

МИНИСТЕРСТВО НАУКИ И ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Рязанский государственный радиотехнический университет
им. В.Ф. Уткина»

Кафедра космических технологий

«К защите»
Заведующий кафедрой
д.т.н., профессор
_____ С.И. Гусев
« ____ » _____ 20 ____ г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

на тему

**«Разработка математического и программного обеспечения генерации
случайных тестовых заданий»**

Направление подготовки: 02.03.01 «Математика и компьютерные науки»
ОПОП «Математика и компьютерные науки»

Студент _____ (Чернобаев Д.А.)

Руководитель работы _____ (Наумов Д.А.)

Руководитель ОПОП _____ (Таганов А.И.)

Рязань, 2022 г.

АННОТАЦИЯ

ABSTRACT

График работы над выпускной квалификационной работой

№ п/п	Планируемая работа	Срок выполнения
1	Обзор литературы, цели и задачи ВКР	18.03 – 31.03
2	Разработка алгоритмов решения поставленной задачи	01.04 – 10.04
3	Разработка программного обеспечение для генерации случайных заданий	11.04 – 31.05
4	Оформление пояснительной записки	01.06 – 14.06

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

СОДЕРЖАНИЕ	
СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ	5
СОДЕРЖАНИЕ	7
ВВЕДЕНИЕ	9
1. ПОСТАНОВКА ЗАДАЧИ.....	10
1.1. Цель РАБОТЫ.....	10
1.2. АКТУАЛЬНОСТЬ РАБОТЫ.....	10
2. ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ	12
2.1. НОРМАЛЬНЫЕ АЛГОРИТМЫ МАРКОВА.....	12
2.2. ВЫВОДЫ К РАЗДЕЛУ 2	14
3. ПРОЕКТИРОВАНИЕ СИСТЕМЫ ГЕНЕРАЦИИ ТЕСТОВЫХ ЗАДАНИЙ.....	15
3.1 НАЗНАЧЕНИЕ И СТРУКТУРА СИСТЕМ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ	15
3.2 ОТЛИЧИТЕЛЬНЫЕ ОСОБЕННОСТИ СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ «КОМПАС-3D»	20
3.3 «КОМПАС-3D» API	28
3.4 КРАТКИЙ ОБЗОР ЯЗЫКА ПРОГРАММИРОВАНИЯ PYTHON	30
3.5 БИБЛИОТЕКИ PYTHON ИСПОЛЬЗУЕМЫЕ В ПРОГРАММЕ	30
3.6 ОПИСАНИЕ ПРОГРАММЫ	31
3.7 ?	31
4. РАЗРАБОТКА ПРОГРАММНОЙ ДОКУМЕНТАЦИИ.....	32
5. ТЕСТИРОВАНИЕ.....	32
ЗАКЛЮЧЕНИЕ.....	32
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	32

ВВЕДЕНИЕ

Все высшие учебные заведения в мире, в том числе и в России, при предоставлении услуг обучения так же должны заниматься и проверкой полученных знаний студентов, следить за тем, как они усваивают материал, предоставленный преподавателем. Одним из способов такой проверки являются различные практические задания, разделенные на варианты, что позволяет в индивидуальном порядке проверить усвоение программы определенным студентом

В выпускной квалификационной работе я ставлю целью разработать такое программное обеспечение, которое позволит быстро и корректно формировать задания для проверок знаний работы в системе автоматизированного моделирования «Компас-3D» по дисциплинам «Инженерная графика» и «Компьютерная графика».

Предоставленная пояснительная записка для достижения упомянутой выше цели разделена на следующие разделы...

1. ПОСТАНОВКА ЗАДАЧИ

1.1. Цель работы

Целью выпускной работы является разработка программного обеспечения генерации случайных тестовых заданий для работы студентов в системе автоматизированного проектирования «Компас-3D». Выполнение поставленной цели заключается в реализации на практике определенных задач:

1) разработать алгоритм выбора типового варианта задания в системе «Компас-3D» для его последующего формирования в индивидуальное задание студента;

2) разработать алгоритм изменения выбранного типового задания при помощи нормальных алгоритмов Маркова;

3) разработать алгоритм визуализации изменений для оператора программы, а также последующего сохранения измененного документа в отдельном файле;

5) провести экспериментальные проверки разработанного программного обеспечения.

1.2. Актуальность работы

Контроль за усвоением изучаемого материала, как и освоение практического применения полученных знаний, являются неотъемлемой частью учебного процесса при освоении любой дисциплины, как в заведениях высшего учебного образования, так и в иных учебных заведениях. Именно поэтому особую важность представляет собой индивидуализация таких заданий. Обучающиеся студенты зачастую имеют огромное количество возможностей, по использованию материалов, подготовленных такими же студентами, которые ранее обучались по их специализации, что приводит к некорректному отображению полученных знаний и не позволяет закрепить

полученный материал на практики. Следствием такой ситуации является малая компетентность выпускаемых специалистов по направлению их обучения.

Моя выпускная квалификационная работа ставит перед собой целью разработку математического и программного обеспечения для генерации случайных заданий студентам по дисциплинам «Инженерная графика» и «Компьютерная графика», созданная мной программа позволит индивидуализировать задания для каждого студента обучающимся этим дисциплинам. Таким образом, разработанное в данной работе ПО позволит в значительной степени решить проблему корректного контроля знаний обучающихся, а также поможет усвоить применение полученных знаний на практике в большей степени чем до этого.

2. ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ

2.1. Нормальные алгоритмы Маркова

Теория нормальных алгоритмов была разработана известным советским математиком А. А. Марковым на рубеже 40-50х годов XX века. Она потребовалась для того, чтобы представить еще одну формулировку такого понятия, как алгоритм и разрешить некоторые проблемы теории ассоциативных исчислений. Сам Марков в своих исследованиях называл их «алгорифмами». Суть алгоритма, представленного ученым, заключается в том, что он представляет собой определенные правила исходя из которых осуществляется переработка «слов» в каком-либо «алфавите». Считаю важным заметить, что в данном контексте «слово» и «алфавит» представляют собой абстрактную сущность, которую можно использовать для определения различных операций, к примеру перевода единичного числа (число которое представляет из себя набор единиц, «5=lllll»), в двоичное. В данном алгоритме, исследователь выделяет элементарную операцию, подстановку в слово в место одной его части другой.

Что же из себя представляют данные подстановки? Будем использовать в качестве нашего алфавита любое непустое множество. Его элементы являются буквами алфавита, а последовательность взятых из алфавита букв – это слово. Также допускается существование пустого слова, которое обозначается λ .

Рассмотрим использование алгоритмов Маркова на примере перевода двоичного не отрицательного числа в единичное, где единичное число характеризуется количеством палочек.

Возьмем алфавит, который будет состоять из трех букв:

$$A = \{0, 1, |\}$$

Далее нам потребуется составить правила, по которым наше двоичного число будет преобразовываться в единичное:

$$\left\{ \begin{array}{l} 1 \rightarrow 0| \\ |0 \rightarrow 0|| \\ 0 \rightarrow (\text{пустая строка}) \end{array} \right.$$

Выберем интересующее нас двоичное число и преобразуем его в единичное:

111

Выполнение:

1. 111 \rightarrow 0|11
2. 0|11 \rightarrow 0|0|1
3. 0|0|1 \rightarrow 0|0|0|
4. 0|0|0| \rightarrow 00|||0|
5. 00|||0| \rightarrow 00||0||
6. 00||0|| \rightarrow 00|0||||
7. 00|0|||| \rightarrow 000||||||
8. 000|||||| \rightarrow 00||||||
9. 00|||||| \rightarrow 0||||||
10. 0|||||| \rightarrow |||||

Как видим из примера, при выполнении алгоритма Маркова, прописанные нами правила применяются определенным образом. Применение алгоритма Маркова имеет следующую последовательность шагов. Сначала алгоритм ищет слева на право «подслово», которое удовлетворяло бы описанному правилу подстановки, после нахождения, он производит подстановку и начинает поиска заново, если же первое правило нельзя выполнить, то алгоритм переходит ко второму, а потом опять начинает сначала, так происходит до тех пор, пока ни одно из написанных правил нельзя

будет применять к имеющемуся слову, в таком случае алгоритм Маркова будет считаться выполненным.

Как мы видим на предоставленном примере, использование такого подхода при выполнении выпускной квалификационной работы по созданию программы, позволит нам четче реализовать процесс изменения типового варианта. Мы можем создать базовый «алфавит», буквами в котором будут являться операции над выбранным чертежом. Когда пользователь программы будет выбирать, что именно он хочет изменить и составлять «слово», то при его реализации одни изменения будут вызывать последующие модификации чертежа, что приведет к созданию нового задания, учитывающего все правила форматирования чертежей с сохранением их дальнейшей пригодности к реализации при создании непосредственно деталей.

2.2. Применение алгоритмов Маркова в задаче генерации случайных тестовых заданий

Теория нормальных алгоритмов была разработана

2.3. Выводы к разделу 2

В данной главе выпускной квалификационной работы

3. ПРОЕКТИРОВАНИЕ СИСТЕМЫ ГЕНЕРАЦИИ ТЕСТОВЫХ стрЗАДАНИЙ

3.1 Архитектура системы генерации тестовых заданий

Программа, которая генерирует случайные тестовые задания для студентов не смогла бы выполнить свою задачу без дополнительных элементов. Данные элементы образуют систему по генерации задания. Которая представлена на рисунке . Входные данные представляют собой информационные сведения для программы, они указывают на файл, который требуется изменить (File.m3d), содержат информацию о переменных, которые могут изменяться в этом файле (File.xml) и дополнительную информацию, например, названия директорий куда нужно сохранять новые файлы. Выходными данными являются наборы различных файлов, среди них: файлы с измененными моделями (File.m3d), и файлы с чертежами, которые построены на основе новых моделей (File.cdw), причем изменяя новые модели в будущем мы можем изменить и чертежи на их основе, а изменяя чертежи изначальные модели изменяться не будут. Сама программа не обладает требуемым функционалом, чтобы напрямую изменять получаемые файлы, поэтому она прибегает к возможностям САПР КОМПАС-3D и с помощью ее API передает нужные команды для открытия, изменения и сохранения файлов. Рассмотрим составляющие нашей системы.



3.2 Хранение информации о модели при помощи файлов XML

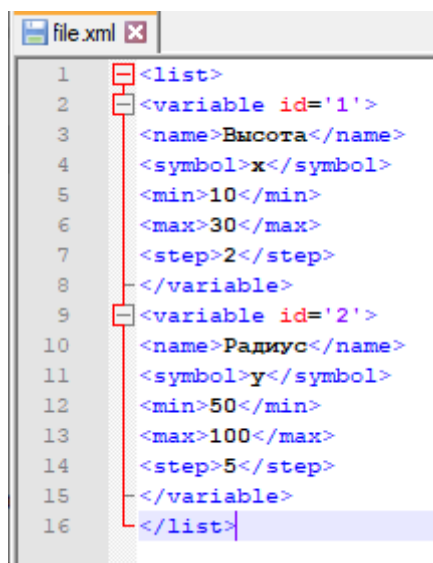
Хранение информации является основой нашей жизни, человек постоянно собирает, сохраняет, изменяет окружающую его информацию. Одним из способов такого взаимодействия с информацией является использование электронных носителей.

Множество программ ежедневно обрабатывают миллионы гигабайт различной информации, которые хранятся на электронных носителях в разных форматах. Графическую информацию хранят в файлах разрешения .JPEG, .PNG и др. Для хранения видео используют форматы .mp4, .avi и др.

В нашей программе нам требуется формат для хранения текстовой информации, которая будет содержать данные о переменных 3D модели. Форматы документов, которые используются для хранения текстовой информации являются форматы: .txt, .doc, .html, .xml и др. Нами был выбран формат .xml, рассмотрим, что представляет из себя этот формат и почему он был выбран нами.

XML или eXtensible Markup Language – это язык расширяемой разметки, основным предназначением данного языка является создание логических структур для данных, хранение этих данных и передача их в том виде, который

будет являться удобным и для компьютера и для человека. На рисунке представлен обычный XML файл.



Структура XML документа очень проста. Она заключается в создании древа элементов, которое потом будет изменяться, храниться, считываться и т.д. Для определения отдельных объектов и их атрибутов в таких файлах используются теги, которые пользователь может задавать самостоятельно. Такие файлы очень распространены среди различных пользователей, так как могут хранить абсолютно любую информацию. В формате XML есть возможность записи различных баз данных, хранение настроек для приложений или, как это сделано в нашей системе, хранение информации о переменных, которые могут быть изменены в модели.

Данный формат представления информации был принят нами для реализации хранения информации о переменных по ряду причин:

- Очень простая и понятная древовидная структура документа
- Возможность самим установить обозначения элементов в структуре
- Наличие у языка программирования Python на котором разработана наша программа очень удобного модуля, для взаимодействия с такими документами, ElementTree.

3.3 Назначение и структура систем автоматизированного проектирования

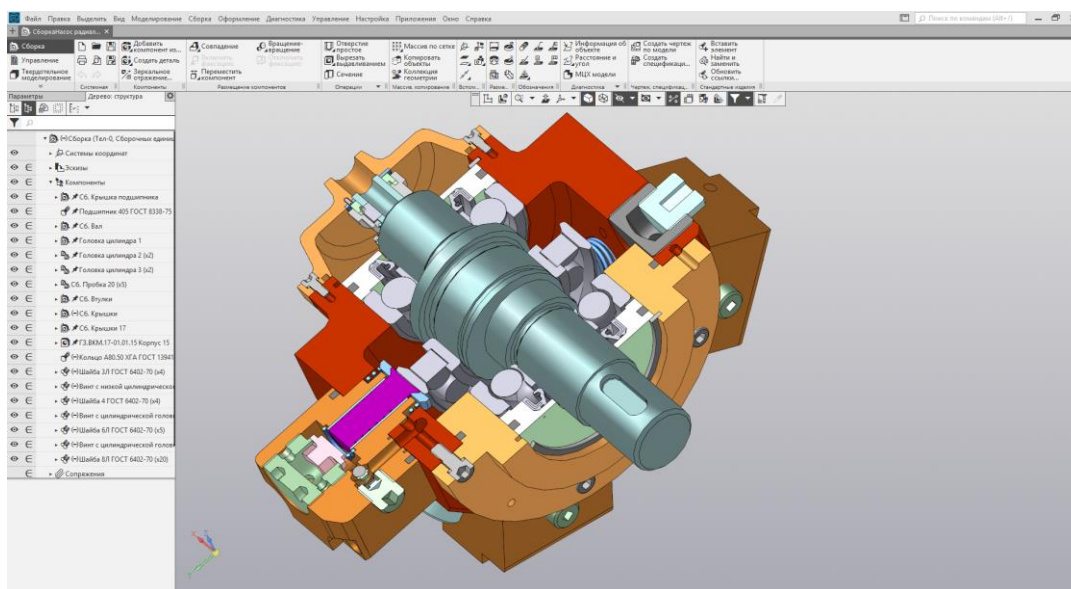
Системы автоматизированного проектирования или же САПР являются специализированным программным обеспечением, которое решает важные задачи оптимизации, унификации и автоматизации стандартных процедур, которые проходят выпускаемые товары. Использование САПР в условиях жесткой конкуренции позволяет производителям более привлекательно в отличии от их конкурентов, сократить время разработки нового продукта, что соответственно увеличит срок, в результате которого изделие морально устареет и ему на смену придут более современные аналоги.

Занимаясь поддержкой и развитием САПР на своем производстве предприятие в первую очередь преследует цель повышение качества своей продукции.

Пользователи современных САПР имеют в своем распоряжении богатый выбор инструментов, а также избавлены от многократного повторения уже выполненных вычислений. Современные САПР имеют мощную математическую составляющую, которая в значительной степени ускоряет производимые вычисления на базе этих систем. Наиболее известными представителями семейства САПР являются следующие программные продукты: AutoCAD (многоцелевая система для работы в различных областях), ArchiCAD (архитектурная система), ArCon (система планировки зданий, дизайна местности и интерьера), Splise (САПР электронных схем) и др.

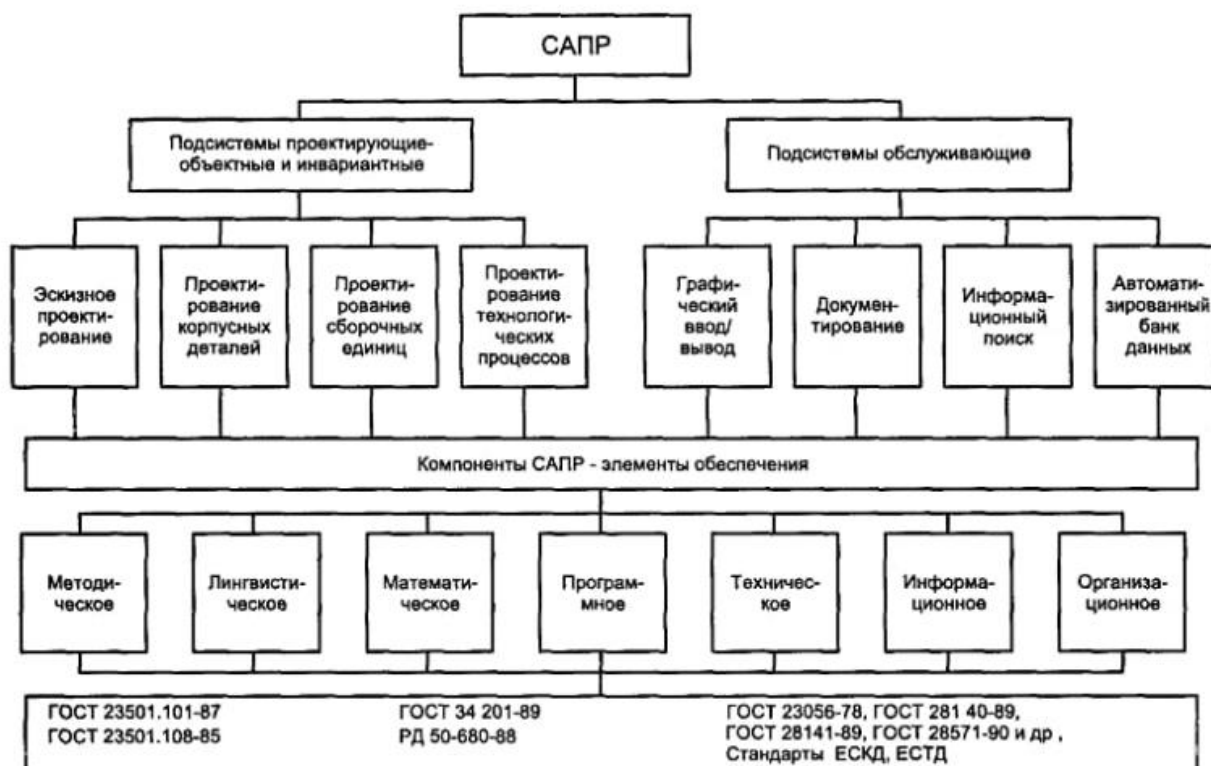
К одной из самых востребованных функций САПР можно отнести возможность построения компьютерных 2-D и 3-D моделей, каталогизация проектной документации. Именно благодаря этой возможности данные системы широко используют в машиностроении. К представителям САПР,

которые акцентируют свое внимание на данных требованиях производителей, так же относится изучаемая на дисциплинах «Инженерная графика» и «Компьютерная графика» система «Компас-3D». На рисунке представлен один из проектов, который был реализован с помощью этой системы.



Структура современных систем автоматизированного проектирования является сложной и многоуровневой, она формируется за счет средств вычислительной техники, различных видов обеспечения и обслуживающего систему персонала. Согласно ГОСТУ 23501. 101-87 структура САПР включает в себя две подсистемы: проектирующую и обслуживающую.

Проектирующие модули отвечают за непосредственное выполнение конкретных проектных задач, в то время, как обслуживающая подсистема берет на себя задачи: управления проектированием, документирование, реализует графический интерфейс, создание и обслуживание базы данных.



Абсолютно все подсистемы и компоненты системы должны быть совместимы друг с другом и решать поставленные задачи взаимодействуя. Также элементы современных САПР унифицируются, чтобы существовала возможность взаимозаменяемости компонентов. Возможность интеграции САПР с другими информационными системами является важной частью данного программного обеспечения, как и модифицируемость и пополнение уже разработанного продукта новыми компонентами.

3.4 Отличительные особенности системы автоматизированного проектирования «Компас-3D»

Как уже было отмечено в предыдущем пункте наиболее популярными являются САПР, перед которыми ставятся задачи по созданию 2-D и 3-D моделей. Одним из представителей данного вида систем является «Компас-3D», для которой мы пишем программной обеспечение по созданию индивидуальных студенческих заданий.

Эта система является очень популярной среди специалистов, занимающихся различным моделированием. В основном программа

ориентирована на разработку моделей промышленного производства, но ее так же часто используют при разработке чертежей зданий и конструкций. Компас позволяет создавать проекты любой степени сложности. Программа предназначена для создания в первую очередь трехмерных параметрических моделей, ориентирована на формирование трехмерных моделей трехмерных тел, содержащих как типичные, так и нестандартные элементы конструкции.

Первая версия этой программы была разработана еще в 1989 году, однако доступ к ней мог получить лишь узкий круг специалистов. Коммерческая история ПО начинается с 1997 года, когда ее впервые представили на коммерческом рынке, с этого времени систему неоднократно улучшали и создавали новые версии. Сейчас к программе так же выпускаются несколько дополнений, благодаря которым система приобрела дополнительные возможности, что еще лучше упростило работу инженеров.

За время своего существования программа получила ряд особенностей, что разительно выделяет ее среди других систем, предназначенных для промышленного проектирования:

- Система имеет собственное математическое ядро, созданное разработчиками АСКОН;
- Благодаря своему простому интерфейсу, программа имеет достаточно низкий порог вхождения, и новички имеют возможность быстро перейти от изучения к созданию реальных моделей;
- Во время взаимодействия с другими программами по проектированию, созданные в Компасе продукты перемещаются без потери данных;
- Система поддерживает достаточно большое количество форматов передачи данных;
- Часть проектирования может происходить автоматически, что значительно упрощает работу специалиста;

- Так же на ПО можно вести разработку различных электрических цепей.

Абсолютно любая программа выделяется на фоне других аналогов различными плюсами, но также не освобождается от недостатков, которые отмечают пользователи САПР используя ее в своих целях.

Плюсы «Компас-3D»:

- Интерфейс данной САПР является очень простым и интуитивно понятным;
- Имеется встроенная библиотека различных моделей, которые можно использовать как при обучении, так и при проектировании;
- Весь интерфейс локализован на русском языке, что является существенным плюсом, это значительно упрощает обучение и использование программы в России;
- Стоимость лицензии является приемлемой, и практически каждый человек имеет возможность приобрести программу для своего пользования
- САПР учитывает свойства различных материалов при проектировании;
- Имеет большое количество форматов для выгрузки и загрузки файлов;
- Можно так же разрабатывать и 2D чертежи.

Недостатки:

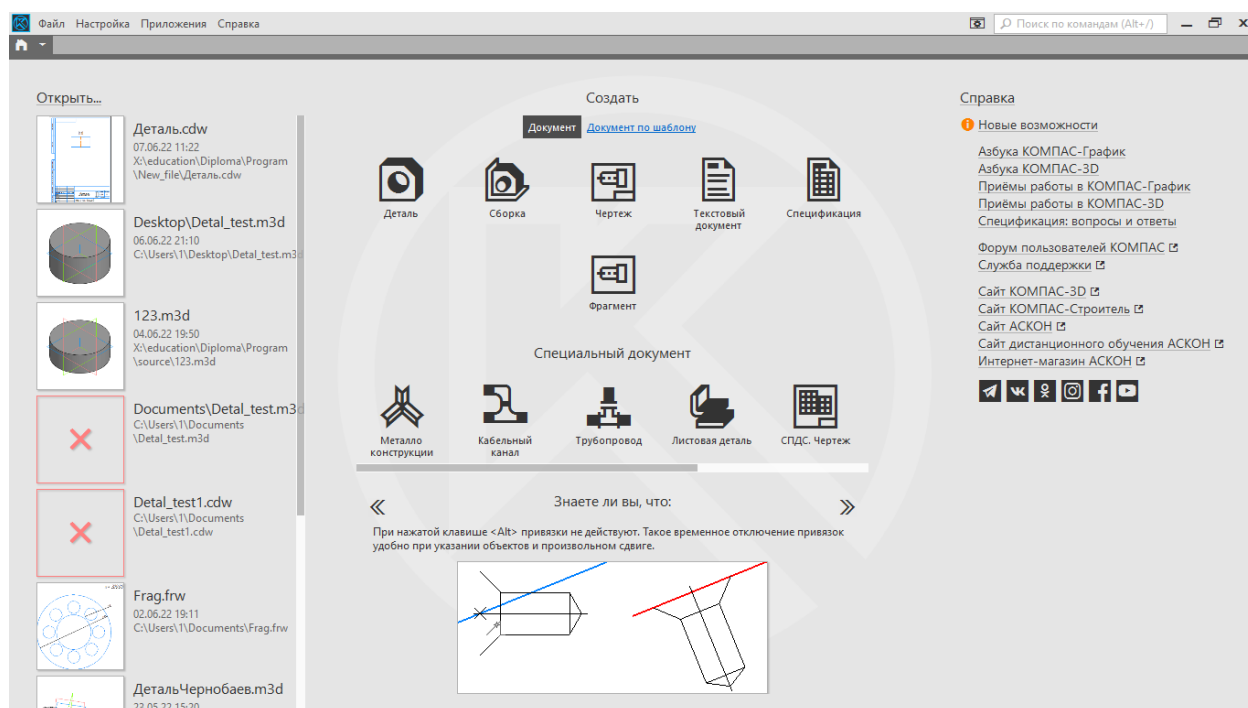
- Визуализация разрабатываемых проектов остается на достаточно низком уровне;
- Иногда возникают проблемы совместимости разрабатываемых моделей на других программах с «Компас»;

- Система поверхностного моделирования так же находится на достаточно низком уровне и имеет определенные недостатки.

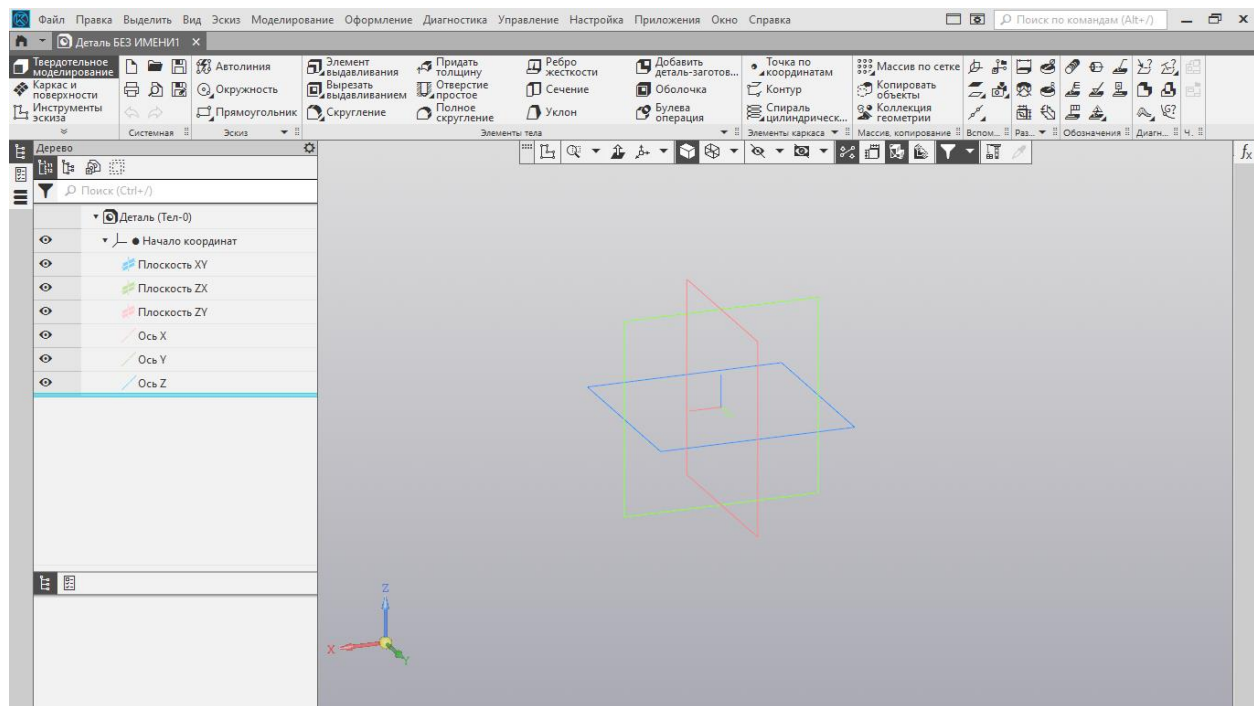
При всех перечисленных недостатках, можно смело сказать, что данная САПР охватывает огромный спектр разработок и очень полезна для широкого круга специалистов.

Рассмотрим интерфейс данной САПР:

При открытии программы Компас-3D, мы попадаем в главное окно, здесь нам предлагают выбрать файл, с которым мы будем работать. Пользователь может выбрать из файлов, с которыми работал недавно, или же нажать на надпись: «Открыть...». Так же пользователь может создать новый проект, для определенных целей, форматы проектов представлены в центральной части.

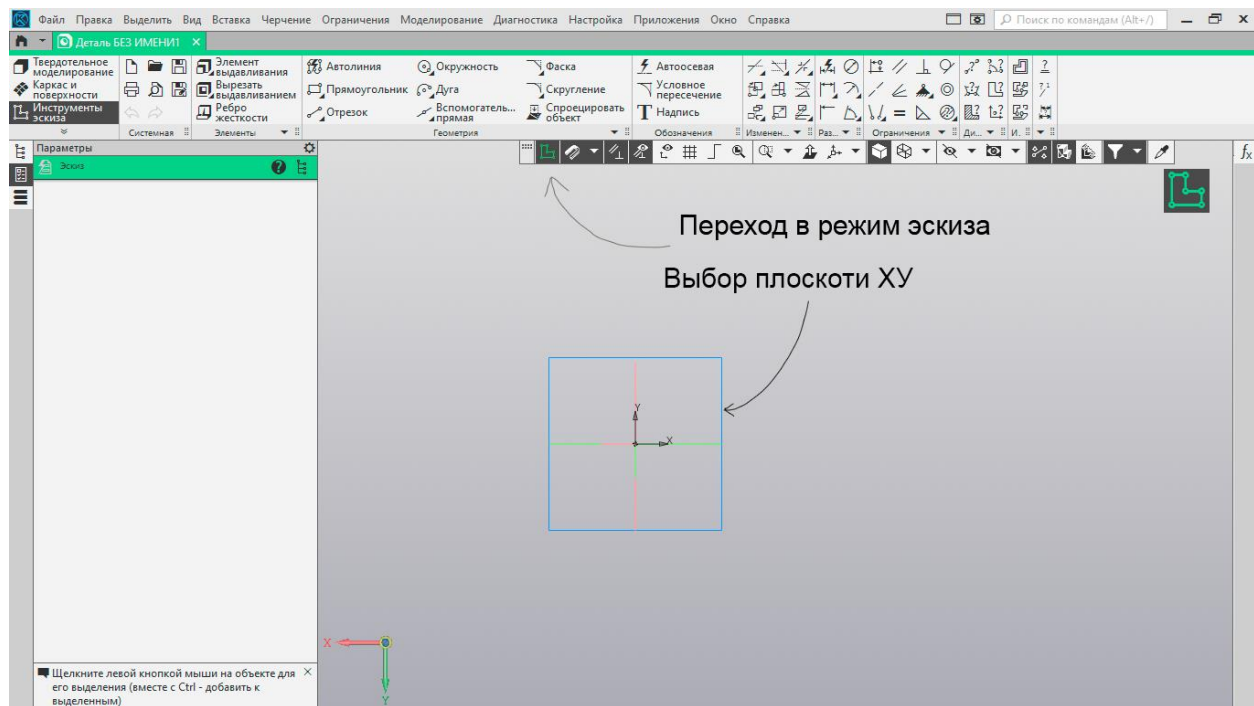


Создадим новый документ, в котором пользователь может создавать деталь. Для этого нажмем на кнопку «Деталь» в центральной части главного окна.

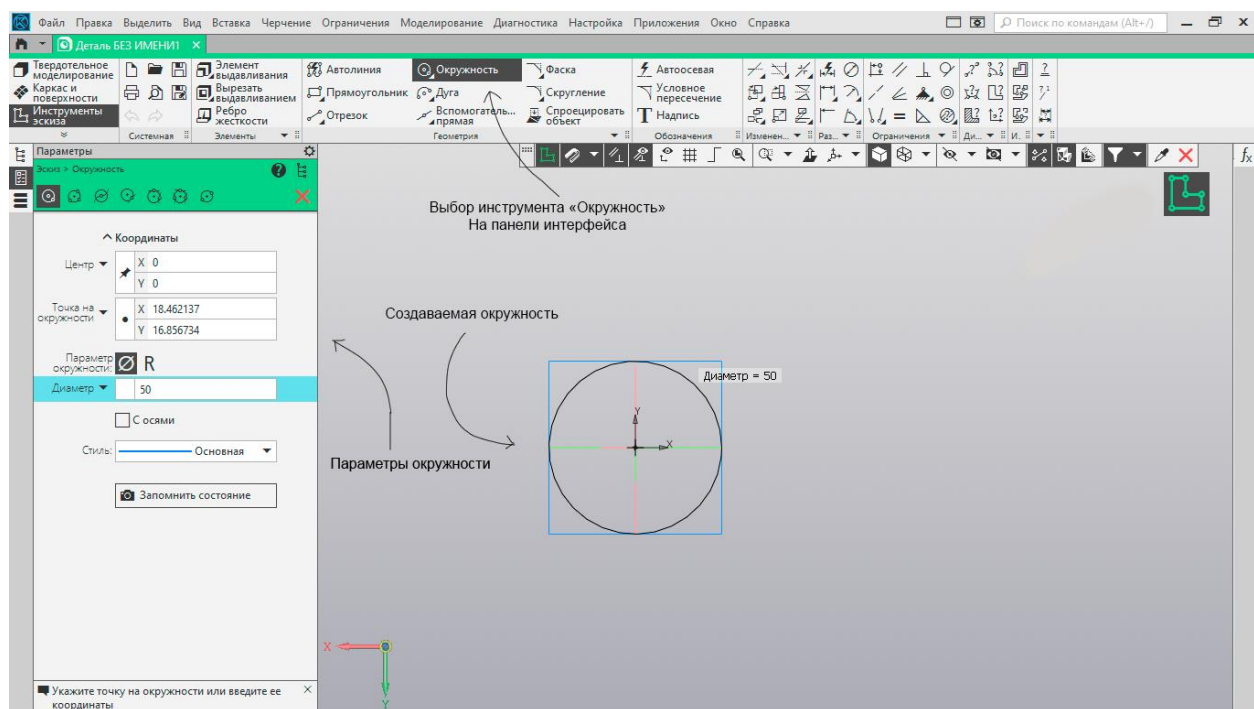


Перед нами открывается окно изменения нашей детали, здесь мы можем создавать 3D-модели различной сложности. Рассмотрим процесс редактирования проекта на примере создания простого цилиндра с переменными, через которые мы сможем редактировать высоту и диаметр модели.

Для создания цилиндра в первую очередь нам требуется создать 2D-эскиз модели, построив окружность, которую в последствии мы сможем вытянуть в цилиндр. Для этого перейдем в режим эскиза нажав соответствующую кнопку на интерактивной модели и выберем плоскость относительно которой наш цилиндр будет перпендикулярен. Мной была выбрана плоскость XY.

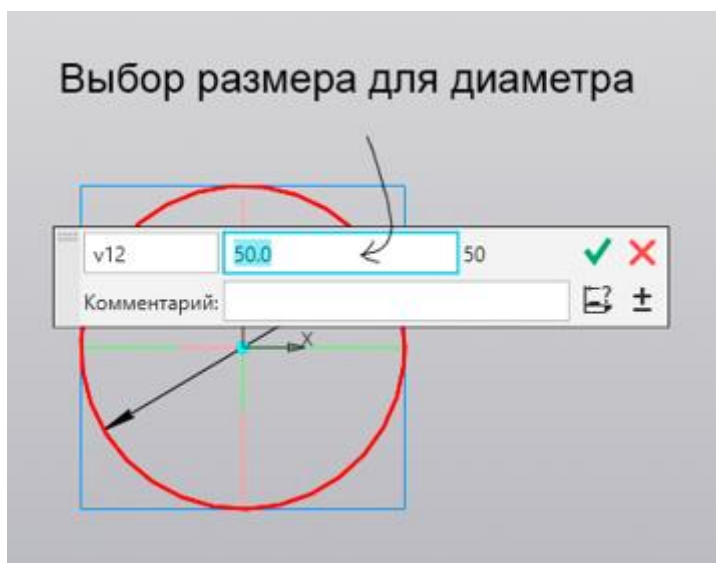
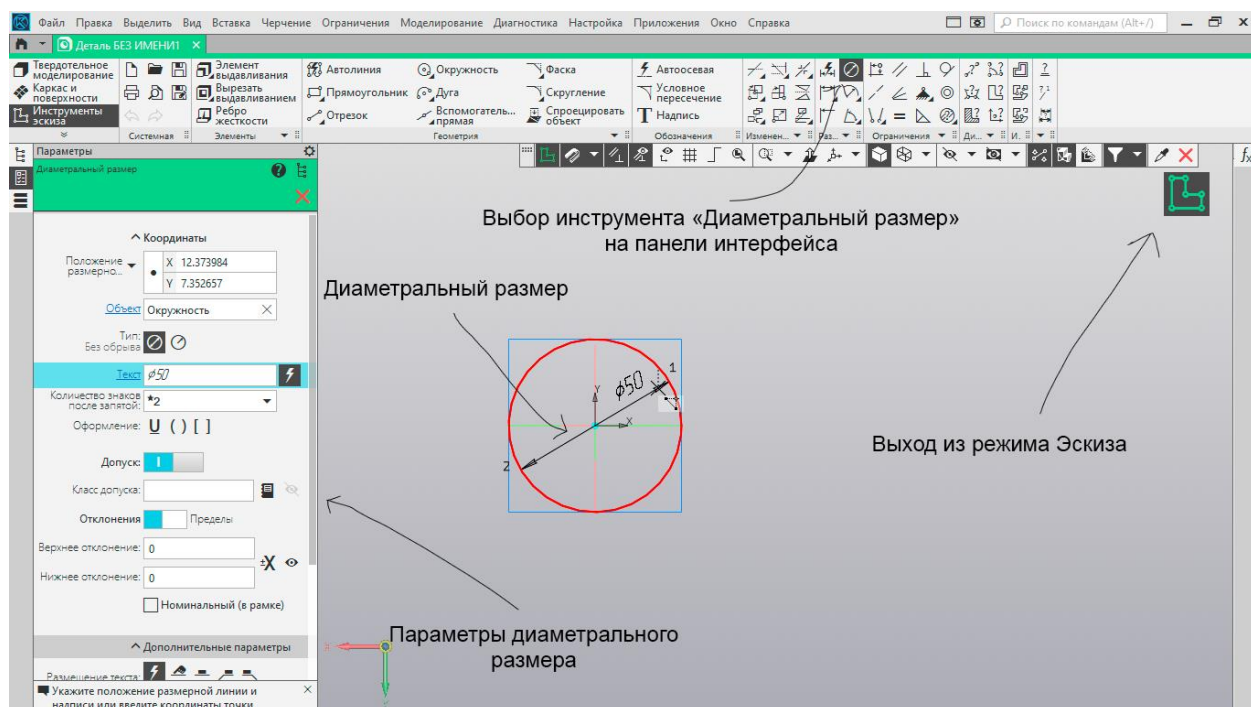


Нарисуем окружность с центром в начале координат. Диаметр окружности возьмем за 50мм.



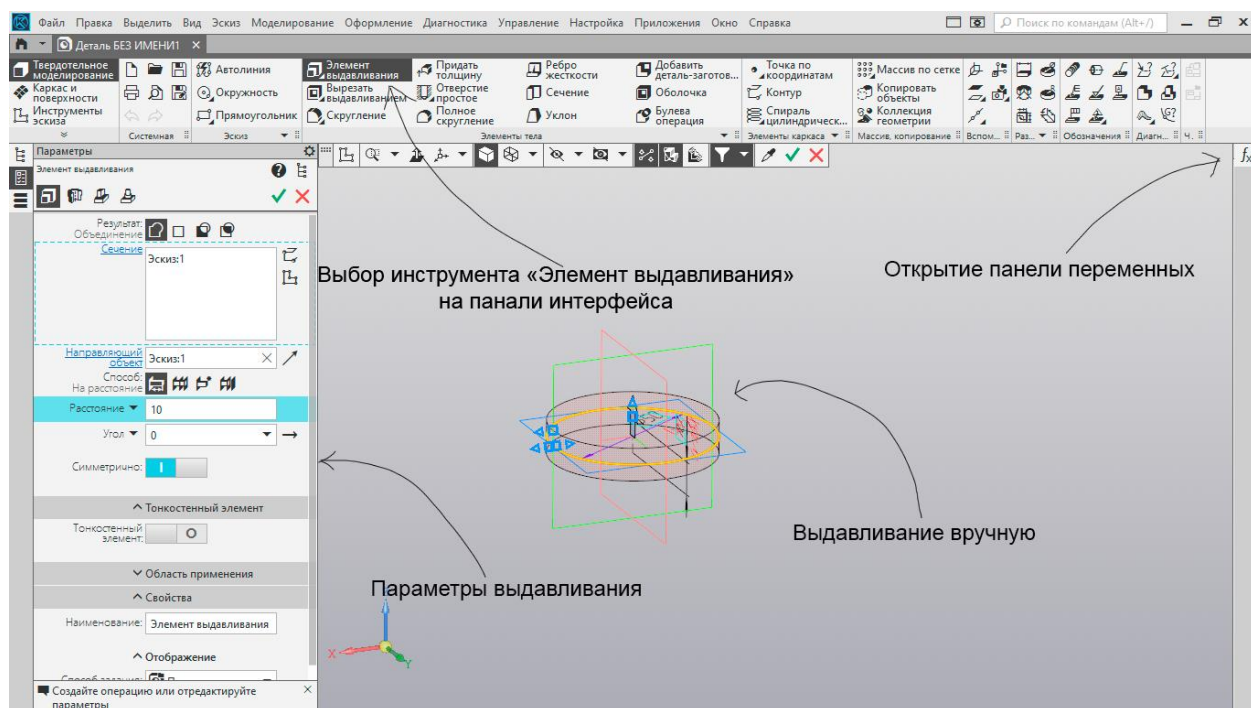
После создания окружности, необходимо установить ее диаметральный размер, в будущем он понадобится нам, чтобы изменять значения диаметра у нашего цилиндра. После установки диаметрального размера, нам предлагают

выбрать его длины, оставим ее такой же, какой она была и выйдем из режима Эскиз.



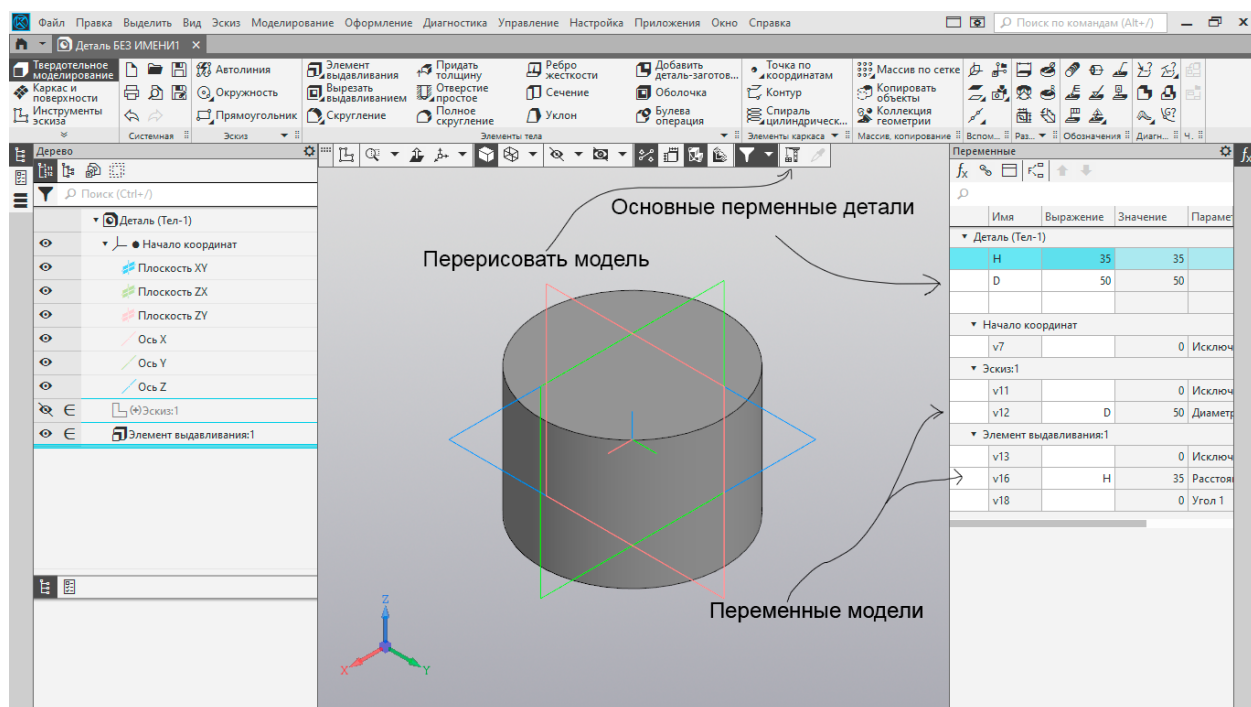
Один из самых простых способов создания объемного объекта в компасе, является его создание с помощью выдавливания. Мы также прибегнем к этому способу для того, чтобы создать нужный нам цилиндр. Мы можем осуществить выдавливание вручную, передвигая стрелочки, которые появятся над нашей окружностью или через панель параметров. Добавим на панели «Параметры» информацию о симметрии выдавливания относительно

окружности, а также расстояние выдавливания в 10мм с каждой стороны. После установки параметров мы завершаем операцию нажатием клавиши «Enter» или же галочки на панели «Параметры». Далее откроем панель переменных, чтобы начать редактировать размеры нашего цилиндра.



На панели «Переменные» изначально указываются только переменные, которые были созданы автоматически при построении модели. Для цилиндра такими переменными будут являться v16(Высота) и v12(Диаметральный размер). Создадим также переменные для детали введя названия переменных в пустой строке раздела «Деталь (Тел-1)», назовем их D и H и укажем произвольные значения 50мм и 35мм соответственно. Далее нам необходимо связать переменные детали и переменные модели. Для этого в ячейке выражение напротив v12 поставим имя переменной D, а напротив v16 поставим имя переменной H. Теперь при изменении переменных детали будет изменяться и наша модель, чтобы увидеть изменения в окне программы пользователь должен перестроить модель, нажав на соответствующую иконку в окне программы. Так же сделаем переменные детали внешними, для этого правой кнопкой мыши щелкнем по переменной и выберем «Внешняя», это

нужно для того, чтобы появилась возможность получать данные об этой переменной из других программ, например, Excel.



Таким образом мы кратко рассмотрели интерфейс САПР «Компас-3D» на примере создания простого компаса. Помимо этого, ПО позволяет выполнить почти все наши действия автоматически создав с помощью API Компас программу.

3.5 «Компас-3D» API

API Компас это возможность для разработчика автоматизировать практически всю работу по построению сложных чертежей и моделей, сведя их к простому нажатию пары кнопок.

Чаще всего к реализации различных задач программным способом прибегают из-за того, что в CAD-системах нет реализации для различных практических задач. Как правило, такие задачи являются узкоспециализированными и применяются лишь на конкретном предприятии или подотрасли. Используя API, как способ реализации часто встречающихся задач, разработчики значительно сокращают время, которое им требуется

затратить на разработку конкретного продукта. Также благодаря интерфейсу, предоставленного API можно интегрировать Компас с другими системами.

API Компаса представлено для разных языков программирования, что позволяет увеличить число программистов, которые могли бы с ним работать. В число языков программирования, которые поддерживает Компас входят: Delphi, C++, C#, Visual Basic, а также Python в виде модуля КОМПАС-Макро.

КОМПАС-Макро представляет из себя уже готовую среду для разработки ПО, которая интегрированная в Компас-3D. По-сути КОМПАС-Макро является обычной библиотекой, которую подключают к Компасу. Для начала работы с макросами, рядовому пользователю совсем не обязательно знать язык программирования Python, потому что среда позволяет просто записать действия пользователя с экрана, например, создание какого-то 3D-объекта, а программа сама интерпретирует его действия в код. После чего, готовый макрос можно применять уже в других проектах, если потребуется выполнить определенные действия.

Однако так же важно отметить, что запись макроса не является универсальным ответом, на задачу создания программ под Компас-3D. Когда мы находимся в состоянии записи макроса, то он не может отследить изменение нами различных переменных или других параметров. И после остановки записи макроса, код для изменения значения переменной не появится. Для изменения переменных придется уже углубиться в программирование и взаимодействие с API.

Так же код записанный с помощью КОМПАС-Макро мы можем переместить в другую стороннюю программу. Использование этого инструмента облегчит работу разработчика, по поиску нужных ему функций при создании своей собственной программы и вместо ее поиска в справочной документации, можно просто записать макрос, который перепишет действия в работающий код.

Рассмотрим базовые модули команды API для взаимодействия программы с САПР:

3.6 Краткий обзор языка программирования Python

Python – представляет из себя язык программирования высокого уровня. Он является одним из самых популярных языков программирования в мире. Благодаря своей много профильности этот язык программирования имеет большую популярность среди специалистов разных секторов IT-индустрии. Помимо этого, данный язык так же обладает простым и лаконичным синтаксисом, что в значительной степени уменьшает порог вхождения для новых программистов. Зачастую код на Python занимает в разы меньше объема, чем на других языках программирования.

Так же стоит выделить огромное количество различных библиотек, которые были написаны разными программистами для этого языка. Практически для любой сферы деятельности можно найти библиотеку, которая будет очень полезна при разработке ПО. Есть библиотеки для работы со структурой файлов Windows, для работы со временем, изображениями, математическими функциями, созданием графического интерфейса и многое другое.

3.7 Библиотеки Python используемые в программе

Рассмотрим библиотеки и модули языка Python, которые были использованы при создании программного обеспечения на выбранную мной тему работы.

Выбранные библиотеки и модули:

- Os – Модуль представляет огромное количество функций, чтобы программа могла взаимодействовать с операционной системой, причем их поведение не зависит от типа ОС.

- ElementTree – Данный модуль реализует работу API для взаимодействия языка программирования и XML файлов.
- win32com – это "компонентная объектная модель" части pywin32, которая в свою очередь является базовым модулем от которого зависит написание COM-компонентов и серверов
- LDefin2D – библиотека для API Компас
- MiscellaneousHelpers – библиотека для API Компас

3.8 Описание программы

В процессе выполнения выпускной квалификационной работы было создано программное обеспечение, которое призвано выполнять задачу по генерации новых индивидуальных заданий для студентов, на основе стандартных моделей.

Данная программа выполнена в одном модуле, код программы представлен в приложении А, схема программы представлена на рисунке.

3.9 ?

4. РАЗРАБОТКА ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

5. ТЕСТИРОВАНИЕ

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ