



## Projektowanie Systemów CAD/CAM

---

### Operacja Shell

Dawid Maksymowski  
Maciej Piętka

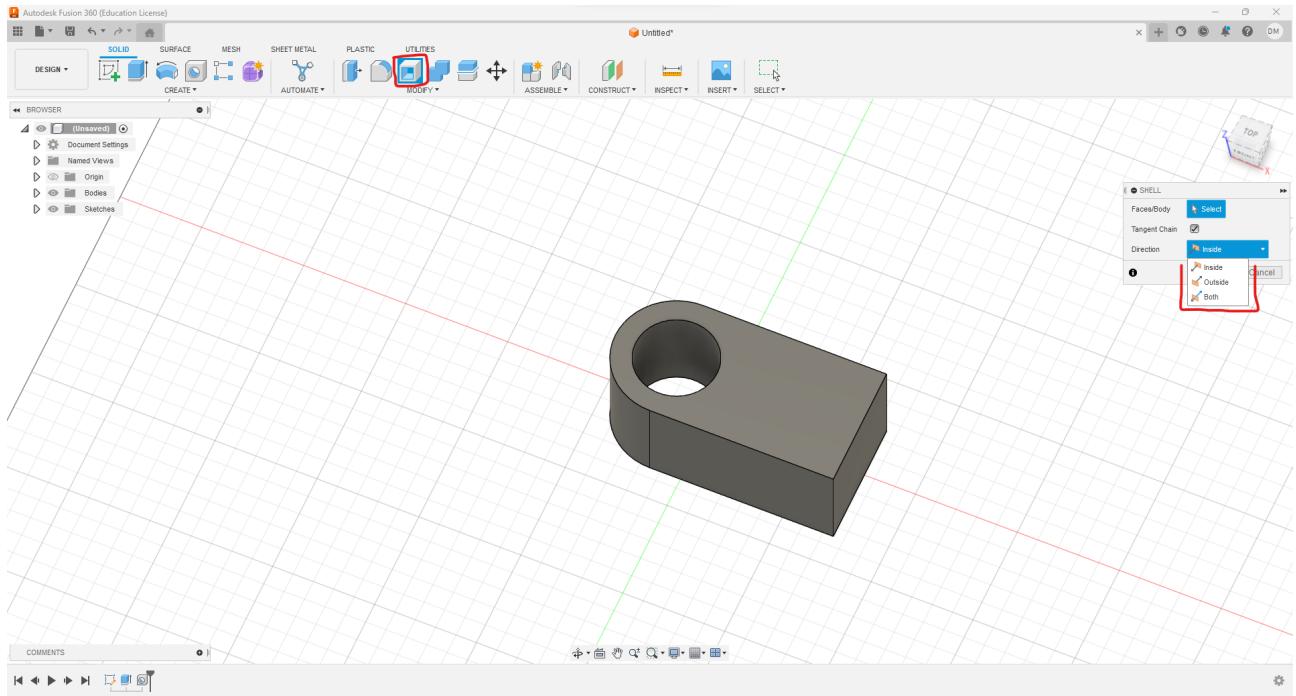
18 czerwca 2023

### Spis treści

<b>1 Operacja Shell w oprogramowaniach CAD</b>	<b>2</b>
1.1 Autodesk Fusion 360 . . . . .	2
1.2 Autodesk Inventor . . . . .	3
1.3 Solidworks . . . . .	4
1.4 FreeCAD . . . . .	6
<b>2 Model testowy</b>	<b>8</b>
<b>3 Operacja Shell w OpenCascade</b>	<b>8</b>
3.1 Parametry . . . . .	8
3.1.1 Mode . . . . .	9
3.1.2 Intersection . . . . .	9
3.1.3 Join . . . . .	9
3.2 Ograniczenia funkcji . . . . .	10
3.3 Aplikacja . . . . .	10
<b>4 Badanie Topologii</b>	<b>11</b>
4.1 Znajdowanie hierarchii . . . . .	11
4.2 Analiza topologii . . . . .	12
4.3 Różnice w topologii . . . . .	12

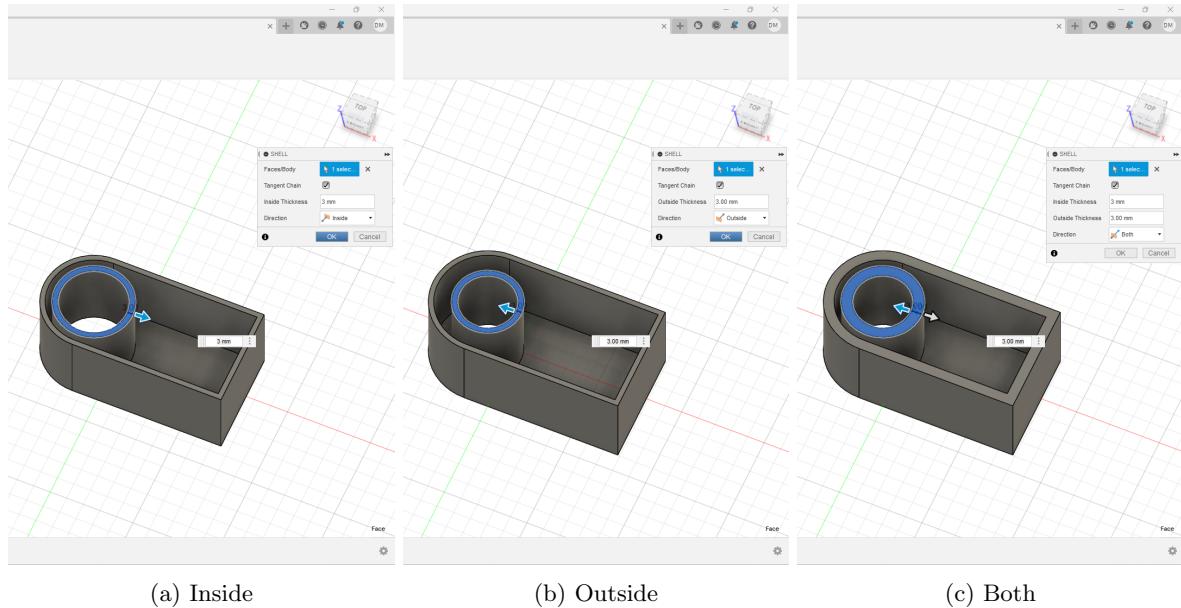
# 1 Operacja Shell w oprogramowaniach CAD

## 1.1 Autodesk Fusion 360



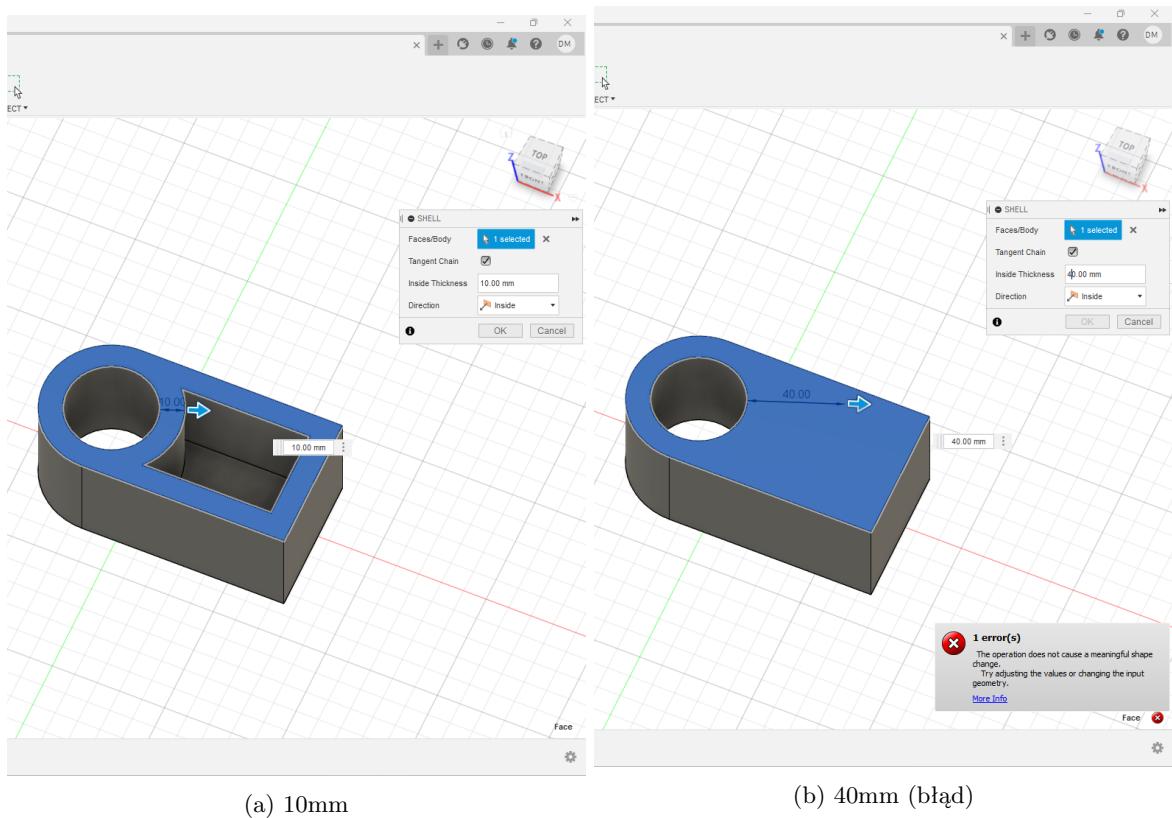
Rysunek 1: Shell w programie Autodesk Fusion 360.

W programie *Autodesk Fusion 360* funkcja Shell jest dostępna z poziomu zakładki „Solid” i ma 3 tryby do wyboru: **Inside**, **Outside**, **Both**. W przypadku trybu **Inside**, skorupa jest tworzona poprzez pogrubienie ściany modelu w kierunku środka modelu; w przypadku trybu **Outside** - ściany są pogrubiane w kierunku na zewnątrz; natomiast w trybie **Both** - w obie strony.



Rysunek 2: Porównanie efektów funkcji Shell dla trybów Inside, Outside, Both dla wartości 3mm.

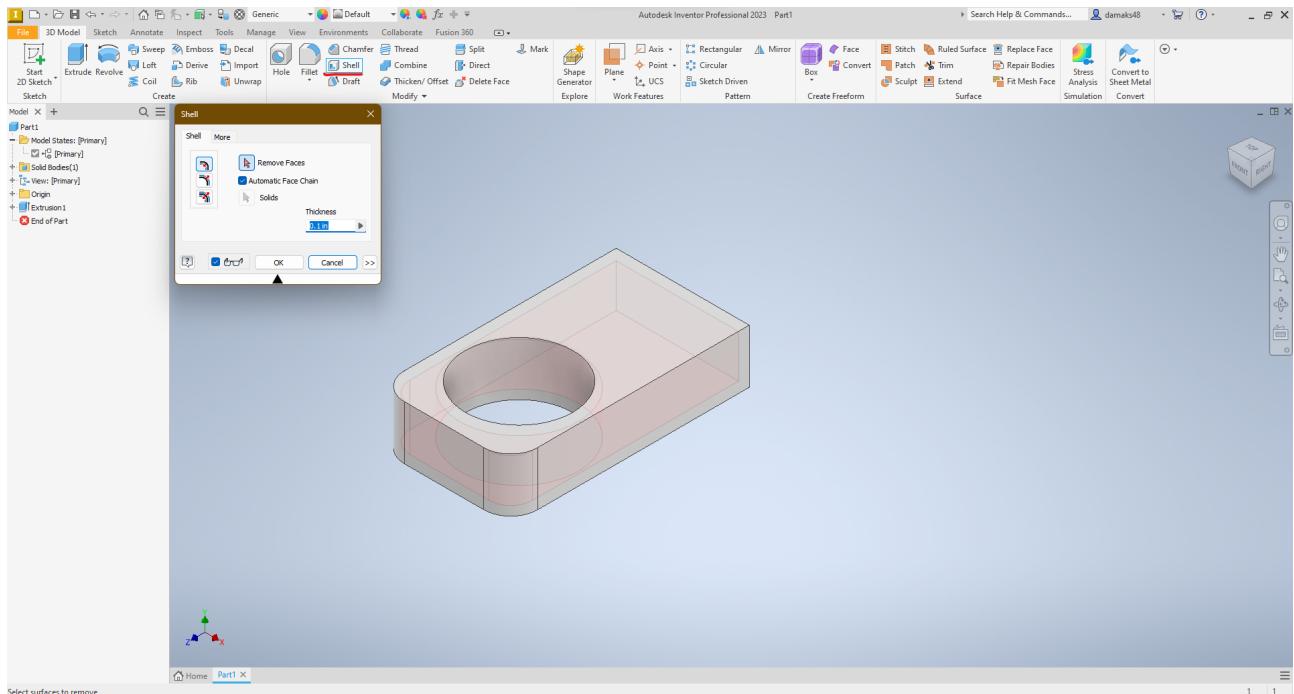
Operacja wykonuje się poprawnie zawsze, jeśli jej konsekwencją jest jakakolwiek zmiana kształtu. Jeżeli podamy zbyt dużą grubość wycięcia, otrzymamy komunikat informujący, że podana wartość jest nieprawidłowa, jak na rys. 3b.



Rysunek 3: Porównanie udanej i nieudanej próby wycięcia skorupy.

## 1.2 Autodesk Inventor

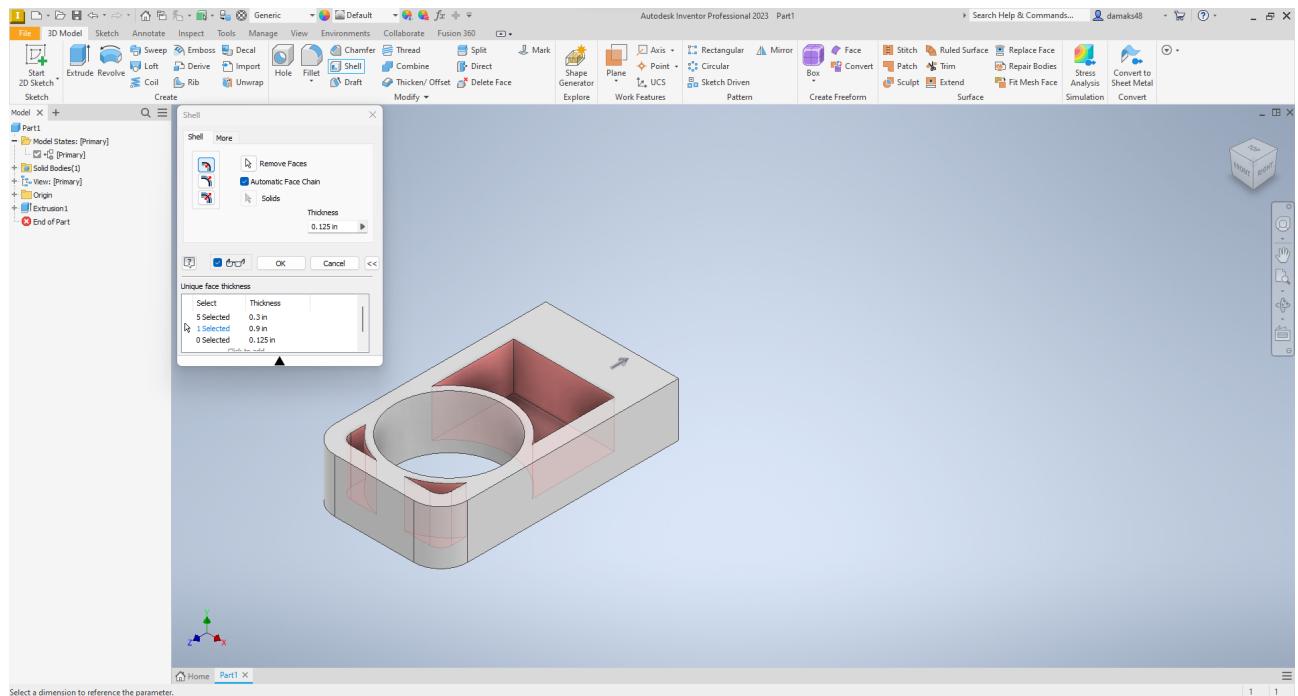
Program Autodesk Inventor wyróżnia się bardzo wyraźnym półprzezroczystym rysowaniem ścian wewnętrznych skorupy. Jest to szczególnie przydatne, jeśli wydrążane jest jedynie wnętrze - tzn. nie wybieramy ściany, która zostanie usunięta (rys. 4).



Rysunek 4: Shell bez wyboru ściany w programie Autodesk Inventor.

Oprócz 3 trybów jak w Fusion 360, program umożliwia również ustawienie grubości osobno dla każdej ze ścian. Na rys. 5 zdjęcie skorupy modelu z ustawionymi grubościami: 0.3in dla tylnej ściany (wskazywanej przez

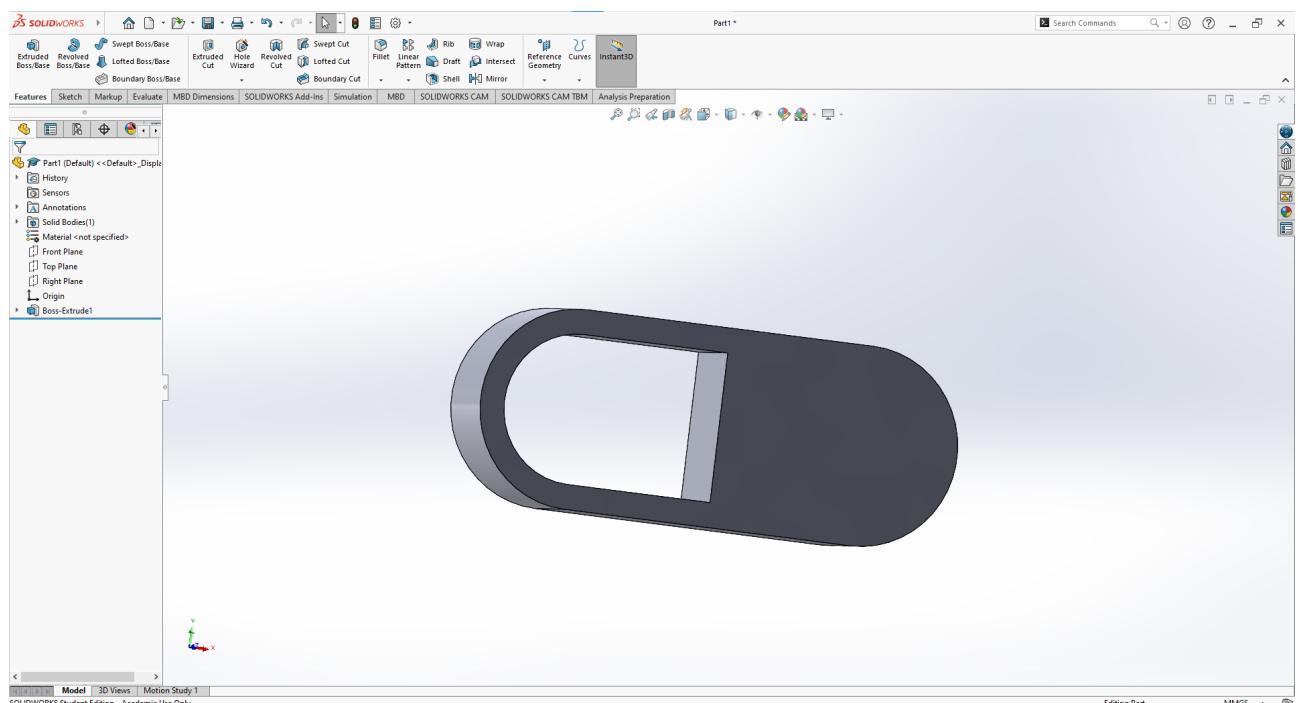
strzałkę), 0.9in dla reszty ścian bocznych oraz 0.125in dla pozostałych ścian (dolna oraz ścianka cylindra w środku).



Rysunek 5: Shell z różnym wyborem grubości ścian w programie Autodesk Inventor.

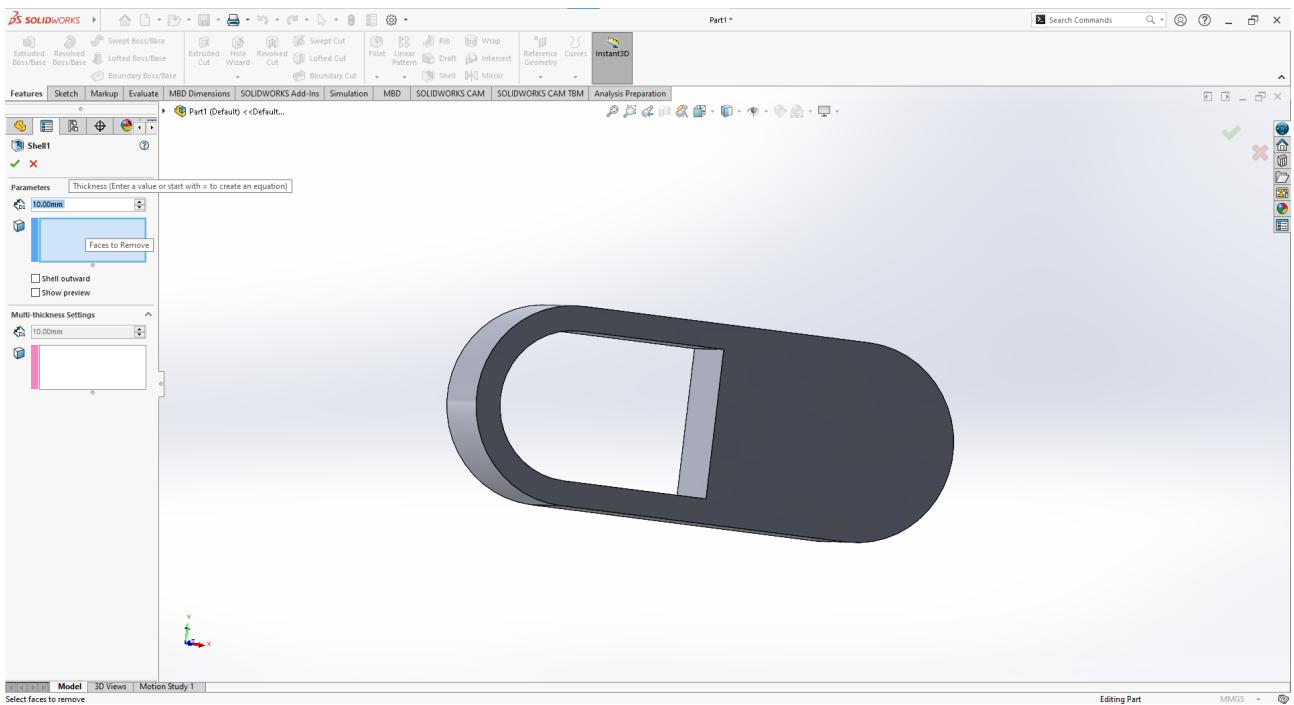
### 1.3 Solidworks

W programie Solidworks testy zostały wykonane na modelu zaprezentowanym na rys. 6.

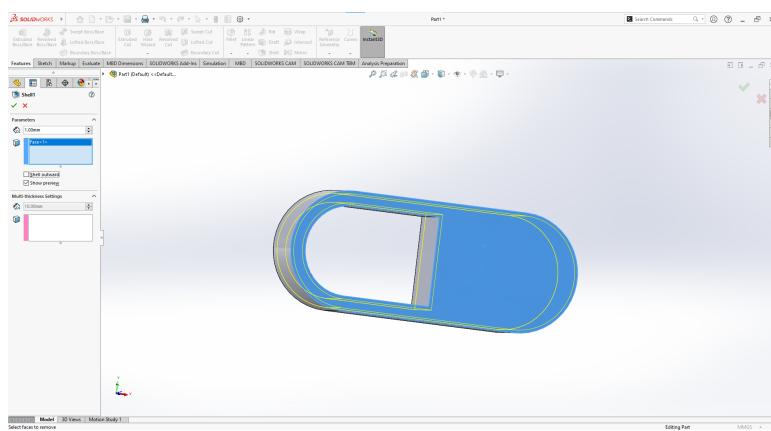


Rysunek 6: Model użyty w programie Solidworks.

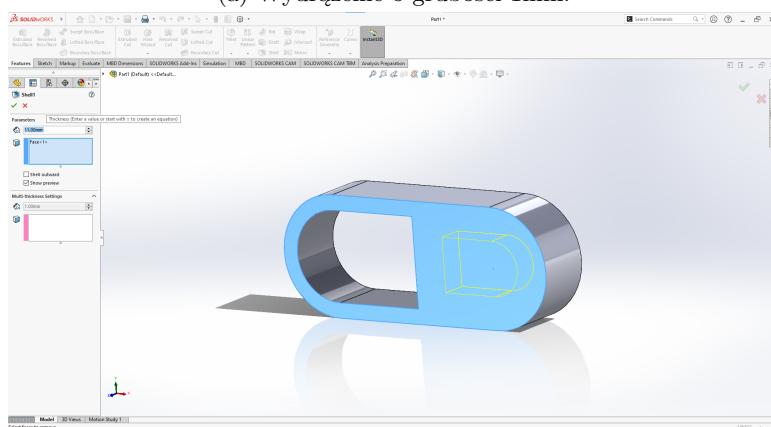
Program udostępnia funkcję tworzenia skorupy obiektu/części. Na rys. 7 przedstawiony jest interfejs do tego przeznaczony – program pozwala wybrać grubość skorupy oraz ścianę od której obiekt ma być wydrążony. Rys. 8 przedstawia wyniki operacji.



Rysunek 7: Solidworks - interfejs funkcji Shell.



(a) Wydrążenie o grubości 1mm.

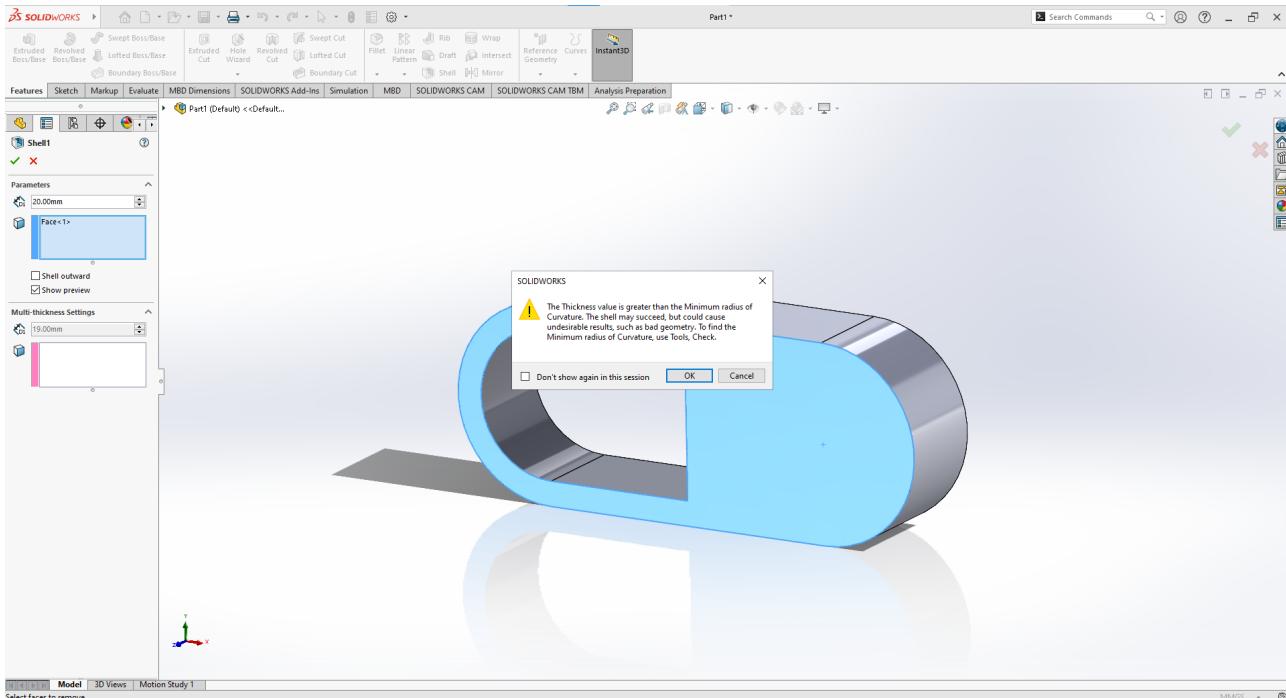


(b) Wydrążenie o grubości 11mm.

Rysunek 8: Wyniki operacji Shell w Solidworks.

Jak można zauważyć, odpowiednia grubość skorupy nie wykona wydrążenia małych części modelu. W przypadku przedstawionym na rys. 8b grubość została ustawiona na 11mm, co sprawiło że cienkie obramowanie nie zostało wydrążone.

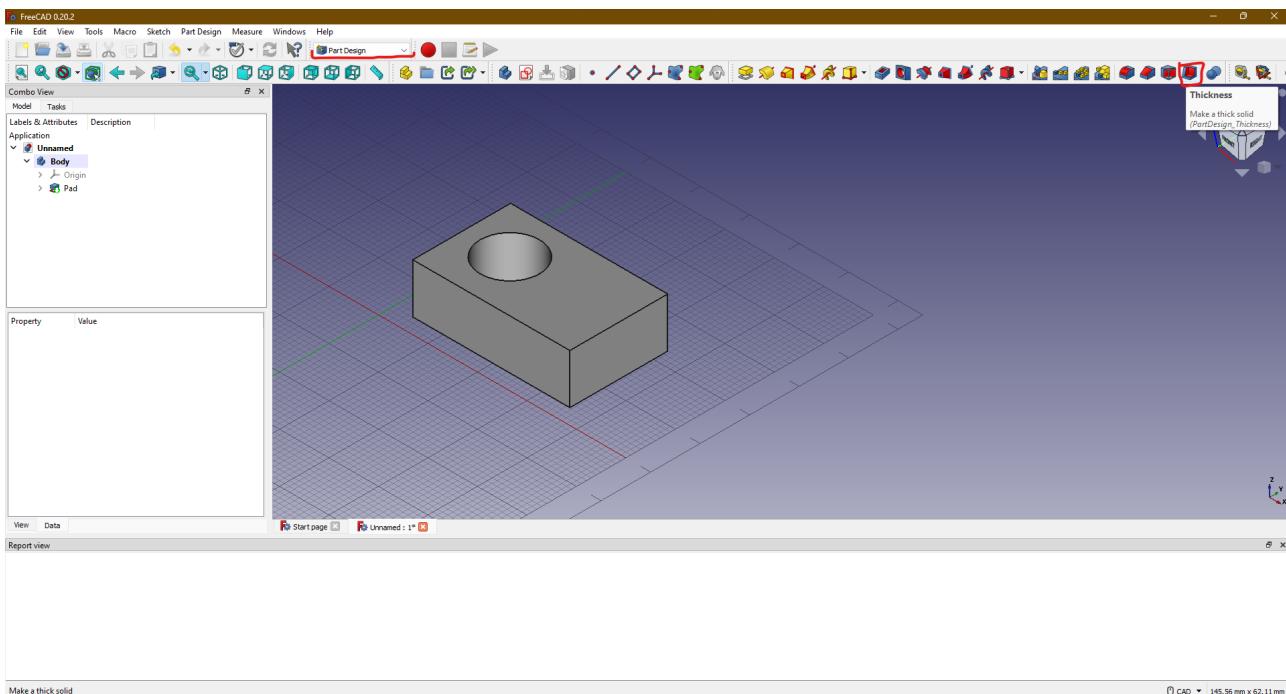
Jeżeli wybrana grubość będzie zbyt duża, pojawi się komunikat informujący o tym użytkownika (rys. 9).



Rysunek 9: Komunikat o błędzie.

## 1.4 FreeCAD

W przypadku oprogramowania FreeCAD funkcjonalność Shell widnieje pod nazwą „Thickness” w narzędziu „Part Design”.



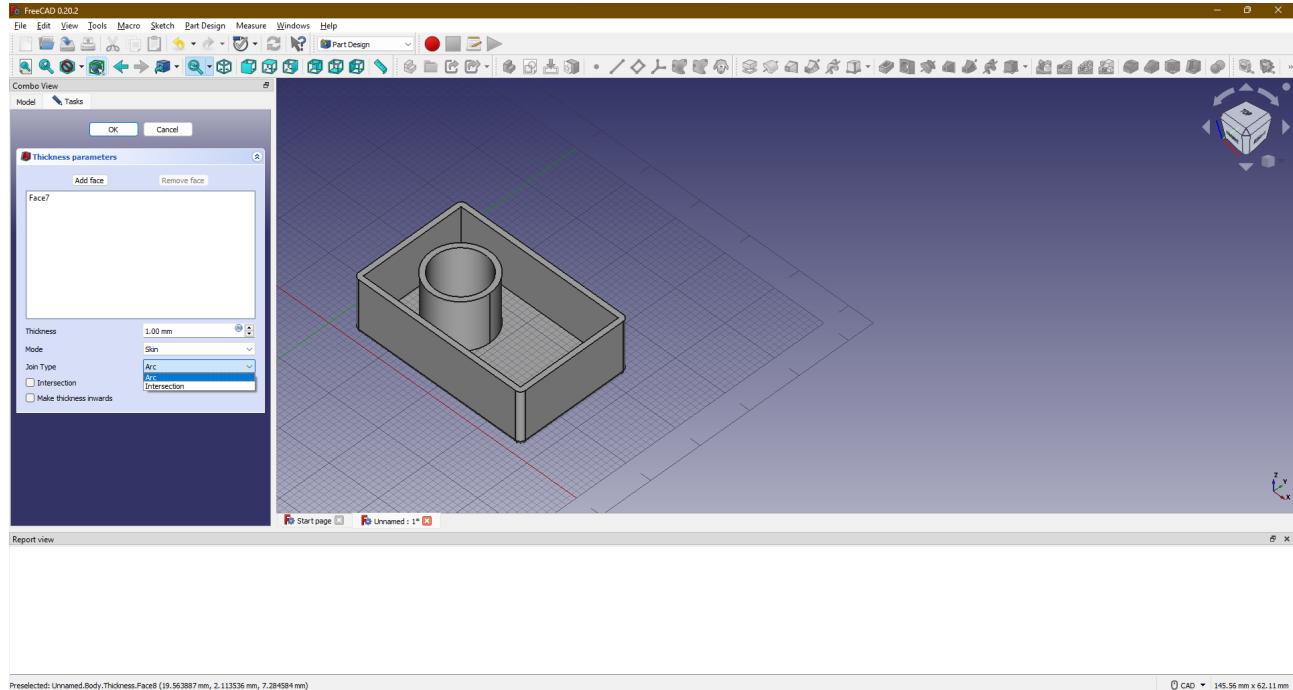
Rysunek 10: Funkcjonalność Thickness w programie FreeCAD.

Stworzenie skorupy jest możliwe tylko po zaznaczeniu którejś ze ścian - niestety nie jest możliwe stworzenie skorupy tylko wewnętrz modelu.

Interfejs domyślnie tworzy skorupę, pogrubiając ściany „do zewnątrz” - mamy tutaj 2 możliwości („Join type”):

- **Arc** (domyślnie) - rogi między ścianami są zaokrąglone,
- **Intersection** - rogi są kanciaste (podobnie jak w innych systemach).

Checkbox „Make thickness inwards” jest odpowiednikiem opcji „Outside” z programu Fusion 360.



Rysunek 11: Wydrażony obiekt w programi FreeCAD.

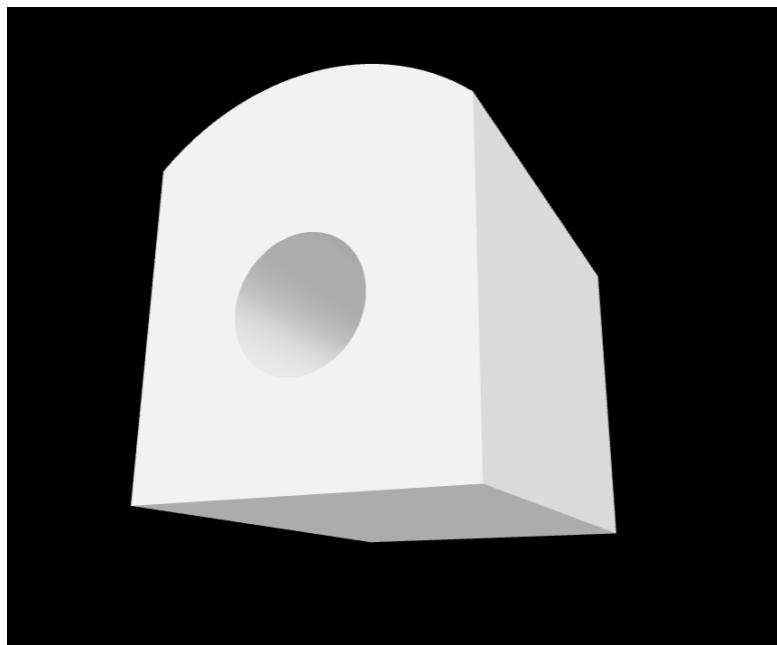
Niestety w tym programie nie jest możliwe wykonanie operacji na bryle, która posiada zaokrąglone rogi funkcją *Fillet*. Ponadto program FreeCAD umożliwia jedynie ustawienie takiej grubości skorupy, która nie spowoduje połączenia się przeciwnie skierowanych ścian obiektu. Podanie większej grubości spowoduje całkowite wypełnienie obiektu (brak efektu funkcjonalności).

## 2 Model testowy

Za pomocą biblioteki OpenCascade, stworzony został proceduralnie model, na którym przeprowadzone zostały testy. Model testowy został stworzony za pomocą następujących kroków:

1. Zostało wyznaczonych pięć punktów  $P_1 = (0, 0, 0)$ ;  $P_2 = (5, 0, 0)$ ;  $P_3 = (5, 5, 0)$ ;  $P_4 = (0, 5, 0)$ ; oraz  $P_c = (2.5, 6, 0)$ ;
2. Utworzono zostały krawędzie  $P_1P_2$ ,  $P_2P_3$ ,  $P_4P_1$ , oraz łuk okręgu przechodzący przez punkty  $P_3$ ,  $P_c$  i  $P_4$
3. Z tak zdefiniowanych krawędzi utworzona została ściana, która została wydłużona w bryłę o 7 jednostek wzduż osi Z.
4. Stworzony został cylinder o promieniu 1, wysokości 10 oraz środku w  $(2.5, 2.5, 0)$
5. Wykonana została operacja odjęcia cylindra od bryły

Wyrenderowany model znajduje się na Rysunku 12.



Rysunek 12: Wyrenderowany model, na którym były przeprowadzane testy.

## 3 Operacja Shell w OpenCascade

Biblioteka OpenCascade udostępnia funkcjonalność operacji Shell pod funkcją o nazwie `MakeThickSolidByJoin()`. W tej sekcji opisane zostały parametry funkcji, ograniczenia użycia oraz zaprezentowana została aplikacja prezentująca tę funkcjonalność.

### 3.1 Parametry

Poniżej znajduje się lista parametrów przyjmowanych przez funkcję, z których najistotniejsze są pierwsze 4.

Operacja Shell jest wykonywana na kształcie `S`; skorupa o grubości `Offset` jest „wydrążana” przez ściany określone w `ClosingFaces` z tolerancją numeryczną `Tol`. Pozostałe parametry dookreślają działanie algorytmu i zostały opisane w kolejnej części dokumentu.

- `TopoDS_Shape& S` – kształt, na którym ma być wykonana operacja,
- `TopTools_ListOfShape& ClosingFaces` – lista ścian bryły, która zostanie usunięta przed wykonaniem operacji.
- `Standard_Real Offset` – grubość ścian w kształcie wynikowym.
- `Standard_Real Tol` – kryterium tolerancji obliczeń.

- `BRepOffset_Mode` Mode
- `Standard_Boolean Intersection`
- `Standard_Boolean SelfInter` – funkcjonalność niezaimplementowana.
- `GeomAbs_JoinType Join`
- `Standard_Boolean RemoveIntEdges` – flaga określająca, czy w kształcie wynikowym mają zostać usunięte wewnętrzne krawędzie.

### 3.1.1 Mode

Parametr `Mode` określa, w jaki sposób są przetwarzane (jeżeli w ogóle) tzw. krawędzie wolne (tj. niebędące częścią żadnej ściany) kształtu wejściowego. Aktualnie parametr jest pomijany, ponieważ zaimplementowana została tylko jedna możliwość; wolne krawędzie po prostu nie generują dodatkowej topologii.

### 3.1.2 Intersection

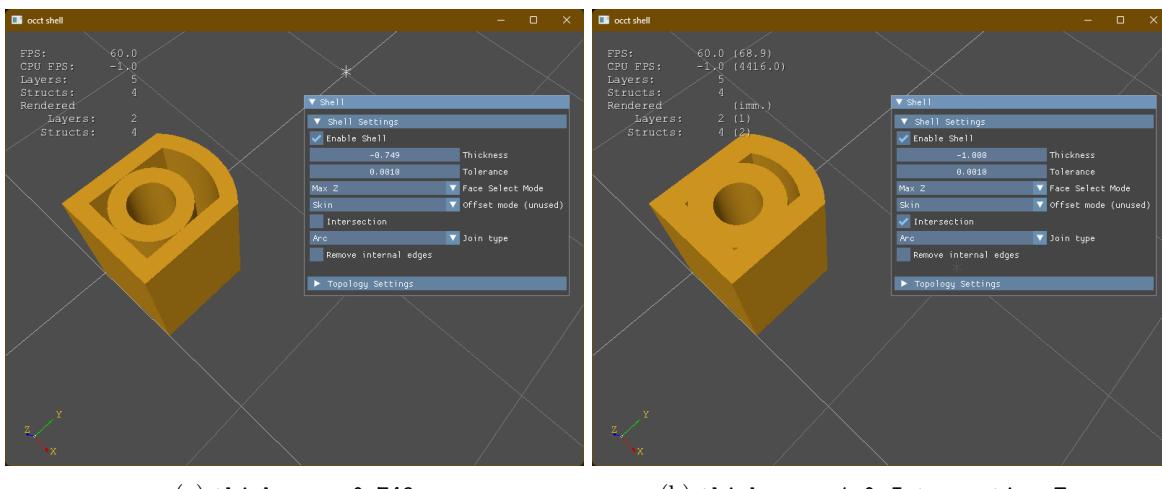
Dla większości brył wydrążenie skorupy o odpowiednio dużej grubości spowoduje, że ściany, które były do siebie równolegle mogą się „spotkać”, tj. wynikowy kształt będzie zawierał samoprzecięcia (samoprzecięcia są potem przez algorytm łączone tak, że są jednolitym kształtem).

Parametr `Intersection` określa, jak ma się zachować algorytm w stosunku do takich sytuacji. Przyjmuje 2 możliwe wartości:

- `False` (domyślnie) – obliczane są tylko przecięcia przy ścianach mających wspólną krawędź. W przypadku wystąpienia jakichś innych przecięć, algorytm kończy się błędem.
- `True` – przecięcie jest obliczane dla wszystkich wygenerowanych ścian.

W ogólności, zachowanie algorytmu dla wartości `True` jest bardziej ogólne i może być w wielu przypadkach bardziej oczekiwane. Niestety wartość ta nie została w pełni zaimplementowana i jej używanie nie jest rekommendowane przez twórców biblioteki.

Na rys. 13 przedstawiono maksymalną grubość skorupy, jaką można uzyskać dla `Intersection = False`. Dla wartości bardziej ujemnych kształtu się nie generuje. Obok również wynik działania dla parametru `Intersection = True`.



Rysunek 13: Porównanie kształtu o maksymalnej grubości skorupy bez samoprzecięć i kształtu zawierającego samoprzecięcia z ustawionym `Intersection=True`.

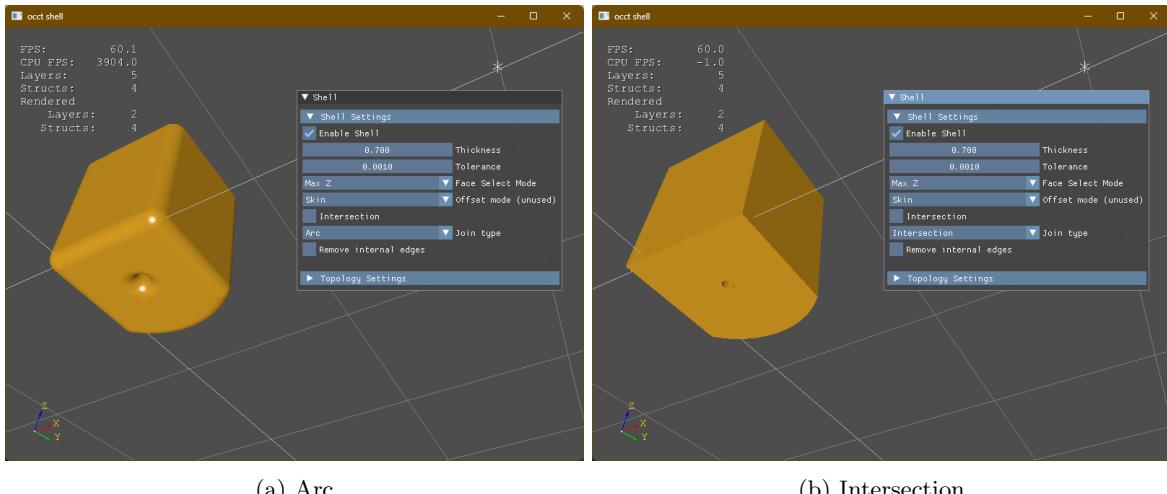
### 3.1.3 Join

W uproszczonym przypadku początkowym etapem konstruowania skorupy jest skopiowanie ścian pierwotnego kształtu i „odsunięcie” ich od środka bryły – tak powstają ściany skorupy. W przypadku tworzenia skorupy „na zewnątrz”, tj. z dodatnim `thickness`, owe skopione ściany przestają mieć wspólne krawędzie.

Parametr `Join` określa, w jaki sposób te nieciągłości występujące między sąsiednimi ścianami skorupy (tj. „na rogach”) są wypełniana. Możliwe są 2 wartości:

- **Arc** – tworzone są „tuby” i sfery, nieciągłości są zaokrąglane.
- **Intersection** – ściany równoległe do ścian pierwotnego kształtu są powiększane, przez co przecięcia powstają naturalnie, są kanciaste.

Efekt jest przedstawiony na rys. 14.



Rysunek 14: Porównanie efektów dla różnych wartości parametru **Join**.

### 3.2 Ograniczenia funkcji

Nie dla każdego kształtu możliwe jest stworzenie jego skorupy. Poniżej przedstawiamy ograniczenia algorytmu oraz wartości, dla których rezultat funkcji może stworzyć nie taki kształt, jaki mógłby być pożądany.

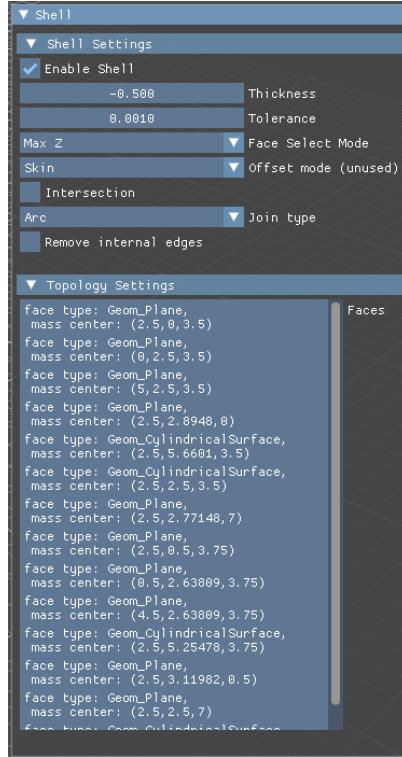
- Wszystkie ściany kształtu wejściowego powinny być oparte o powierzchnie ciągłości przynajmniej C1.
- Algorytm działa najlepiej dla niewielkich wartości grubości skorupy - takich, dla których końcowy kształt nie zawiera samoprzecięć. Dla większych wartości grubości mogą powstać nienaturalne samoprzecięcia lub ściany mogą się odwrócić „na lewą stronę”.
- Algorytm może się zakończyć błędem, jeżeli kształt będzie zawierał więcej niż krawędzie zbiegające się w jednym punkcie.
- Algorytm nie potrafi obliczyć skorupy dla BSpline’ów o ciągłości C0.

### 3.3 Aplikacja

Aplikacja umożliwia przebadanie wszystkich istotnych parametrów metody `MakeThickSolidByJoin()`. Po uruchomieniu pojawia się na ekranie domyślna bryła z możliwością modyfikowania parametrów w GUI. Zmiany w kształcie są obliczane na bieżąco. W przypadku, gdy funkcja nie jest w stanie obliczyć kształtu dla danych wartości (3.2), kształt znika z ekranu.

GUI jest podzielona na 2 sekcje: **Shell Settings** i **Topology Settings**. Sekcja Shell Settings umożliwia, oprócz modyfikacji wspomnianych parametrów, włączanie lub wyłączanie Shella oraz prosty wybór ściany, względem której Shell jest „wydrążany”. Do wyboru jest 6 podstawowych kierunków głównych: dodatnie i ujemne X/Y/Z. Sekcja Topology Settings wypisuje natomiast wszystkie ściany obiektu znajdującego się na ekranie. Umożliwia to podglądarki zmian w topologii podczas eksperymentowania z parametrami (patrz: 4.1).

**Wczytywanie plików** Aplikacja umożliwia wczytywanie również własnych kształtów zapisanych w plikach w formatach IGES, STEP i BREP. Należy to zrobić ręcznie przez modyfikację funkcji `main`. Domyślnie wczytywany jest kształt z rys. 12.



Rysunek 15: GUI aplikacji.

## 4 Badanie Topologii

W bibliotece OpenCascade wykorzystywana jest reprezentacja brył za pomocą ich brzegu. Hierarchię obiektu można przeglądać z poziomu kodu za pomocą obiektu typu `TopExp_Explorer`. Do konstruktora obiektu należy przekazać wcześniej parametr, który powie mu czego ma szukać w topologii sceny. Obiekty topologiczne rozważane w badanym przypadku to:

- `TopAbs_ShapeEnum`:`:TopAbs_EDGE` - krawędź
- `TopAbs_ShapeEnum`:`:TopAbs_FACE` - ściana (powierzchnia lub jej część)
- `TopAbs_ShapeEnum`:`:TopAbs_SHELL` - zamknięty lub otwarty zbiór ścian
- `TopAbs_ShapeEnum`:`:TopAbs_SOLID` - część przestrzeni trójwymiarowej ograniczona przez jeden lub więcej obiekt typu „shell”.

### 4.1 Znajdowanie hierarchii

Klasa `TopExp_Explorer` jest w stanie znaleźć elementy topologii podanego typu, jednak nie jest w stanie znaleźć ich miejsca w hierarchii. Gdy na scenie znajduje się pewna bryła, będzie miała on swój shell, który z kolei będzie składał się z ścian, między którymi będą znajdująły się krawędzie. Aby analizować hierarchię topologii sceny wykorzystane zostało następujące podejście:

- Utworzono zostały mapy typu `TopTools_IndexedDataMapOfShapeListOfShape`, jest to struktura danych, w której kluczami są obiekty typu `TopoDS_Shape` a wartościami są listy obiektów `TopoDS_Shape`. Pozwala to przetrzymywać znalezione relacje pomiędzy nimi.
- Mapy zostały wypełnione za pomocą funkcji `TopExp`:`:MapShapesAndAncestors`. Funkcja ta jako parametry przyjmuje kolejno:
  - Obiekt typu `TopoDS_Shape`, którego hierarchia ma być przeszukana pod kątem relacji
  - Parametr typu `TopAbs_ShapeEnum`, który oznacza typ klucza
  - Parametr typu `TopAbs_ShapeEnum`, który oznacza typ wartości
  - Mapę, do której zapisane zostaną znalezione relacje

Przykładowo, wywołanie takiej funkcji może wyglądać następująco:

Listing 1: Przykładowy plik wyjściowy algorytmu (początkowy fragment pliku wygenerowanego dla kostki)

```
"solid 1" "shell 2"  
"shell 2" "face 3"  
"shell 2" "face 4"  
"shell 2" "face 5"  
"shell 2" "face 6"  
"shell 2" "face 7"  
"shell 2" "face 8"  
"face 3" "edge 9"  
"face 5" "edge 9"  
"face 3" "edge 10"  
...  
.
```

```
TopTools_IndexedDataMapOfShapeListOfShape edgeFaceMap;  
TopExp : MapShapesAndAncestors (shape, TopAbs_EDGE, TopAbs_FACE, edgeFaceMap);
```

Powyższe wywołanie znajdzie w obiekcie `shape` wszystkie relacje pomiędzy krawędziami i ścianami, a następnie zapisze je do mapy `edgeFaceMap`. Jako klucz użyte zostaną krawędzie, a jako wartość użyta zostanie lista ścian, które przylegają do danej krawędzi.

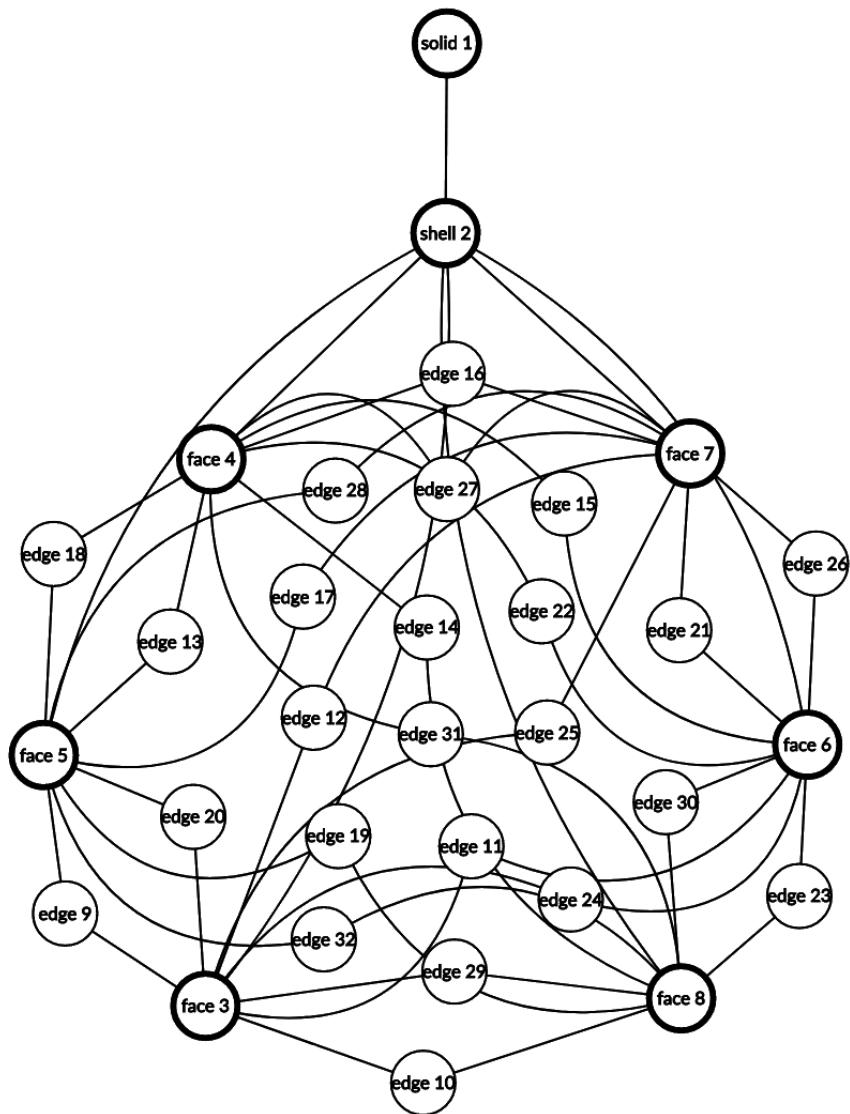
- Dodatkowo została utworzona mapa typu `TopTools_DataMapOfShapeInteger`. Pozwala ona na przypisanie unikatowej liczby całkowitej każdemu elementowi topologii, dzięki czemu można zidentyfikować go i użyć kluczy do wygenerowania czytelnego grafu przedstawiającego topologię.

## 4.2 Analiza topologii

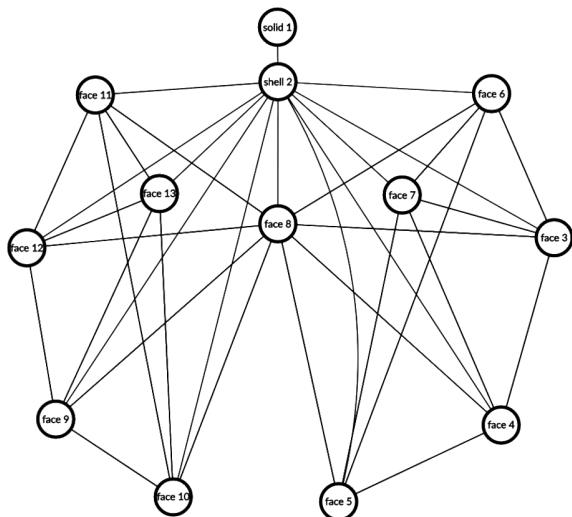
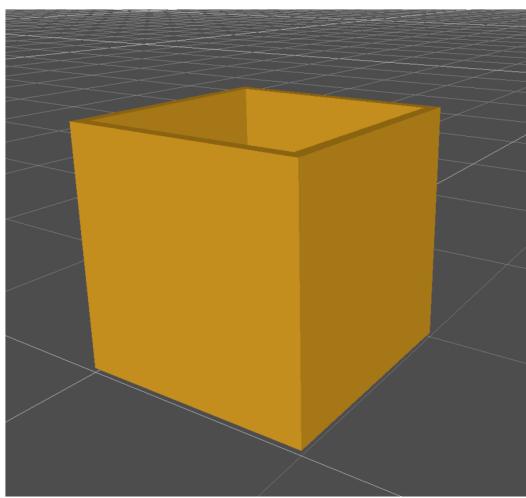
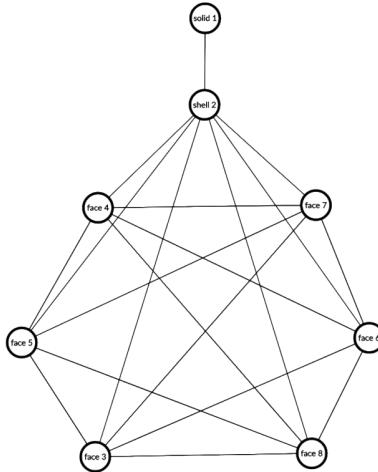
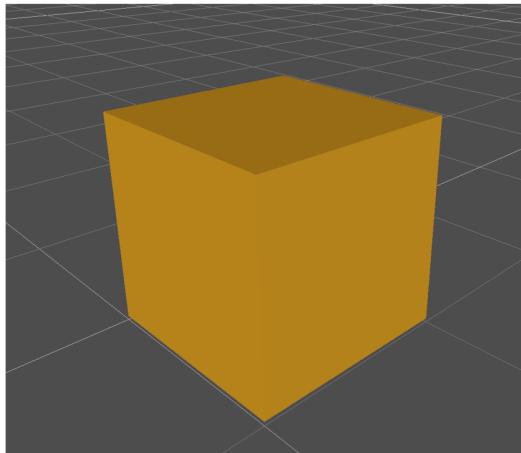
Topologia została wypisana w prostym formacie tekstowym, którego schemat przedstawiony jest na Listingu 1. Odpowiadający mu graf można zobaczyć na Rysunku 16. Jak można zauważyć, czytelność grafu jest znacznie utrudniona, poprzez zawarcie na nim krawędzi, które nie wnoszą nowych informacji, gdyż każda krawędź łączy ze sobą jedynie dwie ściany. O wiele czytelniejszy jest graf kostki na Rysunku 17, w którym prostym skryptem krawędzie łączące dwie ściany zostały zastąpione pojedynczą krawędzią w grafie.

## 4.3 Różnice w topologii

Przekazanie różnych parametrów do algorytmu `BRepOffsetAPI_MakeThickSolid` powoduje pewne różnice w topologii. Ilość powłok, brył oraz ścian pozostaje taka sama w każdym przypadku, jednak zmienia się ilość krawędzi. Dla niektórych parametrów w wyjściu algorytmu pojawiają się zduplikowane krawędzie. Zdarza się tak na przykład dla parametru `Join` ustawionego na `GeomAbs_Intersection` oraz `Mode` ustawionego na `BRepOffset_Mode::BRepOffset_Pipe`.



Rysunek 16: Przykładowy graf wygenerowany dla prostej kostki



Rysunek 17: Przykładowe grafy wygenerowane dla prostej kostki oraz tej samej kostki po wykonaniu operacji shell. Grafy zostały uproszczone skryptem w celu zwiększenia czytelności.