ALIBRE SCRIPT

INSTRUKCJA WERSJA 1.0

© Copyright Datacomp 2018

OŚWIADCZENIE

Informacje zawarte w tym dokumencie mogą ulec zmianie bez powiadomienia i nie stanowią zobowiązania ze strony producenta. Oprogramowanie opisane w tym dokumencie jest dostarczane na podstawie umowy licencyjnej lub umowy o nieujawnianiu i może być używane lub kopiowane zgodnie z warunkami umowy. Kopiowanie oprogramowania na jakimkolwiek nośniku jest niezgodne z prawem, z wyjątkiem przypadków wyraźnie dozwolonych w umowie licencyjnej lub umowie o nieujawnianiu. Żadna część tego podręcznika nie może być powielana ani przesyłana w jakiejkolwiek formie ani za pomocą jakichkolwiek środków, elektronicznych lub mechanicznych, włączając w to fotokopie, nagrywanie lub przechowywanie i odzyskiwanie informacji, do jakichkolwiek celów innych niż użytkowanie oprogramowania przez kupującego, bez uprzedniej pisemnej zgody.

© Alibre, LLC 2018, Wszelkie prawa zastrzeżone

Microsoft® i Windows ™ są znakami towarowymi lub zastrzeżonymi znakami towarowymi firmy Microsoft Corporation.

PC® jest zastrzeżonym znakiem towarowym firmy International Business Machines Corporation.

SPIS TREŚCI

Oświadczenie 2

Spis treści 3

Rozdział 1 Wstęp 4

Rozdział 2: Szybki start 5

Rozdział 3: Jednostki 15

Rozdział 4: Polilinie 16

- Tworzenie 17
- Przecinanie się dwóch polilinii 18
- Obracanie 18
- Łączenie 18
- Przesuwanie 18
- Kopiowanie 18

Rozdział 5: Manipulacja szkicami 19

- Kopiowanie 19
- Obracanie 20
- Odsunięcie 20
- Skalowanie 21
- Połączenie 21

Rozdział 6: Koła zębate 22

- Teoria 22
- Tworzenie kół zębatych 23
- Zaawansowana funkcjonalność 25

Rozdział 7: Modyfikowanie istniejących części i szkiców 29

Rozdział 8: Wskazówki i porady 31

© Copyright Datacomp 2018

RORDZIAŁ 1: WSTĘP

Alibre Script unosi możliwość pisania skryptów a Alibre Design. Jest to potężne narzędzie do tworzenia szkiców i części opartych na zmiennych, powtórzeniach i algorytmach. Na przykład z pojedynczego skryptu można uzyskać 50 części, które są podobne lub nieznacznie się różnią. Modelowanie każdej części ręcznie byłoby uciążliwe i czasochłonne. Skrypty są odpowiednie w generowaniu precyzyjnych matematycznych kształtów, na przykład ewolwenty zęba koła zębatego.

Alibre Script używa języka programowania Python, który jest szeroko obsługiwany i jest idealny dla szybkiego tworzenia skryptów przez osoby niebędące programistami. Python jest dostarczany razem z dużą biblioteką funkcji, które są gotowe do użycia natychmiast po uruchomieniu.

Prosty przykład skryptu:

```
1  Units.Current = UnitTypes.Millimeters
2  Test = Part("Test")
3  XYPlane = Test.GetPlane("XY-Plane")
4  MySketch = Test.AddSketch("MySketch", XYPlane)
5  MySketch.AddCircle(0, 0, 10, False)
6  Cylinder = Test.AddExtrudeBoss("Object", MySketch, 5, False)
```

Powyższy skrypt tworzy nową część o nazwie „Test”, która jest cylindrem o średnicy 10 mm i wysokości 5 mm.

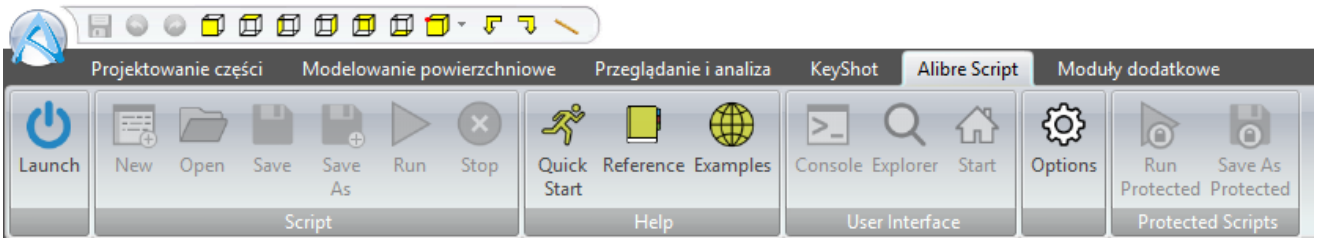
Wymagana jest znajomość Alibre Design. Znajomość języka Python również jest zalecana. Więcej szczegółów na temat Pythona można znaleźć na stronie http://www.python.org/. Polecamy „Python Essential Reference” publikowany przez Addisona Wesleya.

Aby uzyskać pełną listę wszystkich funkcji udostępnionych przez Alibre Script, zobacz oddzielną instrukcję obsługi.

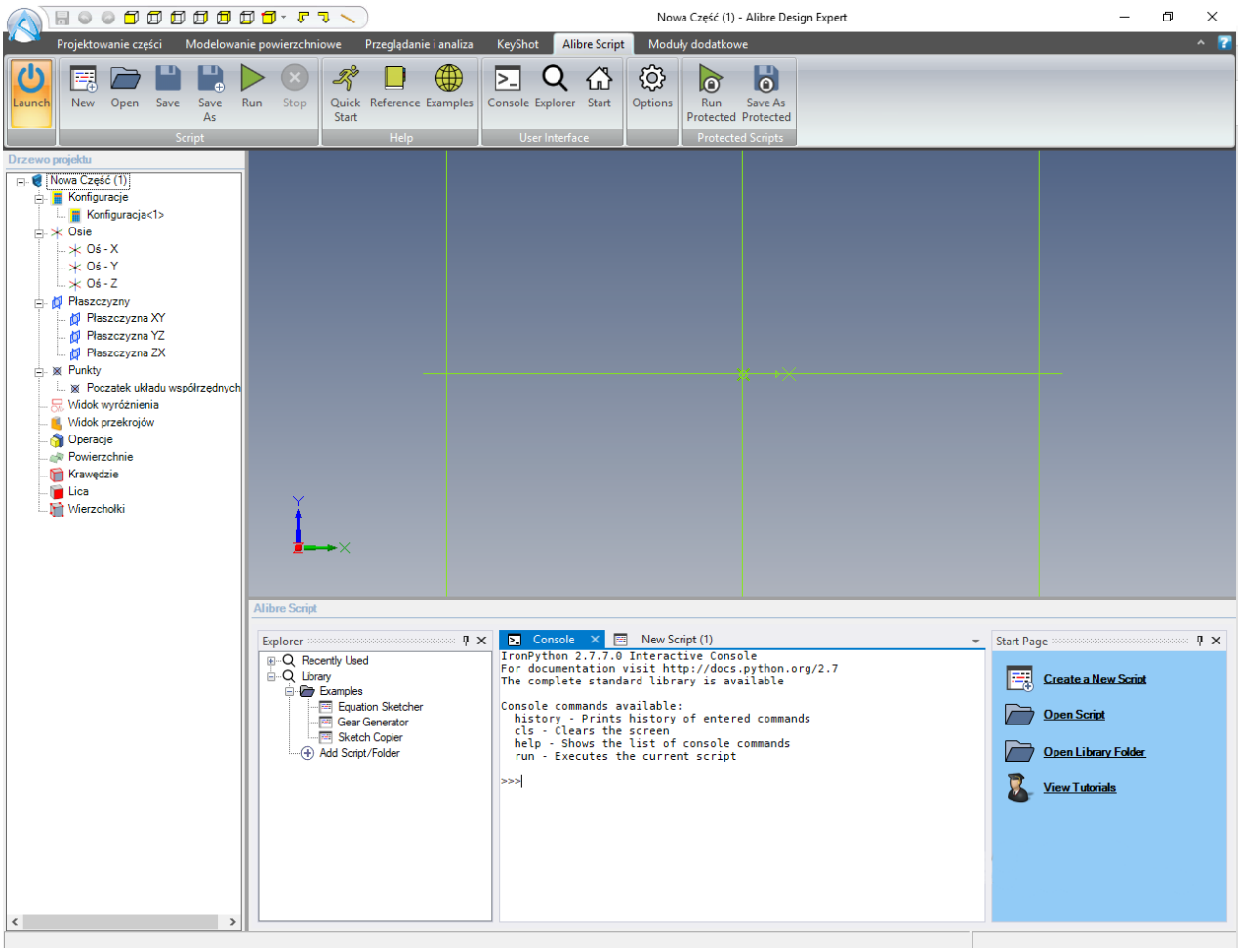
© Copyright Datacomp 2018

ROZDZIAŁ 2: SZYBKI START

1. Uruchom Alibre Design i otwórz zakładkę Alibre Script. Kliknij „Launch” aby otworzyć okienko Alibre Script.



Okno Alibre Script wygląda następująco:

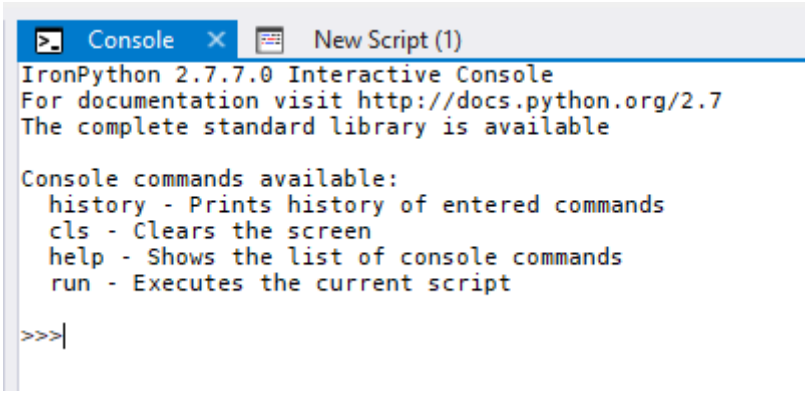


© Copyright Datacomp 2018

Na górze znajduje się pasek narzędzi zapewniający dostęp do plików, edytowania i dokonywania operacji. Główna część okna posiada dwie zakładki: New Script (1) i Console.

Zakładka skryptu pozwala edycję bieżącego kodu. Zakładka konsoli zapewnia natychmiastowy dostęp do środowiska Python.

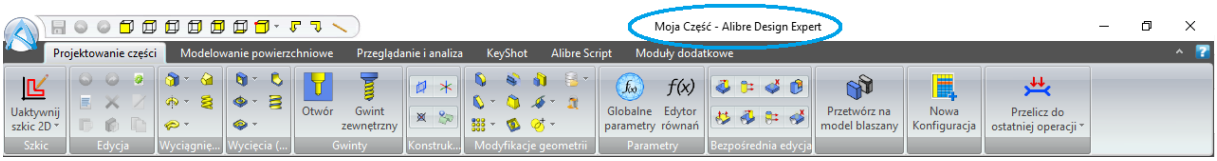
2. Kliknij na zakładkę „Console”, by zobaczyć konsolę skryptu.



3. Kliknij na zakładkę skryptu i wpisz:

```
1    MojaCzesc = Part('Moja Część')
```

4. Kliknij „Run”. Alibre Design uruchomi się i stworzy utworzy nową część gotową do edycji.



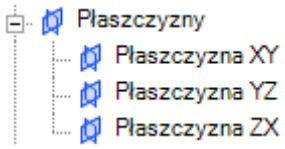
Zwróć uwagę na nazwę części „Moja Część”. Jest ona nazwą wpisaną w poprzedniej linijce kodu.

5. W celu utworzenia szkicu, należy utworzyć dla niego płaszczyznę. Wprowadź poniższe w oknie skryptu aby uzyskać dostęp do płaszczyzny XY.

© Copyright Datacomp 2018

```
2    Plaszczyzna_XY = MojaCzesc.GetPlane('Płaszczyzna XY')
```

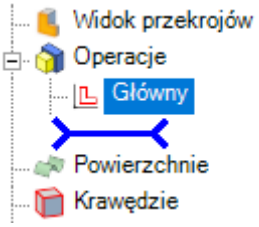
Można otrzymać dostęp do jakiejkolwiek innej płaszczyzny w analogiczny sposób. W tym celu należy podać nazwę wybranej płaszczyzny z eksploratora projektu Alibre Design.



6. Tworzenie nowego szkicu na płaszczyźnie XY przez wprowadzenie:

```
3    Szkic_Glowny = MojaCzesc.AddSketch('Główny', Plaszczyzna_XY)
```

Utworzony szkic można zobaczyć w eksploratorze projektu.

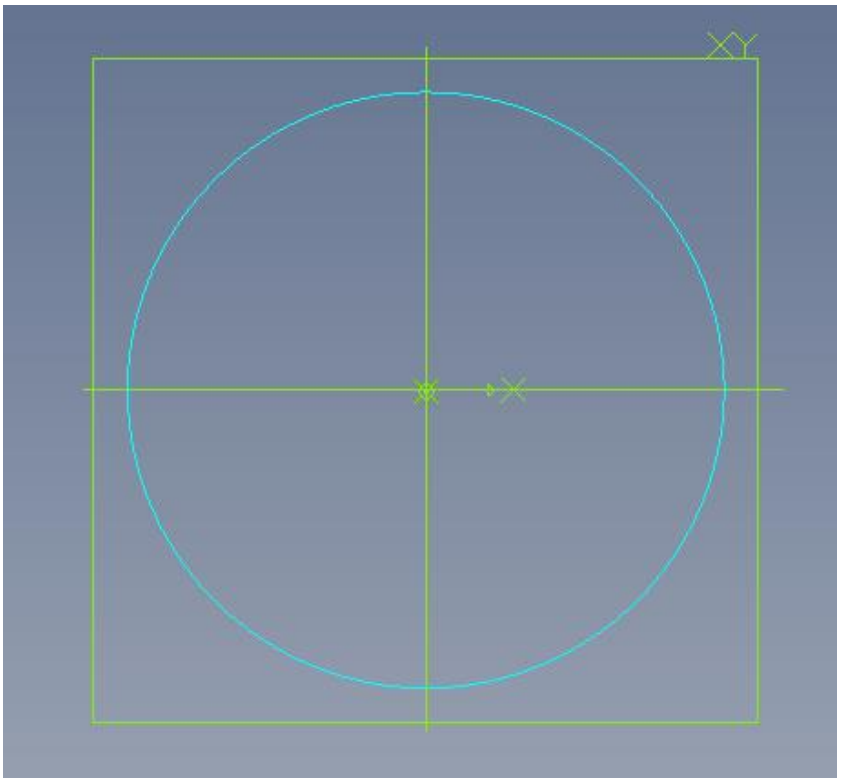


7. Teraz utworzymy okrąg ze środkiem na początku układu współrzędnych. Można to zrobić wpisując:

```
4    Szkic_Glowny.AddCircle(0, 0, 10, False)
```

Polecenie to dodaje okrąg do Szkic_Glowny o średnicy 10mm ze środkiem w punkcie (0, 0). Ostatni parametr ustawiony jest jako False. Jeżeli zostałby ustawiony jako True, zostałby naszkicowany okrąg referencyjny.

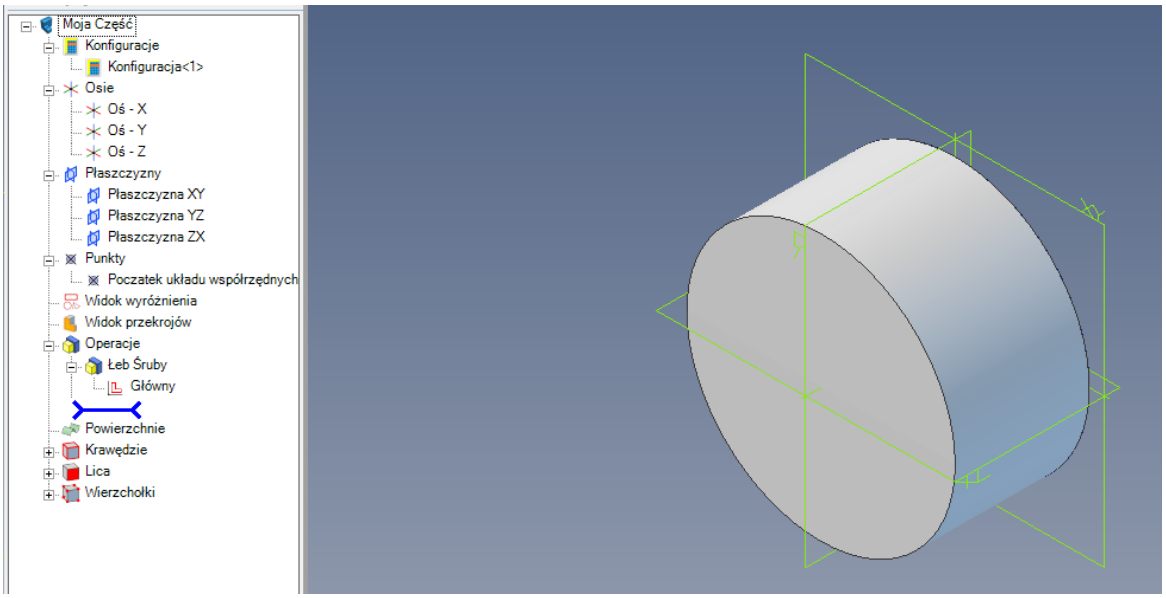
© Copyright Datacomp 2018



8.Teraz można wyciągnąć szkic wprowadzając poniższą linię skryptu w oknie:

```
5    LebSruby = MojaCzesc.AddExtrudeBoss('Łeb Śruby', Szkic_Glowny, 5, False)
```

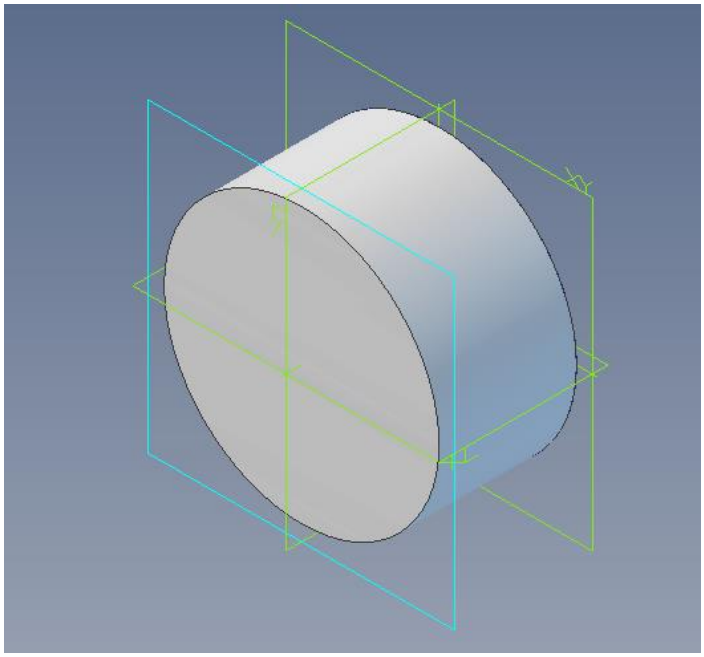
Szkic_Glowny zostaje wyciągnięty na 5mm. Ostatni parametr ustawiony jest jako False. Ustawiając go na True, kierunek wyciągnięcia zostałby ustawiony na przeciwny.



© Copyright Datacomp 2018

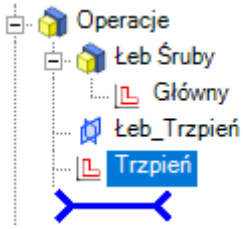
9. Teraz wygenerujemy płaszczyznę odsunięta o 5mm od płaszczyzny XY. Wprowadź poniższą linię w oknie skryptu:

```
6    Plaszczyzna_Trzpien = MojaCzesc.AddPlane('Łeb_Trzpień', Plaszczyzna_XY, 5)
```



10. Na nowo utworzonej płaszczyźnie możemy utworzyć nowy szkic.

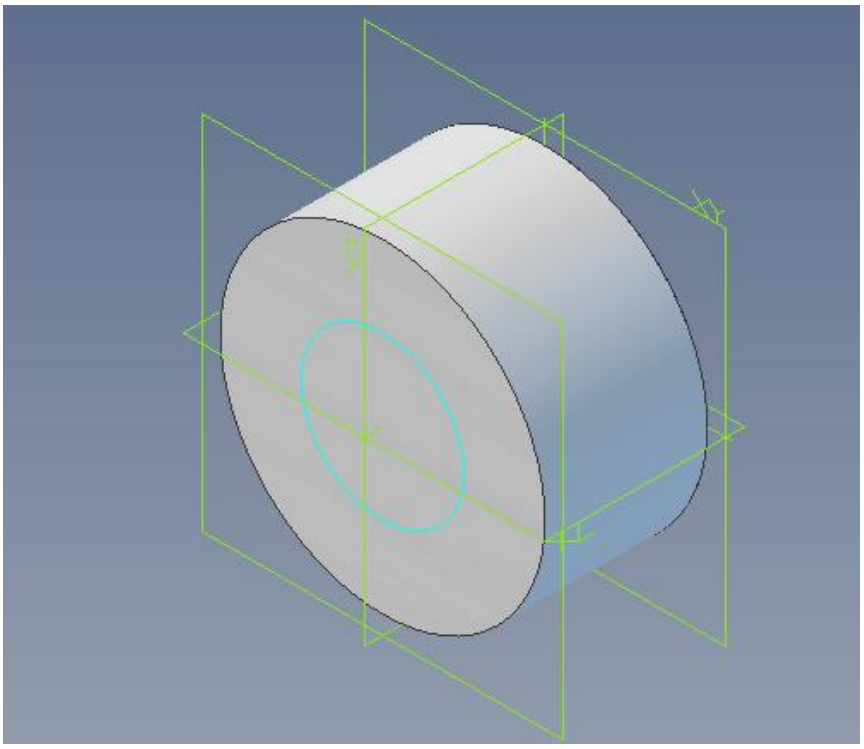
```
7    TrzpienSzkic = MojaCzesc.AddSketch('Trzpień', Plaszczyzna_Trzpien)
```



© Copyright Datacomp 2018

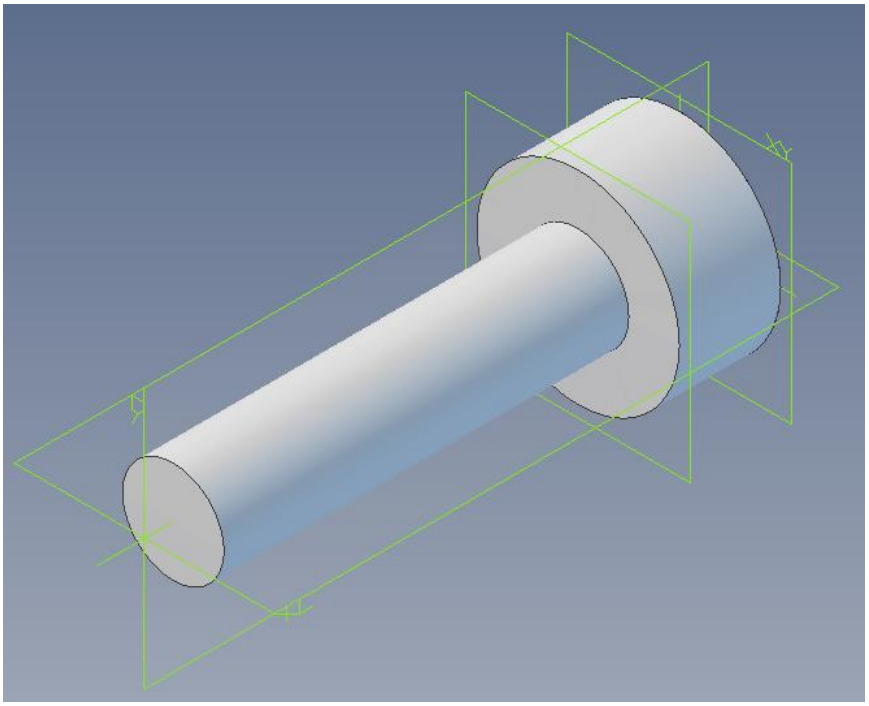
11.Szkic okręgu o średnicy 5mm:

```
8    TrzpienSzkic.AddCircle(0, 0, 5, False)
```



12.Teraz należy wyciągnąć szkic na 20mm w celu utworzenia trzpienia śruby.

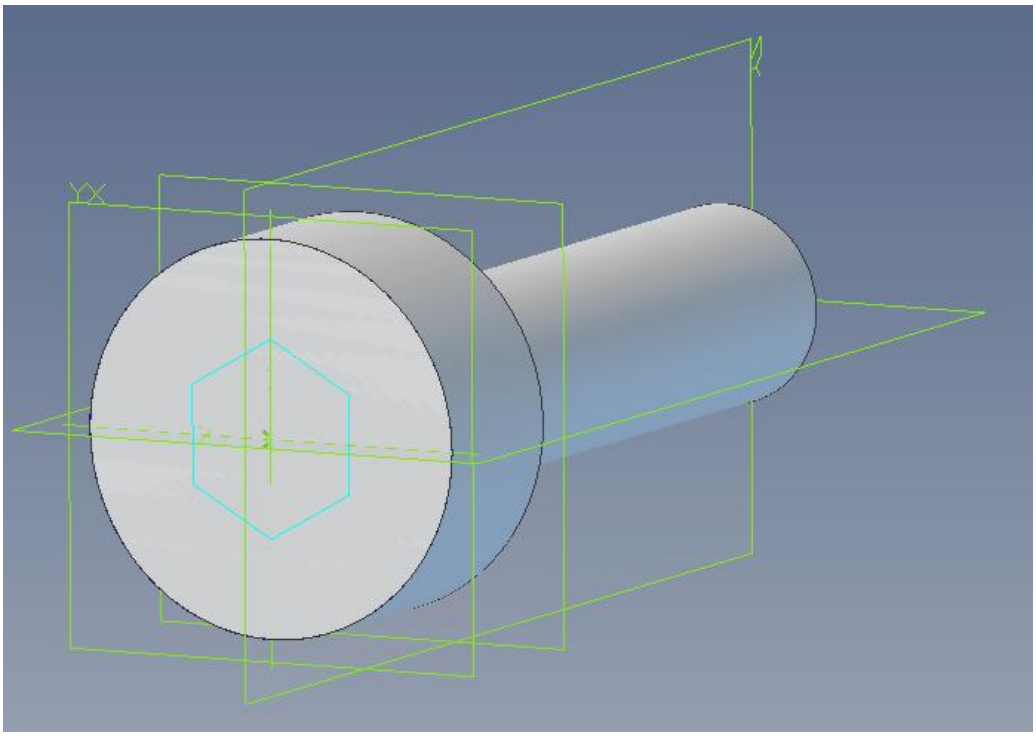
```
9    Trzpien = MojaCzesc.AddExtrudeBoss('Trzpień', TrzpienSzkic, 20, False)
```



© Copyright Datacomp 2018

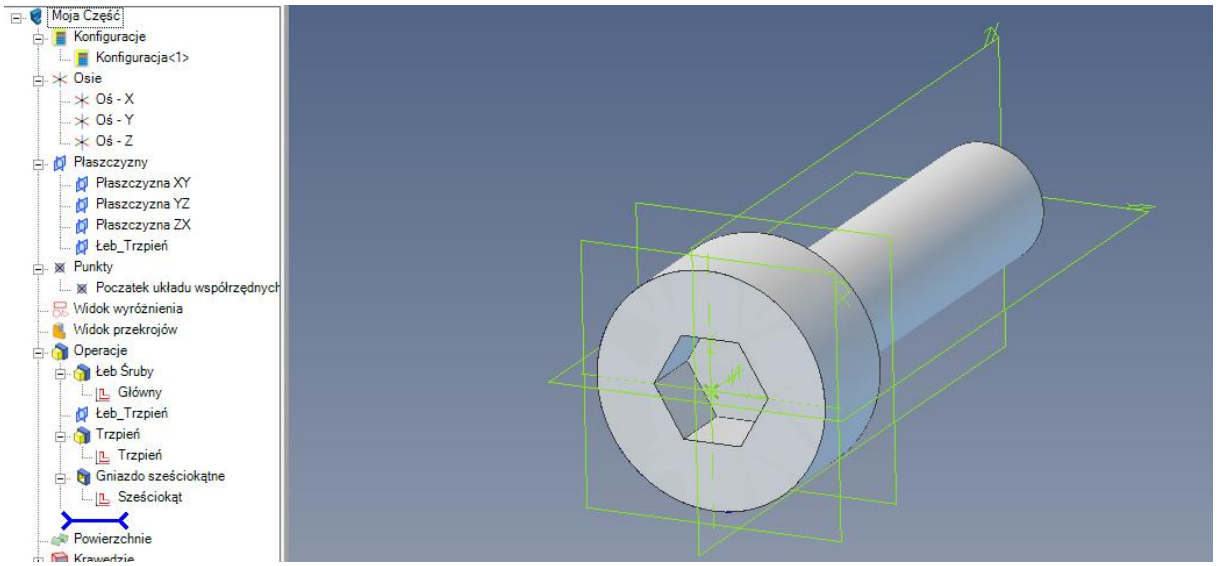
13. Teraz utworzymy gniazdo w łbie śruby. Zaczniemy od dodania nowego szkicu na płaszczyźnie XY, zawierającego sześciokąt wpisany w okrąg o średnicy 5mm.

```
10      Szesciokat = MojaCzesc.AddSketch('Sześciokąt', Plaszczyzna_XY)
11      Szesciokat.AddPolygon(0, 0, 5, 6, False)
```



14. Dodanie wycięcia w celu utworzenia gniazda w łbie śruby.

```
12      Gniazdo6k = MojaCzesc.AddExtrudeCut('Gniazdo sześciokątne', Szesciokat, 3, False)
```



15. W końcu można zapisać naszą część, eksportować do STL i zamknąć okna. Zastąp poniższe linie kodu swoimi traktując je jako szablon.

```
13      MojaCzesc.Save('C:\Modele Alibre')
14      MojaCzesc.ExportSTL('C:\Modele Alibre\Moja Część.stl')
15      MojaCzesc.Close()
```

© Copyright Datacomp 2018

16. Oto cały skrypt:

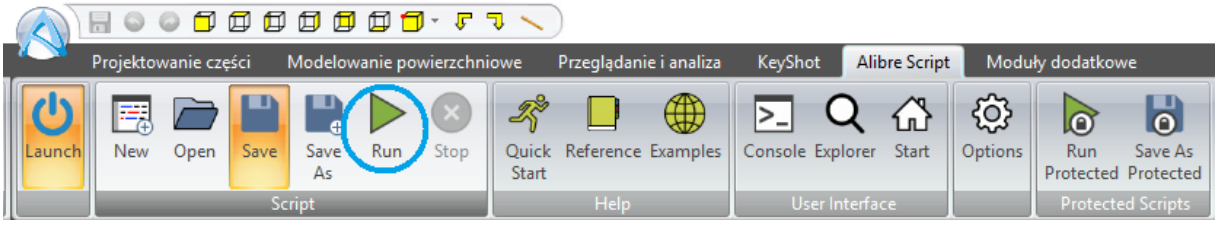
```
MojaCzesc = Part('Moja Część')

Plaszczyzna_XY = MojaCzesc.GetPlane("Płaszczyzna XY")

Szkic_Glowny = MojaCzesc.AddSketch("Główny", Plaszczyzna_XY)

Szkic_Glowny.AddCircle(0, 0, 10, False)

Leb_Sruby = MojaCzesc.AddExtrudeBoss("Łeb śruby", Szkic_Glowny, 5,
False)

Plaszczyzna_Trzpien = MojaCzesc.AddPlane('Łeb_Trzpień',
Plaszczyzna_XY, 5)

TrzpienSzkic = MojaCzesc.AddSketch('Trzpień', Plaszczyzna_Trzpien)

TrzpienSzkic.AddCircle(0, 0, 5, False)

Trzpien = MojaCzesc.AddExtrudeBoss('Trzpień', TrzpienSzkic, 20,
False)

Szesciokat = MojaCzesc.AddSketch('Sześciokąt', Plaszczyzna_XY)

Szesciokat.AddPolygon(0, 0, 5, 6, False)

Gniazdo6k = MojaCzesc.AddExtrudeCut('Gniazdo sześciokątne',
Szesciokat, 3, False)

MojaCzesc.Save('C:\Modele Alibre')

MojaCzesc.ExportSTL('C:\Modele Alibre\Moja Część.stl')

MojaCzesc.Close()
```

New Script (1) *

```
1   MojaCzesc = Part('Moja Część')
2   Plaszczyzna_XY = MojaCzesc.GetPlane("Płaszczyzna XY")
3   Szkic_Glowny = MojaCzesc.AddSketch("Główny", Plaszczyzna_XY)
4   Szkic_Glowny.AddCircle(0, 0, 10, False)
5   Leb_Sruby = MojaCzesc.AddExtrudeBoss("Łeb śruby", Szkic_Glowny, 5, False)
6   Plaszczyzna_Trzpien = MojaCzesc.AddPlane('Łeb_Trzpień', Plaszczyzna_XY, 5)
7   TrzpienSzkic = MojaCzesc.AddSketch('Trzpień', Plaszczyzna_Trzpien)
8   TrzpienSzkic.AddCircle(0, 0, 5, False)
9   Trzpien = MojaCzesc.AddExtrudeBoss('Trzpień', TrzpienSzkic, 20, False)
10  Szesciokat = MojaCzesc.AddSketch('Sześciokąt', Plaszczyzna_XY)
11  Szesciokat.AddPolygon(0, 0, 5, 6, False)
12  Gniazdo6k = MojaCzesc.AddExtrudeCut('Gniazdo sześciokątne', Szesciokat, 3, False)
13  MojaCzesc.Save('C:\Modele Alibre')
14  MojaCzesc.ExportSTL('C:\Modele Alibre\Moja Część.stl')
15  MojaCzesc.Close()
```

© Copyright Datacomp 2018

17. Zapisz skrypt, a następnie uruchom klikając przycisk „Run”. Część zostanie utworzona, zapisana, wyeksportowana i zamknięta w jednym kroku.



© Copyright Datacomp 2018

ROZDZIAŁ 3: JEDNOSTKI

Jednostki używane w skryptach są niezależne od jednostek używanych Alibre Design. Na przykład Alibre Design może był skonfigurowane w calach, a skrypt może używać milimetrów i odwrotnie.

Jednostki używane w skrypcie mogą byś ustawione przez dodanie jednej z poniższych linii na początku skryptu:

```
Units.Current = UnitTypes.Millimeters
Units.Current = UnitTypes.Centimeters
Units.Current = UnitTypes.Inches
```

Jednostki mogą być zmienione w jakimkolwiek punkcie skryptu. Wszystkie wartości po zmianie będą w nowych jednostkach. Na przykład:

```
Units.Current = UnitTypes.Millimeters
HeadSketch.AddCircle(0, 0, 10.2, False)
Units.Current = UnitTypes.Inches
HeadSketch.AddCircle(3, 2.6, 4.1, False)
```

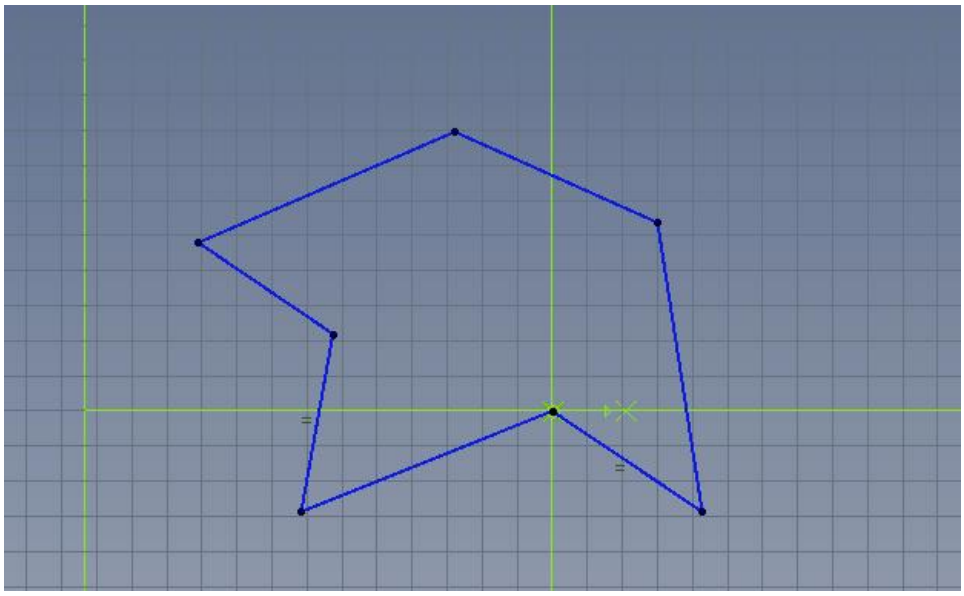
Zostaną narysowane dwa okręgi w szkicu. Środek pierwszego zlokalizowany jest w punkcie (0, 0) i ma 10.2mm średnicy. Drugi zlokalizowany jest w punkcie (3”, 2.6”) i ma 4.1” średnicy.

Kąty zawsze podawane są w stopniach.

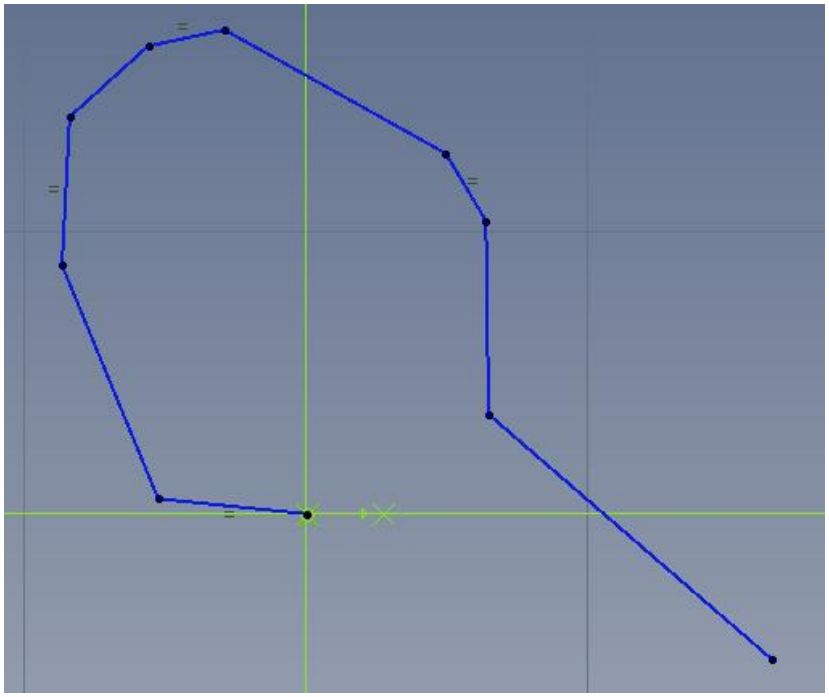
© Copyright Datacomp 2018

ROZDZIAŁ 4: POLILINIE

Linia zawiera punkt początkowy i końcowy. Szkice zwykle składają się z wielu linii połączonych razem. Poniżej przykład szkicu składającego się z 7 linii.



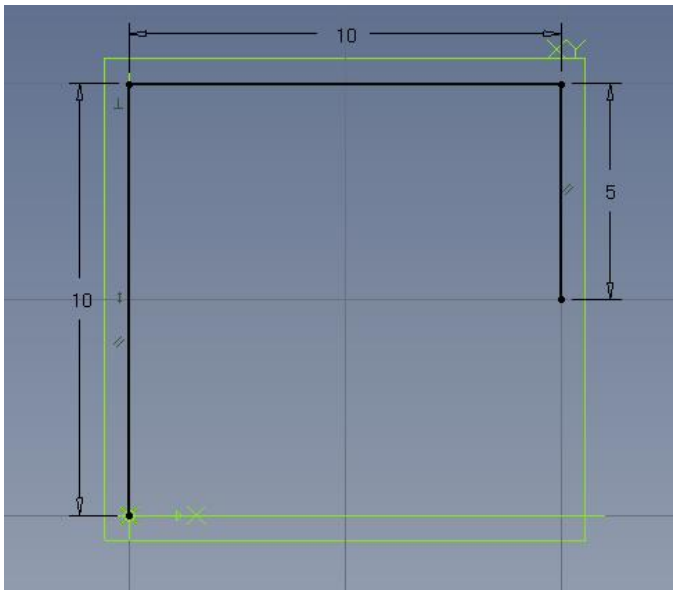
Alibre Script wprowadza pojęcie polilinii. Polilinia jest zestawem łączonych z sobą linii. Na przykład powyższy szkic może być reprezentowany przez pojedynczą polilinię. Oto przykład kolejnej polilinii:



© Copyright Datacomp 2018

Polilinia jest definiowana poprzez zestaw punktów przechodzących z jednego końca na drugi. Dla wielokątów pierwszy i ostatni punkt znajdują się w tym samym miejscu. Poniżej przedstawiony został przykład polilinii zdefiniowanej przez punkty, a następnie jej wygląd na ekranie.

Punkt 1 = (0, 0)
Punkt 1 = (0, 10)
Punkt 1 = (10, 10)
Punkt 1 = (10, 5)



TWORZENIE

Tworzenie polilinii w Alibre Script jest łatwe. Poniżej przykład jak naszkicować polilinię:

```
MojaLinia = Polyline()
MojaLinia.AddPoint(PolylinePoint(0, 0))
MojaLinia.AddPoint(PolylinePoint(0, 10))
MojaLinia.AddPoint(PolylinePoint(10, 10))
MojaLinia.AddPoint(PolylinePoint(10, 5))
```

Raz utworzona polilinia może być dodana do szkicu:

```
MojSzkic.AddPolyline(MojaLinia, False)
```

© Copyright Datacomp 2018

Polilinie mają przydatne właściwości, które pomagają w tworzeniu szkiców opartych na matematyce oraz algorytmach. Na przykład wykryte może przecinanie się dwóch polilinii. Polilinia może być przycięta do punktu. Polilinie mogą być obracane przenoszone, scalane i powielane. Istnieje także możliwość wstawiania punktów w dowolnych miejscach polilinii.

Ważne jest, by pamiętać, że polilinie mogą być manipulowane wiele razy przed dodaniem ich do szkicu.

PRZECINANIE SIĘ DWÓCH POLILINII

Poniżej znajduje się przykład znajdowania punktu przecięcia się dwóch polilinii i przycinania pierwszej polilinii do tego punktu:

```
Przeciecie = Polyline.FindIntersection(PierwszaLinia, DrugaLinia)
PrzycietaLinia = PierwszaLinia.SplitAtPoint(Przeciecie, 0.001)[0]
```

OBRACANIE

Aby obrócić linię względem punktu (4, 7) o 15.3 stopnia:

```
MojaLinia.RotateZ(4, 7, 15.3)
```

ŁĄCZENIE

Łączenie dwóch polilinii razem tworzy nową polilinię:

```
DlugaPolilinia = MyLine.Append(MojaInnaLinia)
```

PRZESUWANIE

Przesuwanie linii jest proste. Na przykład aby przenieść linię o 4.7mm wzdłuż osi X i -8.9mm wzdłuż osi Y:

```
MojaLinia.Offset(4.7, -8.9)
```

KOPIOWANIE

Linia może zostać zduplikowana:

```
NowaLinia = MojaLinia.Clone()
```

Teraz NowaLinia może być poddana manipulacji bez zmiany oryginalnej linii PierwszaLinia, na której bazowała.

© Copyright Datacomp 2018

ROZDZIAŁ 5: MANIPULOWANIE SZKICEM

Alibre Script zawiera funkcjonalność dla manipulowania szkicami takie jak kopiowanie, skalowanie, obracanie, czy przesuwanie. Otwiera on możliwości dla ponownego wykorzystywania szkiców na nowatorskie sposoby.

Funkcja manipulacji jest zapewniona przez funkcję CopyFrom. Oto jak zdefiniowana jest funkcja:

```
CopyFrom(Sketch SketchtoCopy)
```

i:

```
CopyFrom(Sketch SketchtoCopy, double Angle, double RotationCenterX,
double RotationCenterY, double TranslateX, double TranslateY, double ScaleOriginX,
double ScaleOriginY, double ScaleFactor)
```

Kąt (Angle) to kąt obrotu kopiowanego szkicu w stopniach. Wartości dodatnie poskutkują obrotem w kierunku zgodnym z ruchem wskazówek zegara. Obrót odbywa się wokół punktu zdefiniowanego przez RotationCenterX i RotationCenterY (należy zastąpić wartościami współrzędnych).

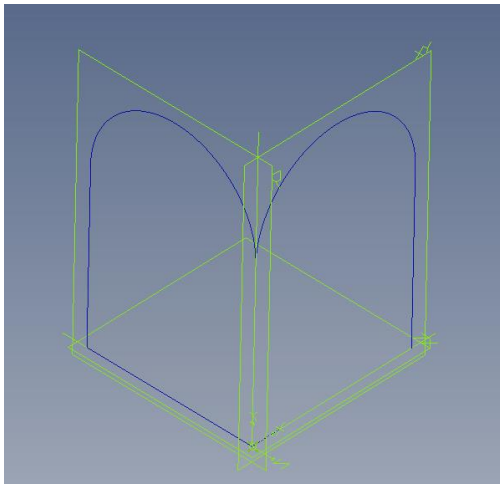
TranslateX i TranslateY (należy zastąpić wartościami liczbowymi) pozwalają na przesunięcie szkicu.

ScaleFactor – współczynnik skali ustawiony na przykład na 50.0 redukuje rozmiar szkicu do 50%, a ustawiony na 150.0 zwiększa go o 50%. Skalowanie oparte jest na punkcie zdefiniowanym w miejscach ScaleOriginY i ScaleOriginX.

KOPIOWANIE

Tworzenie kopii szkicu na innej płaszczyźnie:

```
NowySzkic = MojaCzesc.AddSketch('Nowy', MojaCzesc.GetPlane('Płaszczyzna YZ'))
NowySzkic.CopyFrom(SzkicGl)
```

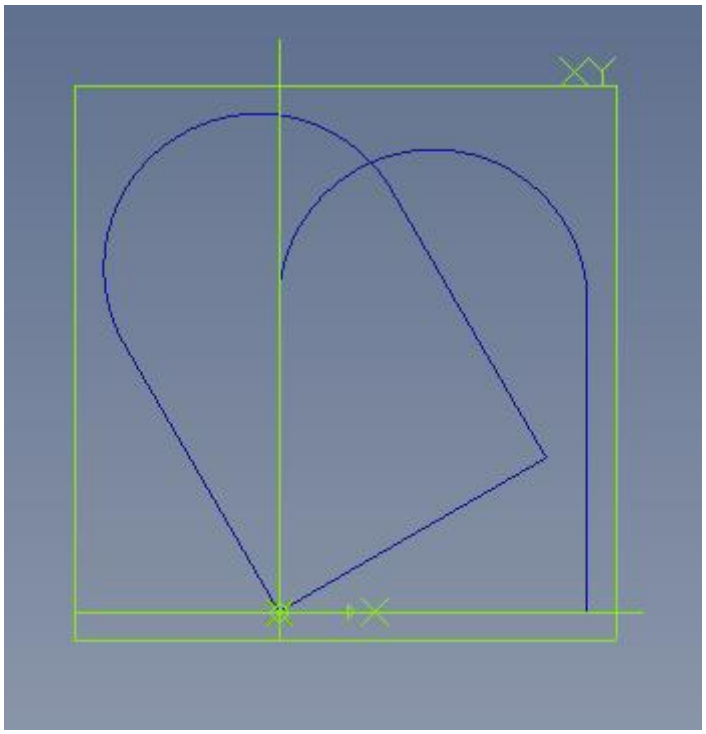


© Copyright Datacomp 2018

OBRÓT

Tworzenie kopii szkicu obróconego o 30 stopni w kierunku przeciwnym do ruchu wskazówek zegara względem punktu (0, 0).

```
NowySzkic = MyPart.AddSketch("Nowy", MyPart.GetPlane("Płaszczyzna XY"))
NowySzkic.CopyFrom(InnySzkic, 30, 0, 0, 0, 0, 0, 0, 100.0)
```

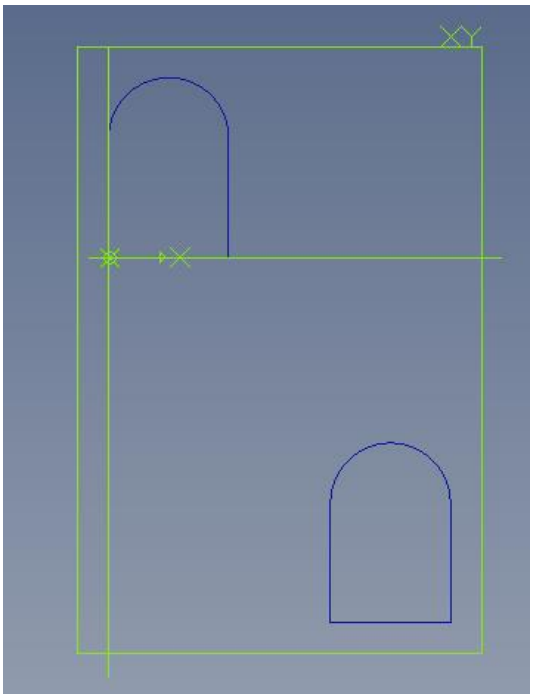


ODSUNIĘCIE

Tworzenie kopii szkicu odsuniętego o 3.7 w kierunku X i -6.1 w kierunku Y:

```
NowySzkic = MyPart.AddSketch("Nowy", MyPart.GetPlane("Płaszczyzna XY"))
NowySzkic.CopyFrom(InnySzkic, 0, 0, 0, 3.7, -6.1, 0, 0, 100.0)
```

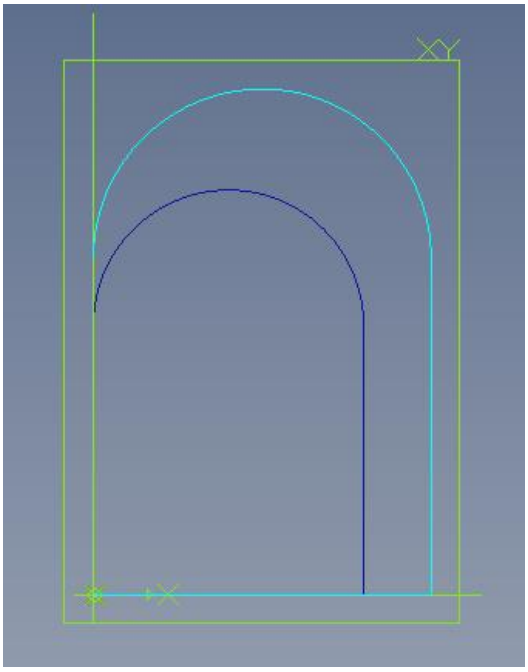
© Copyright Datacomp 2018



SKALOWANIE

Tworzenie kopii szkicu powiększonej o 25%, skalowanej względem punktu początkowego:

```
NowySzkic = MyPart.AddSketch("Nowy", MyPart.GetPlane("Płaszczyzna XY"))
NowySzkic.CopyFrom(InnySzkic, 0, 0, 0, 0, 0, 0, 0, 125.0)
```

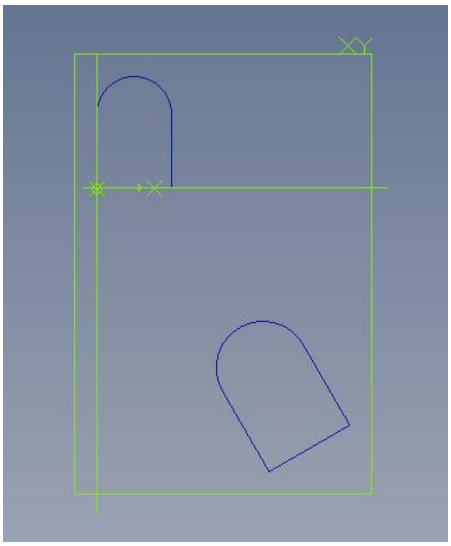


KOMBINACJA

Tworzenie kopii szkicu będącego obróconym, odsuniętym i zeskalowanym:

```
NowySzkic = MyPart.AddSketch("Nowy", MyPart.GetPlane("Płaszczyzna XY"))
NowySzkic.CopyFrom(InnySzkic, 30, 0, 0, 3.7, -6.1, 0, 0, 125.0)
```

© Copyright Datacomp 2018



ROZDZIAŁ 6: KOŁA ZĘBATE

Alibre Script zapewnia podstawową funkcjonalność do tworzenia kół zębatych, o zębach z bokami o zarysie ewolwentowym. Ewolwenta zapewnia stałą siłę i kierunek siły w całym zazębieniu dwóch zębów, co zapewnia płynną i wydajną pracę.

TEORIA

Koła zębate są definiowane przez trzy powiązane z sobą parametry:

- Liczba zębów na cal średnicy podziałowej / Moduł (rozmiar zęba) (D)
- Średnica podziałowa (rozmiar koła zębatego) (P)
- Liczba zębów (N)

Liczba zębów na cal średnicy podziałowej stosowana jest w imperialnych / angielskich systemach pomiarowych, a moduł jest używany w metrycznych systemach pomiarowych (mm średnicy podziałowej na ząb). Alibre Script obsługuje jednostkę imperialną, jednak wartość modułu można łatwo przekonwertować:

Liczba zębów na cal (Diametral Pitch) = 25.4/Moduł(Module)

Wszystkie jednostki w skrypcie można konfigurować - zobacz rozdział 3 - z wyjątkiem liczby zębów na cal. Ta jednostka jest niezmienna.

Relacja między tymi trzema parametrami to:

Liczba zębów = Średnica podziałowa (w calach) x Liczba zębów na cal

Czwarty parametr jest również potrzebny, nazywany kątem przyporu. Jest to kąt, w którym siła jednego koła wywierana jest na drugie. Typowe wartości to 20 stopni i 25 stopni.

Aby dwa koła zębate (A i B) zazębiały się odpowiednio, muszą być spełnione następujące warunki:

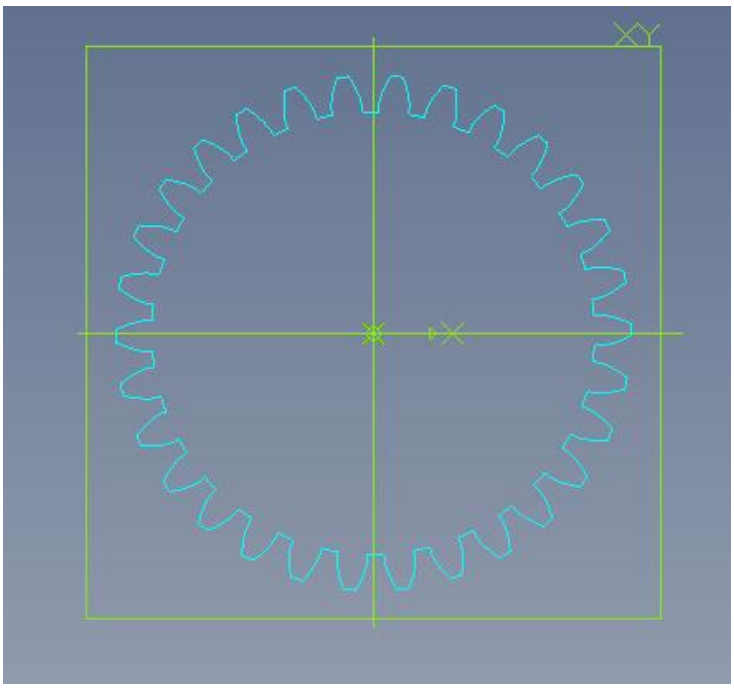
- Moduł A = Moduł B
- Kąt przyporu A = Kąt przyporu B
- Odległość środków kół zębatych = (Średnica podziałowa A + Średnica podziałowa B) / 2

TWORZENIE KÓŁ ZĘBATYCH

Alibre Script traktuje profil koła zębatego jako specjalny typ szkicu. Tworzenie koła zębatego jest zatem jak tworzenie szkicu z już wygenerowanym i dodanym profilem. Oto przykład:

```
Kolo_Zebate_Profil = Kolo_Zebate.AddGearNP("Koło zębate", 30, 38, 20, 0, 0, PlaszczyznaXY)
```

Kod ten tworzy profil koła zębatego o średnicy podziałowej 38mm, 30 zębach i kącie przyporu 20 stopni ze środkiem w punkcie (0, 0).



© Copyright Datacomp 2018

Do zdefiniowania koła zębatego użyliśmy dwóch z trzech parametrów - liczby zębów (N) i średnicy podziałowej (P). Trzeci parametr można wypisać w konsoli i odczytać:

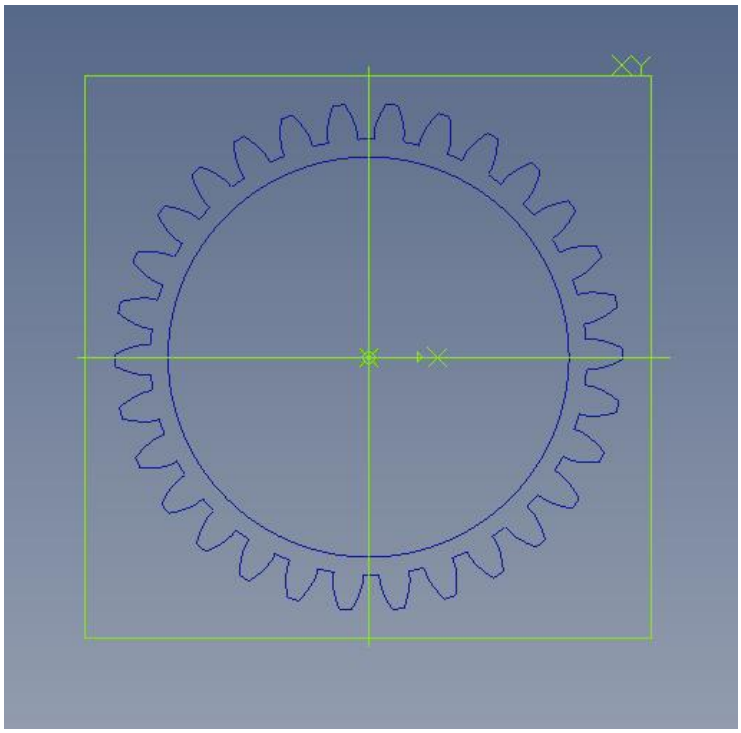
```
D = Kolo_Zebate_Profil.DiametralPitch
print "Moduł = %f" % D
```

Alibre Script definiuje cztery funkcje do tworzenia kół zębatych:

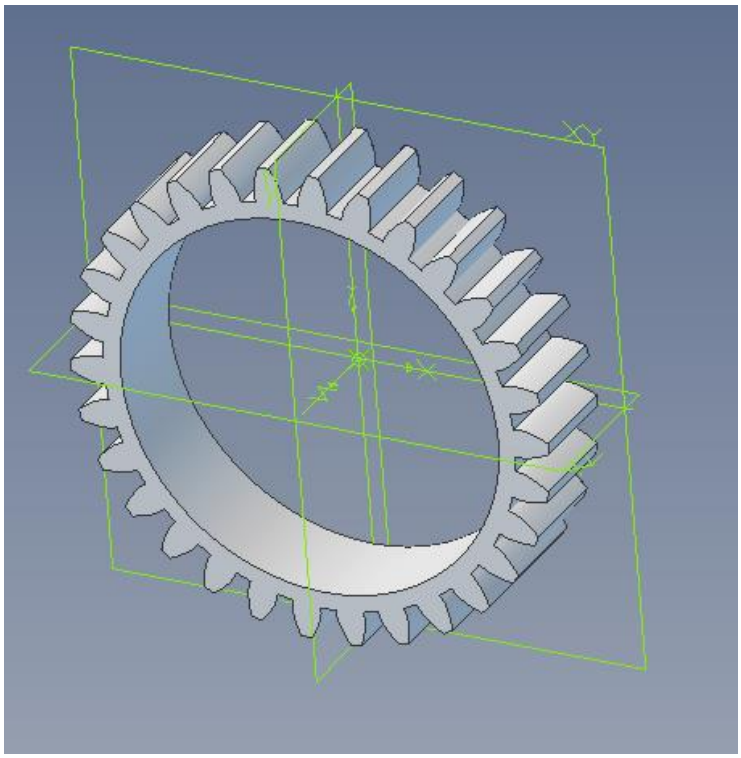
Funkcja	Parametry	Obliczenia dla Ciebie
AddGearNP	Liczba zębów(N) Średnica podziałowa(P)	Moduł(D)
AddGearDP	Moduł(D) Średnica podziałowa(P)	Liczba zębów(N)
AddGearDN	Moduł(D) Liczba zębów(N)	Średnica podziałowa(P)
AddGear	Liczba zębów(N) Średnica podziałowa(P) Moduł(D)	Brak

Zaraz po utworzeniu szkicu koła zębatego, można go traktować jak każdy inny szkic, na przykład dodając okrąg i wyciągając profil:

```
Kolo_Zebate_Profil.AddCircle(0, 0, 32, False)
Kolo_Zebate_Profil = Kolo_Zebate("Wyciągnięcie KZ", 30, 30, 25.4, 20, True, 0, 0, 12, PlaszczyznaXY)
```



© Copyright Datacomp 2018



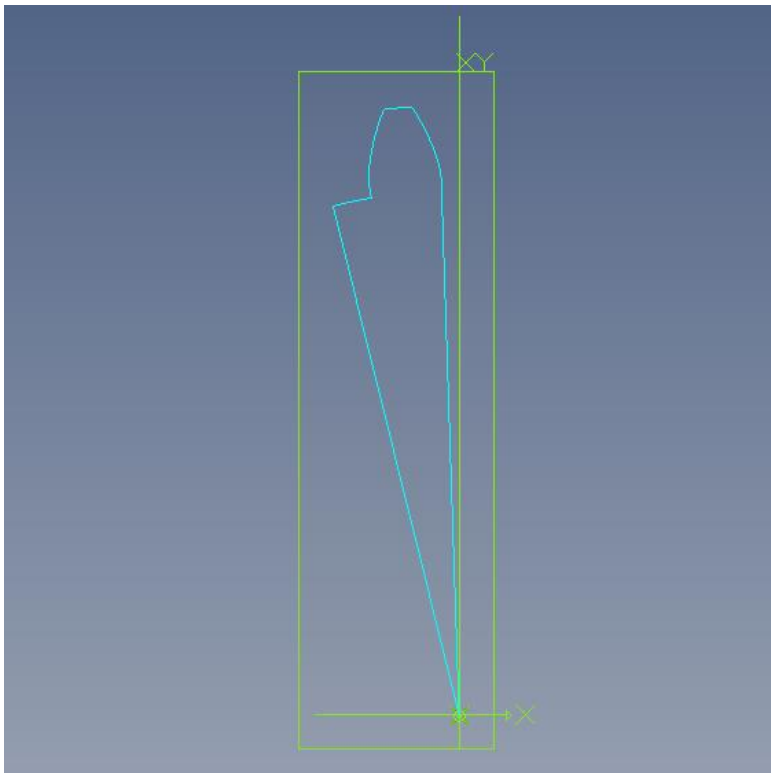
ZAAWANSOWANA FUNKCJONALNOŚĆ

Czasami może być konieczna modyfikacja kształtu zęba ze względu na wymagania projektowe. Na przykład Alibre Script nie dodaje podcięcia. Aby to ułatwić, możliwe jest wygenerowanie pojedynczego zęba, który można zmodyfikować, a następnie zastosować w szyku kołowym. Oto jak to się robi.

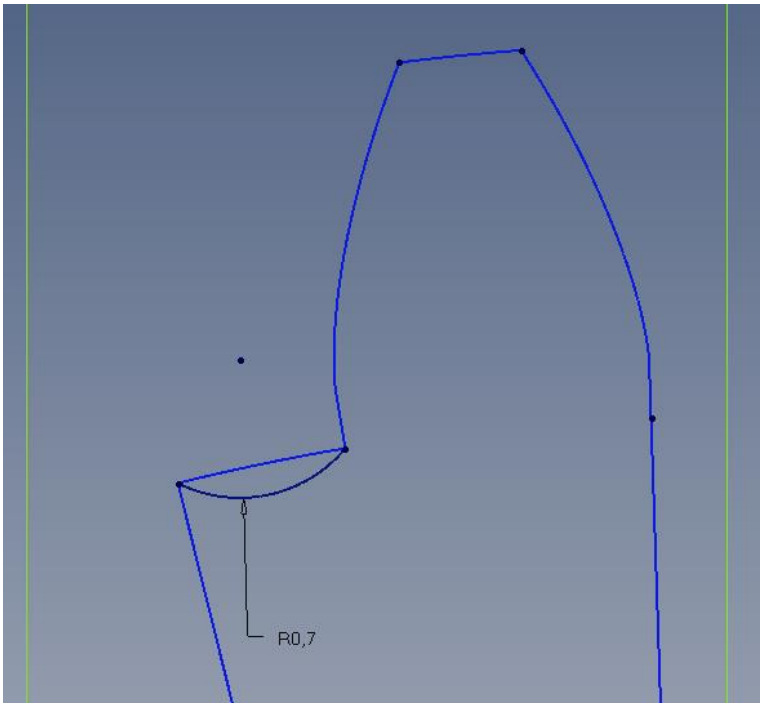
Zaczynamy od użycia funkcji AddGear, aby wygenerować pojedynczy ząb ze środkiem w punkcie (0, 0). W tym przykładzie średnica podziałowa wynosi 25,4 mm, a liczba zębów 20, a kąt przyporu 20 stopni:

```
Kolo_Zebate_Profil = Kolo_Zebate.AddGear("Koło zębate", 30, 30,
25.4, 20, True, 0, 0, 30, PlaszczyznaXY)
```

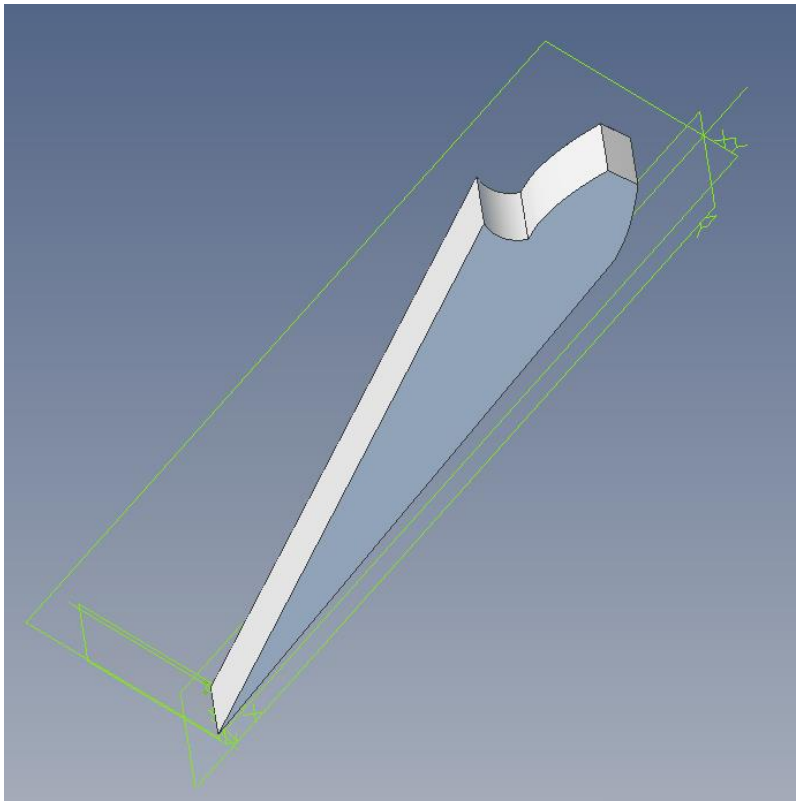
© Copyright Datacomp 2018



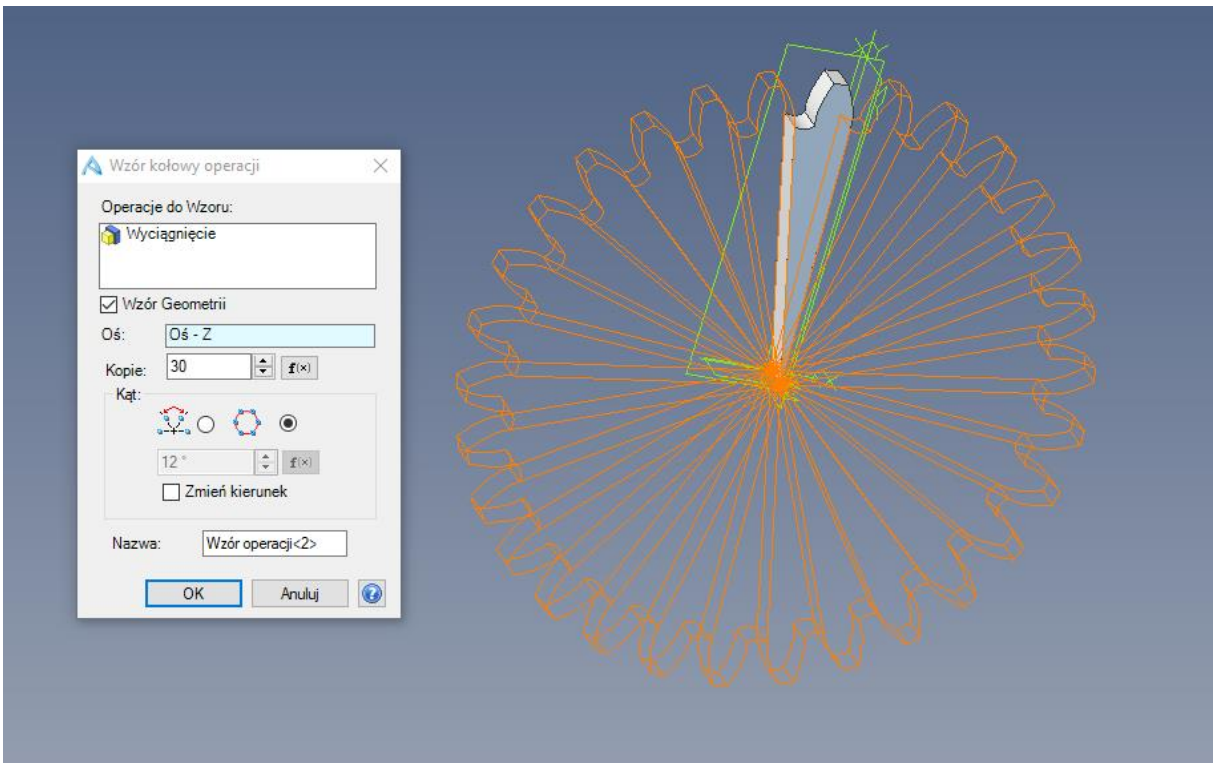
Teraz przejdź bezpośrednio do Alibre Design. Edytuj szkic dodając pożądane podcięcie:



Wyciągnij szkic do na żądaną grubość:

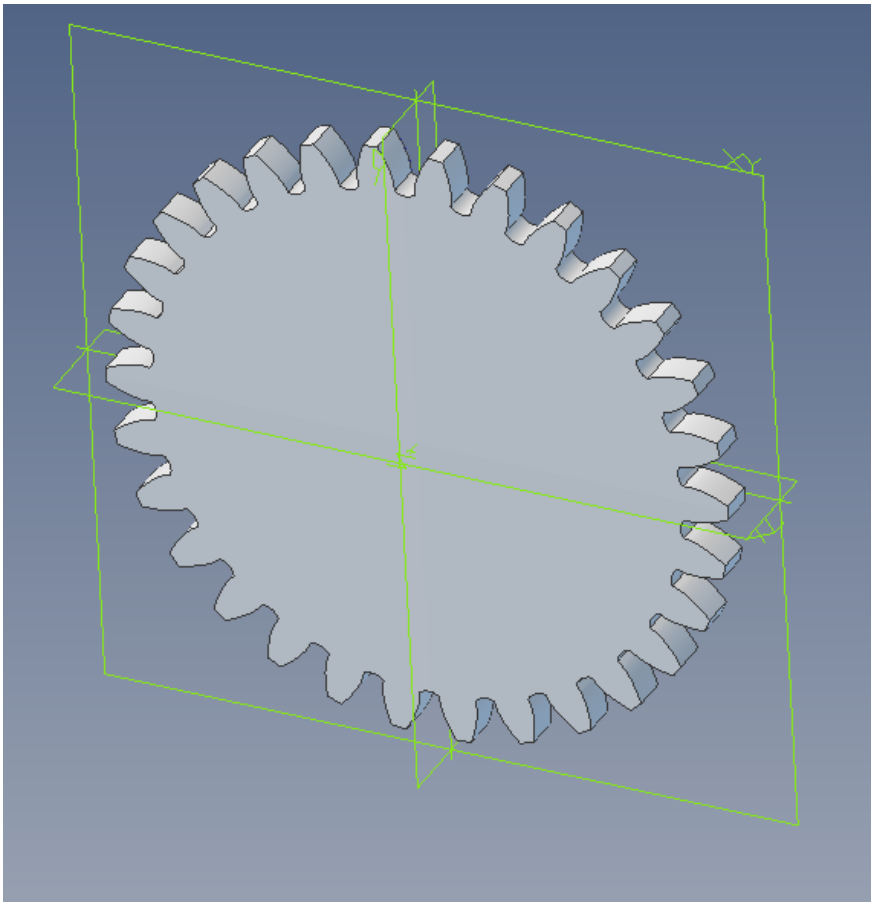


Stwórz szyb kołowy wokół osi Z. Wiemy, że koło zębate ma 30 zębów, zatem potrzebujemy 30 kopii.

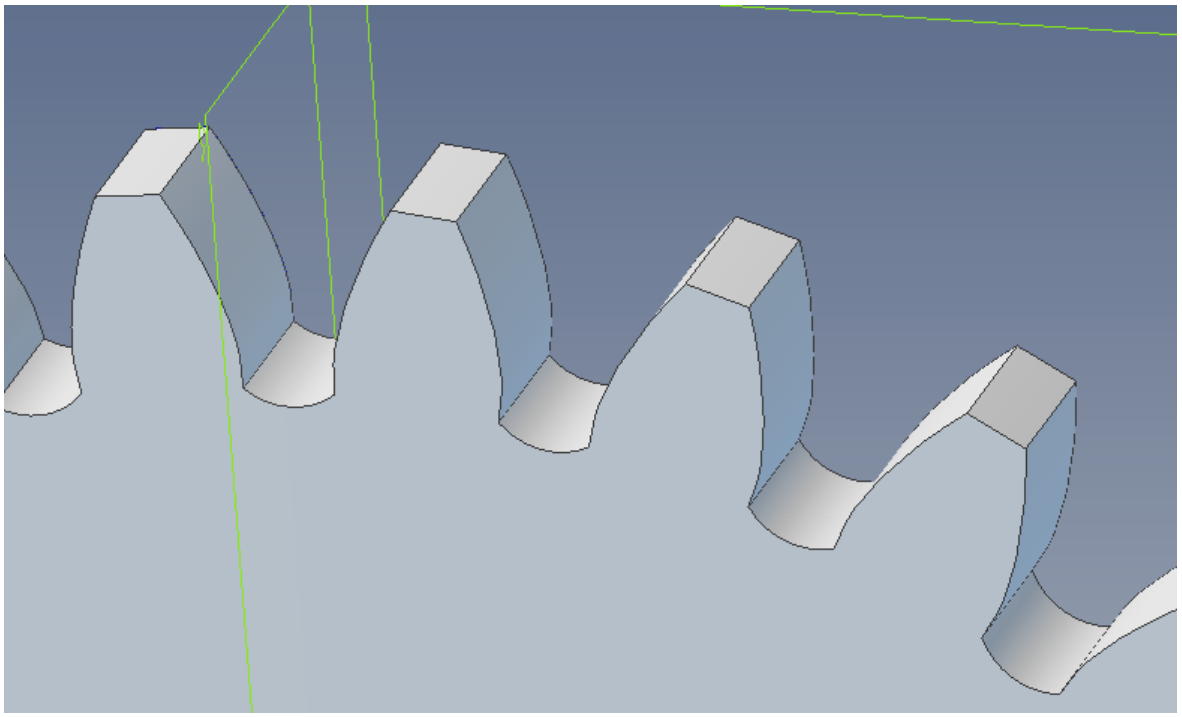


© Copyright Datacomp 2018

Kompletne koło zębate:



Zbliżenie na zmodyfikowane zęby:



© Copyright Datacomp 2018

ROZDZIAŁ 7:
MODYFIKACJA ISTNIEJĄCYCH CZĘŚCI I SZKICÓW

Czasami przydatne może być użycie skryptu do modyfikacji części lub szkicu, które zostały już utworzone. Na przykład skrypt może zostać napisany w celu wykonania powtarzalnego zadania, które nie jest dostępne w interfejsie użytkownika Alibre Design.

Aby uzyskać dostęp do istniejącej części, otwórz część w Alibre Design, a następnie:

```
MojaCzesc = Part('Nowa Część (1)', False)
```

Wstaw nazwę części, jak pokazano w eksploratorze projektu. Drugi parametr jest ustawiony na False, co oznacza, że Alibre Script używa części, która została już utworzona lub otwarta.

Od teraz mamy normalny dostęp do części w dalszym ciągu skryptu.

Aby uzyskać dostęp do szkicu części:

```
MojSzkic = MojaCzesc.GetSketch('Szkic<1>')
```

Wstaw nazwę szkicu, jak pokazano w eksploratorze projektu. Szkic może być teraz modyfikowany przez resztę skryptu.

Linie, okręgi, łuki itp. Zdefiniowane w szkicu nazywane są "figurami"(Figures). Liczby są dostępne jako lista w Pythonie. Na przykład:

```
print len(MojSzkic.Figures)
```

Wypisze całkowitą liczbę figur zawartych w szkicu. Możemy zobaczyć szczegóły konkretnej figury:

```
print MojSzkic.Figures[0]
Circle centered at -15,-19 with radius 19
```

© Copyright Datacomp 2018

Możemy uzyskać dostęp do tej figury i wykorzystywać ją w innych szkicach i innych częściach:

```
Fig = MojSzkic.Figures[0]
NowaCzesc = Part("Nowa")
NowySzkic = NowaCzesc.AddSketch("Szkic", NowaCzesc.GetPlane("Płaszczyzna XY"))
NowySzkic.AddFigure(Fig)
```

Korzystając z tej techniki, można tworzyć szkice, uzyskiwać do nich dostęp i wykorzystywać je wielokrotnie, nawet z niewielkimi zmianami dla każdej nowej części, w której są używane:

```
MojOkrag = MojSzkic.Figures[0]
MojOkrag.Center = [4, -3]
NowySzkic.AddCircle(MojOkrag)
```

```
MojaBspline = MojSzkic.Figures[3]
MojaBspline.IsReference = True
NowySzkic.AddBspline(MojaBspline)
```

Zwróć uwagę, że obecnie elipsy i łuki eliptyczne nie są obsługiwane z powodu ograniczeń w Alibre Design.

© Copyright Datacomp 2018

ROZDZIAŁ 8: PORADY I WSKAZÓWKI

1. Możliwe jest tworzenie i edytowanie wielu części na raz. Na przykład:

```
Ramka = Part("Ramka")
Belka = Part("Belka")
Szkic_Glowny = Ramka.AddSketch("Podstawa", Ramka.GetPlane("Płaszczyzna XY"))
Profil_Szkic = Belka.AddSketch("Profil", Belka.GetPlane("Płaszczyzna XY"))
```

2. Nie jest koniecznie zapisywanie skryptu przed uruchomieniem go. Ułatwia to szybszą edycję i testowanie skryptów.

3. Szkic tymczasowy może zostać utworzony i użyty jako szablon dla szkiców na innych częściach, a następnie usunięty. Na przykład:

```
# tworzy szkic tymczasowy
TymczCzesc = Part("Czymczasowa")
TymczSzkic = TymczCzesc.AddSketch("Tymczasowy", TymczCzesc.GetPlane("Płaszczyzna XY"))
TymczSzkic.AddCircle(...)
TymczSzkic.AddLine(...)

# kopiowanie do szkicu na część A
SzkicNaCzescA.CopyFrom(TymczSzkic)

# kopiowanie do szkicu na część B, zwiększając rozmiar o 25%
SzkicNaCzescB.CopyFrom(TymczSzkic, 0, 0, 0, 0, 0, 0, 0, 125.0)

# kopiowanie jednego elementu do szkicu na część C najpierw modyfikując część
Figura = TymczSzkic.Figures[1]
Figura.IsReference = True
SzkicNaCzescC.AddFigure(Figura)

# delete temporary sketch
TymczSzkic.Close()
```

© Copyright Datacomp 2018