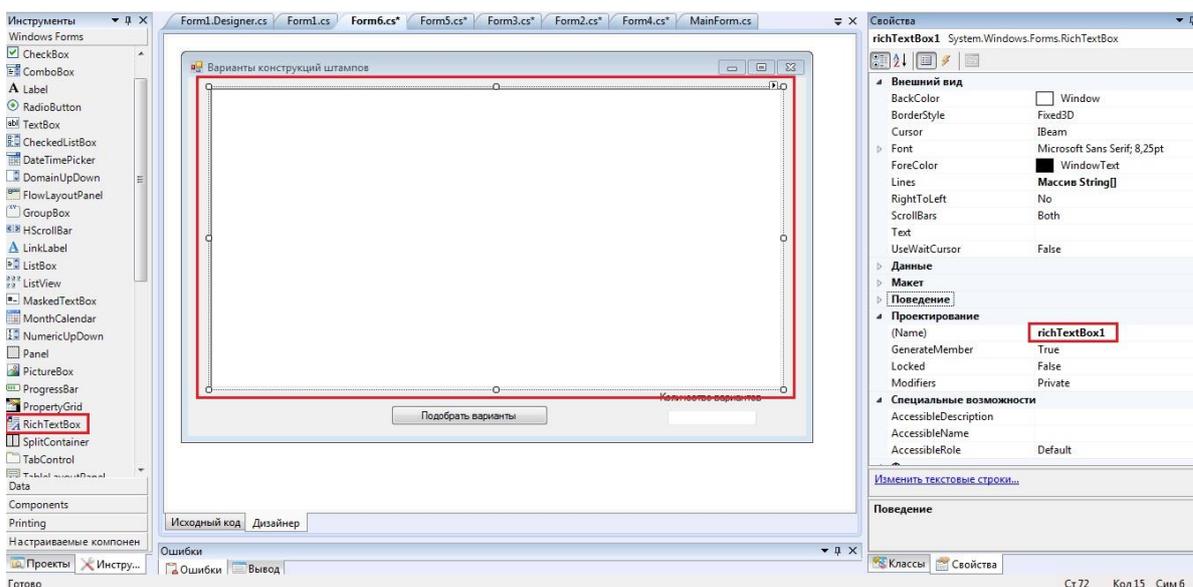


Рисунок 3.12 – Инструмент RichTextBox

В окне есть кнопка созданная с помощью функции Button и окно TextBox (рисунок 3.14).

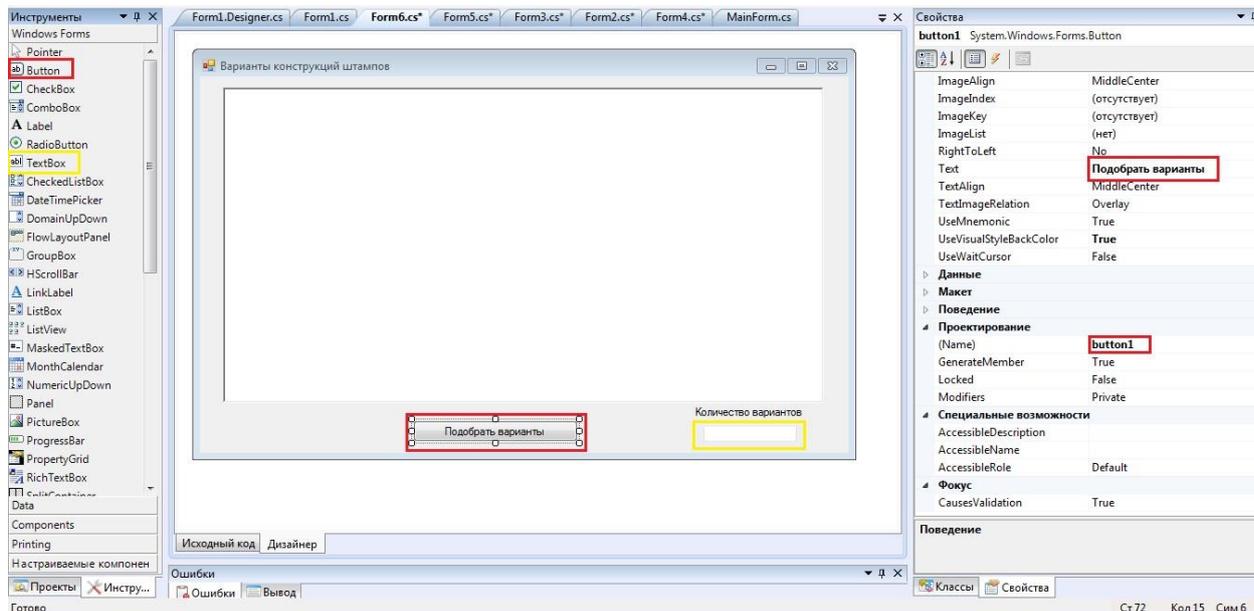


Рисунок 3.12 – Инструмент Button и TextBox

3.2 Разработка кода программы.

Разработка кода программы начинается исходя из дизайна первого окна (рисунок 3.1). Множество признаков конструкций и их связи которые были выявлены в результате построения И/ИЛИ дерева конструкции штампов (рисунок 2.1), было представлено в коде программы в виде инициализации массивов, содержащих целочисленные данные соответствующие деталям и признакам конструкции.

Публично объявляем массивы для их последующего переноса:

```
double a1,a2,a3,a4,a5;
```

```
public int [] B;
```

```
public int [] G;
```

```
public int [] M;  
public int [] F;  
public int [] W;
```

При нажатии кнопки «Подобрать» срабатывает алгоритм в зависимости от введенных пользователем данных в TextBox и нажатых кнопках RadioButton, например если пользователь при вводе данных укажет тип операции вытяжка, то код будет выглядеть следующим образом:

```
public void Button1Click(object sender, EventArgs e)  
{ if (radioButton1.Checked)
```

Инициализируются данные введенные в TextBox:

```
a1 = Convert.ToDouble (textBox1.Text); //толщина материала  
a2 = Convert.ToDouble (textBox2.Text); // предел прочности  
a3 = Convert.ToDouble (textBox3.Text); // длина линиигиба  
a4 = Convert.ToDouble (textBox4.Text); //длина линии периметра
```

Рассчитывается переменная a5 :

```
a5 = a3 * a2 * a1*1.5;
```

Если значение переменной меньше значения 63 то массивам присваиваются значения:

```
{ if ( a5<=63 )  
{ B = new int[] {21};  
G = new int[] {11};  
M = new int[] {31};  
F = new int[] {41};  
W = new int[] {121};}
```

Если значение переменной больше значения 63, массивам присваиваются другие значения:

```
else
{ B = new int[] {21};
G = new int[] {12};
M = new int[] {32};
F = new int[] {42};
W = new int[] {122};}}
```

Далее открывается следующее окно (форма) и в него переносятся элементы массивов, это реализовано так:

```
Form1 obj = new Form1();
obj.M1 = G;
obj.M2 = B ;
obj.M3 = M;
obj.M4 = F;
obj.M12 = W;
obj.ShowDialog();
```

Если пользователю необходимо очистить ранее введённые данные в окне присутствует кнопка «Сброс», при нажатии на которую значения в TextBox и RadioButton стираются это выполняется за счёт реализации следующего кода:

```
void Button3Click(object sender, EventArgs e)
{ radioButton1.Checked = false;
radioButton2.Checked = false;
radioButton3.Checked = false;
radioButton4.Checked = false;
textBox1.Clear();
textBox2.Clear();
textBox3.Clear();
```

```
textBox4.Clear(); }
```

Когда открывается второе окно в нём публично объявляются массивы перенёсённые из предыдущего окна:

```
public int [] M1 ;
```

```
public int [] M2 ;
```

```
public int [] M3 ;
```

```
public int [] M4 ;
```

```
public int [] M12 ;
```

Так же объявляется и массив который добавится в этом окне:

```
public int [] M15;
```

Код для окна «Укладка детали» (рисунок 3.6) срабатывает при нажатии кнопки «Применить»:

```
void Button1Click(object sender, EventArgs e)
```

Если пользователь выберет «Вручную» то сработает следующая часть кода:

```
{ if (radioButton1.Checked)
```

```
{ M15 = new int[] { 151 };}
```

Если пользователь выберет «Вручную» то код будет выглядеть так:

```
if (radioButton2.Checked)
```

```
{ M15 = new int[] { 152 };}
```

После выбора программа автоматически перенёсет элементы массив и откроет следующее окно в коде это реализовано так:

```
Form2 obj = new Form2();  
obj.G1=M1;  
obj.G2=M2;  
obj.G3=M3;  
obj.G4=M4;  
obj.G5=M12;  
obj.G6=M15;  
obj.ShowDialog();
```

Открывается третье окно «Удаление деталей» (рисунок 3.6), в нём публично объявляются массивы перенесённые из второго окна и массив значение которого будет присваиваться в этом окне:

```
public int [] G2;  
public int [] G3;  
public int [] G4;  
public int [] G5;  
public int [] G6;  
public int [] M16;
```

Если пользователь выбирает «Вручную» то срабатывает следующий код:

```
void Button1Click(object sender, EventArgs e)  
{ if (radioButton1.Checked  
{M16 = new int[] {161};}
```

При нажатии на «Пневмоздувом и подбрасывателями» срабатывает код:

```
void Button1Click(object sender, EventArgs e)  
{ if (radioButton2.Checked)  
{ M16 = new int[] {162};}
```

Перенос массивов и открытие следующего окна производится:

```
Form3 obj = new Form3();  
obj.F1=G1;  
obj.F2=G2;  
obj.F3=G3;  
obj.F4=G4;  
obj.F5=G5;  
obj.F6=G6;  
obj.F7=M16;  
obj.ShowDialog();
```

После этого открывается четвёртое окно «Фиксация» После нажатия кнопки «Применить» в нем объявляются массивы перенесённые из предыдущего окна.

```
public int [] F1 ;  
  
public int [] F2;  
public int [] F3;  
  
public int [] F4;  
public int [] F5;  
public int [] F6;  
public int [] F7;  
public int [] M17;
```

При выборе пользователем «По отверстиям» срабатывает следующая часть кода:

```
void Button1Click(object sender, EventArgs e)  
{ { if (radioButton1.Checked)  
  {M17 = new int[] {171};}}
```

При выборе пользователем «По отверстиям» срабатывает следующий код:

```
void Button1Click(object sender, EventArgs e)
{ if (radioButton2.Checked)
{ M17 = new int[] { 172};}}
```

Если выбран вариант «По форме» код выглядит следующим образом :

```
void Button1Click(object sender, EventArgs e)
{ if (radioButton3.Checked)
{ M17 = new int[] { 173};}}
```

После производится перенос данных и открытие следующего окна:

```
Form4 obj = new Form4();
obj.H1=F1;
obj.H2=F2;
obj.H3=F3;
obj.H4=F4;
obj.H5=F5;
obj.H6=F6;
obj.H7=F7;
obj.H8=M17;
obj.ShowDialog();
```

Открывается пятое окно «Усилие прижима» в его коде объявляются массивы перенесёнными из предыдущего окна и :

```
public int [] H1;
public int [] H2;
public int [] H3;
public int [] H4;
public int [] H5;
```

```
public int [] H6;  
public int [] H7;  
public int [] H8;  
public int [] M18;  
public int [] M19;
```

Если пользователь вберет «Усилие прижима за счёт пружин» срабатывает код:

```
Void Button1Click(object sender, EventArgs e)  
{if (radioButton1.Checked)  
{ M18 = new int[] {192};  
M19 = new int[] {201,202};}}
```

Если выбрано «Усилие за счет пружин» код реализуется следующим образом:

```
Void Button1Click(object sender, EventArgs e)  
{if (radioButton2.Checked)//усилие за счет маркета  
{ M18 = new int[] {191};  
M19 = new int[] {202};}}
```

Вывод массивов и открытие следующего окна:

```
Form5 obj = new Form5();  
obj.V1=H1;  
obj.V2=H2;  
obj.V3=H3;  
obj.V4=H4;  
obj.V5=H5;  
obj.V6=H6;  
obj.V7=H7;  
obj.V8=H8;
```

```
obj.V9=M18;  
obj.V10=M19;  
obj.Show();
```

Открывается шестое окно «Транспортировка штампа», в нём публично объявляются массивы перенесённые из пятого окна и массив значение которого будет присваиваться в этом окне:

```
public int [] V1;  
public int [] V2;  
public int [] V3;  
public int [] V4;  
public int [] V5;  
public int [] V6;  
public int [] V7;  
public int [] V8;  
public int [] V9;  
public int [] V10;  
public int [] M13;
```

При нажатии на кнопку «Цапфы» код выглядит следующим образом:

```
{ if (void Button1Click(object sender, EventArgs e)  
radioButton1.Checked)//цапфы  
{M13 = new int[] {131};}
```

Если пользователь вберет «Накладки» срабатывает код:

```
void Button1Click(object sender, EventArgs e)  
{ if (radioButton1.Checked)//цапфы  
{M13 = new int[] {132};}
```

При выборе пользователем «Рым - болты» срабатывает следующая часть кода:

```
void Button1Click(object sender, EventArgs e)
{ if (radioButton1.Checked)//цапфы
  {M13 = new int[] {133};}
```

Далее открывается следующее окно (форма) и в него переносятся элементы массивов, это реализовано так:

```
obj.L1=V1;
obj.L2=V2;
obj.L3=V3;
obj.L4=V4;
obj.L5=V5;
obj.L6=V6;
obj.L7=V7;
obj.L8=V8;
obj.L9=V9;
obj.L10=V10;
obj.L11=M13;
obj.ShowDialog();
```

После открывается седьмое окно «Варианты конструкций штампов» в нём при нажатии кнопки «Подобрать» выводятся все варианты конструкций с учётом ограничений и параметров заданных пользователем.

В форму переходят все элементы массивов которые были заданы ранее:

```
public int [] L1
public int [] L2;
public int [] L3;
public int [] L4;
public int [] L5;
public int [] L6;
```

```
public int [] L7;  
public int [] L8;  
public int [] L9;  
public int [] L10;  
public int [] L11;
```

Множество признаков конструкций штампа не вошедших в ранее в код было представлено в коде программы в виде инициализации массивов:

```
int[] M5 = {51};  
int[] M6 = {61};  
int[] M7 = {71,72};  
int[] M8 = {81,82};  
int[] M9 = {91,92};  
int[] M10 = {102,101};  
int[] M11 = {111};  
int[] M14 = {141,142};  
int[] M21 = {211,212,213,214};
```

Связи между структурными единицами И-ИЛИ дерева были заложены в качестве циклов:

```
ArrayList ITOG = new ArrayList();  
for (i1=0;i1<=L1.Length-1;i1++){  
for (i2=0;i2<=L2.Length-1;i2++){  
for (i3=0;i3<=L3.Length-1;i3++){  
for (i4=0;i4<=L4.Length-1;i4++){  
for (i5=0;i5<=M5.Length-1;i5++){  
for (i6=0;i6<=M6.Length-1;i6++){  
for (i7=0;i7<=M7.Length-1;i7++){  
for (i8=0;i8<=M8.Length-1;i8++){  
for (i9=0;i9<=M9.Length-1;i9++){
```

```

for (i10=0;i10<=M10.Length-1;i10++){
for (i11=0;i11<=M11.Length-1;i11++){
for (i12=0;i12<=L5.Length-1;i12++){
for (i13=0;i13<=L11.Length-1;i13++){
for (i14=0;i14<=M14.Length-1;i14++){
for (i15=0;i15<=L6.Length-1;i15++){
for (i16=0;i16<=L7.Length-1;i16++){
for (i17=0;i17<=L8.Length-1;i17++){
for (i19=0;i19<=L9.Length-1;i19++){
for (i20=0;i20<=L10.Length-1;i20++){
for (i21=0;i21<=M21.Length-1;i21++){

int[] PROMMAS={L1[i1],L2[i2],L3[i3],L4[i4],M5[i5],M6[i6],M7[i7],M8[i8],M9[
i9],M10[i10],M11[i11],L5[i12],L11[i13],M14[i14],L6[i15],L7[i16],L8[i17],L9[i19
],L10[i20],M21[i21]};

```

Дополнительные ограничения отсекающие невозможные варианты конструкций задаются в коде в виде условий. В качестве примера представлены ограничения:

```

if (((PROMMAS[2]==31)&&(PROMMAS[3]==42))||((PROMMAS[2]==32)&&(
PROMMAS[3]==41))||((PROMMAS[8]==91)&&(PROMMAS[9]==101))
||((PROMMAS[8]==92)&&(PROMMAS[9]==102)));

```

Вывод списка возможных вариантов конструкций выполняется за счёт реализации следующего кода:

```

foreach(int h in k)
{ c++;
richTextBox1.Text+= t1 + " . ";
if(c==20)
{ richTextBox1.Text+=Environment.NewLine;
c=0;}}

```

Перевод числового описания конструкции в текстовое реализуется за счёт:

```
if (h==11)
{ t1 = "1) Усилие штампа < 63 МН ";
richTextBox1.Text += t1;}
if (h==12)
{ t1 = "1) Усилие штампа > 63 МН ";
richTextBox1.Text += t1;}
if (h==21)
{ t1 = "2) Штамп гибки ";
richTextBox1.Text += t1;}
if (h==22)
{ t1 = "2) Штамп вытяжки ";
richTextBox1.Text += t1;}
if (h==23)
{ t1 = "2) Штамп формовки ";
richTextBox1.Text += t1;}
и т.д.
```

Счетчик вариантов технических решений представлен в виде:

```
foreach(int[] k in ИТОГ)
{
b++;
textBox1.Text = b.ToString();}
```

В результате срабатывании кода программы пользователь получает текстовое описание возможных вариантов конструкции штампов полученное на основании введенных данных (рисунок 3.13).

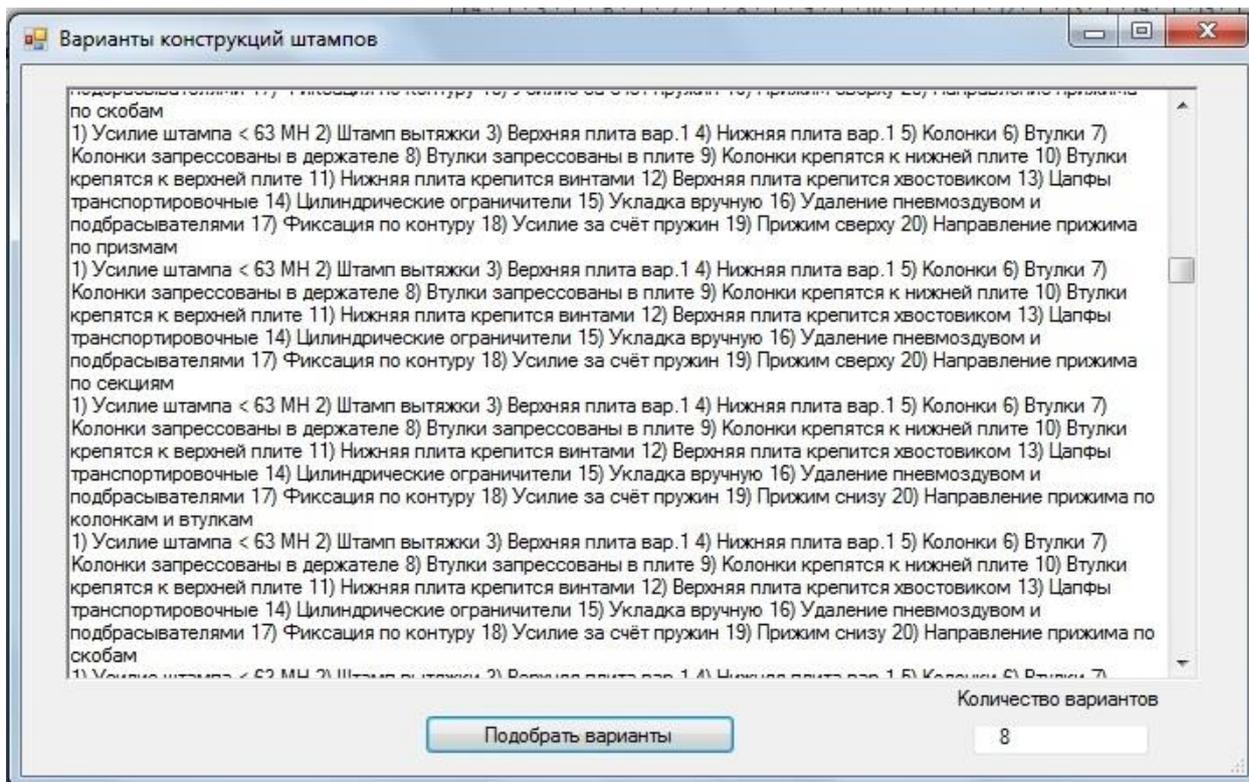


Рисунок 3.13 – Окно вывода текстового описания конструкции штампов

3.3 Реализация связи морфологического И-ИЛИ дерева и кода программы

Элементы конструкций и свойства выявленные в результате морфологического анализа и построенное И-ИЛИ дерева конструкций штампов были реализованы в коде программы следующим образом:

Ограничения в коде которые соответствуют И/ИЛИ - вершинам (рисунок 3.14) могут быть заданы пользователем и реализованы в коде как показано на рисунке 3.15 – рисунке 3.16.

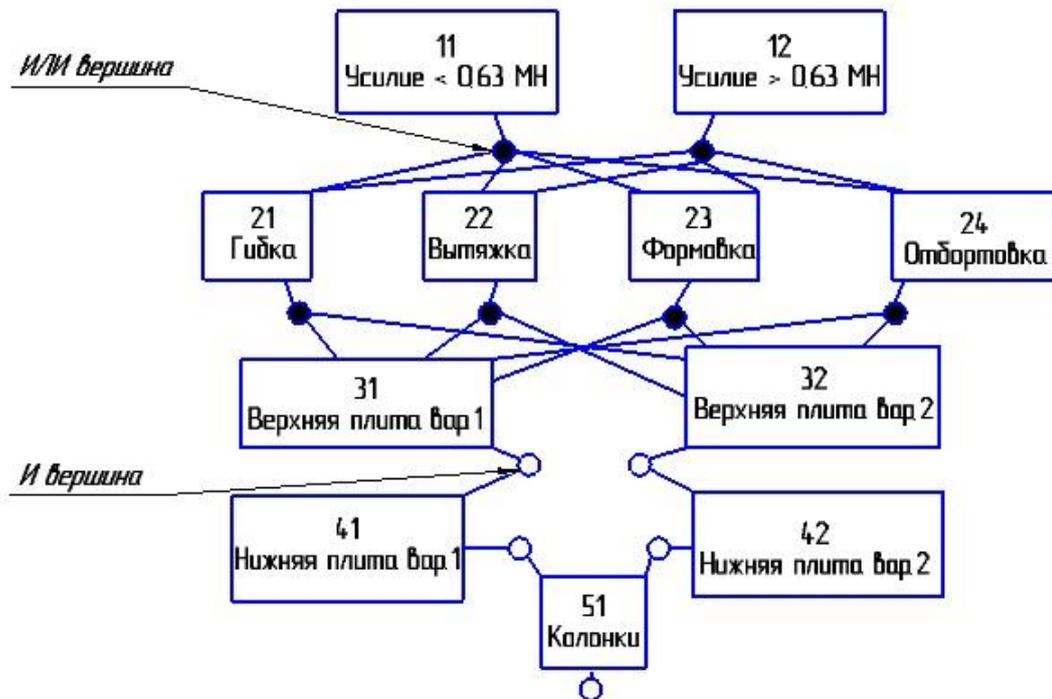


Рисунок 3.14 – Вершины И/ИЛИ морфологического дерева

```

a5 = a3 * a2 * a1*1.5;
if (a5<=63)
{
    B = new int[] {21};
    G = new int[] {11};
    M = new int[] {31};
    F = new int[] {41};
    W = new int[] {121};
}
else
{
    B = new int[] {21};
    G = new int[] {12};
    M = new int[] {32};
    F = new int[] {42};
    W = new int[] {122};
}

```

Рисунок 3.15 – Реализация ограничений задаваемых пользователем

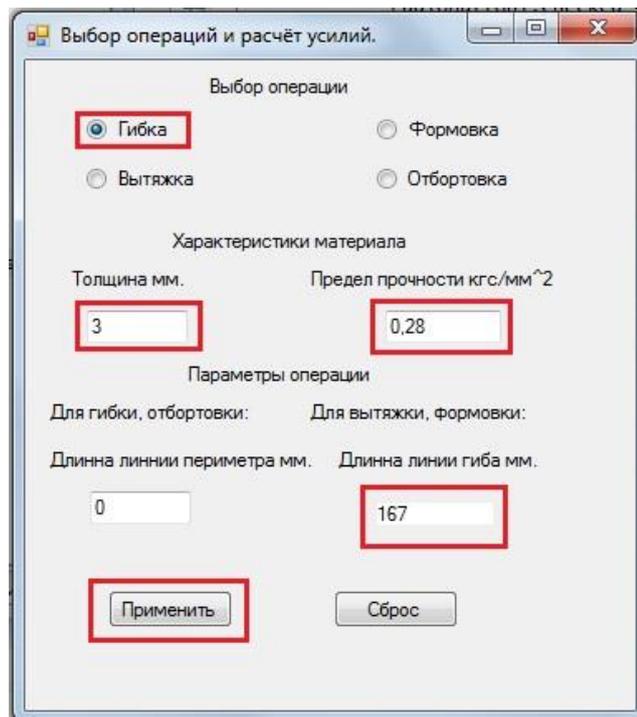


Рисунок 3.16 – Параметры вводимые пользователем

В результате введённых ограничений дерево приобретает вид показанный на рисунке 3.17

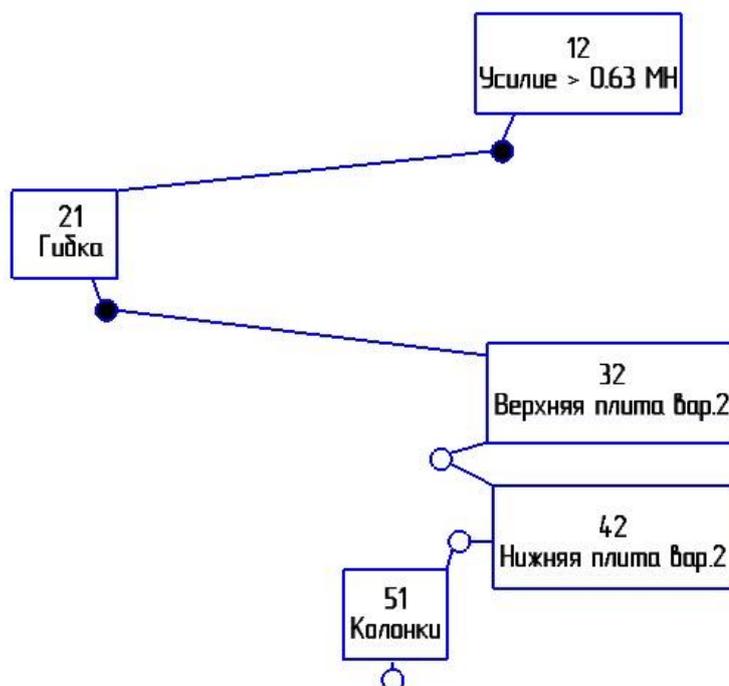


Рисунок 3.17 – И/ИЛИ дерево с учётом введенных пользователем ограничений.

Ограничения в коде так же задаются системно рисунку 3.18

```
if (((PROMMAS[2]==31)&&(PROMMAS[3]==42))||((PROMMAS[2]==32)&&(PROMMAS[3]==41))||((PROMMAS[8]==91)&&(PROMMAS[9]==101))  
    ||((PROMMAS[8]==92)&&(PROMMAS[9]==102)))  
{  
    Resh = false;  
}
```

Рисунок 3.18 – Системное задание ограничений

Пользователь в окне диалога вводит необходимые параметры (рисунок 3.19). Номера элементов дерева разбитого на уровни (рисунок 3.20) образуют массивы количество которых равно количеству уровней (рисунок 3.21).

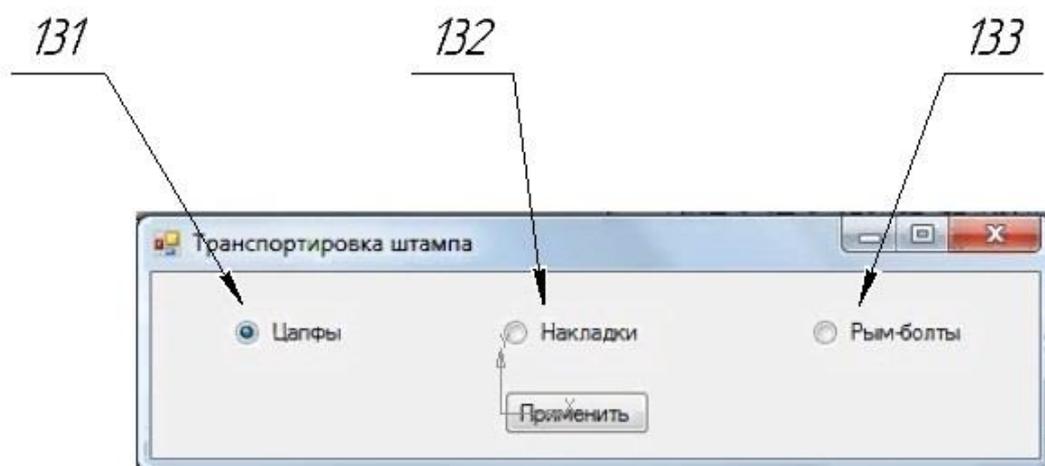


Рисунок 3.19 – Ввод пользователем параметров

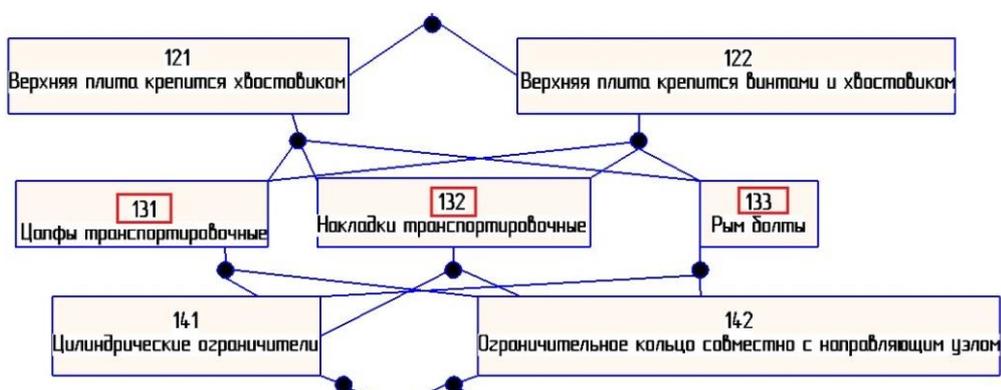


Рисунок 3.20 – Номера элементов морфологического дерева

```

if (radioButton1.Checked)
{
    M13 = new int[] {131};
}
if (radioButton2.Checked)
{
    M13 = new int[] {132};
}
if (radioButton3.Checked)
{
    M13 = new int[] {133};
}

```

Рисунок 3.21 – Реализация объявления массива

Преобразование числового элемента И/ИЛИ дерева обратно в текстовое описание конструкции происходит в последнем окне за счёт реализации следующей части кода (рисунок 3.22)

```

}
    if (h==201)
    {
        t1 = "19) Прижим сверху ";
        richTextBox1.Text += t1;
    }
    if (h==202)
    {
        t1 = "19) Прижим снизу ";
        richTextBox1.Text += t1;
    }
    if (h==211)
    {
        t1 = "20) Направление прижима по колонкам и втулкам ";
        richTextBox1.Text += t1;
    }
}

```

Рисунок 3.22 – Преобразование числового в текстовое описание.

Заключение

1) В результате проделанной работы был проведен анализ конструкций мелких и средних формообразующих штампов позволивший определить их конструктивные особенности и область применения.

2) Конструктивный анализ позволил выявить основные признаки, отвечающие за отличительные особенности конструкций рассматриваемой группы штампов и по которым была осуществлена классификация штамповой оснастки.

3) Проведённый морфологический анализ на основании выявленных признаков, позволил разработать таблицу вариантов технических решений на базе которой было построено И/ИЛИ дерево конструкций штампа с учётом условий ограничения.

4) Разработан метод формализации данных процесса проектирования мелких и средних штампов для переноса их в область автоматизированного проектирования.

5) На языке программирования C# в программе SharpDevelop был разработан интерфейс при помощи которого пользователь имеет возможность задавать параметры проектирования конструкций штампов.

6) Программный код был построен по алгоритму сортировки (пузырьковому алгоритму) с наложением на него ограничений.

7) Разработанная программа позволяет получить текстовое описание конструкции штампа на основании введенных пользователем данных в окнах диалога.

8) Тестирование программы было осуществлено при разработке штамповой оснастки для деталей представленных на рисунках 1.1 – 1.3.

Список используемой литературы

1. Быков, В.П. Методическое обеспечение САПР в машиностроении / В.П. Быков. – Л.: Машиностроение, 1989. - С. 153-155.
2. Соломенцев, Ю.М., Митрофанов В.Г., Прохоров А.Ф. Автоматизированное проектирование и производство в машиностроении / под общ. ред. Ю.М. Соломенцева. – М.: Машиностроение, 1986. – 256 с.
3. Шилдт, Г. С # 4.0. Полное руководство / Г. Шилдт. – ИД «Вильямс», 2015.-1056.
4. Хейлсберг, А. Торгерсен, М. Вилтамут , С. Голд, П. Язык программирования C#. Классика Computers Science. 4-е изд. / Хейлсберг А - ИД «Питер», 2014. - 784 с.
5. Беркли, Э. Символическая логика и разумные машины / Беркли Э. – М.: Изд. иностр. литературы, 1961.– 260 с.
6. Бир, Ст. Кибернетика и управление производство / Бир Ст. - М.: Изд. физ.-мат. литературы, 1963. - 276 с.
7. Грувер, М. САПР и автоматизация производства / М. Грувер, Э. Зиммерс. М.: Мир, 1987. - 528 с.
8. Поспелов, Г.С. Искусственный интеллект — основа новой информационной технологии / Поспелов Г.С. - М.: Высшая школа, 1988. - 278 с.
9. Романовский, В.П. Справочник по холодной штамповке / Романовский В. - М.: Машиностроение, 1979. – 520 с.
10. Рудман, Л.И. Справочник конструктора штампов / Рудман Л. И. - М.: Машиностроение, 1988. – 494 с.
11. Лафоре, Р. Объектно-ориентированное программирование в C++/ Лафоре Р. - ИД «Питер», 2015. – 928 с.
12. Берлинер, Э. Таратынов, О. САПР в машиностроении / Берлинер Э - ИД «Форум», 2012. – 448 с.

13. Скворцов, В.Г. Основы конструирования штампов для холодной штамповки / Скворцов В.Г - М.: Машиностроение, 1974. – 320 с.
14. Сторожев, М.В. Попов, Е.А. Теория обработки металлов давлением / М.В.Сторожев - М.: Машиностроение, 1977. – 423 с.
15. Банкетов, А. Н. Кузнечно-штамповочное оборудование / Банкетов А. Н - М.: Машиностроение, 1982. - 576с.
16. Дурандин, М.М., Рымзин, Н.П. Шихов, Н.А. Штампы для холодной штамповки мелких деталей. Альбом конструкций и схем / Дурандин М.М. - М.: Машиностроение, 1978. – 108 с.
17. Бегг, К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика / К. Бегг, Т. Коннолли - ИД «Вильямс», 2003. - 1436 с.
18. К.Дж.Дейт, Дарвен, Х. Основы будущих систем баз данных. Третий манифест / К.Дж.Дейт - ИД «Янус-К», 2004.- 656 с.
19. Ватсон, Б. С# 4.0 на примерах / Ватсон Б. - ИД «БХВ-Петербург», 2011. - 604 с.
20. Рассел, Д. Морфологический анализ (изобретательство) / Рассел Д - ИД «VSD», 2013. - 87 с.
21. Gurevich, Y. Formalizing Database Recovery [Текст] / Gurevich, Y. Soparkar, N. Wallace, C.// Journal of Universal Computer Science, vol. 3, no. 4 – 1997 – С. 321–340. – Библиогр.: с. 340.;
22. Bjorner, D. FORMALIZATION OF DATABASE SYSTEMS - AND A FORMAL DEFINITION OF IMS [Текст] / Bjorner, D. Lbvengreen, H. // DEPARTMENT OF COMPUTER SCIENCE, TECHNICAL UNIVERSITY OF DENMARK – 1982 – С. 334–347. – Библиогр.: с. 347.;
23. Соломенцев, Ю.М. Митрофанов. В.Г. Прохоров А.Ф. Автоматизированное проектирование и производство в машиностроении/ Соломенцев, Ю.М. – М.: Машиностроение, 1986. 256 с.
24. Вендров, А.М. Проектирование программного обеспечения экономических информационных систем/ Вендров А.М. – М.: Финансы и статистика, 2005. 544 с.

25. Гагарина, Л.Г. Кокорева, Е.В. Виснадул, Б.Д. Технология разработки программного обеспечения./ Гагарина, Л.Г. – М.:ИНФРА-М, 2008. 400 с.
26. Буч, Г. Рамбо, Д. Джекобсон, А. Язык UML: руководство пользователя/ Буч, Г. – М.: ДМК Пресс, 2004. 429 с.
27. Норенков, И.П. Основы автоматизированного проектирования./ Норенков, И.П. – М.: МГТУ им. Баумана, 2002. 336 с.
28. Сурмин, Ю.П. Теория систем и системный анализ./ Сурмин, Ю.П. – Киев: МАУП, 2003. 368 с.
29. Аникин, В.М. Лукашин, В.М. Справочник конструктора штампов для холодной штамповки./ Аникин, В.М. – М.: Машгиз, 1960. 295 с.
30. Почекуев, Е.Н. Системное проектирование последовательных разделительных штампов вырубки листовых заготовок // Почекуев Е.Н., Скрипачев А.В., Шенбергер П.Н. Вестник СГАУ им. академика С.П. Королева (национального исследовательского университета). 2012. № 1. С. 170–177.

ПРИЛОЖЕНИЕ А

Код программы.

```
MainForm.  
using System;  
using System.Collections.Generic;  
using System.Drawing;  
using System.Windows.Forms;  
using System.IO;  
using System.Collections;  
namespace sortirovka2  
{  
    /// <summary>  
    /// Description of MainForm.  
    /// </summary>  
    public partial class MainForm : Form  
    {  
        public MainForm()  
        {  
            //  
            // The InitializeComponent() call is required for Windows Forms designer  
support.  
            //  
            InitializeComponent();  
            //  
            // TODO: Add constructor code after the InitializeComponent() call. }  
            double a1,a2,a3,a4,a5;// Данные которые вводит пользователь в Text-  
Box  
            public int [] B;  
            public int [] G;  
            public int [] M;
```

```

public int [] F;
public int [] W;
public Form1 obj;
void Button3Click(object sender, EventArgs e)
{
radioButton1.Checked = false;
radioButton2.Checked = false;
radioButton3.Checked = false;
radioButton4.Checked = false;
textBox1.Clear();
textBox2.Clear();
textBox3.Clear();
textBox4.Clear();
}
public void Button1Click(object sender, EventArgs e)
{
a1 = Convert.ToDouble(textBox1.Text); //толщина материала
a2 = Convert.ToDouble(textBox2.Text); // предел прочности
a3 = Convert.ToDouble(textBox3.Text); // длинна линиигиба
a4 = Convert.ToDouble(textBox4.Text); //длинна линии периметра
if (radioButton1.Checked)
{
a5 = a3 * a2 * a1*1.5;
if (a5<=63)
{
B = new int[] {21};
G = new int[] {11};
M = new int[] {31};
F = new int[] {41};
W = new int[] {121};
}
}
}

```

```

}
else
{
B = new int[] {21};
G = new int[] {12};
M = new int[] {32};
F = new int[] {42};
W = new int[] {122};
}
}
if(radioButton2.Checked)//Штампы формовки
{
B = new int[] {22};
a5 = a4 * a1 * a2 * (1+a2) * 1.3;
if (a5<=63)
{
B = new int[] {22};
G = new int[] {11};
M = new int[] {31};
F = new int[] {41};
W = new int[] {121};
}
else
{
B = new int[] {22};
G = new int[] {12};
M = new int[] {32};
F = new int[] {42};
W = new int[] {122};
}
}

```

```

}
if (radioButton3.Checked)//ШТАМПЫ ВЫТЯЖКИ
{
a5 = a4 * a1 * a2 * (1+a2) * 1.3;
if (a5<=63)
{
B = new int[] {23};
G = new int[] {11};
M = new int[] {31};
F = new int[] {41};
W = new int[] {121};
}
else
{
B = new int[] {23};
G = new int[] {12};
M = new int[] {32};
F = new int[] {42};
W = new int[] {122};
}
}
if (radioButton4.Checked)// ШТАМПЫ ОТБОРТОВКИ
{
a5 = a3 * a2 * a1*1.5;
if (a5<=63)
{
B = new int[] {24};
G = new int[] {11};
M = new int[] {31};
F = new int[] {41};

```

```
W = new int[] {121};
}
else
{
B = new int[] {24};
G = new int[] {12};
M = new int[] {32};
F = new int[] {42};
W = new int[] {122};
Form1 obj = new Form1();//Открывает и переносит массивы в следующую-
щее окно
```

```
obj.M1 = G;
obj.M2 = B ;
obj.M3 = M;
obj.M4 = F;
obj.M12 = W;
obj.ShowDialog();
}
}
}
```

```
Form1
using System;
using System.Drawing;
using System.Windows.Forms;
namespace sortirovka2
{
/// <summary>
/// Description of Form1.
/// </summary>
```

```

public partial class Form1 : Form
{
public Form1()
{
//
// The InitializeComponent() call is required for Windows Forms designer
support.
//
InitializeComponent();
//
// TODO: Add constructor code after the InitializeComponent() call.
//
}
public int [] M1 ;
public int [] M2 ;
public int [] M3 ;
public int [] M4 ;
public int [] M12 ;
public int [] M15;
void Button1Click(object sender, EventArgs e)
{
if (radioButton1.Checked)//Укладка вручную
{
M15 = new int[] {151};
}
if (radioButton2.Checked)//Укладка пневмоукладчиком
{
M15 = new int[] {152};
}
Form2 obj = new Form2();

```

```
obj.G1=M1;
obj.G2=M2;
obj.G3=M3;
obj.G4=M4;
obj.G5=M12;
obj.G6=M15;
obj.ShowDialog();
}
}
}
```

Form2

```
using System;
using System.Drawing;
using System.Windows.Forms;
namespace sortirovka2
{
    /// <summary>
    /// Description of Form2.
    /// </summary>
    public partial class Form2 : Form
    {
        public Form2()
        {
            //
            // The InitializeComponent() call is required for Windows Forms designer
support.
            //
            InitializeComponent();
            //

```

```

// TODO: Add constructor code after the InitializeComponent() call.
//
}
public int [] G1 ;
public int [] G2;
public int [] G3;
public int [] G4;
public int [] G5;
public int [] G6;
public int [] M16;
void Button1Click(object sender, EventArgs e)
{
if (radioButton1.Checked)//удаление вручную
{
M16 = new int[] {161};
}
if (radioButton2.Checked)//удаление сдувом
{
M16 = new int[] {162};
}
Form3 obj = new Form3();
obj.F1=G1;
obj.F2=G2;
obj.F3=G3;
obj.F4=G4;
obj.F5=G5;
obj.F6=G6;
obj.F7=M16;
obj.ShowDialog();

```

```
}  
}  
}
```

Form 3

```
using System;  
using System.Drawing;  
using System.Windows.Forms;  
namespace sortirovka2  
{  
    /// <summary>  
    /// Description of Form3.  
    /// </summary>  
    public partial class Form3 : Form  
    {  
        public Form3()  
        {  
            //  
            // The InitializeComponent() call is required for Windows Forms designer  
support.  
            //  
            InitializeComponent();  
            //  
            // TODO: Add constructor code after the InitializeComponent() call.  
            //  
        }  
        public int [] F1 ;  
        public int [] F2;  
        public int [] F3;  
        public int [] F4;
```

```

public int [] F5;
public int [] F6;
public int [] F7;
public int [] M17;
void Button1Click(object sender, EventArgs e)
{
if (radioButton1.Checked)//фиксация по отверстиям
{
M17 = new int[] {171};
}
if (radioButton2.Checked)//фиксация по контуру
{
M17 = new int[] {172};
}
if (radioButton3.Checked)//фиксация по форме
{
M17 = new int[] {173};
}
Form4 obj = new Form4();
obj.H1=F1;
obj.H2=F2;
obj.H3=F3;
obj.H4=F4;
obj.H5=F5;
obj.H6=F6;
obj.H7=F7;
obj.H8=M17;
obj.ShowDialog();
}
}

```

```
}
```

Form 4

```
using System;
```

```
using System.Drawing;
```

```
using System.Windows.Forms;
```

```
using System.IO;
```

```
using System.Collections;
```

```
namespace sortirovka2
```

```
{
```

```
/// <summary>
```

```
/// Description of Form4.
```

```
/// </summary>
```

```
public partial class Form4 : Form
```

```
{
```

```
public Form4()
```

```
{
```

```
//
```

```
// The InitializeComponent() call is required for Windows Forms designer  
support.
```

```
//
```

```
InitializeComponent();
```

```
//
```

```
// TODO: Add constructor code after the InitializeComponent() call.
```

```
//
```

```
}
```

```
public int [] H1;
```

```
public int [] H2;
```

```
public int [] H3;
```

```
public int [] H4;
```

```

public int [] H5;
public int [] H6;
public int [] H7;
public int [] H8;
public int [] M18;
public int [] M19;
void Button1Click(object sender, EventArgs e)
{
if (radioButton1.Checked)//уилие за счет пружин
{
M18 = new int[] {192};
M19 = new int[] {201,202};
}
if (radioButton2.Checked)//уилие за счет маркета
{
M18 = new int[] {191};
M19 = new int[] {202};
}
Form5 obj = new Form5();
obj.V1=H1;
obj.V2=H2;
obj.V3=H3;
obj.V4=H4;
obj.V5=H5;
obj.V6=H6;
obj.V7=H7;
obj.V8=H8;
obj.V9=M18;
obj.V10=M19;
obj.Show();

```

```
}  
}  
}
```

Form5

```
using System;
```

```
using System.Drawing;
```

```
using System.Windows.Forms;
```

```
namespace sortirovka2
```

```
{
```

```
/// <summary>
```

```
/// Description of Form5.
```

```
/// </summary>
```

```
public partial class Form5 : Form
```

```
{
```

```
public Form5()
```

```
{
```

```
//
```

```
// The InitializeComponent() call is required for Windows Forms designer  
support.
```

```
//
```

```
InitializeComponent();
```

```
//
```

```
// TODO: Add constructor code after the InitializeComponent() call.
```

```
//
```

```
}
```

```
public int [] V1;
```

```
public int [] V2;
```

```
public int [] V3;
```

```
public int [] V4;
```

```

public int [] V5;
public int [] V6;
public int [] V7;
public int [] V8;
public int [] V9;
public int [] V10;
public int [] M13;
void Button1Click(object sender, EventArgs e)
{
if (radioButton1.Checked)
{
M13 = new int[] {131};
}
if (radioButton2.Checked)
{
M13 = new int[] {132};
}
if (radioButton3.Checked)
{
M13 = new int[] {133};
}
Form6 obj = new Form6();
obj.L1=V1;
obj.L2=V2;
obj.L3=V3;
obj.L4=V4;
obj.L5=V5;
obj.L6=V6;
obj.L7=V7;
obj.L8=V8;

```

```
obj.L9=V9;
obj.L10=V10;
obj.L11=M13;
obj.ShowDialog();
}
}
}
```

Form6

```
using System;
using System.Collections;
using System.Drawing;
using System.Windows.Forms;
using System.IO;
namespace sortirovka2
{
    /// <summary>
    /// Description of Form6.
    /// </summary>
    public partial class Form6 : Form
    {
        public Form6()
        {
            //
            // The InitializeComponent() call is required for Windows Forms designer
support.
            //
            InitializeComponent();
            //
            // TODO: Add constructor code after the InitializeComponent() call.
```

```

//
}
public int [] L1;
public int [] L2;
public int [] L3;
public int [] L4;
public int [] L5;
public int [] L6;
public int [] L7;
public int [] L8;
public int [] L9;
public int [] L10;
public int [] L11;
void Button1Click(object sender, EventArgs e)
{
int[] M5 = {51};
int[] M6 = {61};
int[] M7 = {71,72};
int[] M8 = {81,82};
int[] M9 = {91,92};
int[] M10 = {102,101};
int[] M11 = {111};
int[] M14 = {141,142};
int[] M21 = {211,212,213,214};
int
i1,i2,i3,i4,i5,i6,i7,i8,i9,i10,i11,i12,i13,i14,i15,i16,i17,i19,i20,i21;//Объявляем
пременные
int c = 0;//объявляем переменную c
int b = 0;
string t1;

```

```

ArrayList ITOG = new ArrayList();//задаем параметры цикла .
for (i1=0;i1<=L1.Length-1;i1++)
{
for (i2=0;i2<=L2.Length-1;i2++)
{
for (i3=0;i3<=L3.Length-1;i3++)
{
for (i4=0;i4<=L4.Length-1;i4++)
{
for (i5=0;i5<=M5.Length-1;i5++)
{
for (i6=0;i6<=M6.Length-1;i6++)
{
for (i7=0;i7<=M7.Length-1;i7++)
{
for (i8=0;i8<=M8.Length-1;i8++)
{
for (i9=0;i9<=M9.Length-1;i9++)
{
for (i10=0;i10<=M10.Length-1;i10++)
{
for (i11=0;i11<=M11.Length-1;i11++)
{
for (i12=0;i12<=L5.Length-1;i12++)
{
for (i13=0;i13<=L11.Length-1;i13++)
{
for (i14=0;i14<=M14.Length-1;i14++)
{
for (i15=0;i15<=L6.Length-1;i15++)

```

```

{
for (i16=0;i16<=L7.Length-1;i16++)
{
for (i17=0;i17<=L8.Length-1;i17++)
{
for (i19=0;i19<=L9.Length-1;i19++)
{
for (i20=0;i20<=L10.Length-1;i20++)
{
for (i21=0;i21<=M21.Length-1;i21++)
{
int[] PROM-
MAS={L1[i1],L2[i2],L3[i3],L4[i4],M5[i5],M6[i6],M7[i7],M8[i8],M9[i9],M10[i1
0],M11[i11],L5[i12],L11[i13],M14[i14],
L6[i15],L7[i16],L8[i17],L9[i19],L10[i20],M21[i21]};
int v = 0;
bool Resh;
if (((PROM-
MAS[2]==31)&&(PROMMAS[3]==42))||((PROMMAS[2]==32)&&(PROMMA
S[3]==41))||((PROMMAS[8]==91)&&(PROMMAS[9]==101))
||((PROMMAS[8]==92)&&(PROMMAS[9]==102)))
{
Resh = false;
}
else
{
if (v<=0)
{
ITOG.Add(PROMMAS);
v = v+1;

```

```

}
}
}}}}}}}}}}}}}}}}}}}}}}
foreach(int[] k in ITOG)
{
b++;
textBox1.Text = b.ToString();
foreach(int h in k)
{
if (h==11)
{
t1 = "1) Усилие штампа < 63 МН ";
richTextBox1.Text += t1;
}
if (h==12)
{
t1 = "1) Усилие штампа > 63 МН ";
richTextBox1.Text += t1;
}
if (h==21)
{
t1 = "2) Штамп гибки ";
richTextBox1.Text += t1;
}
if (h==22)
{
t1 = "2) Штамп вытяжки ";
richTextBox1.Text += t1;
}
if (h==23)

```

```
{
t1 = "2) Штамп формовки ";
richTextBox1.Text += t1;
}
if (h==24)
{
t1 = "2) Штамп отбортовки ";
richTextBox1.Text += t1;
}
if (h==31)
{
t1 = "3) Верхняя плита вар.1 ";
richTextBox1.Text += t1;
}
if (h==32)
{
t1 = "3) Верхняя плита вар.2 ";
richTextBox1.Text += t1;
}
if (h==41)
{
t1 = "4) Нижняя плита вар.1 ";
richTextBox1.Text += t1;
}
if (h==42)
{
t1 = "4) Нижняя плита вар.1 ";
richTextBox1.Text += t1;
}
if (h==51)
```

```
{
t1 = "5) Колонки ";
richTextBox1.Text += t1;
}
if (h==61)
{
t1 = "6) Втулки ";
richTextBox1.Text += t1;
}
if (h==71)
{
t1 = "7) Колонки запрессованы в держателе ";
richTextBox1.Text += t1;
}
if (h==72)
{
t1 = "7) Колонки запрессованы в плите ";
richTextBox1.Text += t1;
}
if (h==81)
{
t1 = "8) Втулки запрессованы в держателе ";
richTextBox1.Text += t1;
}
if (h==82)
{
t1 = "8) Втулки запрессованы в плите ";
richTextBox1.Text += t1;
}
if (h==91)
```

```
{
t1 = "9) Колонки крепятся к нижней плите ";
richTextBox1.Text += t1;
}
if (h==92)
{
t1 = "9) Колонки крепятся к верхней плите ";
richTextBox1.Text += t1;
}
if (h==101)
{
t1 = "10) Втулки крепятся к нижней плите ";
richTextBox1.Text += t1;
}
if (h==102)
{
t1 = "10) Втулки крепятся к верхней плите ";
richTextBox1.Text += t1;
}
if (h==111)
{
t1 = "11) Нижняя плита крепится винтами ";
richTextBox1.Text += t1;
}
if (h==121)
{
t1 = "12) Верхняя плита крепится хвостовиком ";
richTextBox1.Text += t1;
}
if (h==122)
```

```

{
t1 = "12) Верхняя плита крепится винтами и хвостовиком ";
richTextBox1.Text += t1;
}
if (h==131)
{
t1 = "13) Цапфы транспортировочные ";
richTextBox1.Text += t1;
}
if (h==132)
{
t1 = "13) Накладки транспортировочные ";
richTextBox1.Text += t1;
}
if (h==133)
{
t1 = "13) РЫМ болты ";
richTextBox1.Text += t1;
}
if (h==141)
{
t1 = "14) Цилиндрические ограничители ";
richTextBox1.Text += t1;
}
if (h==142)
{
t1 = "14) Ограничительное кольцо совместно с направляющим узлом
";
richTextBox1.Text += t1;
}

```

```
if (h==151)
{
t1 = "15) Укладка вручную ";
richTextBox1.Text += t1;
}
if (h==152)
{
t1 = "15) Укладка пневмоукладчиком ";
richTextBox1.Text += t1;
}
if (h==161)
{
t1 = "16) Удаление вручную ";
richTextBox1.Text += t1;
}
if (h==162)
{
t1 = "16) Удаление пневмоздувом и подбрасывателями ";
richTextBox1.Text += t1;
}
if (h==171)
{
t1 = "17) Фиксация на отверстия ";
richTextBox1.Text += t1;
}
if (h==172)
{
t1 = "17) Фиксация по контуру ";
richTextBox1.Text += t1;
}
```

```
if (h==173)
{
t1 = "17) Фиксация по форме ";
richTextBox1.Text += t1;
}
if (h==191)
{
t1 = "18) Усилие за счёт маркета ";
richTextBox1.Text += t1;
}
if (h==192)
{
t1 = "18) Усилие за счёт пружин ";
richTextBox1.Text += t1;
}
if (h==201)
{
t1 = "19) Прижим сверху ";
richTextBox1.Text += t1;
}
if (h==202)
{
t1 = "19) Прижим снизу ";
richTextBox1.Text += t1;
}
if (h==211)
{
t1 = "20) Направление прижима по колонкам и втулкам ";
richTextBox1.Text += t1;
}
```

```
if (h==212)
{
t1 = "20) Направление прижима по скобам ";
richTextBox1.Text += t1;
}
if (h==213)
{
t1 = "20) Направление прижима по призмам ";
richTextBox1.Text += t1;
}
if (h==214)
{
t1 = "20) Направление прижима по секциям ";
richTextBox1.Text += t1;
}
c++;
//richTextBox1.Text+ = t1 + " . ";
if(c==20)
{
richTextBox1.Text+=Environment.NewLine;
c=0;
}
}
}
}
}
```