

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе по курсу «Архитектура и проектирование графических систем»

Тема работы: «Разработка программы для моделирования параметризованного трёхмерного объекта в среде AutoCAD»

Руководители:

Карабчевский В. В.

Доценко Г. В.

Бабакина А. А.

(подпись)

(дата)

Разработал:

ст. гр. ПИ-19а

Сычева Е.В.

(подпись)

(дата)

Донецк – 2022

РЕФЕРАТ

Пояснительная записка содержит 45 страниц, 43 рисунка, 3 источника и 2 приложения.

Объектом исследования данной работы является программное обеспечение для моделирования параметризованного [1] трёхмерного объекта в среде AutoCAD.

Цель работы – создать трёхмерную модель выбранного объекта средствами твердотельного моделирования с помощью программы. Исходными данными для работы программы будут являться параметры модели, которые должен вводить пользователь.

Результатом работы является разработанная программа, моделирующая твердотельный объект на основе введенных пользователем параметров.

Для разработки программы был использован специализированный диалект языка Lisp – AutoLISP[2], созданный специально для использования внутри AutoCAD. За взаимодействие с пользователем отвечает всплывающие окна, необходимые для ввода параметров модели.

AUTOLISP, ТВЕРДОТЕЛЬНАЯ МОДЕЛЬ, ПОВЕРХНОСТНАЯ МОДЕЛЬ, ПРОЕКТИРОВАНИЕ ГРАФИЧЕСКИХ СИСТЕМ, МОДЕЛИРОВАНИЕ, ПАРАМЕТРИЗОВАННЫЙ ТРЁХМЕРНЫЙ ОБЪЕКТ, AUTOCAD

СОДЕРЖАНИЕ

| | |
|---|----|
| 1 РАЗРАБОТКА ПОВЕРХНОСТЕЙ И ТВЕРДОТЕЛЬНОЙ МОДЕЛИ ОБЪЕКТА | 4 |
| 1.1 Поверхностные модели объектов ферзя | 4 |
| 1.2 Твердотельные модели объектов ферзя | 13 |
| 2 ОПИСАНИЕ ВЫБРАННЫХ МЕТОДОВ И АЛГОРИТМОВ | 20 |
| 3 ОПИСАНИЕ ПРОГРАММЫ..... | 21 |
| 4 ПРИМЕР ВЫПОЛНЕНИЯ ПРОГРАММ | 29 |
| ЗАКЛЮЧЕНИЕ | 33 |
| СПИСОК ЛИТЕРАТУРЫ | 34 |
| ПРИЛОЖЕНИЕ А. ТЕХНИЧЕСКОЕ ЗАДАНИЕ | 35 |
| ПРИЛОЖЕНИЕ Б. ИСХОДНЫЙ КОД..... | 39 |

1 РАЗРАБОТКА ПОВЕРХНОСТЕЙ И ТВЕРДОТЕЛЬНОЙ МОДЕЛИ ОБЪЕКТА

1.1 Поверхностные модели объектов ферзя

При поверхностном моделировании моделирование объекта осуществляется постепенно: сначала он разбивается на поверхности (плоские грани), затем собирается в единый трёхмерный объект. AutoCAD имеет множество способов поверхностного моделирования:

- с помощью трёхмерных поверхностей;
- с помощью трёхмерных поверхностных объектов;
- с помощью выдавливания или задания толщины объектам.

Для формирования поверхностных моделей существуют различные пространственные примитивы:

- 1) параллелепипед;
- 2) пирамида;
- 3) купол (полусфера);
- 4) клин;
- 5) тор;
- 6) сфера;
- 7) конус.

Формирование поверхностной модели «Ферзь» происходит в несколько этапов.

Первым этапом создаем основание. Создаем 2 плоских многоугольника с определенным количеством сторон (используя команды «многоугольник» и «область») на расстоянии равном высоте основания и из верхнего вырезаем окружность равную нижнему радиусу подставки. Также создаем отрезками наш многоугольник (команда «отрезок») и выдавливаем на нужную нам высоту (команда «выдавить») (рис. 1.1-1.2).

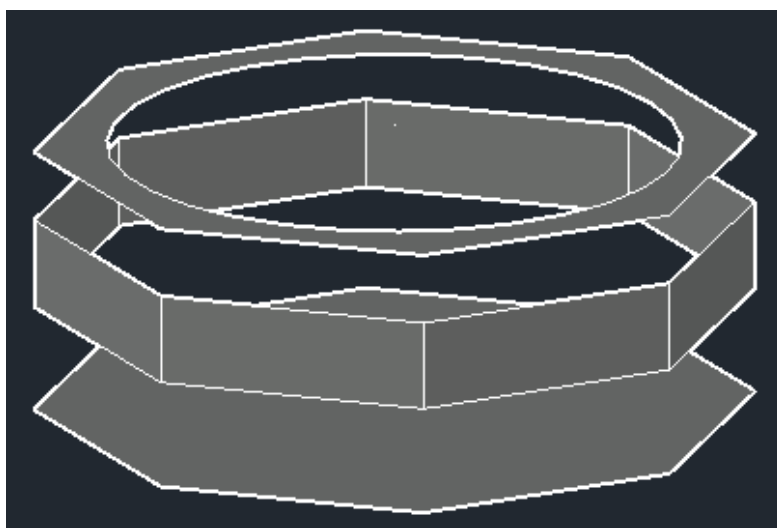


Рисунок 1.1 – Формирование основания

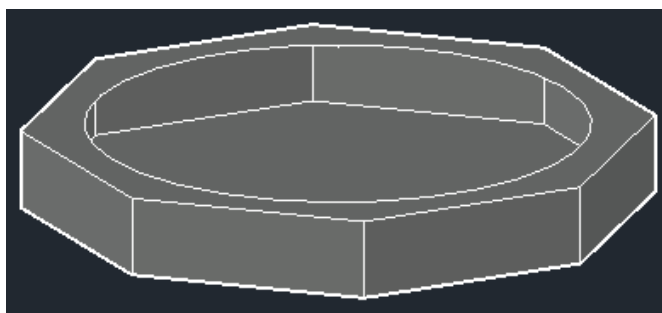


Рисунок 1.2 – Сформированное основание

Вторым этапом создаем подставку. Создаем 2 окружности (командой «окружность») и с помощью команды «лофт» объединяем их, также с помощью «область» сформируем верхнюю область (круг) и вырежем окружность равную нижнему радиусу юбочки(рис. 1.3-1.4).

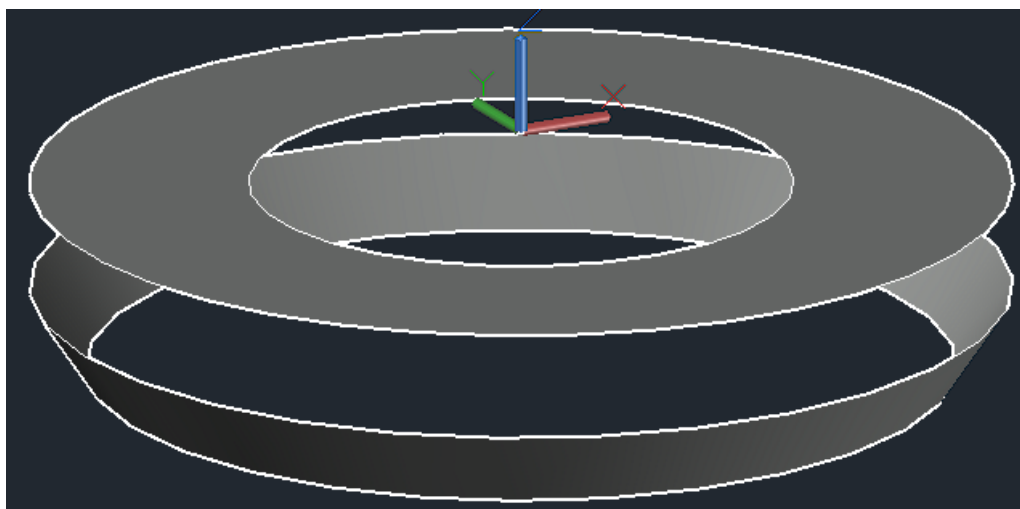


Рисунок 1.3 – Формирование подставки

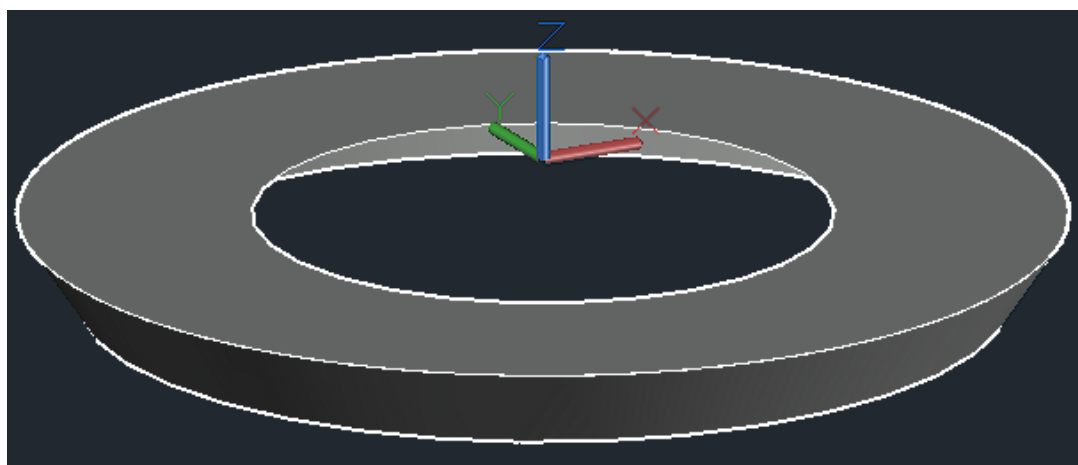


Рисунок 1.4 – Сформированная подставка

Третьим этапом создаем юбочку. Создаем 2 окружности (командой «окружность») и с помощью команды «лофт» объединяем их, также с помощью «область» сформируем верхнюю область (круг) и вырежем окружность равную нижнему радиусу юбочки(рис. 1.5-1.6).

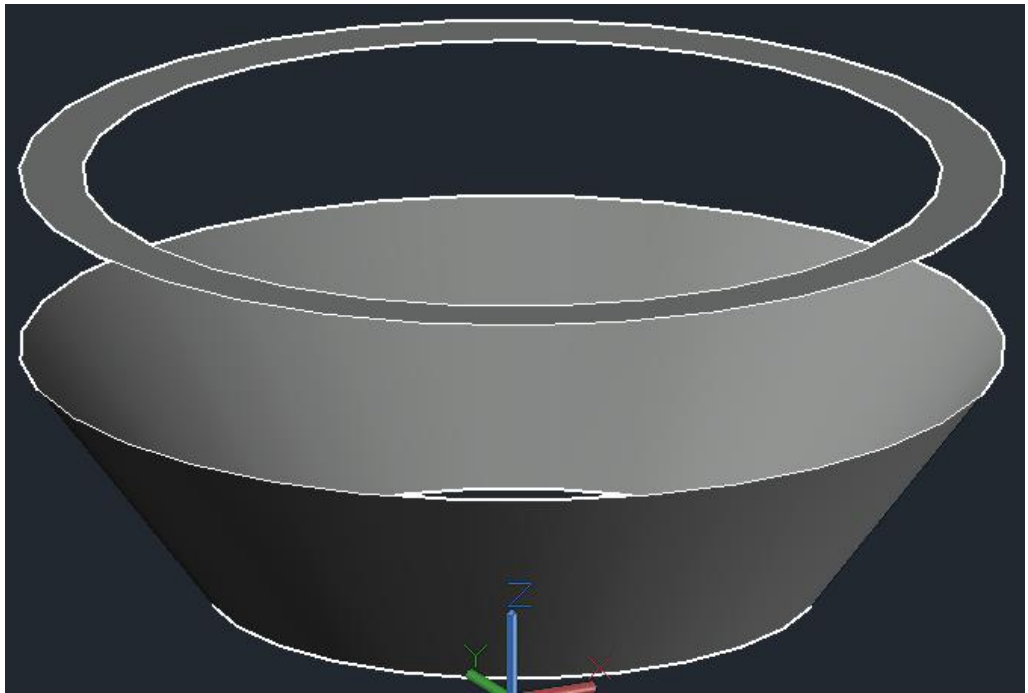


Рисунок 1.5 – Формирование юбочки

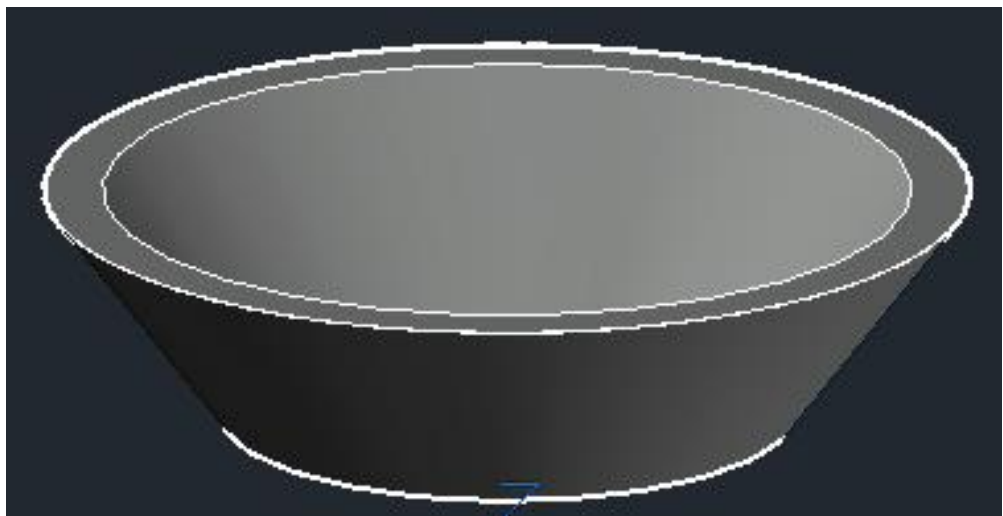


Рисунок 1.6 – Сформированный юбочка

Четвертым этапом создаем среднюю часть. Создаем 3 окружности (командой «окружность») и с помощью команды «лофт» объединяем их (рис. 1.7).



Рисунок 1.7 – Сформированная средняя часть

Пятым этапом создаем одно кольцо. Создаем 2 плоских окружности с вырезанными окружностями, величиной радиуса верхней окружности средней части и нижней окружности верхней части (используя команды «окружность» и «область») на расстоянии равном высоте кольца. Также создаем 2 дуги радиусом нашего кольца (команда «окружность» «разорвать») и выдавливаем на нужную нам высоту (команда «выдавить») (рис. 1.8-1.9).



Рисунок 1.8 – Формирование 1-го кольца

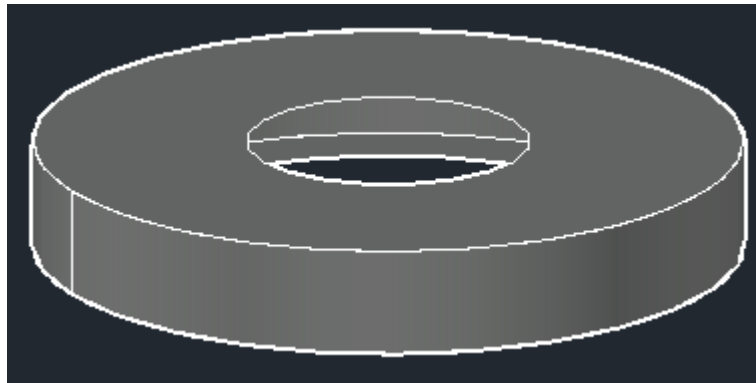


Рисунок 1.9 – Сформированное кольцо

Шестым этапом создаем верхнюю часть. Создаем 4 окружности (командой «окружность») и с помощью команды «лофт» объединяем их, также с помощью «область» сформируем верхнюю область (круг) и вырежем окружность равную нижнему радиусу сферы (рис. 1.10-1.11).



Рисунок 1.10 – Формирование верхней части



Рисунок 1.11 – Сформированная верхняя дуга

Седьмым этапом создаем купол(сферу). Создаем четверть окружности (командой «окружность» и «разорвать»), с помощью команды «вращать» создадим купол и с помощью команды «3d-поворот» повернем его по ОХ на 90 и вырежем окружность равную нижнему радиусу пимпочки (рис. 1.12).

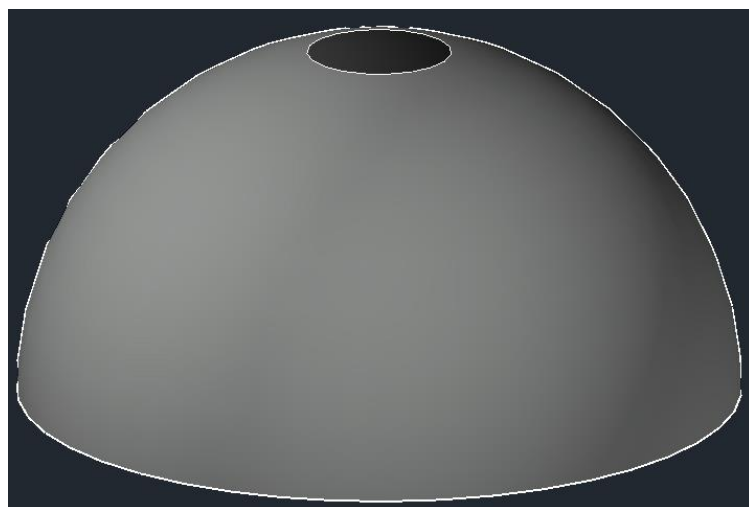


Рисунок 1.12 – Сформированная сфера

И последним восьмым этапом строим пимпочку, состоящую из ножки и верхушки. Создаем 2 плоских окружности (у верхней вырежем окружность

радиусом нижней окружности верхушки, используя команды «окружность» и «область») на расстоянии равном высоте ножки и создаем 2 дуги радиусом ножки (команда «окружность» «разорвать») и выдавливаем на нужную нам высоту (команда «выдавить»). Создаем 3 окружности (командой «окружность») и с помощью команды «лофт» объединяем их, также с помощью «область» сформируем верхнюю область (круг) (рис. 1.13-1.14)

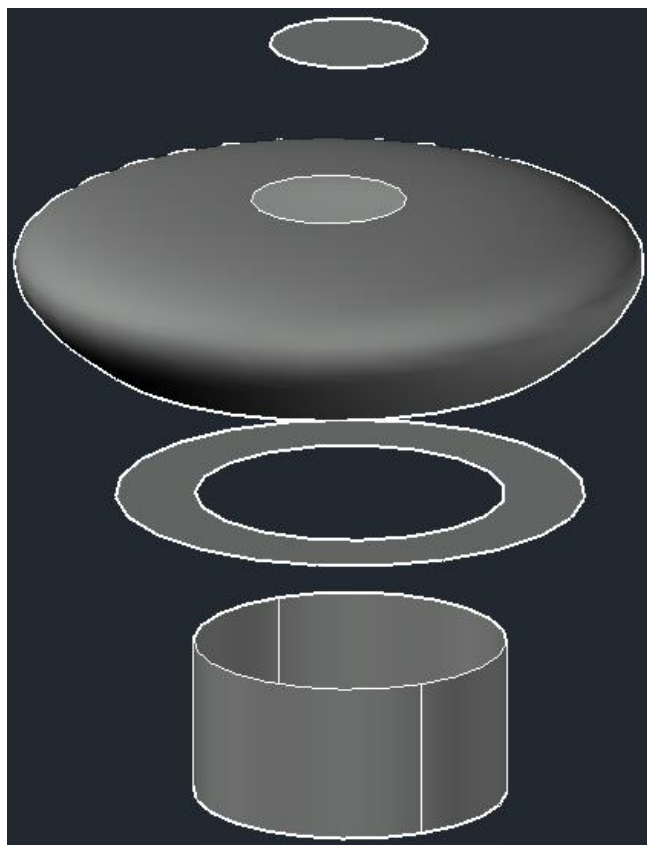


Рисунок 1.13 – Формирование пимпочки

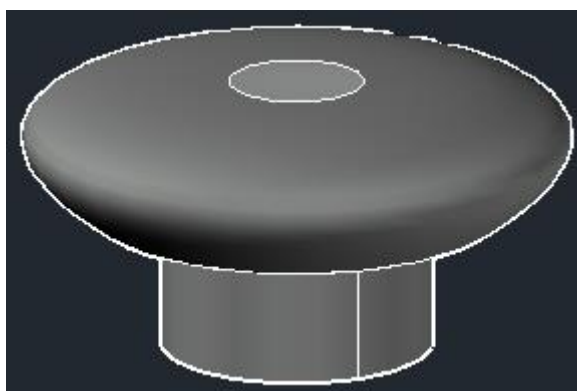


Рисунок 1.14 – Сформированная пимпочка

Разобранная модель в полной форме представлена на рисунке 1.15.



Рисунок 1.15 – Разобранная модель в полной форме

1.2 Твердотельные модели объектов ферзя

В отличие от поверхностного моделирования, твердотельное наиболее полно отражает реальные свойства моделируемых объектов, так как создаются с помощью твердотельных объектов[3].

Их можно разделить на основные группы:

- 1) простые тела (параллелепипед, конус, тор, клин, сфера, цилиндр);
- 2) выдавленные тела;
- 3) тела вращения;
- 4) составные тела.

В категорию выдавленных тел входят объекты, созданные при помощи команды «EXTRUDE», которая выдавливает контур объекта по заданному пути.

Что касается третьей группы, то они получаются при помощи команды «REVOLVE». Она позволяет вращать замкнутый объект вокруг оси.

Последняя группа получается путём модификации существующих простых объектов с помощью логических команд: «UNION» (объединение), «SUBTRACT» (вычитание), «INTERSECT» (пересечение).

В рамках курсовой работы для выбранной модели «ферзь» использовались такие примитивы, как цилиндр (кольца), лофтинг, а также выдавленное тело (основание).

Этапы создания твердотельной модели схожи с этапами создания поверхностной модели.

Первый этап заключается в постройке основания модели из выдавленного многоугольника с помощью команд “_POLYGON” и “_EXTRUDE” (рис. 1.16). Сначала в начальной точке строим многоугольник,

количество сторон и радиус которого указывает пользователь. И выдавливаем его на указанную пользователем высоту.

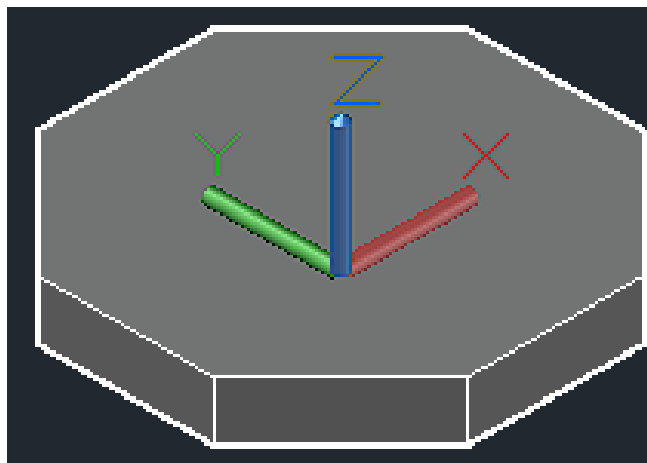


Рисунок 1.16 – Построение основания модели (количество сторон равняется 8)

Вторым этапом поверх основания модели строим подставку, состоящую из 2-х окружностей, радиусы (нижний радиус должен быть меньше верхнего) и расстояние между которыми указывает пользователь, построенных с помощью команды “_CIRCLE” и объединенных с помощью команды “_LOFT” (рис. 1.17).

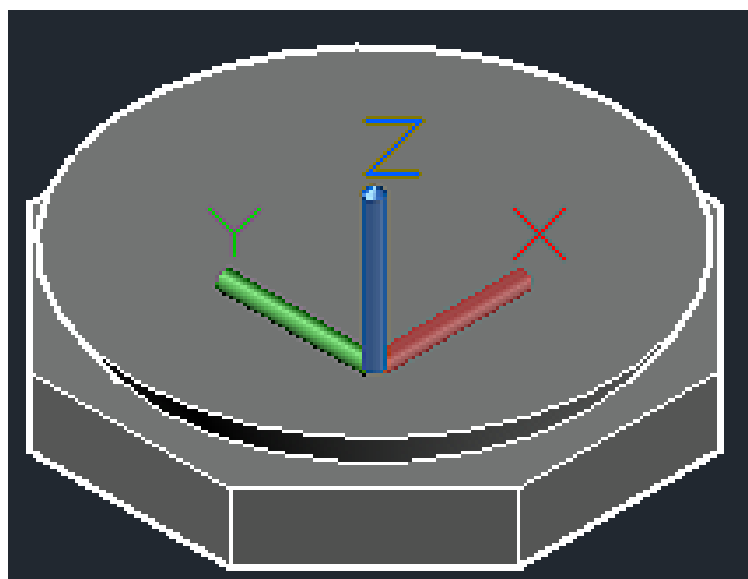


Рисунок 1.17 – Построение подставки

Третьим этапом поверх подставки модели строим юбочку, состоящую также из 2-х окружностей, радиусы (нижний радиус должен быть меньше верхнего) и расстояние между которыми указывает пользователь, построенных с помощью команды “_CIRCLE” и объединенных с помощью команды “_LOFT” (рис. 1.18).

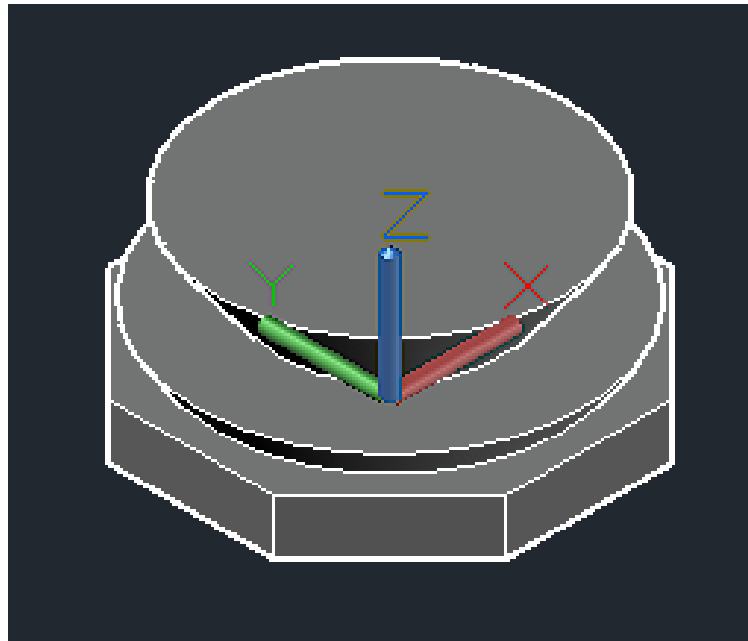


Рисунок 1.18 – Построение юбочки

Четвертым этапом поверх юбочки модели строим среднюю часть, состоящую из 3-х окружностей (две из которых одинаковые), радиусы (нижний радиус должен быть больше верхних) и 2 расстояния между которыми указывает пользователь, построенных с помощью команды “_CIRCLE” и объединенных с помощью команды “_LOFT” (рис. 1.19).

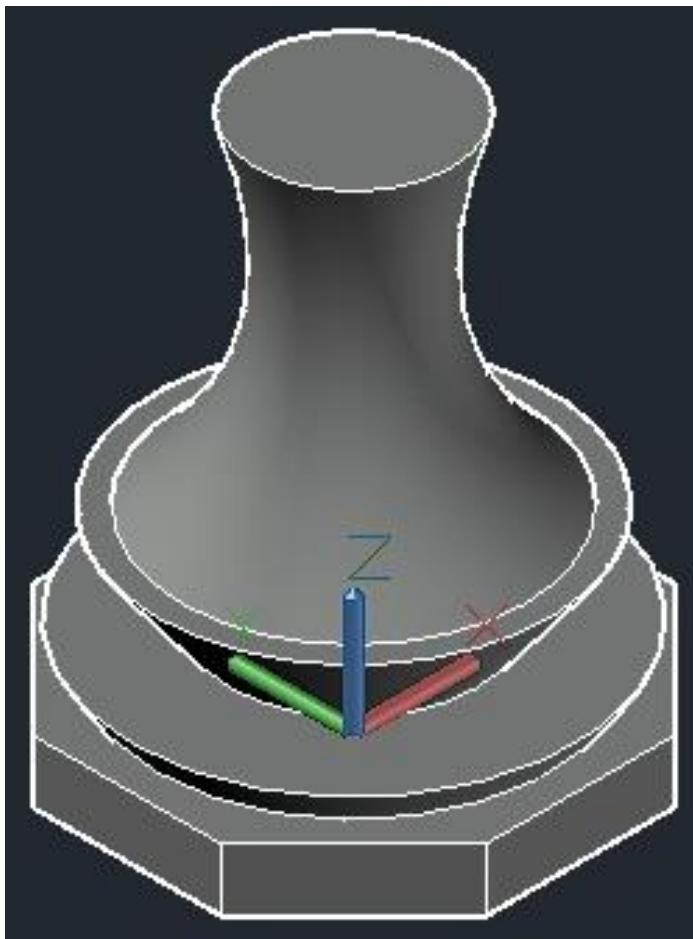


Рисунок 1.19 – Построение средней части

Пятым этапом поверх средней части модели строим кольца, 1, 2 или 3 штуки, в зависимости от того сколько выберет пользователь. Кольца строятся с помощью команд “_CYLINDER”, радиус и высоту которых указывает пользователь (рис. 1.20).

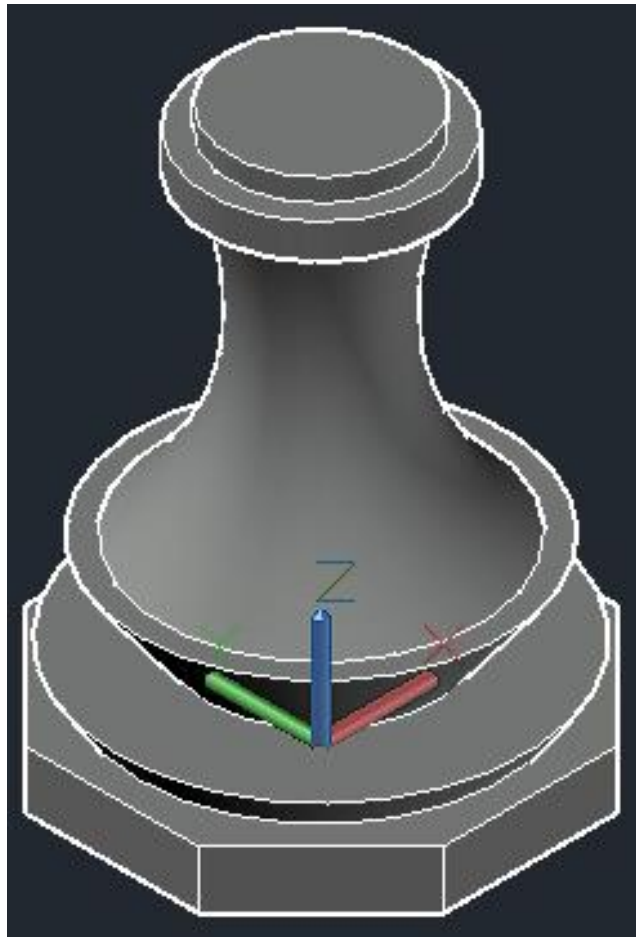


Рисунок 1.20 – Построение 2-х колец

Шестым этапом поверх колец (не зависима от количества) модели строим верхнюю часть, состоящую из 4-х окружностей, радиусы (средний радиус должен быть больше остальных) и 3 расстояния между которыми указывает пользователь, построенных с помощью команды “_CIRCLE” и объединенных с помощью команды “_LOFT” (рис. 1.21).

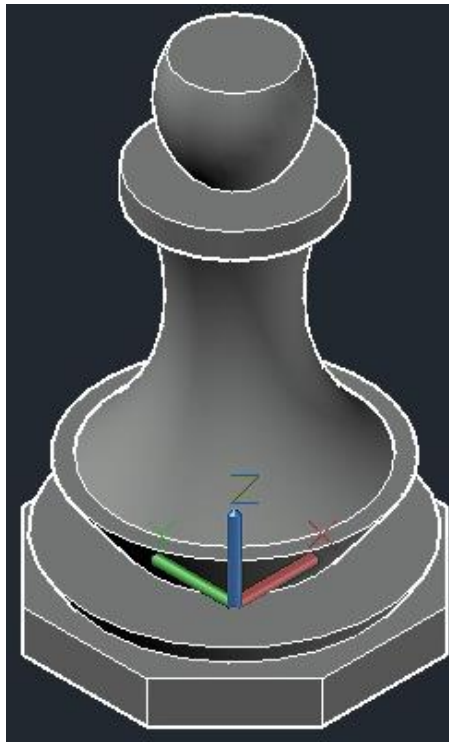


Рисунок 1.21 – Построение верхней части

Седьмым этапом поверх верхней части строим сферу, которая строится с помощью команды “_ SPHERE”, радиус которой указывает пользователь (рис. 1.22).

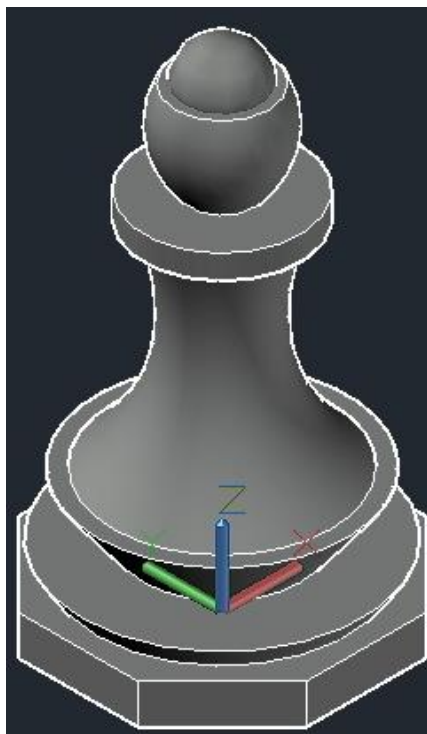


Рисунок 1.22 – Построение сферы

И последним восьмым этапом поверх сферы строим пимпочку, состоящую из ножки (цилиндра, радиус и высоту которого указывает пользователь) и верхушки (состоящую из 3-х окружностей, радиусы (средний радиус должен быть больше остальных) и 2 расстояния между которыми указывает пользователь), построенных с помощью команд “_ SPHERE”, “_ CIRCLE” и объединенных с помощью команды “_ LOFT” (рис. 1.23)



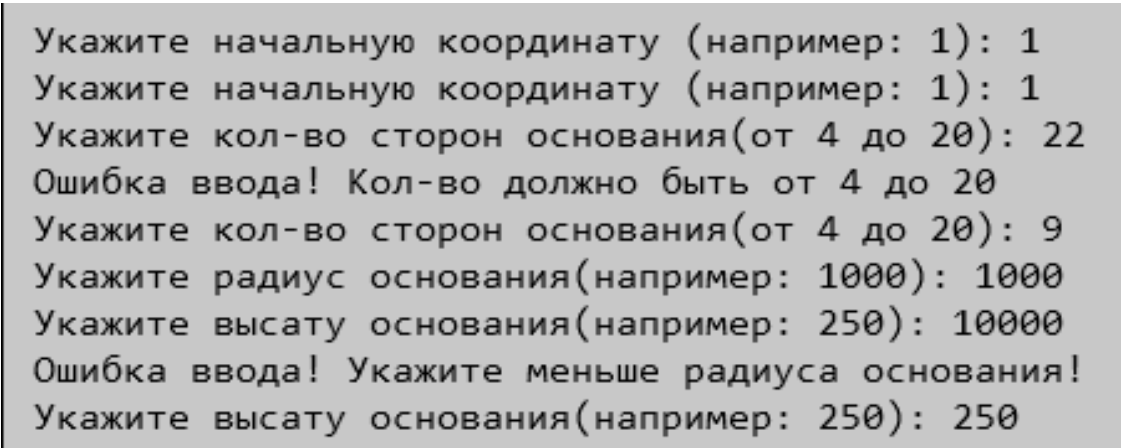
Рисунок 1.23– Построение пимпочки

2 ОПИСАНИЕ ВЫБРАННЫХ МЕТОДОВ И АЛГОРИТМОВ

Для отслеживания корректности вводимых данных были созданы две функции: `get_real` и `format`. Первая функция принимает в качестве параметров выводимую строку, а также диапазон допустимых значений (`min`, `max`). В случае, если вместо `min` или `max` передано -1, значит проверка на этот критерий пропускается. Внутри функции описан цикл, который каждый раз выводит пользователю форматированное сообщение, после чего запрашивается ограниченный с помощью команды «`initget (+ 1 2 4)`» ввод. Введённые данные сверяются с переданным диапазоном и в случае, если ввод корректный, цикл завершается, иначе в консоль выводится ошибка и цикл начинается с начала (рис 2.1).

Вторая функция создана для форматирования выводимых пользователю сообщений. Внутри неё производится проверка на наличие ограничений и формирование строки вида: «`Input {parameter} (from {min} to {max}):`».

В случае, когда проверка не требуется (например, при вводе координат), используется классическая функция AutoLISP «`getreal`».



```
Укажите начальную координату (например: 1): 1
Укажите начальную координату (например: 1): 1
Укажите кол-во сторон основания(от 4 до 20): 22
Ошибка ввода! Кол-во должно быть от 4 до 20
Укажите кол-во сторон основания(от 4 до 20): 9
Укажите радиус основания(например: 1000): 1000
Укажите высоту основания(например: 250): 10000
Ошибка ввода! Укажите меньше радиуса основания!
Укажите высоту основания(например: 250): 250
```

Рисунок 2.1 – Пример проверки корректности ввода

Программа начинает работу с функции «ferz». Вначале у пользователя запрашиваются все необходимые для пошагового моделирования данные (рис. 3.1):

- начальные координаты;
- количество сторон, радиус и высота основания;
- радиусы и высота подставки;
- радиусы и высота юбочки;
- радиусы и высоты средней части;
- количество колец (1-3);
- радиусы и высоты колец;
- радиусы и высоты верхней части;
- радиус сферы;
- радиус и высоты ножки(пимпочки);
- радиусы и высоты верхушки(пимпочки).

```
(initget 7)
(setq nx0 (getreal "\n Укажите начальную координату (например: 1): "))
(initget 7)
(setq ny0 (getreal "\n Укажите начальную координату (например: 1): "))
(initget 7)
(setq nz0 (getreal "\n Укажите начальную координату (например: 1): "))
(initget 7)
(setq k_s (getreal "\n Укажите кол-во сторон основания(от 4 до 20): "))
; (initget 7)
  (while (or (= 1 k_s) (= 2 k_s) (= 3 k_s) (< 20 k_s))
    (princ "\n Ошибка ввода! Кол-во должно быть от 4 до 20")
    (initget 7)
    (setq k_s (getreal "\n Укажите кол-во сторон основания(от 4 до 20): "))
    (initget 7)
  )
(setq ro_1 (getreal "\n Укажите радиус основания(например: 1000): "))
(initget 7)
(setq ho_1 (getreal "\n Укажите высату основания(например: 250): "))
  (while (or (= ro_1 ho_1) (< ro_1 ho_1))
    (princ "\n Ошибка ввода! Укажите меньше радиуса основания!")
    (initget 7)
    (setq ho_1 (getreal "\n Укажите высату основания(например: 250): "))
    (initget 7)
  )
```

Рисунок 3.1 – Ввод данных основания

После ввода строится призма с указанными размерами (рис. 3.2).

```
(command "_polygon" k_s (list nx0 ny0 nz0) "" ro_1)
(command "Выдавить" (entlast) "" ho_1)
(setq polygon1 (entlast))
(initget 7)
```

Рисунок 3.2 – Команды построения основания

Ввод всех данных проверяется на корректность (рис. 3.3).

```
(setq r_okr_1 (getreal "\n Укажите радиус нижней окружности подставки(например: 800): "))
(initget 7)
(setq r_okr_2 (getreal "\n Укажите радиус верхней окружности подставки(например: 900): "))
(while (or (= r_okr_1 r_okr_2)(< r_okr_2 r_okr_1))
  (princ "\n Ошибка ввода! Укажите больше нижней окружности подставки!")
  (initget 7)
  (setq r_okr_2 (getreal "\n Укажите радиус верхней окружности подставки(например: 900): ")))
(initget 7)
(setq h_pod (getreal "\n Укажите высоту подставки(например: 150): "))
(while (or (= h_pod ho_1)(< ho_1 h_pod))
  (princ "\n Ошибка ввода! Укажите меньше высоты основания!")
  (initget 7)
  (setq h_pod (getreal "\n Укажите высоту подставки(например: 150): ")))
(initget 7)
```

Рисунок 3.3 – Ввод данных подставки с проверкой на ошибки

После ввода строится подставка с указанными размерами (рис. 3.4).

```
(setq ho_1_1 (+ ny0 ho_1))
(setq ho_1_2 (+ ho_1_1 h_pod))
(command "_circle" (list nx0 ny0 ho_1_1) r_okr_1)
(setq circle_1 (entlast))
(command "_circle" (list nx0 ny0 ho_1_2) r_okr_2)
(setq circle_2 (entlast))

(command "_loft" circle_1 circle_2 "" "C")
(setq circle_1a (entlast))
(initget 7)
```

Рисунок 3.4 – Команды построения подставки

Юбочка, средняя и верхняя части и пимпочка строятся по тому же принципу что и подставка (рис. 3.5-3.8).

```
(setq r_okr_3 (getreal "\n Укажите радиус нижней окружности юбочки(например: 500): "))
(initget 7)
(setq r_okr_4 (getreal "\n Укажите радиус верхней окружности юбочки(например: 800): "))
(while (or (= r_okr_3 r_okr_4)(< r_okr_4 r_okr_3))
(princ "\n Ошибка ввода! Укажите больше радиуса нижней окружности юбочки!")
(initget 7)
(setq r_okr_4 (getreal "\n Укажите радиус верхней окружности юбочки(например: 800): "))
(initget 7)
(setq h_yb(getreal "\n Укажите высоту подставки(например: 400): "))
;-----
(setq h_yb1 (+ ho_1_2 h_yb))
(command "_circle" (list nx0 ny0 ho_1_2) r_okr_3)
(setq circle_4 (entlast))
(command "_circle" (list nx0 ny0 h_yb1) r_okr_4)
(setq circle_5 (entlast))
(command "_loft" circle_4 circle_5 "" "C")
(setq circle_2a (entlast))
```

Рисунок 3.5 – Ввод и построение юбочки

```
(setq r_okr_5 (getreal "\n Укажите радиус нижней окружности средней части(например: 700): "))
(initget 7)
(setq r_okr_6 (getreal "\n Укажите радиус верхней средней части(например: 400): "))
(while (or (= r_okr_5 r_okr_6)(< r_okr_5 r_okr_6))
(princ "\n Ошибка ввода! Укажите меньше радиуса нижней окружности средней части!")
(initget 7)
(setq r_okr_6 (getreal "\n Укажите радиус верхней средней части(например: 400): "))
(initget 7)
(setq h_sr_1(getreal "\n Укажите 1-ю высоту средней части(например: 400): "))
(initget 7)
(setq h_sr_2(getreal "\n Укажите 2-ю высоту средней части(например: 1000): "))
(while (or (= h_sr_1 h_sr_2)(< h_sr_2 h_sr_1))
(princ "\n Ошибка ввода! Укажите больше 1-й высоты средней части!")
(initget 7)
(setq h_sr_2(getreal "\n Укажите 2-ю высоту средней части(например: 1000): "))
(initget 7)
;-----
(setq h_sr_1_1 (+ h_yb1 h_sr_1))
(setq h_sr_2_1 (+ h_sr_1_1 h_sr_2))
(command "_circle" (list nx0 ny0 h_yb1) r_okr_5)
(setq circle_6 (entlast))
(command "_circle" (list nx0 ny0 h_sr_1_1) r_okr_6)
(setq circle_7 (entlast))
(command "_circle" (list nx0 ny0 h_sr_2_1) r_okr_6)
(setq circle_8 (entlast))
(command "_loft" circle_6 circle_7 circle_8 "" "C")
(setq circle_3a (entlast))
(initget 7)
```

Рисунок 3.6 – Ввод и построение средней части

```

(setq r_okr_7 (getreal "\n Укажите 1-й радиус верхней чати(например: 200): "))
(while (or (= r_okr_7 r_k_1)(< r_k_1 r_okr_7))
(princ "\n Ошибка ввода! Укажите меньше радиуса кольца")
(initget 7)
(setq r_okr_7 (getreal "\n Укажите 1-й радиус верхней чати(например: 200): ")))
(initget 7)
(setq h_okr_7 (getreal "\n Укажите 1-ю высоту верхней чати(например: 300): "))
(initget 7)
(setq r_okr_8 (getreal "\n Укажите 2-й радиус верхней чати(например: 250): "))
(while (or (= r_okr_7 r_okr_8)(< r_okr_8 r_okr_7))
(princ "\n Ошибка ввода! Укажите больше 1-го радиуса верхней чати")
(initget 7)
(setq r_okr_8 (getreal "\n Укажите 2-й радиус верхней чати(например: 250): ")))
(initget 7)
(setq h_okr_8 (getreal "\n Укажите 2-ю высоту верхней чати(например: 150): "))
(initget 7)
(setq r_okr_9 (getreal "\n Укажите 3-й радиус верхней чати(например: 350): "))
(while (or (= r_okr_9 r_okr_8)(< r_okr_9 r_okr_8))
(princ "\n Ошибка ввода! Укажите больше 2-го радиуса верхней чати")
(initget 7)
(setq r_okr_9 (getreal "\n Укажите 3-й радиус верхней чати(например: 350): ")))
(initget 7)
(setq h_okr_9 (getreal "\n Укажите 3-ю высоту верхней чати(например: 150): "))
(initget 7)
(setq r_okr_10 (getreal "\n Укажите 4-й радиус верхней чати(например: 300): "))
(while (or (= r_okr_9 r_okr_10)(< r_okr_9 r_okr_10))
(princ "\n Ошибка ввода! Укажите меньше 3-го радиуса верхней чати")
(initget 7)
(setq r_okr_10 (getreal "\n Укажите 3-й радиус верхней чати(например: 300): ")))
(initget 7)
(command "_circle" (list nx0 ny0 h_chast_1) r_okr_7)
(setq circle_9 (entlast))
(setq h_chast_2 (+ h_chast_1 h_okr_7))
(setq h_chast_3 (+ h_chast_2 h_okr_8))
(setq h_chast_4 (+ h_chast_3 h_okr_9))
(command "_circle" (list nx0 ny0 h_chast_2) r_okr_8)
(setq circle_10 (entlast))
(command "_circle" (list nx0 ny0 h_chast_3) r_okr_9)
(setq circle_11 (entlast))
(command "_circle" (list nx0 ny0 h_chast_4) r_okr_10)
(setq circle_12 (entlast))
(command "_loft" circle_9 circle_10 circle_11 circle_12 "" "C")
(setq circle_4a (entlast))

```

Рисунок 3.7 – Ввод и построение верхней части


```

(setq r_pim_1 (getreal "\n Укажите 1-й радиус пимпочки(например: 75): "))
(while (or (<= (* r_nogi 2) r_pim_1))
  (princ "\n Ошибка ввода! Укажите меньше радиуса сферы")
  (initget 7)
  (setq r_pim_1 (getreal "\n Укажите 1-й радиус пимпочки(например: 75): ")))
  (initget 7)
(setq h_pim_1_1 (getreal "\n Укажите 1-ю высоту пимпочки(например: 25): "))
(while (or (< (/ h_nogi 2) h_pim_1_1))
  (princ "\n Ошибка ввода! Укажите не больше половины высоты ножки")
  (initget 7)
  (setq h_pim_1_1 (getreal "\n Укажите 1-ю высоту пимпочки(например: 25): ")))
  (initget 7)
(setq r_pim_2 (getreal "\n Укажите 2-й радиус пимпочки(например: 100): "))
(while (or (< r_pim_2 r_pim_1))
  (princ "\n Ошибка ввода! Укажите больше 1-го радиуса пимпочки!")
  (initget 7)
  (setq r_pim_2 (getreal "\n Укажите 2-й радиус пимпочки(например: 100): ")))
  (initget 7)
(setq h_pim_2_1 (getreal "\n Укажите 2-ю высоту пимпочки(например: 25): "))
(while (or (< h_pim_1_1 h_pim_2_1))
  (princ "\n Ошибка ввода! Укажите не больше 1-й высоты пимпочки!")
  (initget 7)
  (setq h_pim_2_1 (getreal "\n Укажите 2-ю высоту пимпочки(например: 25): ")))
  (initget 7)
(setq r_pim_3 (getreal "\n Укажите 3-й радиус пимпочки(например: 50): "))
(while (or (< r_pim_2 r_pim_3))
  (princ "\n Ошибка ввода! Укажите меньше 2-го радиуса пимпочки!")
  (initget 7)
  (setq r_pim_3 (getreal "\n Укажите 3-й радиус пимпочки(например: 50): ")))
  (initget 7)
(setq h_pim_1 (+ h_nogi h_nogi_1))
(setq h_pim_2 (+ h_pim_1 h_pim_1_1))
(setq h_pim_3 (+ h_pim_2 h_pim_2_1))
(command "_circle" (list nx0 ny0 h_pim_1) r_pim_1)
(setq circle_13 (entlast))
(command "_circle" (list nx0 ny0 h_pim_2) r_pim_2)
(setq circle_14 (entlast))
(command "_circle" (list nx0 ny0 h_pim_3) r_pim_3)
(setq circle_15 (entlast))
(command "_loft" circle_13 circle_14 circle_15 "" "C")
(setq circle_5a (entlast))

```

Рисунок 3.8 – Ввод и построение пимпочки

После построения средней части строятся кольца, в зависимости от их количества (от 1 до 3) (рис. 3.9-3.12).

```

(setq k_k (getreal "\n Укажите кол-во колец(от 1 до 3): "))
(while (or (= 0 k_k)(< 3 k_k))
  (princ "\n Ошибка ввода! Кол-во должно быть от 1 до 3")
  (initget 7)
  (setq k_k (getreal "\n Укажите кол-во колец(от 1 до 3): ")))
  (initget 7)

```

Рисунок 3.9 – Ввод количества колец

```

(if (= k_k 1)
  (prompt
    (setq r_k_1 (getreal "\n Укажите радиус 1-го кольца(например: 500): "))
    (initget 7)
    (setq h_k_1 (getreal "\n Укажите высоту 1-го кольца(например: 150): "))
    (command "_cylinder" (list nx0 ny0 h_sr_2_1) r_k_1 h_k_1)
    (setq cyl_1 (entlast)))

```

Рисунок 3.10 – Ввод и построение 1-го кольца

```

(if (= k_k 2)
  (prompt
    (setq r_k_1 (getreal "\n Укажите радиус 1-го кольца(например: 500): "))
    (initget 7)
    (setq h_k_1 (getreal "\n Укажите высоту 1-го кольца(например: 150): "))
    (command "_cylinder" (list nx0 ny0 h_sr_2_1) r_k_1 h_k_1)
    (setq cyl_1 (entlast)))

  (setq h_kol_1 (+ h_sr_2_1 h_k_1))
  (setq r_k_2 (getreal "\n Укажите радиус 2-го кольца(например: 400): "))
  (while (or (< r_k_1 r_k_2))
    (princ "\n Ошибка ввода! Укажите меньше радиуса 1-го кольца!")
    (initget 7)
    (setq r_k_2 (getreal "\n Укажите радиус 2-го кольца(например: 400): "))
    (initget 7))
  (setq h_k_2 (getreal "\n Укажите высоту 2-го кольца(например: 100): "))
  (command "_cylinder" (list nx0 ny0 h_kol_1) r_k_2 h_k_2)
  (setq cyl_2 (entlast)))

```

Рисунок 3.11 – Ввод и построение 2-х колец

```

(if (= k_k 3)
  (prompt
    (setq r_k_1 (getreal "\n Укажите радиус 1-го кольца(например: 500): "))
    (initget 7)
    (setq h_k_1 (getreal "\n Укажите высоту 1-го кольца(например: 150): "))
    (command "_cylinder" (list nx0 ny0 h_sr_2_1) r_k_1 h_k_1)
    (setq cyl_1 (entlast))

    (setq h_kol_1 (+ h_sr_2_1 h_k_1))
    (setq r_k_2 (getreal "\n Укажите радиус 2-го кольца(например: 400): "))
    (while (or (< r_k_1 r_k_2))
      (princ "\n Ошибка ввода! Укажите меньше радиуса 1-го кольца!")
      (initget 7)
      (setq r_k_2 (getreal "\n Укажите радиус 2-го кольца(например: 400): ")))
      (initget 7)
    (setq h_k_2 (getreal "\n Укажите высоту 2-го кольца(например: 100): "))
    (command "_cylinder" (list nx0 ny0 h_kol_1) r_k_2 h_k_2)
    (setq cyl_2 (entlast))

    (setq h_kol_2 (+ h_kol_1 h_k_2))
    (setq r_k_3 (getreal "\n Укажите радиус 3-го кольца(например: 300): "))
    (while (or (< r_k_2 r_k_3))
      (princ "\n Ошибка ввода! Укажите меньше радиуса 2-го кольца!")
      (initget 7)
      (setq r_k_3 (getreal "\n Укажите радиус 3-го кольца(например: 300): ")))
      (initget 7)
    (setq h_k_3 (getreal "\n Укажите высоту 3-го кольца(например: 50): "))
    (command "_cylinder" (list nx0 ny0 h_kol_2) r_k_3 h_k_3)
    (setq cyl_3 (entlast))
  )
)

```

Рисунок 3.12 – Ввод и построение 3-х колец

Не зависимо от количества колец далее по порядку вводятся и строятся: верхняя часть (рис. 3.7), сфера (3.13), ножка (рис. 3.14) и пимпочка (рис. 3.8).

```

;
  (setq r_sph_1 (getreal "\n Укажите радиус сферы(например: 250): "))
  (initget 7)
  (while (or (= r_sph_1 r_okr_10)(< r_okr_10 r_sph_1))
    (princ "\n Ошибка ввода! Укажите меньше 4-го радиуса верхней чати")
    (initget 7)
    (setq r_sph_1 (getreal "\n Укажите радиус сферы(например: 250): ")))
    (initget 7)
  (command "_sphere" (list nx0 ny0 h_chast_4) r_sph_1)
  (setq sphere_1 (entlast))
;

```

Рисунок 3.13 – Ввод и построение сферы

```
(setq r_nogi (getreal "\n Укажите радиус ножки(например: 50): "))
  (while (or (= r_sph_1 r_nogi)(< r_sph_1 r_nogi))
    (princ "\n Ошибка ввода! Укажите меньше радиуса сферы чати")
    (initget 7)
    (setq r_nogi (getreal "\n Укажите радиус ножки(например: 50): ")))
  (initget 7)
(setq h_nogi_1 (getreal "\n Укажите высоту ножки(например: 50): "))
(while (or (= r_sph_1 h_nogi_1)(< r_sph_1 h_nogi_1))
  (princ "\n Ошибка ввода! Укажите меньше радиуса сферы")
  (initget 7)
  (setq h_nogi_1 (getreal "\n Укажите высоту ножки(например: 50): ")))
  (initget 7)
(setq h_nogi (+ h_chast_4 r_sph_1))
(command "_cylinder" (list nx0 ny0 h_nogi) r_nogi h_nogi_1)
(setq cyl_4 (entlast))
```

Рисунок 3.14 – Ввод и построение ножки

В процессе загрузки и выполнения программы выдаются сообщения о выполнении моделирования (рис. 4.1).

```
Укажите кол-во колец(от 1 до 3): 3
Укажите радиус 1-го кольца(например: 500): 500
Укажите высоту 1-го кольца(например: 150): 150
_cylinder
Центр основания или [3Т/2Т/ККР/Эллиптический]:
Радиус основания или [Диаметр] <25.0000>: 500
Высота или [2Точки/Конечная точка оси] <250.0000>: 150
Команда:
Укажите радиус 2-го кольца(например: 400): 400
Укажите высоту 2-го кольца(например: 100): 100
_cylinder
Центр основания или [3Т/2Т/ККР/Эллиптический]:
Радиус основания или [Диаметр] <500.0000>: 400
Высота или [2Точки/Конечная точка оси] <150.0000>: 100
Команда:
Укажите радиус 3-го кольца(например: 300): 300
Укажите высоту 3-го кольца(например: 50): 50
_cylinder
Центр основания или [3Т/2Т/ККР/Эллиптический]:
```

Рисунок 4.1 – Скриншот команд из консоли во время моделирования

Результатом выполнения функции является построенная модель ферзя (рис. 4.2).



Рисунок 4.2 – Результат выполнения функции

Так как в некоторых параметрах учитывается условия ввода значения, программой выполняется контроль значений (рис. 4.3).

```

Укажите кол-во колец(от 1 до 3): 3
Укажите радиус 1-го кольца(например: 500): 500
Укажите высоту 1-го кольца(например: 150): 150
_cylinder
Центр основания или [ЗТ/2Т/ККР/Эллиптический]:
Радиус основания или [Диаметр] <25.0000>: 500
Высота или [2Точки/Конечная точка оси] <250.0000>: 150
Команда:
    Укажите радиус 2-го кольца(например: 400): 400
    Укажите высоту 2-го кольца(например: 100): 100
    _cylinder
Центр основания или [ЗТ/2Т/ККР/Эллиптический]:
Радиус основания или [Диаметр] <500.0000>: 400
Высота или [2Точки/Конечная точка оси] <150.0000>: 100
Команда:
    Укажите радиус 3-го кольца(например: 300): 300
    Укажите высоту 3-го кольца(например: 50): 50
    _cylinder
Центр основания или [ЗТ/2Т/ККР/Эллиптический]:

```

Рисунок 4.3 – Проверка ввода

На рисунках 4.4–4.5 Представлены примеры фигур с различными сочетаниями размеров и параметров.

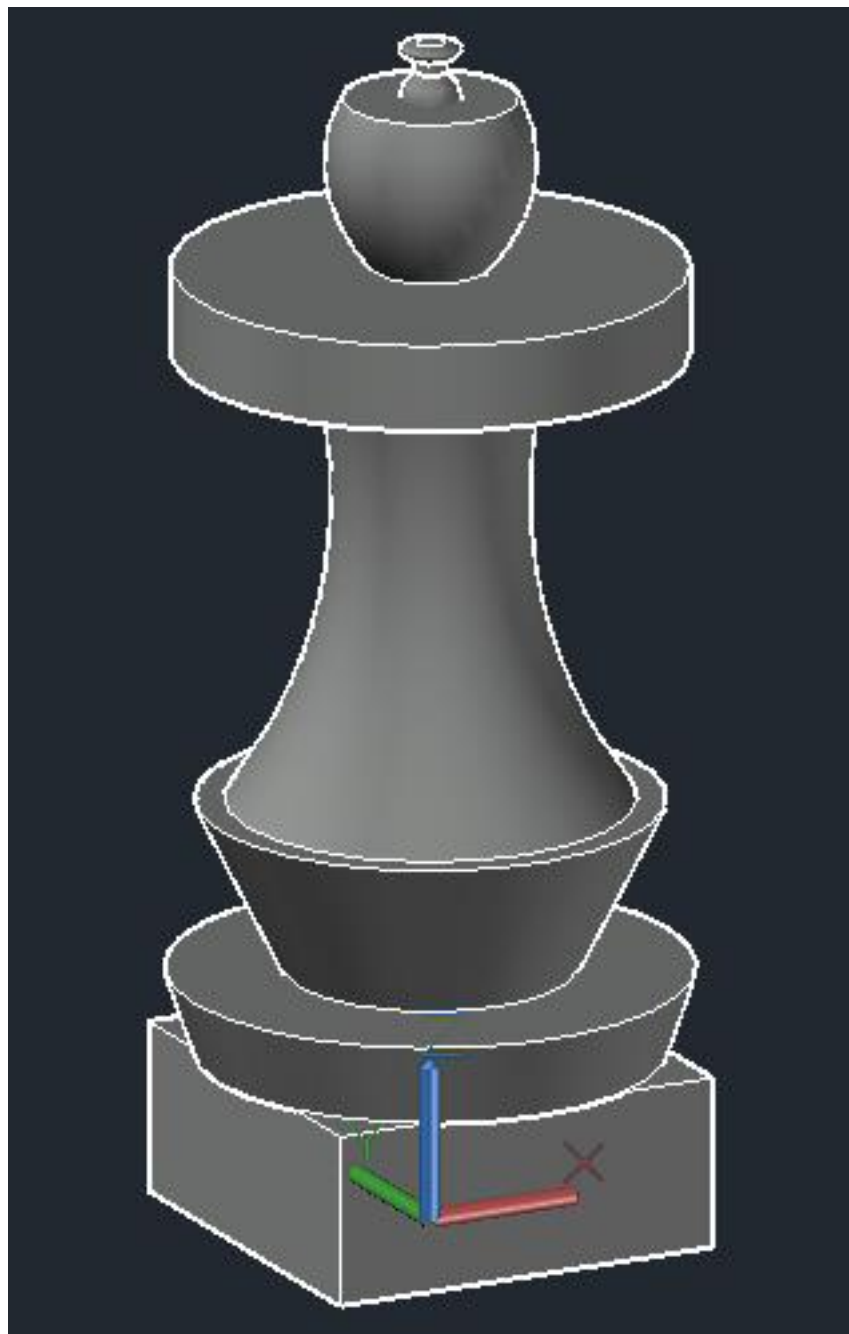


Рисунок 4.4 – Не стандартный ферзь с одним кольцом и четырехугольным основанием



Рисунок 4.5 – Не стандартный ферзь с двумя кольцами и 16-угольным основанием

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы были изучены и освоены методы создания трёхмерных поверхностных и твердотельных геометрических объектов с помощью AutoCAD, разработана интерактивная программа, автоматизирующая построение параметризованного твердотельного объекта (ферзь) с помощью средств AutoLISP.

В программе была реализована система для предотвращения некорректного ввода, а также добавлены ограничения для моделируемых объектов в зависимости от параметров, введенных ранее.

В ходе отладки были протестированы различные вариации параметров, в том числе ввод некорректных данных или значений, выходящих за границы диапазонов.

СПИСОК ЛИТЕРАТУРЫ

1. AutoCAD 2016. Самоучитель, Жарков (2011 г.);
2. AutoCAD Platform Customization: AutoLISP, Ambrosius Lee (2014 г.);
3. Трёхмерное моделирование деталей в САД-системах, Большаков, Бочков, Лячек (2015 г.).

ПРИЛОЖЕНИЕ А
ТЕХНИЧЕСКОЕ ЗАДАНИЕ

ГОУ ВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
ФАКУЛЬТЕТ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ И ПРОГРАММИРОВАНИЯ КАФЕДРА
КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ И ДИЗАЙНА

Направление подготовки: «Программная инженерия»

Курс 3

Группа ПИ-19а

Семестр 6

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

к курсовому проекту

по курсу «Архитектура и проектирование графических систем»

по теме: «Разработка программы для моделирования параметризованного
трёхмерного объекта в среде AutoCAD»



Рисунок А.1 – Объект «Ферзь»

Объект задаётся следующими параметрами:

- 1) Количество сторон основания;
- 2) Радиус основания;
- 3) Высота основания;
- 4) Нижний радиус подставки;
- 5) Верхний радиус подставки;
- 6) Высота подставки;
- 7) Нижний радиус юбочки;
- 8) Верхний радиус юбочки;
- 9) Высота юбочки;
- 10) Нижний радиус средней части;
- 11) Верхний радиус средней части;
- 12) Нижняя высота средней части;
- 13) Верхняя высота средней части;
- 14) Количество колец (1-3);
- 15) Радиусы колец;

- 16) Высоты колец;
- 17) Нижний радиус верхней части;
- 18) 2-й радиус верхней части;
- 19) 3-й радиус верхней части;
- 20) Верхний радиус верхней части;
- 21) Нижняя высота верхней части;
- 22) Средняя высота верхней части;
- 23) Верхняя высота верхней части;
- 24) Радиус сферы
- 25) Радиус ножки;
- 26) Высота ножки;
- 27) Нижний радиус пимпочки;
- 28) Средний радиус пимпочки;
- 29) Верхний радиус пимпочки;
- 30) Нижняя высота пимпочки;
- 31) Верхняя высота пимпочки.

Требования к модели:

- 1) Наличие базовой точки;
- 2) Проверка значений параметров на допустимость и непротиворечивость.

Содержание пояснительной записки:

- 1) Разработка поверхностей и твердотельной модели объекта;
- 2) Описание выбранных методов и алгоритмов;
- 3) Описание программы;
- 4) Пример выполнения программы.

Дата выдачи задания:

Задание принял:

ст. гр.ПИ-19а Сычева Е.В.

Руководители проекта:

зав. каф. Карабчевский В.В.

доц. каф. Доценко Г.В.

асс. каф. Бабакина А.А.

ПРИЛОЖЕНИЕ Б ЛИСТИНГ ПРОГРАММЫ

```
(defun c:ferz ()
  (command "_osnap" "_off")
  (setq os (getvar "osmode"))
  (setq os3 (getvar "3dosmode"))
  (setvar "osmode" 0)
  (setvar "3dosmode" 0)
  ;(command "_view" "_SWISO")
  ;-----начальная точки и основание-----
  (initget 7)
  (setq nx0 (getreal "\n Укажите начальную координату
(например: 1): "))
  (initget 7)
  (setq ny0 (getreal "\n Укажите начальную координату
(например: 1): "))
  (initget 7)
  (setq nz0 (getreal "\n Укажите начальную координату
(например: 1): "))
  (initget 7)
  (setq k_s (getreal "\n Укажите кол-во сторон
основания(от 4 до 20): "))
  ;(initget 7)
  (while (or (= 1 k_s)(= 2 k_s)(= 3 k_s)(< 20 k_s))
    (princ "\n Ошибка ввода! Кол-во должно быть от 4
до 20"))
  (initget 7)
  (setq k_s (getreal "\n Укажите кол-во сторон
основания(от 4 до 20): "))
  (initget 7)
  (setq ro_1 (getreal "\n Укажите радиус
основания(например: 1000): "))
  (initget 7)
  (setq ho_1 (getreal "\n Укажите высоту
основания(например: 250): "))
  (while (or (= ro_1 ho_1)(< ro_1 ho_1))
    (princ "\n Ошибка ввода! Укажите меньше радиуса
основания!"))
  (initget 7)
  (setq ho_1 (getreal "\n Укажите высоту
основания(например: 250): "))
  (initget 7)
  ;-----
  (command "_polygon" k_s (list nx0 ny0 nz0) "" ro_1)
  (command "Выдавить" (entlast) "" ho_1)
  (setq polygon1 (entlast))
  (initget 7)
  ;-----подставка-----
  (setq r_okr_1 (getreal "\n Укажите радиус нижней
окружности подставки (например: 800): "))
  (initget 7)
  (setq r_okr_2 (getreal "\n Укажите радиус верхней
окружности подставки (например: 900): "))
  (while (or (= r_okr_1 r_okr_2)(< r_okr_2 r_okr_1))
    (princ "\n Ошибка ввода! Укажите больше нижней
```

```
окружности подставки!"))
  (initget 7)
  (setq r_okr_2 (getreal "\n Укажите радиус верхней
окружности подставки(например: 900): "))
  (initget 7)
  (setq h_pod(getreal "\n Укажите высоту
подставки(например: 150): "))
  (while (or (= h_pod ho_1)(< ho_1 h_pod))
    (princ "\n Ошибка ввода! Укажите меньше высоты
основания!"))
  (initget 7)
  (setq h_pod(getreal "\n Укажите высоту
подставки(например: 150): "))
  (initget 7)
  ;-----
  (setq ho_1_1 (+ ny0 ho_1))
  (setq ho_1_2 (+ ho_1_1 h_pod))
  (command "_circle" (list nx0 ny0 ho_1_1) r_okr_1)
  (setq circle_1 (entlast))
  (command "_circle" (list nx0 ny0 ho_1_2) r_okr_2)
  (setq circle_2 (entlast))
  (command "_loft" circle_1 circle_2 "" "C")
  (setq circle_1a (entlast))
  (initget 7)
  ;-----
  (setq r_okr_3 (getreal "\n Укажите радиус нижней
окружности юбочки(например: 500): "))
  (initget 7)
  (setq r_okr_4 (getreal "\n Укажите радиус верхней
окружности юбочки(например: 800): "))
  (while (or (= r_okr_3 r_okr_4)(< r_okr_4 r_okr_3))
    (princ "\n Ошибка ввода! Укажите больше радиуса
нижней окружности юбочки!"))
  (initget 7)
  (setq r_okr_4 (getreal "\n Укажите радиус верхней
окружности юбочки(например: 800): "))
  (initget 7)
  (setq h_yb(getreal "\n Укажите высоту
подставки(например: 400): "))
  ;-----
  (setq h_yb1 (+ ho_1_2 h_yb))
  (command "_circle" (list nx0 ny0 ho_1_2) r_okr_3)
  (setq circle_4 (entlast))
  (command "_circle" (list nx0 ny0 h_yb1) r_okr_4)
  (setq circle_5 (entlast))
  (command "_loft" circle_4 circle_5 "" "C")
  (setq circle_2a (entlast))
  ;-----
  (setq r_okr_5 (getreal "\n Укажите радиус нижней
окружности средней части(например: 700): "))
  (initget 7)
  (setq r_okr_6 (getreal "\n Укажите радиус верхней
```

```

средней части(например: 400): ")
    (while (or (= r_okr_5 r_okr_6)(< r_okr_5 r_okr_6))
      (princ "\n Ошибка ввода! Укажите меньше радиуса
нижней окружности средней части!")
      (initget 7)
      (setq r_okr_6 (getreal "\n Укажите радиус верхней
средней части(например: 400): ")
      (initget 7)
      (setq h_sr_1(getreal "\n Укажите 1-ю высоту средней
части(например: 400): ")
      (initget 7)
      (setq h_sr_2(getreal "\n Укажите 2-ю высоту средней
части(например: 1000): ")
      (while (or (= h_sr_1 h_sr_2)(< h_sr_2 h_sr_1))
        (princ "\n Ошибка ввода! Укажите больше 1-й
высоты средней части!")
        (initget 7)
        (setq h_sr_2(getreal "\n Укажите 2-ю высоту средней
части(например: 1000): ")
        (initget 7)
        ;-----
        (setq h_sr_1_1 (+ h_yb1 h_sr_1))
        (setq h_sr_2_1 (+ h_sr_1_1 h_sr_2))
        (command "_circle" (list nx0 ny0 h_yb1) r_okr_5)
        (setq circle_6 (entlast))
        (command "_circle" (list nx0 ny0 h_sr_1_1) r_okr_6)
        (setq circle_7 (entlast))
        (command "_circle" (list nx0 ny0 h_sr_2_1) r_okr_6)
        (setq circle_8 (entlast))
        (command "_loft" circle_6 circle_7 circle_8 "" "C")
        (setq circle_3a (entlast))
        (initget 7)
        ;-----
        -----
        ;-----
        -----
        (setq k_k (getreal "\n Укажите кол-во колец(от 1 до 3):
"))
        (while (or (= 0 k_k)(< 3 k_k))
          (princ "\n Ошибка ввода! Кол-во должно быть от от
1 до 3")
          (initget 7)
          (setq k_k (getreal "\n Укажите кол-во колец(от 1 до
3): ")
          (initget 7)
          ;-----
          -----
          (if (= k_k 1)
            (prompt
              (setq r_k_1 (getreal "\n Укажите радиус 1-го
кольца(например: 500): ")
              (initget 7)
              (setq h_k_1 (getreal "\n Укажите высоту 1-го
кольца(например: 150): ")
              (command "_cylinder" (list nx0 ny0 h_sr_2_1) r_k_1
h_k_1)

```

```

      (setq cyl_1 (entlast))
      ;-----
      (setq h_chast_1 (+ h_sr_2_1 h_k_1))
      (initget 7)
      (setq r_okr_7 (getreal "\n Укажите 1-й радиус
верхней чати(например: 200): ")
      (while (or (= r_okr_7 r_k_1)(< r_k_1 r_okr_7))
        (princ "\n Ошибка ввода! Укажите меньше
радиуса кольца")
        (initget 7)
        (setq r_okr_7 (getreal "\n Укажите 1-й радиус
верхней чати(например: 200): )))
        (initget 7)
        (setq h_okr_7 (getreal "\n Укажите 1-ю высоту
верхней чати(например: 300): ")
        (initget 7)
        (setq r_okr_8 (getreal "\n Укажите 2-й радиус
верхней чати(например: 250): ")
        (while (or (= r_okr_7 r_okr_8)(< r_okr_8
r_okr_7))
          (princ "\n Ошибка ввода! Укажите больше 1-
го радиуса верхней чати")
          (initget 7)
          (setq r_okr_8 (getreal "\n Укажите 2-й радиус
верхней чати(например: 250): )))
          (initget 7)
          (setq h_okr_8 (getreal "\n Укажите 2-ю высоту
верхней чати(например: 150): ")
          (initget 7)
          (setq r_okr_9 (getreal "\n Укажите 3-й радиус
верхней чати(например: 350): ")
          (while (or (= r_okr_9 r_okr_8)(< r_okr_9 r_okr_8))
            (princ "\n Ошибка ввода! Укажите больше 2-
го радиуса верхней чати")
            (initget 7)
            (setq r_okr_9 (getreal "\n Укажите 3-й радиус
верхней чати(например: 350): )))
            (initget 7)
            (setq h_okr_9 (getreal "\n Укажите 3-ю высоту
верхней чати(например: 150): ")
            (initget 7)
            (setq r_okr_10 (getreal "\n Укажите 4-й радиус
верхней чати(например: 300): ")
            (while (or (= r_okr_9 r_okr_10)(< r_okr_9
r_okr_10))
              (princ "\n Ошибка ввода! Укажите меньше
3-го радиуса верхней чати")
              (initget 7)
              (setq r_okr_10 (getreal "\n Укажите 3-й
радиус верхней чати(например: 300): )))
              (initget 7)
              ;-----
              (command "_circle" (list nx0 ny0 h_chast_1) r_okr_7)
              (setq circle_9 (entlast))
              (setq h_chast_2 (+ h_chast_1 h_okr_7))
              (setq h_chast_3 (+ h_chast_2 h_okr_8))

```



```

(setq h_chast_4 (+ h_chast_3 h_okr_9))
(command "_circle" (list nx0 ny0 h_chast_2) r_okr8)
(setq circle_10 (entlast))
(command "_circle" (list nx0 ny0 h_chast_3) r_okr_9)
(setq circle_11 (entlast))
(command "_circle" (list nx0 ny0 h_chast_4)
r_okr_10)

(setq circle_12 (entlast))
(command "_loft" circle_9 circle_10 circle_11
circle_12 "" "C")
(setq circle_4a (entlast))
;-----
(setq r_sph_1 (getreal "\n Укажите радиус
сферы(например: 250): "))
(initget 7)
(while (or (= r_sph_1 r_okr_10)(< r_okr_10
r_sph_1))

(princ "\n Ошибка ввода! Укажите меньше
4-го радиуса верхней чати")

(initget 7)
(setq r_sph_1 (getreal "\n Укажите радиус
сферы(например: 250): )))
(initget 7)
(command "_sphere" (list nx0 ny0 h_chast_4)
r_sph_1)

(setq sphere_1 (entlast))
;-----
(setq r_nogi (getreal "\n Укажите радиус
ножки(например: 50): "))
(while (or (= r_sph_1 r_nogi)(< r_sph_1
r_nogi))

(princ "\n Ошибка ввода! Укажите меньше
радиуса сферы чати")

(initget 7)
(setq r_nogi (getreal "\n Укажите радиус
ножки(например: 50): )))
(initget 7)
(setq h_nogi_1 (getreal "\n Укажите высоту
ножки(например: 50): "))
(while (or (= r_sph_1 h_nogi_1)(< r_sph_1 h_nogi_1))
(princ "\n Ошибка ввода! Укажите меньше
радиуса сферы")

(initget 7)
(setq h_nogi_1 (getreal "\n Укажите высоту
ножки(например: 50): )))
(initget 7)
(setq h_nogi (+ h_chast_4 r_sph_1))
(command "_cylinder" (list nx0 ny0 h_nogi) r_nogi
h_nogi_1)
(setq cyl_4 (entlast))
;-----
(setq r_pim_1 (getreal "\n Укажите 1-й радиус
пимпочки(например: 75): "))
(while (or (<= (* r_nogi 2) r_pim_1))
(princ "\n Ошибка ввода! Укажите меньше
радиуса сферы")

```

```

(initget 7)
(setq r_pim_1 (getreal "\n Укажите 1-й
радиус пимпочки(например: 75): )))
(initget 7)
(setq h_pim_1_1 (getreal "\n Укажите 1-ю высоту
пимпочки(например: 25): "))
(while (or (< (/ h_nogi 2) h_pim_1_1))
(princ "\n Ошибка ввода! Укажите не
больше половины высоты ножки")
(initget 7)
(setq h_pim_1_1 (getreal "\n Укажите 1-ю
высоту пимпочки(например: 25): )))
(initget 7)
(setq r_pim_2 (getreal "\n Укажите 2-й радиус
пимпочки(например: 100): "))
(while (or (< r_pim_2 r_pim_1))
(princ "\n Ошибка ввода! Укажите больше 1-
го радиуса пимпочки!")
(initget 7)
(setq r_pim_2 (getreal "\n Укажите 2-й
радиус пимпочки(например: 100): )))
(initget 7)
(setq h_pim_2_1 (getreal "\n Укажите 2-ю высоту
пимпочки(например: 25): "))
(while (or (< h_pim_1_1 h_pim_2_1))
(princ "\n Ошибка ввода! Укажите не
больше 1-й высоты пимпочки!")
(initget 7)
(setq h_pim_2_1 (getreal "\n Укажите 2-ю
высоту пимпочки(например: 25): )))
(initget 7)
(setq r_pim_3 (getreal "\n Укажите 3-й радиус
пимпочки(например: 50): "))
(while (or (< r_pim_2 r_pim_3))
(princ "\n Ошибка ввода! Укажите меньше
2-го радиуса пимпочки!")
(initget 7)
(setq r_pim_3 (getreal "\n Укажите 3-й
радиус пимпочки(например: 50): )))
(initget 7)

(setq h_pim_1 (+ h_nogi h_nogi_1))
(setq h_pim_2 (+ h_pim_1 h_pim_1_1))
(setq h_pim_3 (+ h_pim_2 h_pim_2_1))
(command "_circle" (list nx0 ny0 h_pim_1) r_pim_1)
(setq circle_13 (entlast))
(command "_circle" (list nx0 ny0 h_pim_2) r_pim_2)
(setq circle_14 (entlast))
(command "_circle" (list nx0 ny0 h_pim_3) r_pim_3)
(setq circle_15 (entlast))
(command "_loft" circle_13 circle_14 circle_15 "" "C")
(setq circle_5a (entlast))
)
)
;-----

```

```

(if (= k_k 2)
  (prompt
    (setq r_k_1 (getreal "\n Укажите радиус 1-го
кольца(например: 500): "))
    (initget 7)
    (setq h_k_1 (getreal "\n Укажите высоту 1-го
кольца(например: 150): "))
    (command "_cylinder" (list nx0 ny0 h_sr_2_1) r_k_1
h_k_1)
    (setq cyl_1 (entlast))

    (setq h_kol_1 (+ h_sr_2_1 h_k_1))
    (setq r_k_2 (getreal "\n Укажите радиус 2-го
кольца(например: 400): "))
    (while (or (< r_k_1 r_k_2))
      (princ "\n Ошибка ввода! Укажите меньше
радиуса 1-го кольца!")
      (initget 7)
      (setq r_k_2 (getreal "\n Укажите радиус 2-го
кольца(например: 400): "))
      (initget 7)
      (setq h_k_2 (getreal "\n Укажите высоту 2-го
кольца(например: 100): "))
      (command "_cylinder" (list nx0 ny0 h_kol_1) r_k_2
h_k_2)
      (setq cyl_2 (entlast))

      (setq h_chast_1 (+ h_kol_1 h_k_2))
      (initget 7)
      (setq r_okr_7 (getreal "\n Укажите 1-й радиус
верхней чати(например: 200): "))
      (while (or (= r_okr_7 r_k_1)(< r_k_1 r_okr_7))
        (princ "\n Ошибка ввода! Укажите меньше
радиуса кольца")
        (initget 7)
        (setq r_okr_7 (getreal "\n Укажите 1-й радиус
верхней чати(например: 200): "))
        (initget 7)
        (setq h_okr_7 (getreal "\n Укажите 1-ю высоту
верхней чати(например: 300): "))
        (initget 7)
        (setq r_okr_8 (getreal "\n Укажите 2-й радиус
верхней чати(например: 250): "))
        (while (or (= r_okr_7 r_okr_8)(< r_okr_8
r_okr_7))
          (princ "\n Ошибка ввода! Укажите больше 1-
го радиуса верхней чати")
          (initget 7)
          (setq r_okr_8 (getreal "\n Укажите 2-й радиус
верхней чати(например: 250): "))
          (initget 7)
          (setq h_okr_8 (getreal "\n Укажите 2-ю высоту
верхней чати(например: 150): "))
          (initget 7)
          (setq r_okr_9 (getreal "\n Укажите 3-й радиус

```

```

верхней чати(например: 350): "))
      (while (or (= r_okr_9 r_okr_8)(< r_okr_9 r_okr_8))
        (princ "\n Ошибка ввода! Укажите больше 2-
го радиуса верхней чати")
        (initget 7)
        (setq r_okr_9 (getreal "\n Укажите 3-й радиус
верхней чати(например: 350): )))
        (initget 7)
        (setq h_okr_9 (getreal "\n Укажите 3-ю высоту
верхней чати(например: 150): "))
        (initget 7)
        (setq r_okr_10 (getreal "\n Укажите 4-й радиус
верхней чати(например: 300): "))
        (while (or (= r_okr_9 r_okr_10)(< r_okr_9
r_okr_10))
          (princ "\n Ошибка ввода! Укажите меньше
3-го радиуса верхней чати")
          (initget 7)
          (setq r_okr_10 (getreal "\n Укажите 3-й
радиус верхней чати(например: 300): )))
          (initget 7)
          ;-----
          (command "_circle" (list nx0 ny0 h_chast_1) r_okr_7)
          (setq circle_9 (entlast))
          (setq h_chast_2 (+ h_chast_1 h_okr_7))
          (setq h_chast_3 (+ h_chast_2 h_okr_8))
          (setq h_chast_4 (+ h_chast_3 h_okr_9))
          (command "_circle" (list nx0 ny0 h_chast_2) r_okr_8)
          (setq circle_10 (entlast))
          (command "_circle" (list nx0 ny0 h_chast_3) r_okr_9)
          (setq circle_11 (entlast))
          (command "_circle" (list nx0 ny0 h_chast_4)
r_okr_10)
          (setq circle_12 (entlast))
          (command "_loft" circle_9 circle_10 circle_11
circle_12 "" "C")
          (setq circle_4a (entlast))
          ;-----
          (setq r_sph_1 (getreal "\n Укажите радиус
сферы(например: 250): "))
          (initget 7)
          (while (or (= r_sph_1 r_okr_10)(< r_okr_10
r_sph_1))
            (princ "\n Ошибка ввода! Укажите меньше
4-го радиуса верхней чати")
            (initget 7)
            (setq r_sph_1 (getreal "\n Укажите радиус
сферы(например: 250): )))
            (initget 7)
            (command "_sphere" (list nx0 ny0 h_chast_4)
r_sph_1)
            (setq sphere_1 (entlast))
            ;-----
            (setq r_nogi (getreal "\n Укажите радиус
ножки(например: 50): "))
            (while (or (= r_sph_1 r_nogi)(< r_sph_1

```

```

r_nogi))

(princ "\n Ошибка ввода! Укажите меньше
радиуса сферы чати")

(initget 7)

(setq r_nogi (getreal "\n Укажите радиус
ножки(например: 50): "))

(initget 7)

(setq h_nogi_1 (getreal "\n Укажите высоту
ножки(например: 50): "))

(while (or (= r_sph_1 h_nogi_1)(< r_sph_1 h_nogi_1))

(princ "\n Ошибка ввода! Укажите меньше
радиуса сферы")

(initget 7)

(setq h_nogi_1 (getreal "\n Укажите высоту
ножки(например: 50): "))

(initget 7)

(setq h_nogi (+ h_chast_4 r_sph_1))

(command "_cylinder" (list nx0 ny0 h_nogi) r_nogi
h_nogi_1)

(setq cyl_4 (entlast))

;-----

(setq r_pim_1 (getreal "\n Укажите 1-й радиус
пимпочки(например: 75): "))

(while (or (<= (* r_nogi 2) r_pim_1))

(princ "\n Ошибка ввода! Укажите меньше
радиуса сферы")

(initget 7)

(setq r_pim_1 (getreal "\n Укажите 1-й
радиус пимпочки(например: 75): "))

(initget 7)

(setq h_pim_1_1 (getreal "\n Укажите 1-ю высоту
пимпочки(например: 25): "))

(while (or (< (/ h_nogi 2) h_pim_1_1))

(princ "\n Ошибка ввода! Укажите не
больше половины высоты ножки")

(initget 7)

(setq h_pim_1_1 (getreal "\n Укажите 1-ю
высоту пимпочки(например: 25): "))

(initget 7)

(setq r_pim_2 (getreal "\n Укажите 2-й радиус
пимпочки(например: 100): "))

(while (or (< r_pim_2 r_pim_1))

(princ "\n Ошибка ввода! Укажите больше 1-
го радиуса пимпочки!")

(initget 7)

(setq r_pim_2 (getreal "\n Укажите 2-й
радиус пимпочки(например: 100): "))

(initget 7)

(setq h_pim_2_1 (getreal "\n Укажите 2-ю высоту
пимпочки(например: 25): "))

(while (or (< h_pim_1_1 h_pim_1_2))

(princ "\n Ошибка ввода! Укажите не
больше 1-й высоты пимпочки!")

(initget 7)

(setq h_pim_2_1 (getreal "\n Укажите 2-ю
высоту пимпочки(например: 25): "))

```

```

(initget 7)

(setq r_pim_3 (getreal "\n Укажите 3-й радиус
пимпочки(например: 50): "))

(while (or (< r_pim_2 r_pim_3))

(princ "\n Ошибка ввода! Укажите меньше
2-го радиуса пимпочки!")

(initget 7)

(setq r_pim_3 (getreal "\n Укажите 3-й
радиус пимпочки(например: 50): "))

(initget 7)

(setq h_pim_1 (+ h_nogi h_nogi_1))

(setq h_pim_2 (+ h_pim_1 h_pim_1_1))

(setq h_pim_3 (+ h_pim_2 h_pim_2_1))

(command "_circle" (list nx0 ny0 h_pim_1) r_pim_1)

(setq circle_13 (entlast))

(command "_circle" (list nx0 ny0 h_pim_2) r_pim_2)

(setq circle_14 (entlast))

(command "_circle" (list nx0 ny0 h_pim_3) r_pim_3)

(setq circle_15 (entlast))

(command "_loft" circle_13 circle_14 circle_15 "" "C")

(setq circle_5a (entlast))

)

)

;-----

(if (= k_k 3)

(prompt

(setq r_k_1 (getreal "\n Укажите радиус 1-го
кольца(например: 500): "))

(initget 7)

(setq h_k_1 (getreal "\n Укажите высоту 1-го
кольца(например: 150): "))

(command "_cylinder" (list nx0 ny0 h_sr_2_1) r_k_1
h_k_1)

(setq cyl_1 (entlast))

(setq h_kol_1 (+ h_sr_2_1 h_k_1))

(setq r_k_2 (getreal "\n Укажите радиус 2-го
кольца(например: 400): "))

(while (or (< r_k_1 r_k_2))

(princ "\n Ошибка ввода! Укажите меньше
радиуса 1-го кольца!")

(initget 7)

(setq r_k_2 (getreal "\n Укажите радиус 2-го
кольца(например: 400): "))

(initget 7)

(setq h_k_2 (getreal "\n Укажите высоту 2-го
кольца(например: 100): "))

(command "_cylinder" (list nx0 ny0 h_kol_1) r_k_2
h_k_2)

(setq cyl_2 (entlast))

(setq h_kol_2 (+ h_kol_1 h_k_2))

(setq r_k_3 (getreal "\n Укажите радиус 3-го

```

```

кольца(например: 300): ")
    (while (or (< r_k_2 r_k_3))
      (princ "\n Ошибка ввода! Укажите меньше
радиуса 2-го кольца!")
      (initget 7)
      (setq r_k_3 (getreal "\n Укажите радиус 3-го
кольца(например: 300): ")))
    (initget 7)
    (setq h_k_3 (getreal "\n Укажите высоту 3-го
кольца(например: 50): "))
    (command "_cylinder" (list nx0 ny0 h_kol_2) r_k_3
h_k_3)
    (setq cyl_3 (entlast))

    (setq h_chast_1 (+ h_kol_2 h_k_3))
    (initget 7)
    (setq r_okr_7 (getreal "\n Укажите 1-й радиус
верхней чати(например: 200): "))
    (while (or (= r_okr_7 r_k_1)(< r_k_1 r_okr_7))
      (princ "\n Ошибка ввода! Укажите меньше
радиуса кольца")
      (initget 7)
      (setq r_okr_7 (getreal "\n Укажите 1-й радиус
верхней чати(например: 200): ")))
    (initget 7)
    (setq h_okr_7 (getreal "\n Укажите 1-ю высоту
верхней чати(например: 300): "))
    (initget 7)
    (setq r_okr_8 (getreal "\n Укажите 2-й радиус
верхней чати(например: 250): "))
    (while (or (= r_okr_7 r_okr_8)(< r_okr_8
r_okr_7))
      (princ "\n Ошибка ввода! Укажите больше 1-
го радиуса верхней чати")
      (initget 7)
      (setq r_okr_8 (getreal "\n Укажите 2-й радиус
верхней чати(например: 250): ")))
    (initget 7)
    (setq h_okr_8 (getreal "\n Укажите 2-ю высоту
верхней чати(например: 150): "))
    (initget 7)
    (setq r_okr_9 (getreal "\n Укажите 3-й радиус
верхней чати(например: 350): "))
    (while (or (= r_okr_9 r_okr_8)(< r_okr_9 r_okr_8))
      (princ "\n Ошибка ввода! Укажите больше 2-
го радиуса верхней чати")
      (initget 7)
      (setq r_okr_9 (getreal "\n Укажите 3-й радиус
верхней чати(например: 350): ")))
    (initget 7)
    (setq h_okr_9 (getreal "\n Укажите 3-ю высоту
верхней чати(например: 150): "))
    (initget 7)
    (setq r_okr_10 (getreal "\n Укажите 4-й радиус
верхней чати(например: 300): "))
    (while (or (= r_okr_9 r_okr_10)(< r_okr_9

```

```

r_okr_10))
      (princ "\n Ошибка ввода! Укажите меньше
3-го радиуса верхней чати")
      (initget 7)
      (setq r_okr_10 (getreal "\n Укажите 3-й
радиус верхней чати(например: 300): ")))
      (initget 7)
      ;-----
      (command "_circle" (list nx0 ny0 h_chast_1) r_okr_7)
      (setq circle_9 (entlast))
      (setq h_chast_2 (+ h_chast_1 h_okr_7))
      (setq h_chast_3 (+ h_chast_2 h_okr_8))
      (setq h_chast_4 (+ h_chast_3 h_okr_9))
      (command "_circle" (list nx0 ny0 h_chast_2) r_okr8)
      (setq circle_10 (entlast))
      (command "_circle" (list nx0 ny0 h_chast_3) r_okr_9)
      (setq circle_11 (entlast))
      (command "_circle" (list nx0 ny0 h_chast_4)
r_okr_10)
      (setq circle_12 (entlast))
      (command "_loft" circle_9 circle_10 circle_11
circle_12 "" "C")
      (setq circle_4a (entlast))
      ;-----
      (setq r_sph_1 (getreal "\n Укажите радиус
сферы(например: 250): "))
      (initget 7)
      (while (or (= r_sph_1 r_okr_10)(< r_okr_10
r_sph_1))
        (princ "\n Ошибка ввода! Укажите меньше
4-го радиуса верхней чати")
        (initget 7)
        (setq r_sph_1 (getreal "\n Укажите радиус
сферы(например: 250): ")))
      (initget 7)
      (command "_sphere" (list nx0 ny0 h_chast_4)
r_sph_1)
      (setq sphere_1 (entlast))
      ;-----
      (setq r_nogi (getreal "\n Укажите радиус
ножки(например: 50): "))
      (while (or (= r_sph_1 r_nogi)(< r_sph_1
r_nogi))
        (princ "\n Ошибка ввода! Укажите меньше
радиуса сферы чати")
        (initget 7)
        (setq r_nogi (getreal "\n Укажите радиус
ножки(например: 50): ")))
      (initget 7)
      (setq h_nogi_1 (getreal "\n Укажите высоту
ножки(например: 50): "))
      (while (or (= r_sph_1 h_nogi_1)(< r_sph_1 h_nogi_1))
        (princ "\n Ошибка ввода! Укажите меньше
радиуса сферы")
        (initget 7)
        (setq h_nogi_1 (getreal "\n Укажите высоту

```

```

ножки(например: 50): ""))
    (initget 7)
    (setq h_nogi (+ h_chast_4 r_sph_1))
    (command "_cylinder" (list nx0 ny0 h_nogi) r_nogi
h_nogi_1)
    (setq cyl_4 (entlast))
    ;-----
    (setq r_pim_1 (getreal "\n Укажите 1-й радиус
пимпочки(например: 75): "))
    (while (or (<= (* r_nogi 2) r_pim_1))
        (princ "\n Ошибка ввода! Укажите меньше
радиуса сферы")
        (initget 7)
        (setq r_pim_1 (getreal "\n Укажите 1-й
радиус пимпочки(например: 75): "")))
    (initget 7)
    (setq h_pim_1_1 (getreal "\n Укажите 1-ю высоту
пимпочки(например: 25): ""))
    (while (or (< (/ h_nogi 2) h_pim_1_1))
        (princ "\n Ошибка ввода! Укажите не
больше половины высоты ножки")
        (initget 7)
        (setq h_pim_1_1 (getreal "\n Укажите 1-ю
высоту пимпочки(например: 25): "")))
    (initget 7)
    (setq r_pim_2 (getreal "\n Укажите 2-й радиус
пимпочки(например: 100): ""))
    (while (or (< r_pim_2 r_pim_1))
        (princ "\n Ошибка ввода! Укажите больше 1-
го радиуса пимпочки!")
        (initget 7)
        (setq r_pim_2 (getreal "\n Укажите 2-й
радиус пимпочки(например: 100): "")))
    (initget 7)
    (setq h_pim_2_1 (getreal "\n Укажите 2-ю высоту
пимпочки(например: 25): ""))
    (while (or (< h_pim_1_1 h_pim_1_2))

```

```

        (princ "\n Ошибка ввода! Укажите не
больше 1-й высоты пимпочки!")
        (initget 7)
        (setq h_pim_2_1 (getreal "\n Укажите 2-ю
высоту пимпочки(например: 25): "")))
        (initget 7)
        (setq r_pim_3 (getreal "\n Укажите 3-й радиус
пимпочки(например: 50): ""))
        (while (or (< r_pim_2 r_pim_3))
            (princ "\n Ошибка ввода! Укажите меньше
2-го радиуса пимпочки!")
            (initget 7)
            (setq r_pim_3 (getreal "\n Укажите 3-й
радиус пимпочки(например: 50): "")))
        (initget 7)

```

```

        (setq h_pim_1 (+ h_nogi h_nogi_1))
        (setq h_pim_2 (+ h_pim_1 h_pim_1_1))
        (setq h_pim_3 (+ h_pim_2 h_pim_2_1))
        (command "_circle" (list nx0 ny0 h_pim_1) r_pim_1)
        (setq circle_13 (entlast))
        (command "_circle" (list nx0 ny0 h_pim_2) r_pim_2)
        (setq circle_14 (entlast))
        (command "_circle" (list nx0 ny0 h_pim_3) r_pim_3)
        (setq circle_15 (entlast))
        (command "_loft" circle_13 circle_14 circle_15 "" "C")
        (setq circle_5a (entlast))
    )
    )
    ;-----

    (setvar "osmode" os)
    (setvar "3dosmode" os3)
    )

```