

Реферат

Пояснительная записка имеет объём 63 страниц, содержит 2 таблицы, 13 рисунков и список из 26 использованных источников.

Содержание записки характеризуют следующие ключевые слова: автоматизация конструирования, зубчаток зацепление, эвольвента, твердотельное моделирование, геометрические параметры, CADсистема, SolidEdge, позиция, эскиз, грань.

Объект исследования или разработки – «Программная подсистема, обеспечивающая расширение возможностей системы SolidEdge в части построения зубчатых колес внешнего зацепления в соответствии с ГОСТ 16532-70».

Цель дипломного проекта – Разработка программной системы, обеспечивающей расширение возможностей системы SolidEdge в части построения зубчатых колес внешнего зацепления в соответствии с ГОСТ 16532-70.

Полученные результаты их новизна - результатом выполнения дипломного проекта является реализация программной системы, обеспечивающей расширение возможностей системы SolidEdge при построении зубчатых колес внешнего зацепления в соответствии с ГОСТ 16532-70.

Новизна, разработанной САПР, заключается в частичной автоматизации процесса построения механических передач(зубчатых колес внешнего зацепления)на основе использования современного программного и информационного обеспечения, а также новых информационных технологий.

Область применения – машиностроительные предприятия.

Прогнозные предположения о развитии объекта исследования – данное ПС является актуальным, так как при выполнении конструкторских работ в SolidEdge очень важно располагать инструментарием, выполняющим создание конструктивных элементов в соответствии с ГОСТ в частности с ГОСТ 16532-70.

Содержание

Введение	6
1 Характеристика системы	9
1.1 Характеристика библиотек конструирования изделий	9
1.2 Перспективы внедрения и адаптации САПР на предприятии	11
1.3 Основные требования к программной системе	12
2 Задачи проектируемой библиотеки	15
2.1 Требования к программной подсистеме	15
2.2 Входная и выходная информация разрабатываемой программной системы	17
3 Проектирование программы	21
3.1 Функциональная модель системы	21
3.2 Проектирование алгоритма работы программы	21
3.2.1 Проектирование оптимальной последовательности построения	21
3.2.2 Проектирование базы данных	24
3.3 Модульная структура программы	29
3.4 Алгоритмическая структура программной системы	32
4 Реализация подсистемы	34
4.1 Реализация интерфейса программы	34
4.2 Реализация связи приложения с XML данными	34
4.3 Реализация набора методов, обеспечивающих построение зубчатых колес внешнего зацепления	41
4.4 Тестирование программы	44
4.5 Виды обеспечений	47
4.5.1 Программное обеспечение	47
4.5.2 Техническое обеспечение	48
4.5.3 Лингвистическое обеспечение	49
4.5.4 Информационное обеспечение	50
4.5.5 Математическое обеспечение	53

4.5.6 Методическое обеспечение	54
Заключение	57
Список литературы	58

Введение

Уменьшение длительности проектирования новых изделий, рост производительности труда при разработке новых изделий, повышение качества производимой продукции, увеличение общей эффективности производства – все эти аспекты являются важнейшими факторами, определяющими развитие любого современного производства.

Современный уровень развития систем автоматизированного проектирования основан на серьезной научно-технической основе, к которой можно отнести и современные средства вычислительной техники, и новые методы работы с данными и обширный математический аппарат геометрических вычислений. Реализация всех этих средств зачастую базируется на принципах искусственного интеллекта, методах решения задач оптимизации и численных методах.

Машиностроение – является одной из наиболее важных и сложных областей промышленности. От развития именно машиностроения зависит общий экономический рост, в виду того, что повышение качества выпускаемых изделий и снижение их себестоимости позволяет оставаться конкурентоспособной продукцией с высокой добавленной стоимостью на мировом рынке.

В настоящее время системы АПР и системы управления инженерными данными находят все более широкое применение в промышленности. Более того, функционирование современного производства выполняющего производство высокотехнологичной продукции в принципе невозможно без использования систем АПР.

Использование САПР и систем информационной поддержки жизненного цикла изделий в промышленности влечет значительную экономию затрат на конструкторское, технологическое и другие виды проектирования, а также сопровождение, эксплуатацию высокотехнологичных промышленных изделий. При этом все это возможно при общем увеличении объема проектной и сопроводительной документации.

Целью нашего проектирования является создание программной системы, обеспечивающей расширение возможностей системы SolidEdge в части построения зубчатых колес внешнего зацепления в соответствии с ГОСТ 16532-70.

С целью ускорения разработки системы, целесообразно разбить данную задачу на отдельные функциональные классы, которые сосредоточены в одном решении. Работа над проектом осуществляется в среде MS Visual Studio 2019.

В представленной работе будет дано описание реализации программной системы построения зубчатых колес внешнего зацепления в соответствии с ГОСТ 16532-70. Процесс проектирования программной системы включает следующие основные этапы: формулировка требований к программной системе, исследование предметной области, выбор метода решения, разработка или проектирование программной системы, программная реализация задачи, а также ее тестирование.

Созданное программное средство является актуальным и может быть применено на предприятиях, в которых закуплен программный продукт компании Siemens – SolidEdge, а также производится конструирование механических передач в соответствии с ГОСТ. Положительные результаты решения задачи по созданию программной системы позволят повысить эффективность проектных работ, за счет роста производительности труда конструкторов и повышения качества проектных решений и проектируемых изделий.

Развитие новых технологий постоянно ужесточает требования, предъявляемые к инженеру-конструктору. На первое место в современном конструировании выходят скорость и динамичность выполнения проектов, как расчетных составляющих, так и в виде чертежей или моделей, а также возможность быстрого внесения в них изменений, без какого бы то ни было влияния на качество выполняемых работ или проектируемого объекта. Наверное, каждый инженер неоднократно сталкивался с задачей выполнения многократного конструирования однотипных элементов с применением большого количества справочных материалов, когда объекты вроде бы и не очень различаются, но и приходится производить полный пересмотр параметров и характеристик. Для

решения этой проблемы целесообразно применять средства программирования параметризации, посредством которых можно задать определенные связи между отдельными элементами объекта, позволяющие при последующей разработке типовых конструкций не переделывать все его свойства, а изменить лишь несколько параметров, и провести полный перерасчет и моделирование объекта с применением автоматизированных средств. Это дает возможность многократно использовать единожды построенный алгоритм, значительно сокращающий время на создание новых конструкций.

Хорошо владея одним из языков программирования можно научиться самостоятельно разрабатывать различные по структуре программные модули для решения широкого круга узкопрофильных задач конструирования. Такие приложения смогут производить сложные вычисления, самостоятельно выбирать необходимые параметры из баз данных, обмениваться данными с внешними приложениями и, как результат, конструкторские расчеты неограниченной сложности с учетом всех параметров.

1 Характеристика системы

1.1 Характеристика библиотек конструирования изделий

Широкое внедрение компьютеризации в условиях научно-технического прогресса обеспечивает рост производительности труда в различных областях общественного производства. Главное внимание при этом обращается на те области, где рост производительности труда до применения ЭВМ проходил крайне медленно. Это, в первую очередь, области, связанные с применением умственного труда человека, т.е. управление производством, проектирование и исследование объектов и процессов. Если производительность труда в сфере производства с начала века возросла в сотни раз, то в области проектирования только в 1.5 - 2 раза. Это обуславливает большие сроки проектирования новых объектов, что не отвечает потребностям развития экономики.

Прогресс производства в современных условиях связывают с достижениями в области автоматизации производства. Поскольку проектирование и разработка технологии являются ступенью производства (логическим уровнем), то прогресс на этой ступени также должен определяться автоматизацией.

Автоматизированное проектирование позволяет значительно сократить субъективизм при принятии решений, повысить точность расчетов, выбрать наилучшие варианты для реализации на основе строгого математического анализа всех или большинства вариантов проекта с оценкой технических, технологических и экономических характеристик производства и эксплуатации проектируемого объекта, значительно повысить качество конструкторской документации, существенно сократить сроки проектирования и передачи конструкторской документации в производство, эффективнее использовать технологическое оборудование с программным управлением. Автоматизация проектирования способствует более полному использованию унифицированных изделий в качестве стандартных компонентов проектируемого объекта.

Объединение нескольких ПП в единую систему, предназначенную для

реализации вполне определенных функций, позволяет говорить о новом, более высоком уровне в иерархии программных комплексов, т.е. САПР. При этом качественные изменения претерпевают и организация информационного, технического и других видов обеспечения, и, что особенно важно, условия обмена информацией между человеком и ЭВМ. Как правило, эти изменения направлены на повышение гибкости и универсальности системы, улучшение характеристик взаимодействия проектировщика с ЭВМ, повышение качества получаемого результата и снижение времени его получения. Собственно, САПР могут в качестве подсистемы входить в системы более высокого уровня, например, АСУП (автоматизированных систем управления производством).

Формальное определение САПР, определяющее ее главные особенности, — человеко-машинная система, использующая современные математические методы, средства электронно-вычислительной техники и связи, а также новые организационные принципы проектирования для нахождения и практической реализации наиболее эффективного проектного решения существующего объекта.

Прошло время не связанных друг с другом программ и систем, автоматизирующих отдельные звенья технологической цепи производства, как это было на заре компьютерной эры. Теперь пользователь требует от разработчиков законченные решения, обеспечивающие сквозную технологию в рамках единой интегрированной системы автоматизированного проектирования. Такой подход позволяет моделировать изделие на компьютере и выдавать в производство готовые оптимальные решения путем перебора большого числа вариантов на этапе проектирования и сокращать, таким образом, в несколько раз время выпуска готового изделия и создавать информационную основу для качественного принятия решений в управлении производством.

1.2 Перспективы внедрения и адаптации САПР на предприятии

При внедрении и адаптации САПР на предприятии существует общее описание задач, планируемых к решению с помощью САПР. В частности, на предприятии ООО «Бизнес Инвест Трейд», обозначены следующие основные задачи, решаемые автоматизацией технологических и производственных процессов.

Цель деятельности технического отдела (ТО) компании – Конструкторская подготовка производства, разработка конструкторской и проектной документации (КПД) на изделия, выпускаемые предприятием.

Основные задачи, решаемые автоматизацией конструкторских работ:

1. Внедрение программного обеспечения, для поддержки работ ТО по конструированию и подготовке конструкторской документации изготовления изделий в машиностроительном производстве;

2. Создание отлаженной системы документооборота: сбора (создания), хранения и обработки конструкторской и технологической информации, упрощающей взаимодействие между ТО и смежными службами производства, снабжение и др., организующей сквозное прохождение электронной информации через перечисленные службы, средствами Teamcenter;

3. Внедрение системы защиты, создаваемой документации от несанкционированного доступа.

Приоритетами при автоматизации ТО считается:

1. Автоматизация процесса передачи конструкторской документации из ТО в смежные подразделения;

2. Автоматизация процесса конструирования изделий;

3. Автоматизация процесса расчета;

4. Автоматизация электронного документооборота и совместного доступа к базам данных конструкторско-технологических служб.

1.3 Основные требования к программной системе

При разработке программного средства, которое будет не только нужным, но и конкурентоспособным, необходимо учесть требования, которые предъявляют пользователи к программам такого плана.

Удобный и интуитивно понятный интерфейс.

Программное средство должно обладать интерфейсом, пользоваться которым сможет любой человек без специальной подготовки и длительного обучения. Функции системы должны дублироваться в различных элементах управления формы (меню, кнопки и т.д.), чтобы пользователь мог выбирать удобный для него способ работы с программой. Выбор пути открытия и сохранения файлов должен проходить в диалоговом режиме для исключения ошибок ввода.

Выполнение программой своего функционального назначения.

Программная система должна обеспечивать расширение функциональных возможностей SolidEdgeв части расчета и построения зубчатых колес внешнего зацепления в соответствии с ГОСТ 16532-70. В работе программы должен быть предусмотрен механизм автоматизированного расчета и установкimeжосевого расстояния.

Уведомление о возникающих ошибках и обработка исключительных ситуаций.

Ошибки, которые могут возникнуть при работе программы, должны быть предусмотрены программистом и не должны вызывать прерывание работы ПС. Кроме того, должны выводиться понятные сообщения, информирующие о причине сбоя работы программы.

- Вывод краткой информации о текущем состоянии системы.

Блок вывода информации о текущем состоянии системы, поможет пользователю сориентироваться и определить его дальнейшие действия, в текущий момент времени.

- Наличие встроенной справочной системы.

ПС должно иметь удобную и развернутую встроенную справочную систему для решения возникающих в ходе эксплуатации вопросов. Кроме того, справочная система должна также выступать в роли средства обучения пользователя работе с программой.

- Надежность.

Данное ПС должно обеспечивать сохранность исходных данных для формирования нового запроса на создание зубчатого зацепления, т.е. использовать механизм защищенного доступа к данным, обеспечивающим их целостность на каждом этапе работы программы. При непредвиденном прекращении работы программы должен обеспечиваться процесс ведения лога.

Низкие требования к техническим и программным ресурсам компьютера пользователя.

Программа должна занимать небольшой объем на жестком диске пользователя и потреблять минимальное количество ресурсов системы. Максимальное количество компонентов для работы программы (библиотеки, файлы) должно быть включено в установочный файл программы. Программное обеспечение, необходимое для функционирования ПС, должно иметь минимальную стоимость или быть бесплатным.

Высокая скорость работы.

ПС должно обрабатывать большие массивы данных в кратчайшие сроки, это определяется использованием сервис-ориентированной архитектуры при разработке автономного клиента Teamcenter.

Низкая стоимость программы и ее эксплуатации.

Затраты на разработку программного средства должны быть минимальными, что определяется правильным подбором технических и программных средств, которые будут использоваться при проектировании и реализации программных функций.

Легкая установка на компьютер пользователя.

Установка на компьютер пользователя самого программного приложения и всех необходимых программных модулей должна проходить быстро и при минимальном участии человека, т.е. не требовать от пользователя выполнения дополнительных действий (копирования файлов, создания папок, ввода различных сведений и т.д.). Также при установке необходимо привести перечень программных средств, которые должны быть установлены на ПК пользователя.

2 Задачи проектируемой библиотеки

2.1 Требования к программной подсистеме

При разработке программного средства, которое будет не только нужным, но и конкурентоспособным, необходимо учесть требования, которые предъявляют пользователи к программам такого плана.

Удобный и интуитивно понятный интерфейс.

Программное средство должно обладать интерфейсом, пользоваться которым сможет любой человек без специальной подготовки и длительного обучения. Функции системы должны дублироваться в различных элементах управления формы (меню, кнопки и т.д.), чтобы пользователь мог выбирать удобный для него способ работы с программой. Выбор пути открытия и сохранения файлов должен проходить в диалоговом режиме для исключения ошибок ввода.

-Выполнение программой своего функционального назначения.

Модуль должен обеспечивать выполнение указанного процесса для определенного пользователем объекта. В работе модуля должно быть предусмотрено нахождения, шаблонов процессов и запуска выбранного процесса.

Уведомление о возникающих ошибках и обработка исключительных ситуаций.

Ошибки, которые могут возникнуть при работе программы, должны быть предусмотрены программистом и не должны вызывать прерывание работы ПС. Кроме того, должны выводиться понятные сообщения, информирующие о причине сбоя работы программы.

- Вывод краткой информации о текущем состоянии системы.

Блок вывода информации о текущем состоянии системы, поможет пользователю сориентироваться и определить его дальнейшие действия, в текущий момент времени.

- Наличие встроенной справочной системы.

ПС должно иметь удобную и развернутую встроенную справочную систему для решения возникающих в ходе эксплуатации вопросов. Кроме того, справочная система должна также выступать в роли средства обучения пользователя работе с программой.

- Надежность.

Данное ПС должно обеспечивать сохранность исходных данных для формирования нового запроса на создание объекта, т.е. использовать технологию Services, для доступа к объектам Teamcenter. При непредвиденном прекращении работы программы возможно будет снова запустить процесс авторизации пользователя.

- Низкие требования к техническим и программным ресурсам компьютера пользователя.

Программа должна занимать небольшой объем на жестком диске пользователя и потреблять минимальное количество ресурсов системы. Максимальное количество компонентов для работы программы (библиотеки, файлы) должно быть включено в установочный файл программы. Программное обеспечение, необходимое для функционирования ПС, должно иметь минимальную стоимость или быть бесплатным.

- Высокая скорость работы.

ПС должно обрабатывать большие массивы данных в кратчайшие сроки, это определяется использованием сервис-ориентированной архитектуры при разработке автономного клиента Teamcenter.

- Низкая стоимость программы и ее эксплуатации.

Затраты на разработку программного средства должны быть минимальными, что определяется правильным подбором технических и программных средств, которые будут использоваться при проектировании и реализации программных функций.

- Легкая установка на компьютер пользователя.

Установка на компьютер пользователя самого программного приложения и всех необходимых программных модулей должна проходить быстро и при минимальном участии человека, т.е. не требовать от пользователя выполнения дополнительных действий (копирования файлов, создания папок, ввода различных сведений и т.д.). Также при установке необходимо привести перечень программных средств, которые должны быть установлены на ПК пользователя.

2.2 Входная и выходная информация разрабатываемой программной системы

В ходе рассмотрения состава и структуры входной информации программной системы, следует перечислить ее основные составляющие. Поскольку программа считывает параметры зубчатых колес внешнего зацепления из интерфейса программы, то входными параметрами можно считать заданные пользователем значения. В соответствии со стандартом, модули зубчатых колес должны выбираться в соответствии с ГОСТ 9563-60 и таблицей 2.1.

Таблица 2.1 – Таблица модулей зубчатых колес по ГОСТ 9563-60

1 ряд	0,05		0,06		0,08		0,1		0,12		0,15	
2 ряд		0,055		0,07		0,09		0,11		0,14		0,18
1 ряд	0,2		0,25		0,3		0,4		0,5		0,6	
2 ряд		0,22		0,28		0,35		0,45		0,55		0,7
1 ряд	0,8		1		1,25		1,5		2		2,5	
2 ряд		0,9		1,125		1,375		1,75		2,25		2,75
1 ряд	3		4		5		6		8		10	
2 ряд		3,5		4,5		5,5		7		9		11
1 ряд	12		16		20		25		32		40	
2 ряд		14		18		22		28		36		45
1 ряд	50		60		80		100					
2 ряд		55		70		90						

Приведем таблицу исходных параметров зубчатых колес внешнего зацепления в соответствии с ГОСТ 16532-70 и ее эскиз (табл. 2.2):

Таблица 2.2 – Исходные данные для расчета

Наименование параметров		Обо- зна- чения	Расчетные формулы и указания
Число зубьев	шестерни	Z1	-
	колеса	Z2	-
Модуль		M	-
Угол наклона		β	-
Нормальный исходный контур	Угол профиля	α	-
	Коэффициент высоты головки	ha*	-
	Коэффициент радиуса кривизны переходной кривой	ρl^*	-
	Коэффициент граничной высоты	hl*	-
	Коэффициент радиального зазора	c*	-
	Линия модификации головки	-	-
	Коэффициент высоты модификации головки	hg*	-
	Коэффициент глубины модификации головки	Δ^*	-
Межосевое расстояние		a _w	Входит в состав исходных данных, если его значение задано
Коэффициент смещения	у шестерни	X1	Входят в состав исходных данных, если значение межосевого расстояния не задано. Рекомендации по выбору коэффициентов смещения настоящим стандартом не устанавливаются. Величины коэффициентов смещения определяются требуемыми прочностными и геометрическими показателями качества передачи
	у колеса	X2	

Данная совокупность входной информации определяет полный массив входной информации, который требуется для создания решающего алгоритма.

В качестве выходной информации будет выступать готовая спроектированная системой SolidEdge механическая передача внешнего зацепления.

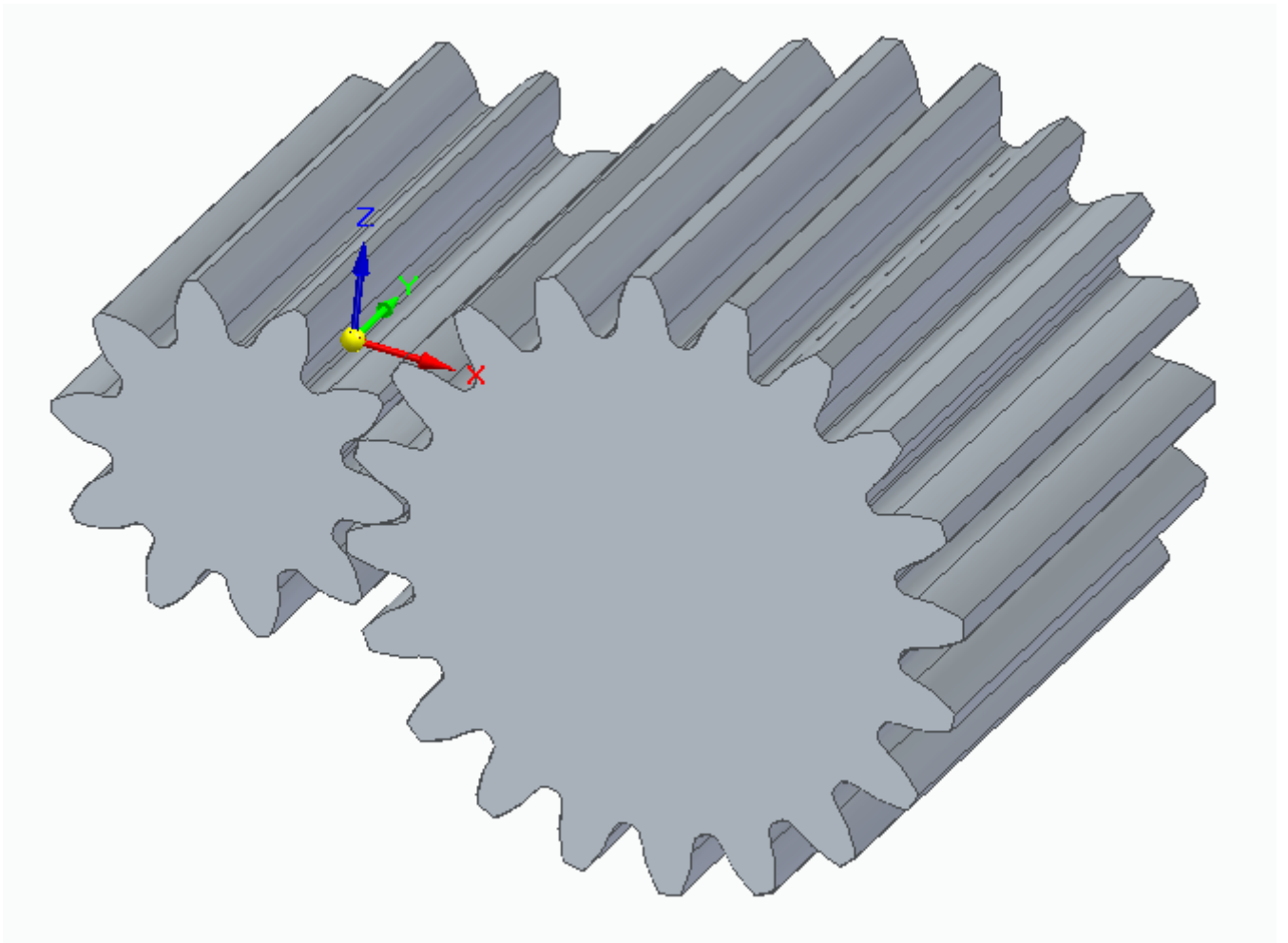


Рисунок 2.1 – Пример построения элементов механической передачи внешнего зацепления

3 Проектирование программы

3.1 Функциональная модель системы

В ходе анализа требований к программе и определения круга задач, подлежащих автоматизации, определяется функциональная модель программы. В ее состав входят функции, связанные как с выполнением сервисных операций, так и операций по построению конструктивного элемента. Решено использовать средства доступа к ядру системы SolidEdge.

Разрабатываемая программная система имеет следующие функции:

- ввод параметров зубчатой передачи для расчета;
- расчет геометрических параметров зубчатых колес и зацепления;
- построение зубчатых колес внешнего зацепления в соответствии с ГОСТ 16532-70;
- организация диалога со списком параметров зубчатого зацепления;
- передача данных о геометрии зубчатого зацепления в тело метода, выполняющего построение;
- загрузка и выгрузка программной системы в память базовой САДсистемы SolidEdge 2020;
- визуализация эскиза профилей колеса и шестерни;
- вызов метода выполняющего построение колес;
- запуск последовательности выполнения SolidEdgeAPI.

3.2 Проектирование алгоритма работы программы

3.2.1 Проектирование оптимальной последовательности построения

Прежде чем приступить к разработке программы, прежде всего, следует уяснить оптимальную последовательность выполнения проектных операций и процедур. Для этого следует проделать построение зубчатых колес в SolidEdgev

режиме ручного моделирования. Построения будем производить для одного из набора конструктивных параметров механической передачи с модулем 5 (табл. 2.1).

Создадим новую деталь SolidEdge и построим выступ расчетным диаметром вершин зубьев колеса, и шириной зубчатого венца.

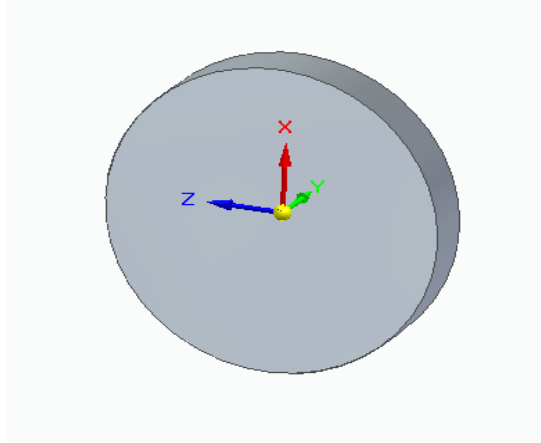


Рисунок 3.1 – Начало построения колеса

Для построения зубьев зубчатых колес целесообразнее всего воспользоваться уже существующей методикой построения эвольвентного профиля зуба, основанного на вычислении координат точек профилей боковой поверхности зуба и переходной кривой. Для построения выреза профиля зуба воспользуемся инструментами кривая и дуга. Таким образом, всего в замкнутом профиле выреза будет 2 кривые составляющие боковую поверхность зуба, 2 кривые производящие переходную кривую и 2 дуги, закрывающие контур диаметрами вершин зубьев и впадин колеса (рис. 3.2).

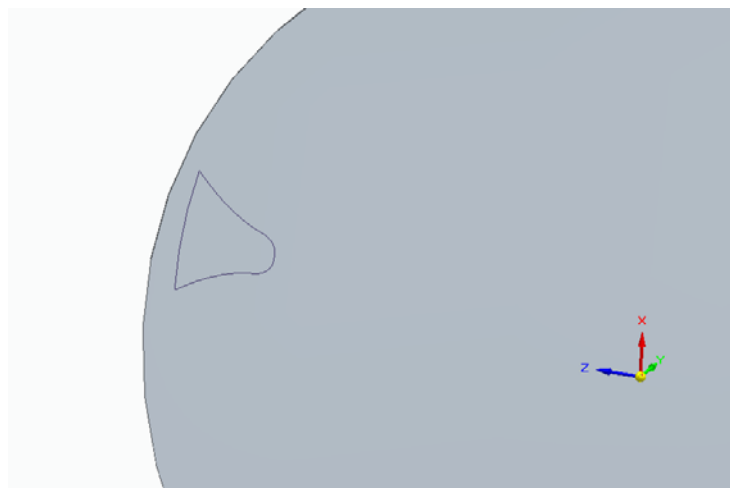


Рисунок 3.2 – Профиль выреза зуба

Вызовем команду «Вырез» и определим его конструктивные характеристики. Укажем построенный контур и зададим нужное направление для построения выреза (рис. 3.3).

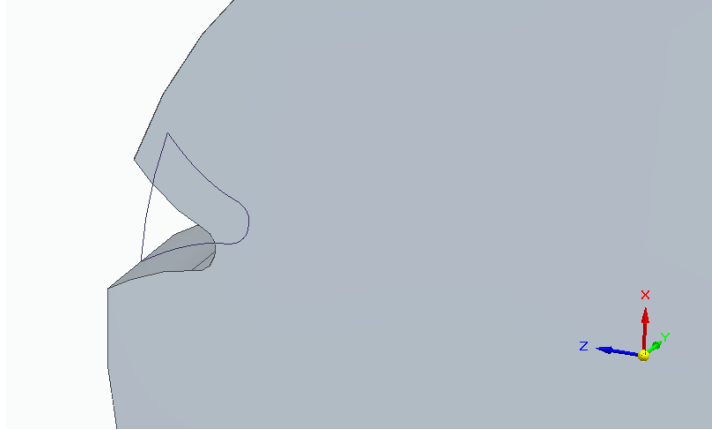


Рисунок 3.3 – Результат команды «Вырез»

Далее создадим круговой массив элементов выреза, с числом элементов, равным числу зубьев колеса.

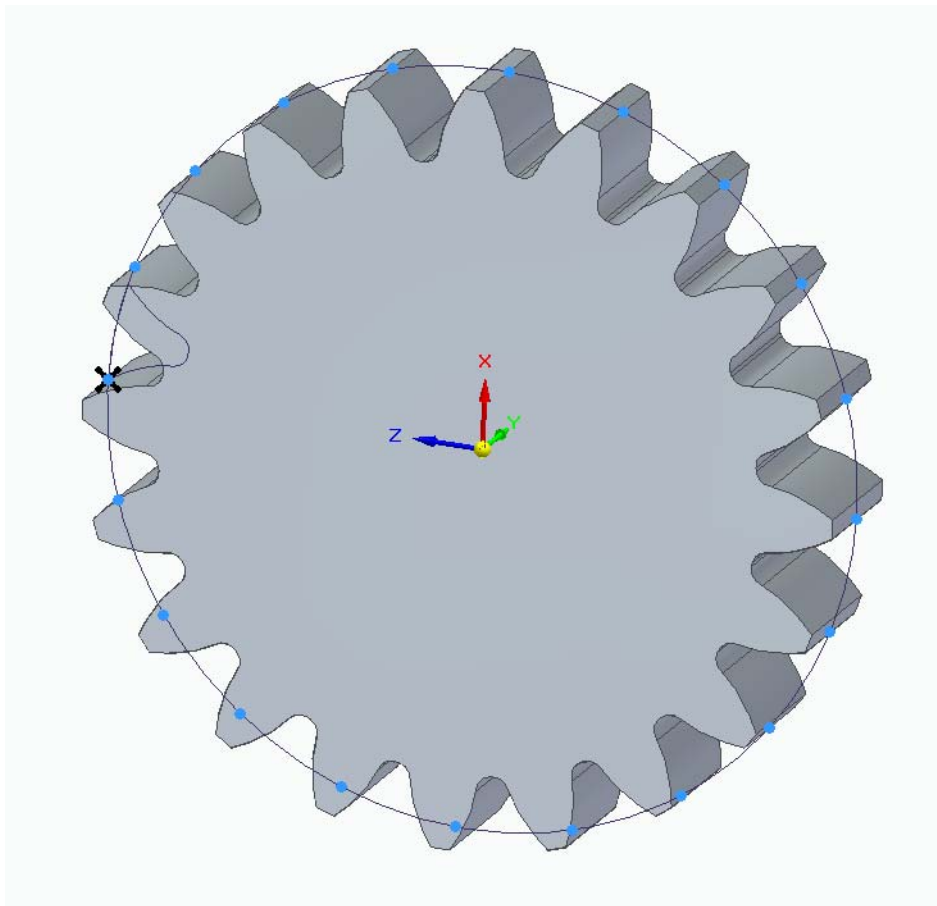


Рисунок 3.4 – Результат команды «Круговой массив»

В результате проделанных операций, был получен оптимальный вариант построения зубчатого колеса и определены основные операции и их последовательность, особенности, которые требуют параметризации.

3.2.2 Проектирование базы данных

Создание базы данных для нашего приложения целесообразнее всего выполнять с использованием XML файла, т.к. данная система хранения не предполагает единого сетевого доступа к данным поскольку для данной задачи в этом просто нет необходимости. Создадим XMLсхему, позволяющую выполнять хранение всех необходимых данных.

```
<?xmlversion="1.0" encoding="utf-8"?>
<xs:schemaattributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:elementname="BaseGearsData">
<xs:complexType>
<xs:sequence>
<xs:elementname="ModuleRows">
<xs:complexType>
<xs:sequence>
<xs:elementmaxOccurs="unbounded" name="ModuleRow">
<xs:complexType>
<xs:attributename="M"type="xs:decimal" use="required" />
<xs:attributename="RowName"type="xs:unsignedByte" use="required" />
<xs:attributename="StandartName"type="xs:string" use="required" />
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:elementname="Rollers">
<xs:complexType>
<xs:sequence>
<xs:elementmaxOccurs="unbounded" name="Roller">
<xs:complexType>
<xs:attributename="D"type="xs:decimal" use="required" />
```

```

<xs:attributename="StandartName" type="xs:string" use="required" />
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:elementname="Tolerances">
<xs:complexTypemixed="true">
<xs:sequence>
<xs:elementmaxOccurs="unbounded" name="TL">
<xs:complexType>
<xs:attributename="Mark" type="xs:string" use="required" />
<xs:attributename="StandartName" type="xs:string" use="required" />
</xs:complexType>
</xs:element>
<xs:elementmaxOccurs="unbounded" name="BZ">
<xs:complexType>
<xs:attributename="Mark" type="xs:string" use="required" />
<xs:attributename="StandartName" type="xs:string" use="required" />
</xs:complexType>
</xs:element>
<xs:elementmaxOccurs="unbounded" name="AW">
<xs:complexType>
<xs:attributename="PairType" type="xs:string" use="required" />
<xs:attributename="MinAW" type="xs:unsignedShort" use="required" />
<xs:attributename="MaxAW" type="xs:unsignedShort" use="required" />
<xs:attributename="Value" type="xs:unsignedShort" use="required" />
<xs:attributename="StandartName" type="xs:string" use="required" />
</xs:complexType>
</xs:element>
<xs:elementmaxOccurs="unbounded" name="Fr">
<xs:complexType>
<xs:attributename="ToleranceLevel" type="xs:unsignedByte" use="required" />
<xs:attributename="MinM" type="xs:decimal" use="required" />
<xs:attributename="MaxM" type="xs:decimal" use="required" />
<xs:attributename="MinD" type="xs:unsignedShort" use="required" />
<xs:attributename="MaxD" type="xs:unsignedShort" use="required" />
<xs:attributename="Value" type="xs:unsignedShort" use="required" />
<xs:attributename="StandartName" type="xs:string" use="required" />
</xs:complexType>
</xs:element>
<xs:elementmaxOccurs="unbounded" name="Ewms1">

```

```

<xs:complexType>
<xs:attributename="PairType"type="xs:string"use="required" />
<xs:attributename="MinToleranceLvl"type="xs:unsignedByte"use="required" />
<xs:attributename="MaxToleranceLvl"type="xs:unsignedByte"use="required" />
<xs:attributename="MinD"type="xs:unsignedShort"use="required" />
<xs:attributename="MaxD"type="xs:unsignedShort"use="required" />
<xs:attributename="Value"type="xs:unsignedShort"use="required" />
<xs:attributename="StandartName"type="xs:string"use="required" />
</xs:complexType>
</xs:element>
<xs:elementmaxOccurs="unbounded"name="Ewms2">
<xs:complexType>
<xs:attributename="MinFr"type="xs:unsignedShort"use="required" />
<xs:attributename="MaxFr"type="xs:unsignedShort"use="required" />
<xs:attributename="Value"type="xs:unsignedByte"use="required" />
<xs:attributename="StandartName"type="xs:string"use="required" />
</xs:complexType>
</xs:element>
<xs:elementmaxOccurs="unbounded"name="Twm">
<xs:complexType>
<xs:attributename="PairType"type="xs:string"use="required" />
<xs:attributename="BZ"type="xs:string"use="required" />
<xs:attributename="MinFr"type="xs:unsignedShort"use="required" />
<xs:attributename="MaxFr"type="xs:unsignedShort"use="required" />
<xs:attributename="Value"type="xs:unsignedShort"use="required" />
<xs:attributename="StandartName"type="xs:string"use="required" />
</xs:complexType>
</xs:element>
<xs:elementmaxOccurs="unbounded"name="Tw">
<xs:complexType>
<xs:attributename="PairType"type="xs:string"use="required" />
<xs:attributename="BZ"type="xs:string"use="required" />
<xs:attributename="MinFr"type="xs:unsignedShort"use="required" />
<xs:attributename="MaxFr"type="xs:unsignedShort"use="required" />
<xs:attributename="Value"type="xs:unsignedShort"use="required" />
<xs:attributename="StandartName"type="xs:string"use="required" />
</xs:complexType>
</xs:element>
<xs:elementmaxOccurs="unbounded"name="Esc">
<xs:complexType>
<xs:attributename="PairType"type="xs:string"use="required" />

```



```

<xs:attributename="MinToleranceLvl" type="xs:unsignedByte" use="required" />
<xs:attributename="MaxToleranceLvl" type="xs:unsignedByte" use="required" />
<xs:attributename="MinD" type="xs:unsignedShort" use="required" />
<xs:attributename="MaxD" type="xs:unsignedShort" use="required" />
<xs:attributename="Value" type="xs:unsignedShort" use="required" />
<xs:attributename="StandartName" type="xs:string" use="required" />
</xs:complexType>
</xs:element>
<xs:elementmaxOccurs="unbounded" name="Tc">
<xs:complexType>
<xs:attributename="PairType" type="xs:string" use="required" />
<xs:attributename="BZ" type="xs:string" use="required" />
<xs:attributename="MinFr" type="xs:unsignedShort" use="required" />
<xs:attributename="MaxFr" type="xs:unsignedShort" use="required" />
<xs:attributename="Value" type="xs:unsignedShort" use="required" />
<xs:attributename="StandartName" type="xs:string" use="required" />
</xs:complexType>
</xs:element>
<xs:elementmaxOccurs="unbounded" name="K1">
<xs:complexType>
<xs:attributename="PairType" type="xs:string" use="required" />
<xs:attributename="MinToleranceLvl" type="xs:unsignedByte" use="required" />
<xs:attributename="MaxToleranceLvl" type="xs:unsignedByte" use="required" />
<xs:attributename="Value" type="xs:decimal" use="required" />
<xs:attributename="StandartName" type="xs:string" use="required" />
</xs:complexType>
</xs:element>
<xs:elementmaxOccurs="unbounded" name="Jnmax">
<xs:complexType>
<xs:attributename="PairType" type="xs:string" use="required" />
<xs:attributename="MinToleranceLvl" type="xs:unsignedByte" use="required" />
<xs:attributename="MaxToleranceLvl" type="xs:unsignedByte" use="required" />
<xs:attributename="MinAw" type="xs:unsignedByte" use="required" />
<xs:attributename="MaxAw" type="xs:unsignedShort" use="required" />
<xs:attributename="MinM" type="xs:decimal" use="required" />
<xs:attributename="MaxM" type="xs:decimal" use="required" />
<xs:attributename="Value" type="xs:unsignedShort" use="required" />

```

```

<xs:attributename="StandartName" type="xs:string" use="required" />
</xs:complexType>
</xs:element>
<xs:elementmaxOccurs="unbounded" name="Jnmin">
<xs:complexType>
<xs:attributename="PairType" type="xs:string" use="required" />
<xs:attributename="MinAw" type="xs:unsignedShort" use="required" />
<xs:attributename="MaxAw" type="xs:unsignedShort" use="required" />
<xs:attributename="Value" type="xs:unsignedByte" use="required" />
<xs:attributename="StandartName" type="xs:string" use="required" />
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attributename="Product" type="xs:string" use="required" />
<xs:attributename="Version" type="xs:unsignedByte" use="required" />
<xs:attributename="Date" type="xs:string" use="required" />
<xs:attributename="Time" type="xs:time" use="required" />
</xs:complexType>
</xs:element>
</xs:schema>

```

НижеданнаясхемапредставленапомощьюинструментаXMLShemaExplorerсред
ыразработкиMSVisualStudio.

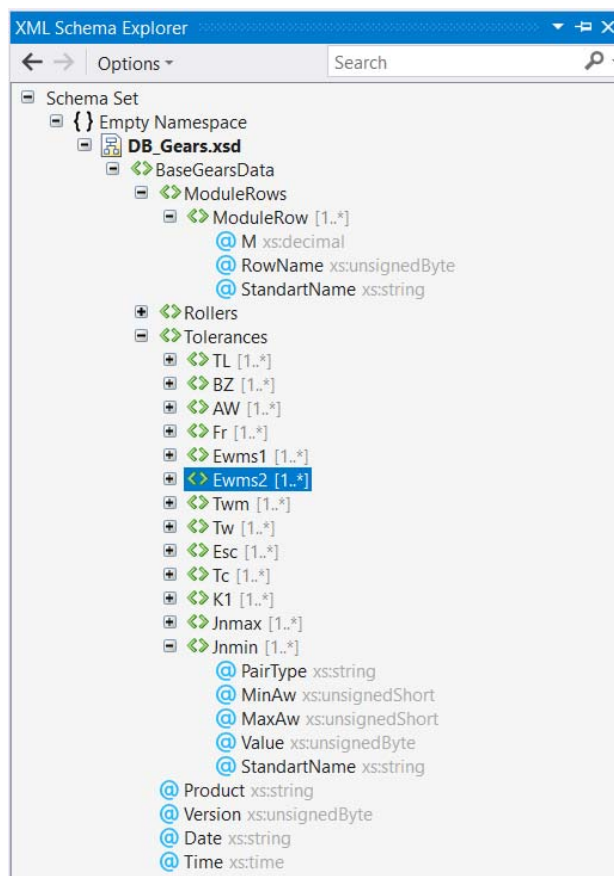


Рисунок 3.4 – Схема данных

3.3 Модульная структура программы

С целью функциональной декомпозиции создаваемого приложения, необходимо создать модульную структуру в древовидной форме. Узлы данного дерева будут содержать программные модули, а направления дуг указывать статическую иерархию модулей, где каждая дуга указывает, что в исходном коде модуля-источника имеется ссылка на указываемый модуль.

Структура плагина SolidEdge построения:

- MyAssembly – пользовательский элемент управления главного окна программы, выполняющий управление ее модулями; обеспечивающий выполнение запросов; их обработку, вызов методов, построение конструктивных элементов, реализующий интерактивность;

- MyAddIn – модуль, отвечающий за регистрацию, создание и инициализацию приложения-клиента системы SolidEdge;

- MyRibbon – модуль, отвечающий за размещения кнопок вызова пользовательского диалога из ленты команд;

- BaseGearsData– модуль-обертка, отвечающий за доступ к данным в файле XML.

- ModuleRow, Rollers, TL, BZ, AW, Fr, Ewms1, Ewms2, Twm, Tw, Esc, Tc, K1, Jnmax, Jnmin – модули-обертки, отвечающие за доступ к конструктивным характеристикам зубчатых колес внешнего зацепления.

- MainForm– модуль, отвечающий за основную форму программы, реализующий ввод исходных данных и вывод результатов расчета.

- CalcInner– модуль, отвечающий за расчет параметров и конструктивных характеристик как отдельных зубчатых колес, так и всей механической передачи внешнего зацепления.

Модульная структура программной системы, обеспечивающей построение зубчатых колес внешнего зацепления по ГОСТ 16532-70 приведена на рисунке 3.7.

Представление программы в виде нескольких отдельных программных модулей имеет ряд преимуществ по сравнению с программой, написанной в виде одного длинного модуля. Каждый программный модуль программируется, компилируется и отлаживается отдельно от других модулей программы, и тем самым, физически разделен с другими модулями программы. Более того, каждый разработанный программный модуль может включаться в состав разных программ, если выполнены условия его использования, декларированные в документации по этому модулю. Таким образом, программный модуль может рассматриваться и как средство борьбы со сложностью программ, и как средство борьбы с дублированием в программировании (т.е. как средство накопления и многократного использования программистских знаний).

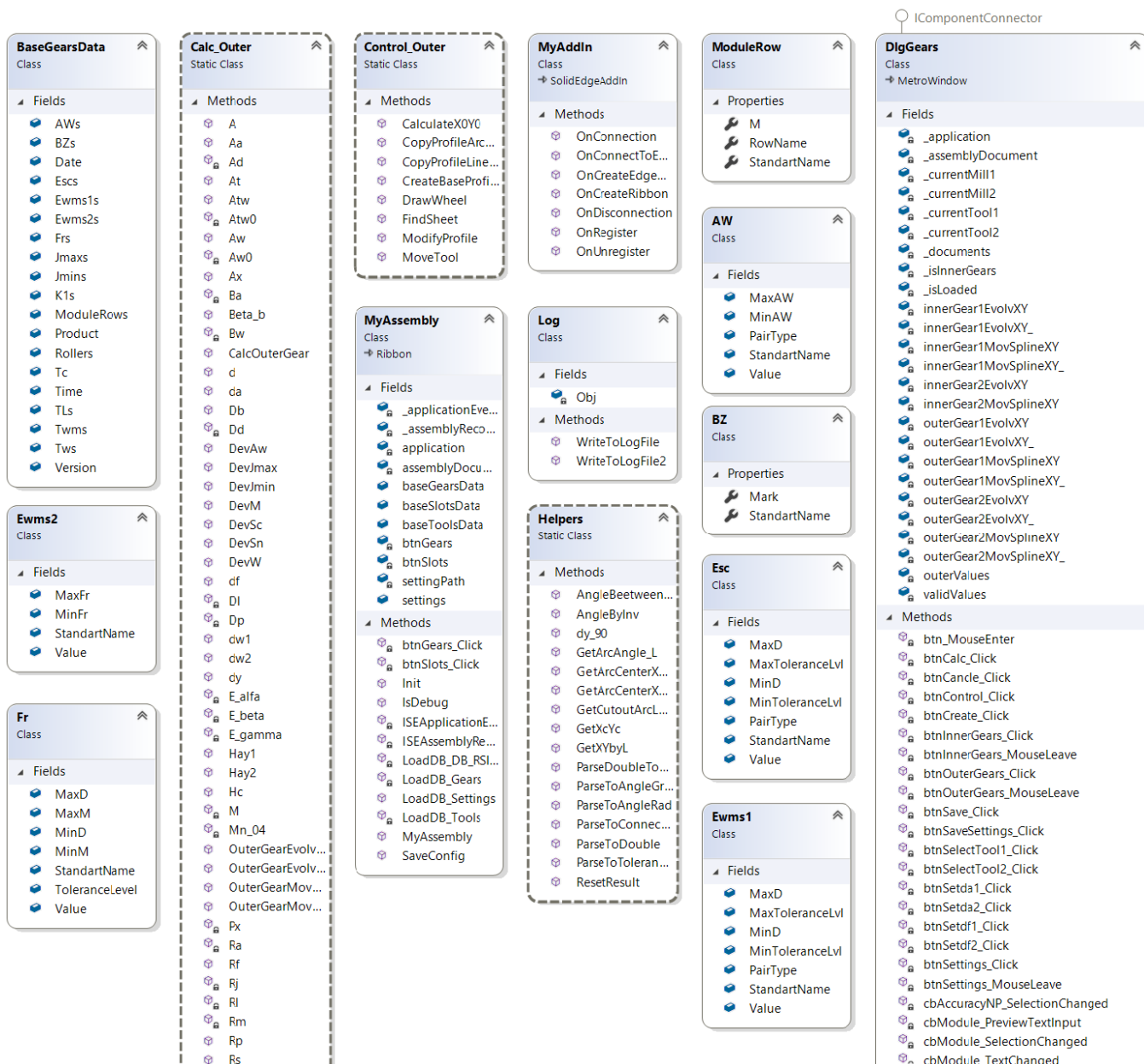


Рисунок 3.5 – Диаграмма классов

3.4 Алгоритмическая структура программной системы

Алгоритмическая структура программной системы по расширению возможностей SolidEdge в части построения зубчатых колес внешнего зацепления по ГОСТ 16532-70, представляется ее алгоритмом, приведенным на рисунке 3.6.

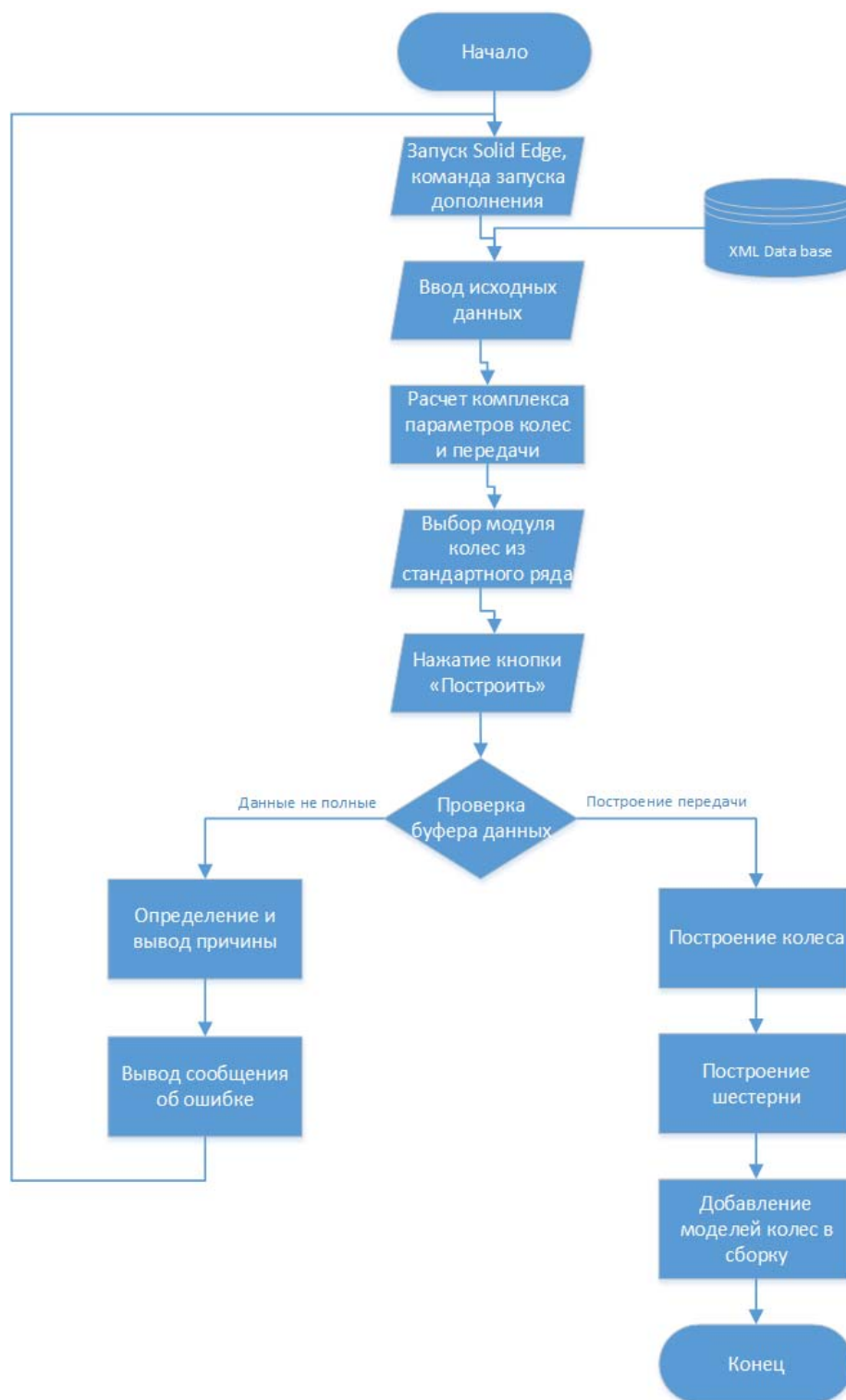


Рисунок 3.6 – Алгоритм

Рассмотрим подробнее алгоритмическую структуру программной системы построения зубчатых колес внешнего зацепления.

- 1) Начало программы.
 - 2) Загрузка формы ввод исходных данных.
 - 3) Загрузка данных из XML-файла.
 - 4) Выбор модуля колеса.
 - 5) Нажатие на кнопку «Расчет».
 - 6) Расчет геометрических параметров механической передачи и проверка передачи по геометрическим параметрам.
 - 7) Нажитие на кнопку «Построение».
 - 8) Проверка буфера данных.
 - 9) Старта процесса построения колес.
 - 10) Сохранение зубчатых колес.
 - 11) Добавление моделей колес в сборку.
 - 12) Обработка исключительных ситуаций.
 - 13) Вывод результата на экран
 - 14) Вывод окна «О программе».
- Конец программы.

4 Реализация подсистемы

4.1 Реализация интерфейса программы

Интерфейс является важнейшим элементом взаимодействия между пользователем и функциональными возможностями информационной системы, с которой он работает. Именно от этого элемента на прямую, зависит эффективность работы всей разрабатываемой системы.

Для реализации всех функциональных возможностей программы и с целью создания дружелюбного интерфейса создадим форму в соответствии с рисунком 4.1.

The image shows a software window titled "Библиотека расчета и построения зубчатых колес внутреннего эвольвентного зацепления ГОСТ 16532-70". It has two tabs: "Параметры" (Parameters) and "Результаты расчета" (Calculation Results). The "Параметры" tab is active and contains a table of input parameters for gear calculation. The parameters are organized into three columns: "Наименование параметра" (Parameter Name), "Сателлит" (Satellite), and "Эпицикл" (Epicycle). The "Сателлит" column has a sub-column for "Модуль" (Module) which is a dropdown menu.

Наименование параметра	Сателлит	Эпицикл
Число зубьев	20	43
Модуль	5	
Угол профиля зуба	20	
Коэффициент высоты головки зуба	1	
Коэффициент радиального зазора	0.25	
Коэффициент радиуса кривизны переходной кривой	0.38	
Ширина зубчатого венца, мм	20	20
Коэффициент смещения исходного контура	0.3	0

At the bottom right of the dialog box is a button labeled "Расчёт" (Calculate).

Рисунок 4.1 – Диалоговое окно программы в режиме конструктора

4.2 Реализация связи приложения с XML данными

Для работы с XMLсхемой, представленной выше, необходимо создание классов-обертки с целью удобного взаимодействия с содержимым XML файла.

Класс BaseGearsData предназначен для работы с общими полями, а также для инициализации других ключевых классов приложения.

```
namespace SEWheelTool.Models
{
public class BaseGearsData
    {
public string Product;
public string Version;
public string Date;
public string Time;

public List<ModuleRow> ModuleRows = new List<ModuleRow>();
public List<Roller> Rollers = new List<Roller>();
public List<TL> TLs = new List<TL>();
public List<AW> AWs = new List<AW>();
public List<Fr> Frs = new List<Fr>();
public List<Ewms1> Ewms1s = new List<Ewms1>();
public List<Ewms2> Ewms2s = new List<Ewms2>();
public List<Twm> Twms = new List<Twm>();
public List<Tw> Tws = new List<Tw>();
public List<Tc> Tc = new List<Tc>();
public List<Esc> Escs = new List<Esc>();
public List<K1> K1s = new List<K1>();
public List<BZ> BZs = new List<BZ>();
public List<Jmax> Jmaxs = new List<Jmax>();
public List<Jmin> Jmins = new List<Jmin>();
    }
}
```

Класс MyAssembly организует работу плагина в системе SolidEdge, обеспечивает загрузку данных из XML, а также взаимодействие окон плагина и SolidEdge.

```
namespace SEWheelTool
{
class MyAssembly : SolidEdgeCommunity.AddIn.Ribbon
{
public static bool IsDebug()
{
#if DEBUG
return true;
#else
return false;
#endif
}

private RibbonButton btnGears;
private RibbonButton btnSlots;

private SolidEdgeFramework.Application application = null;
private AssemblyDocument assemblyDocument = null;

private ISEAssemblyRecomputeEvents_Event _assemblyRecomputeEvents;
private ISEApplicationEvents_Event _applicationEvents;

public static BaseGearsData baseGearsData = null;

private static string settingPath = "";
```

```

public MyAssembly()
    {
var assembly = System.Reflection.Assembly.GetExecutingAssembly();
var _embeddedResourceName = String.Format("{0}.xml", this.GetType().FullName);
this.LoadXml(assembly, _embeddedResourceName);

string curAssemblyPath = System.IO.Path.GetDirectoryName(assembly.Location);

btnGears = GetButton(1);
        btnGears.Click += btnGears_Click;

Init();

baseGearsData = new BaseGearsData();

var db_GearsPath = String.Concat(curAssemblyPath, @"\db\", @"\ ", "DB_Gears.xml");

if (File.Exists(db_GearsPath))
        LoadDB_Gears(db_GearsPath);
    }

private void btnGears_Click(RibbonControl control)
    {
try
        {
var dlg = new MainForm(application, assemblyDocument);
dlg.ShowDialog();
        }
catch (Exception ex)

```

```

    {
Log.WriteToFile(ex, "btnGears_Click");
    }
}

/// <summary>
/// Загрузка БД зубчатых передач
/// </summary>
/// <param name="fullNameBDXml"></param>
private void LoadDB_Gears(string fullNameBDXml)
    {
try
    {
using (XmlReader r = XmlReader.Create(new StreamReader(fullNameBDXml,
Encoding.UTF8)))
    {
r.ReadToFollowing("BaseGearsData");

        baseGearsData.Product = r.GetAttribute("Product");
        baseGearsData.Version = r.GetAttribute("Version");
        baseGearsData.Date = r.GetAttribute("Date");
        baseGearsData.Time = r.GetAttribute("Time");

while (r.Read())
    {
if (r.Name == "ModuleRow")
    {
ModuleRow moduleRow = new ModuleRow();
moduleRow.M = r.GetAttribute("M");
moduleRow.RowName = r.GetAttribute("RowName");

```

```
moduleRow.StandartName = r.GetAttribute("StandartName");
```

```
baseGearsData.ModuleRows.Add(moduleRow);
```

```
}
```

```
else if (r.Name == "Roller")
```

```
{
```

```
Roller roller = new Roller();
```

```
roller.D = r.GetAttribute("D");
```

```
roller.StandartName = r.GetAttribute("StandartName");
```

```
baseGearsData.Rollers.Add(roller);
```

```
}
```

```
else if (r.Name == "TL")
```

```
{
```

```
TL tl = new TL();
```

```
tl.Mark = r.GetAttribute("Mark");
```

```
tl.StandartName = r.GetAttribute("StandartName");
```

```
baseGearsData.TLs.Add(tl);
```

```
}
```

```
else if (r.Name == "BZ")
```

```
{
```

```
BZ bz = new BZ();
```

```
bz.Mark = r.GetAttribute("Mark");
```

```
bz.StandartName = r.GetAttribute("StandartName");
```

```
baseGearsData.BZs.Add(bz);
```

```
}
```

```
.....
```

```
}
```

Данный метод, позволяет осуществить вычисление координат эвольвенты зуба колеса, исходя из конструктивных характеристик, полученных в расчетном методе.

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

```
    /// Вычисление координат эвольвенты шестерни (фреза)
```

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

```
    /// <param name="db1"></param>
```

```
    /// <param name="d1"></param>
```

```
    /// <param name="da1"></param>
```

```
    /// <param name="dl1"></param>
```

```
    /// <param name="S1"></param>
```

```
    /// <param name="alfa"></param>
```

```
    /// <param name="tolerance"></param>
```

```
    /// <returns></returns>
```

```
public static List<double[]> InnerGear1EvolvXY(double db1/*основнойдиаметр*/,  
double d1/*делительныйдиаметр*/, double da1, double dl1, double  
Sn1/*толщиназуба*/, double alfa, int tolerance)
```

```
{
```

```
    List<double[]> XYList = new List<double[]>();
```

```
double _A_angle, dy1_min, dy1_max, Delta_dy1, invA, dy1,
```

```
    Hl1, Ay1, invAy1, Xe, Ye;
```

```
    _A_angle = alfa * Math.PI / 180;
```

```
invA = Math.Tan(_A_angle) - _A_angle;
```

```
dy1_min = dl1;
```

```
    dy1_max = da1;
```

```
    Delta_dy1 = (dy1_max - dy1_min) / tolerance;
```

```
    dy1 = dy1_min;
```

$H1b1 = S1 / d1 - invA;$

```
for (int i = 0; i <= tolerance; i++)
    {
        Ay1 = Math.Acos(db1 / dy1);
        invAy1 = Math.Tan(Ay1) - Ay1;
        Xe = 0.5 * dy1 * Math.Sin(H1b1 - invAy1);
        Ye = 0.5 * dy1 * Math.Cos(H1b1 - invAy1);

        dy1 = dy1 + Delta_dy1;
        XYList.Add(new double[] { Xe, Ye });
    }
return XYList;
}
```

4.3 Реализация набора методов, обеспечивающих построение зубчатых колес внешнего зацепления

Данный метод, позволяет исходя из конструктивных параметров зубчатого колеса, получить набор координат точек переходной кривой.

```
/// <summary>
    /// Вычисление координат переходной кривой колеса
/// </summary>
    /// <param name="df">диаметр впадин колеса</param>
    /// <param name="d2">делительный диаметр</param>
    /// <param name="S2">толщина зуба</param>
    /// <param name="M"></param>
    /// <param name="Z2"></param>
```

```

/// <param name="Z0"></param>
/// <param name="X2"></param>
/// <param name="X0"></param>
/// <param name="alfa"></param>
/// <param name="tolerance"></param>
/// <returns></returns>

public static List<double[]> InnerGear2MovSplineXY(double df, double d2, double
S2, double M, double Z2, double Z0, double X2, double alfa, int tolerance)
{
    List<double[]> XYList = new List<double[]>();
double _A_angle, Aw02, Muw0, Muw0r, Delta_Muw0, Muy, Muy0, gy0, invA,
    Muw0_max, Muw0_min, HIb2, dw0, bd0, Ad0, by, dy, Xe, Ye;

    _A_angle = alfa * Math.PI / 180;

double X0 = (2 * S2 - Math.PI * M) / (4 * M * Math.Tan(_A_angle));

invA = Math.Tan(_A_angle) - _A_angle;
    Aw02 = Helpers.AngleByInv(2 * (X2 - X0) * Math.Tan(_A_angle) / (Z2 - Z0) +
invA);

double Aw02r = Aw02 * Math.PI / 180;

    Muw0_min = Aw02;
    Muw0_max = 90;
    Delta_Muw0 = (Muw0_max - Muw0_min) / tolerance;
    Muw0 = Muw0_min;

    HIb2 = S2 / d2 - invA;

```



```
Ad0 = Math.Acos(M * Z0 * Math.Cos(_A_angle) / df);
```

```
bd0 = Math.Tan(Ad0) - Ad0;
```

```
dw0 = M * Z0 * Math.Cos(_A_angle) / Math.Cos(Aw02r);
```

```
while (Muw0 <= Muw0_max)
```

```
{
```

```
    Muw0r = Muw0 * Math.PI / 180;
```

```
if (Muw0 < 90)
```

```
    gy0 = Math.Acos(dw0 * Math.Cos(Muw0r) / df);
```

```
else
```

```
    gy0 = 90 * Math.PI / 180;
```

```
    Muy0 = Math.Atan(Math.Tan(gy0));
```

```
if (Muw0 < 90)
```

```
    Muy = Math.Atan((Z2 - Z0) * Math.Tan(Muw0r) / Z2 + Z0 *  
Math.Tan(Muy0) / Z2);
```

```
else
```

```
    Muy = 90 * Math.PI / 180;
```

```
if (Muw0 < 90)
```

```
dy = M * Z2 * Math.Cos(_A_angle) * Math.Cos(Muw0r) / (Math.Cos(Aw02r) *  
Math.Cos(Muy));
```

```
else
```

```
dy = df;
```

```

double InvAw02 = Math.Tan(Aw02r) - Aw02r;

if (Muw0 < 90)
by = (Z2 - Z0) * (Muw0r + InvAw02) / Z2 + Z0 * (bd0 + gy0) / Z2 - Muy;
else
by = (Z2 - Z0) * InvAw02 / Z2 + Z0 * bd0 / Z2;

        Xe = 0.5 * dy * Math.Sin(HIb2 + by);
        Ye = 0.5 * dy * Math.Cos(HIb2 + by);

XYList.Add(new double[] { Xe, Ye });
if (Muw0 + Delta_Muw0 > 90)
        Muw0 = Math.Round(Muw0 + Delta_Muw0);
else
        Muw0 = Math.Round(Muw0 + Delta_Muw0, 6);
}

return XYList;
}

```

4.5 Тестирование программы

С целью проверки программного средства на наличие в нем ошибок, функциональных недоработок, и критических несоответствий следует провести процесс тестирования.

Безошибочность работы программы и его функциональность в общем зависят от оптимальности выбранного способа построения конструктивных элементов, организации самого приложения, стабильности передачи информации основному приложению, от качества самой программы. Необходимо провести создание

механической передачи внешнего зацепления с заданными параметрами стандартными средствами CAD-системы SolidEdge, запомнить полученные результаты и выполнить ту же функцию с использованием разработанного приложения. После чего необходимо сравнить полученные результаты и сделать вывод о правильности функционирования программы.

В процессе тестирования проводилась проверка на наличие возможности появления сбоев в работе программы при некорректных действиях пользователя.

Для проверки работоспособности приложения выполним запуск программы из интерфейса CAD-системы SolidEdge, и выполним построение механической передачи. В результате данного действия в CAD-системе SolidEdge в среде сборки должны появиться колеса и шестерня. Работа созданного приложения представлена на рисунках 4.2-4.5.

Наименование параметра	Колесо	Шестерня
Число зубьев	<input type="text" value="20"/>	<input type="text" value="43"/>
Модуль	<input type="text" value="1"/>	
Угол профиля зуба	<input type="text" value="20"/>	
Коэффициент высоты головки зуба	<input type="text" value="1"/>	
Коэффициент радиального зазора	<input type="text" value="0.25"/>	
Коэффициент радиуса кривизны переходной кривой	<input type="text" value="0.38"/>	
Ширина зубчатого венца, мм	<input type="text" value="20"/>	<input type="text" value="20"/>
Коэффициент смещения исходного контура	<input type="text" value="0.3"/>	<input type="text" value="0"/>

Рисунок 4.2 – Вид диалогового окна после запуска (закладка Параметры)

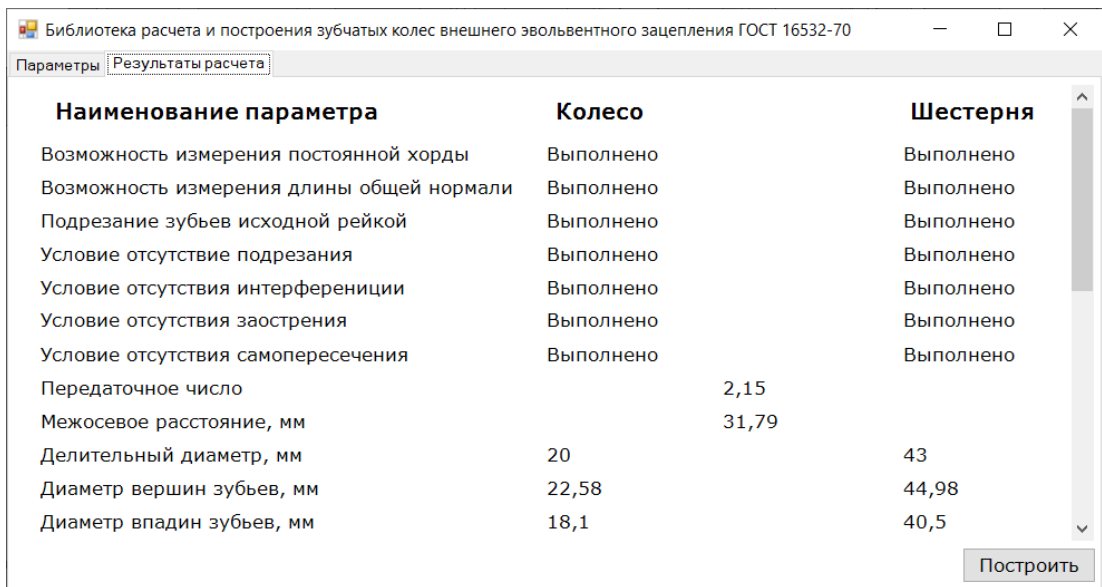


Рисунок 4.3 – Вид диалогового окна с результатами расчета и проверки колес по геометрическим параметрам (закладка Результаты расчета)

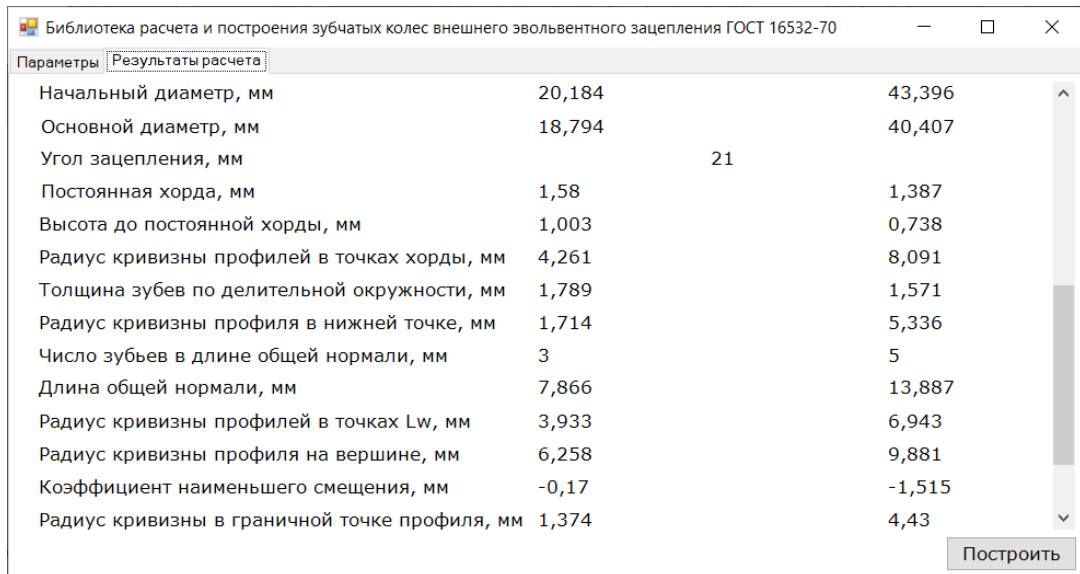


Рисунок 4.4 – Вид диалогового окна с результатами расчета и проверки колес по геометрическим параметрам (закладка Результаты расчета - продолжение)

Наименование параметра	Колесо	Шестерня
Возможность измерения постоянной хорды	Выполнено	Выполнено
Возможность измерения длины общей нормали	Выполнено	Выполнено
Подрезание зубьев исходной рейкой	Не выполнено	Выполнено
Условие отсутствие подрезания	Не выполнено	Выполнено
Условие отсутствия интерференции	Выполнено	Выполнено
Условие отсутствия заострения	Выполнено	Выполнено
Условие отсутствия самопересечения	Выполнено	Выполнено
Передаточное число	2	
Межосевое расстояние, мм	15,281	
Делительный диаметр, мм	10	20
Диаметр вершин зубьев, мм	12,563	21,963
Диаметр впадин зубьев, мм	8,1	17,5

Рисунок 4.5 – Вид диалогового окна с результатами расчета и проверки колес по геометрическим параметрам (проверка не пройдена)

Таким образом, в ходе тестирования разработанного программного средства, было определено, его правильное функционирование на ряде контрольных примеров. Более того, была также проверена корректность его работы в условиях беспорядочного и хаотичного использования всех возможных элементов управления одновременно. Даже такая эксплуатация никогда не приводила к фатальным сбоям.

4.6 Виды обеспечений

4.6.1 Программное обеспечение

Программное обеспечение (ПО) – это множество машинных программ, которые необходимы для выполнения автоматизированного проектирования.

При разработке ПО были использованы следующие программные средства:

- OS MSWindows10;
- MSOfficeVisio 2013, Word 2013;
- SolidEdge2019.

Microsoft Visual Studio 2015 это полный набор средств разработки для создания веб-приложений ASP.NET, XML (веб-службы), настольных и мобильных приложений.

Visual Basic, Visual C# и Visual C++ используют единую среду разработки (IDE), с помощью которой появляется возможность совместно использовать средства и упрощать создание решений на базе нескольких языков. В данных языках используются функциональные возможности платформы .NET Framework 4.5, с помощью которой открывается доступ к ключевым технологиям, упрощающим разработку веб-службы.

Visual Studio содержит в себе редактор исходного кода с поддержкой технологии IntelliSense и возможностью простого преобразования кода. Встроенный отладчик может быть использован как отладчик уровня исходного кода. Другие встраиваемые инструменты включают в себя редактор форм для упрощения создания графического функционального объекта приложений, веб-редактор, дизайнер классов и схем баз данных. С помощью Visual Studio удастся создать и подключить сторонние дополнения для расширения функциональности на каждом уровне, добавить поддержку систем контроля версий исходного кода, добавить новый набор инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования).

4.6.2 Техническое обеспечение

Средства программной обработки данных представляют собой процессоры и запоминающие устройства, устройства ЭВМ, с помощью которых удается реализовать данные и программное управление. Средства подготовки, ввода, отображения и документирования данных используются для работы пользователя с ЭВМ. Средства архива проектных решений представляют собой внешние запоминающие устройства.

Техническое обеспечение САПР — это совокупность взаимосвязанных и взаимодействующих технических средств, служащих для выполнения автоматизированного проектирования. Техническое обеспечение можно разделить на группы программной обработки данных, подготовки и ввода данных,

отображения и документирования, архивации проектных решений, передачи данных.

Для работы системы необходим минимальный набор программного обеспечения:

Windows 7/8/10;

.Net Framework 4.5 и выше;

В компонентах используются аппаратные ресурсы. Каждый компонент имеет свои требования. Самый ресурсоёмкий компонент программного обеспечения - операционная система.

В качестве операционной системы мы используем Microsoft Windows 7. Microsoft Windows 7 имеет аппаратные требования:

- 32-разрядный (x86) или 64-разрядный (x64) процессор с тактовой частотой 1 гигагерц (ГГц) или более;
- 1 гигабайт (ГБ) (для 32-разрядной системы) или 2 ГБ (для 64-разрядной системы) оперативной памяти (ОЗУ);
- 16 гигабайт (ГБ) (для 32-разрядной системы) или 20 ГБ (для 64-разрядной системы) места на жестком диске;
- графическое устройство DirectX 9 с использованием драйвера WDDM версии 1.0 или выше.

4.6.3 Лингвистическое обеспечение

Лингвистическое обеспечение это совокупность представленных в определенной форме языков (термины, определения, правила формализации языка, методы сжатия и развертывания языков), которые необходимы для автоматизированного проектирования. Языки, использующиеся в вычислительной технике - алгоритмические. Они необходимы для задания алгоритмов переработки информации и построены в качестве набора символов и системы правил

определенного языка. В информационных технологиях применяется множество языков программирования для работы пользователя с ЭВМ.

Для реализации информационной подсистемы был использован язык программирования C#. Этот язык имеет огромную функциональность. Язык был создан для быстрой разработки сложных приложений. Он сочетает в себе лучшие черты других языков, таких как Java и C++, C# подходит для разработки библиотеки, он обладает очень высокой производительностью и надежностью.

К числу одним из важных решений, которые реализованы корпорацией Microsoft в языке программирования C#, относятся следующие:

- компонентно-ориентированный подход к программированию (характерно для идеологии Microsoft .NET);
- свойства инкапсуляции данных (характерно для объектно-ориентированного программирования);
- обработка событий (имеются расширения, обработки исключений, оператор try);
- модульная система типизации (соответствует идеологии Microsoft .NET);
- делегаты (развитие указателя на функцию в языках C и C++);
- индексаторы (операторы индекса для обращения к элементам класса-контейнера);
- развитие объектно-ориентированного программирования
- оператор foreach (аналог Visual Basic);
- механизмы boxing, unboxing используются для преобразования типов;
- инструменты (средство оперирования метаданными в COM-модели);
- прямоугольные массивы

4.6.4 Информационное обеспечение

Информационное обеспечение (ИО) – представляют собой сведения, необходимые для реализации автоматизированного проектирования и представления результатов в заданной форме.

Информационное обеспечение это множество различных документов, описывающих проектные процедуры, решения, элементы и комплектующие изделия, материалы и другие данные.

Основная задача информационного обеспечения это достоверное и надежное, своевременное поступление всей необходимой информации при минимальном затрачиваемом времени и наименьших затратах на создание и эксплуатацию системы.

В ИО входят: справочные данные об элементах, ГОСТ, сведения о проектных решениях, результаты предыдущих этапов проектирования и т.п.

В качестве информационного обеспечения проекта применяется база данных с типоразмерами отклонений размеров по ГОСТ 16532-70и рядами зубчатых колес в виде XMLфайла, схема которого и реализация взаимодействия с которым, представлены в соответствующих разделах. Ниже представлен фрагмент XMLфайла с рядами модулей зубчатых колес.

```
<?xml version="1.0" encoding="UTF8"?>
<BaseGearsData Product="Gears Bilder" Version="1" Date="04.12.2019"
Time="11:26:04">
  <ModuleRows>
    <ModuleRow M="1" RowName="1" StandartName="ГОСТ 9563-60"/>
    <ModuleRow M="1.25" RowName="1" StandartName="ГОСТ 9563-60"/>
    <ModuleRow M="1.5" RowName="1" StandartName="ГОСТ 9563-60"/>
    <ModuleRow M="2" RowName="1" StandartName="ГОСТ 9563-60"/>
    <ModuleRow M="2.5" RowName="1" StandartName="ГОСТ 9563-60"/>
    <ModuleRow M="3" RowName="1" StandartName="ГОСТ 9563-60"/>
    <ModuleRow M="4" RowName="1" StandartName="ГОСТ 9563-60"/>
    <ModuleRow M="5" RowName="1" StandartName="ГОСТ 9563-60"/>
  </ModuleRows>
</BaseGearsData>
```

<ModuleRow M="6" RowName="1" StandartName="ГOCT 9563-60"/>
<ModuleRow M="8" RowName="1" StandartName="ГOCT 9563-60"/>
<ModuleRow M="10" RowName="1" StandartName="ГOCT 9563-60"/>
<ModuleRow M="12" RowName="1" StandartName="ГOCT 9563-60"/>
<ModuleRow M="16" RowName="1" StandartName="ГOCT 9563-60"/>
<ModuleRow M="20" RowName="1" StandartName="ГOCT 9563-60"/>
<ModuleRow M="25" RowName="1" StandartName="ГOCT 9563-60"/>
<ModuleRow M="32" RowName="1" StandartName="ГOCT 9563-60"/>
<ModuleRow M="40" RowName="1" StandartName="ГOCT 9563-60"/>
<ModuleRow M="50" RowName="1" StandartName="ГOCT 9563-60"/>
<ModuleRow M="60" RowName="1" StandartName="ГOCT 9563-60"/>
<ModuleRow M="80" RowName="1" StandartName="ГOCT 9563-60"/>
<ModuleRow M="100" RowName="1" StandartName="ГOCT 9563-60"/>
<ModuleRow M="1.125" RowName="2" StandartName="ГOCT 9563-60"/>
<ModuleRow M="1.375" RowName="2" StandartName="ГOCT 9563-60"/>
<ModuleRow M="1.75" RowName="2" StandartName="ГOCT 9563-60"/>
<ModuleRow M="2.25" RowName="2" StandartName="ГOCT 9563-60"/>
<ModuleRow M="2.75" RowName="2" StandartName="ГOCT 9563-60"/>
<ModuleRow M="3.5" RowName="2" StandartName="ГOCT 9563-60"/>
<ModuleRow M="4.5" RowName="2" StandartName="ГOCT 9563-60"/>
<ModuleRow M="5.5" RowName="2" StandartName="ГOCT 9563-60"/>
<ModuleRow M="7" RowName="2" StandartName="ГOCT 9563-60"/>
<ModuleRow M="9" RowName="2" StandartName="ГOCT 9563-60"/>
<ModuleRow M="11" RowName="2" StandartName="ГOCT 9563-60"/>
<ModuleRow M="14" RowName="2" StandartName="ГOCT 9563-60"/>
<ModuleRow M="18" RowName="2" StandartName="ГOCT 9563-60"/>
<ModuleRow M="22" RowName="2" StandartName="ГOCT 9563-60"/>
<ModuleRow M="28" RowName="2" StandartName="ГOCT 9563-60"/>
<ModuleRow M="36" RowName="2" StandartName="ГOCT 9563-60"/>

```
<ModuleRow M="45" RowName="2" StandartName="ГОСТ 9563-60"/>
<ModuleRow M="55" RowName="2" StandartName="ГОСТ 9563-60"/>
<ModuleRow M="70" RowName="2" StandartName="ГОСТ 9563-60"/>
<ModuleRow M="90" RowName="2" StandartName="ГОСТ 9563-60"/>
</ModuleRows>
</BaseGearsData>
```

4.6.5 Математическое обеспечение

Математическое обеспечение (МО) – представляет собой множество математических методов, моделей и алгоритмов проектирования, которые используются для выполнения автоматизированного проектирования ТП. В математическое обеспечение входят модели конкретных объектов, методы их проектирования, методы и алгоритмы выполнения различных проектных операций, связанных с оптимизацией, поиском информации.

За основу математического обеспечения взяты алгоритмы, с помощью которых разрабатывается программное обеспечение и осуществляется процесс автоматизирования проектирования информационных систем. Математическое обеспечение при автоматизированном проектировании не используется, а применяется его компонент – программное обеспечение.

МО включает методы и алгоритмы, которые не связаны с особенностями математических моделей они используются при решении множества задач проектирования. Такие методы и алгоритмы многовариантного анализа и параметрической оптимизации. Проектируемые программные модули используют математические, логические операторы. (операторы сравнения \lt , \leq , \geq , операторы булевой алгебры AND, OR, NOT)

Математическое Обеспечение выделяет специфику объекта проектирования, физические, информационные особенности функционирования. Эта часть

охватывает математические модели, методы, алгоритмы получения, огромную важную часть используемых алгоритмов синтеза.

Математическое обеспечение разработанного модуля, представлено набором расчетов позиционирования образующего отрезка на касательной плоскости.

4.6.6 Методическое обеспечение

Методическое обеспечение – представляет собой множество документов, определяющих состав, правила отбора и эксплуатации средств обеспечения автоматизированного проектирования, которые используются для решения проектных задач.

К компонентам методического обеспечения можно отнести:

- утверждение документации инструктивно-методического характера, определение порядка работ по автоматизированному проектированию технологической документации;

- правила использования ПМ комплекса;

- нормативы, стандарты, документы, регламентирующие процесс и объект проектирования.

Методическое обеспечение программы включает стандарты, приведенные в списке литературы к дипломному проекту.

Методическое обеспечение включает руководство пользователя (по ГОСТ 19.505) и руководство разработчика (по ГОСТ 19.504).

Руководство пользователя

1) Назначение программы

Назначение программы – построение зубчатых колес внешнего зацепления по ГОСТ 16532-70.

2) Процесс выполнения программы

В ПК должны входить следующих программные средства:

- операционная система Microsoft Windows 8/7;

- платформа Microsoft .NET Framework 4.5;

ПК, на который устанавливается программа, должен следовать характеристикам:

- процессор Intel с тактовой частотой 2,0 ГГц;

- ОЗУ 2 Гбайт;

- жесткий диск 250 Гбайт.

Состав периферийных устройств

- монитор 17";

- клавиатура, мышь

3) Выполнение программы

Руководство для пользователя - описание процесса построения механической передачи с применением программы рассмотрено в разделе тестирования. Следуя данной последовательности действий пользователь может запустить программу и выполнить построение.

Руководство разработчика

Руководство по реализации программного средства представлено в п. 4 данного дипломного проекта.

Заключение

Целью выпускной квалификационной работы была разработка программной системы осуществляющей расширение функционала системы SolidEdgeв части построения зубчатых колес внешнего зацепления в соответствии с ГОСТ 16532-70.

При выполнении проекта было решено:

1. Создание приложения в виде дополнения к системе Solid Edge.
2. Создание интерфейса программы.
3. Организация и наполнение содержимым базы данных с типоразмерами конструктивных элементов в виде XML файла.
4. Создание алгоритма автоматического расчета и построения механической передачи в соответствии с выбранными расчетными характеристиками.
5. Создание алгоритма проверки конструкции по геометрическим параметрам.
6. Выполнено тестирование разработанного модуля и доказано его правильное и надежное функционирование.

Список литературы

- 1 Робинсон С. С# для профессионалов: в 2 т. / С. Робинсон, О. Корнес, Д. Глинн и др. – М.: Изд-во «ЛОРИ», 2018. – 1002 с.
- 2 Уотсон К. Visual C# 2012: полный курс / К. Уотсон, К. Нейгел, Я.Х. Педерсен и др. – М.: ООО «И.Д. Вильямс», 2021. – 960 с.
- 3 Гарольд Э. XML. Справочник / Э. Гарольд, С. Минс. – СПб.: Символ-Плюс, 2022. – 576 с.
- 4 Обзор системы Teamcenter, 2017. – 20 с.
- 5 Пауэрс Л. Microsoft Visual Studio 2008 / Л. Пауэрс, М. Снелл. – СПб.: БХВ-Петербург, 2019. – 1200 с.
- 6 Раттц-мл. Д. С. LINQ: язык интегрированных запросов в С# 2008 для профессионалов / Д.С. Раттц-мл. – М.: ООО «И.Д. Вильямс», 2018. – 560 с.
- 7 Тороп Д.Н. Teamcenter. Начало работы / Д.Н. Тороп, В.В. Терликов. – М.: ДМК Пресс, 2021. – 280 с.
- 8 Фролов А.В. Язык С#. Самоучитель / А.В. Фролов, Г.В. Фролов. – М.: ДИАЛОГ-МИФИ, 2016. – 560 с.
- 9 Хабибуллин И.Ш. Самоучитель XML / И.Ш. Хабибуллин. – СПб.: БХВ-Петербург, 2021. – 336 с.
- 10 Шумаков П.В. ADO .NET и создание приложений баз данных в среде Microsoft Visual Studio .NET. Руководство разработчика с примерами на С# / П.В. Шумаков. – М.: ДИАЛОГ-МИФИ, 2020. – 528 с.
- 11 White Paper. Open product lifecycle data sharing using XML, 2011. – 14 с.
- 12 Teamcenter для поддержки процессов технологической подготовки производства Введение в TeamcenterManufacturing, 2019. – 41 с.
- 13 ГОСТ 16532-70. Соединения шпоночные с призматическими шпонками. Размеры шпонок и сечений зубчатых колес внешнего зацепления. Допуски и посадки – М.: ИПК Издательство стандартов, 2019. – 9 с.

14 ГОСТ 19.505-79. Единая система программной документации. Руководство оператора. Требования к содержанию и оформлению. – М.: ИПК Издательство стандартов, 1979. – 3 с.

15 Электронный каталог #define VISUAL_STUDIO. – Электрон. дан. – Режим доступа: [http://msdn.microsoft.com/ru-ru/library/vstudio/fx6bk1f4\(v=vs.100\).aspx](http://msdn.microsoft.com/ru-ru/library/vstudio/fx6bk1f4(v=vs.100).aspx).

16 Электронный каталог ADSCProect. – Электрон. дан. – Режим доступа: http://adsc.ru/linq_to_xml.

17 Электронный каталог Bourabai Research. – Электрон. дан. – Режим доступа: <http://bourabai.kz/dhtml/index.html>.

18 Электронный каталог Fast Reports. – Электрон. дан. – Режим доступа: <http://www.fast-report.com/ru/product/fast-report-net/>

19 Электронный каталог MSDN. – Электрон. дан. – Режим доступа: <http://msdn.microsoft.com/ru-ru/library/bb943852.aspx>.

20 Электронный каталог MSDN. – Электрон. дан. – Режим доступа: <http://msdn.microsoft.com/ru-ru/library/bb943914.aspx>.

21 Электронный каталог NSlabs. – Электрон. дан. – Режим доступа: <http://www.nslabs.ru/info/sapr/teamcenter/print/>

22 Электронный каталог OSzone.net. – Электрон. дан. – Режим доступа: <http://www.oszone.net/3789>.

23 Электронный каталог PLMXMLSchema. – Электрон. дан. – Режим доступа: <http://www.plm.automation.siemens.com/legacy/products/open/plmxml/docs/v2/documentation/plmxml.html>

24 Электронный каталог Softkey.info. – Электрон. дан. – Режим доступа: <http://www.softkey.info/reviews/review6353.php>.

25 Электронный каталог Wikipedia. – Электрон. дан. – Режим доступа: http://ru.wikipedia.org/wiki/XML_Schema.

26 Электронный каталог «Клуб программистов». – Электрон. дан. – Режим доступа: <http://club.shelek.ru/viewart.php?id=182>.