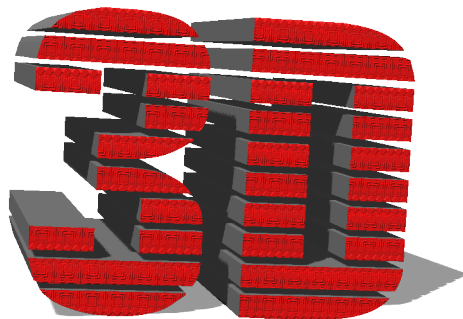


W kolejnym odcinku naszego kursu zaprezentujemy kilka rozwiązań, które pozwolą poznać nietypowe możliwości oprogramowania OpenSCAD, a także umożliwią automatyzację i uproszczenie wielu zadań. Na początek zaprojektujemy i wykonamy oryginalne etui na klucze, w formie szwajcarskiego scyzoryka.

# PRAKTYCZNY KURS DRUKU



## Lekcja 7

### OpenSCAD, program o dużych możliwościach

Kształt scyzoryka musi być dopasowany do kluczy, dlatego warto najpierw dokonać pomiaru standardowego klucza, a dopiero potem przystąpić do projektowania.

W naszym przypadku został zmierzony klucz typu YALE jako podstawowy oraz inne klucze zbliżonych systemów. Na bazie tych danych oraz przy założeniu, że klucze będą chowane po dwa i po obu końcach, długość etui została ustalona na ok. 93 mm. Najlepiej wybrać trzy punkty mocowania – dwa na końcach służą do spinania kluczy, a jeden znajdzie się na środku, będąc jednocześnie ogranicznikiem podczas składania kluczy w scyzoryku. Grubość każdej z okładek w przypadku użycia PLA

powinna wynosić ok. 3 mm, natomiast w przypadku ABS co najmniej 5 mm.

Na początek należy zdefiniować podstawowy kształt okładziny, złożony z prostopadłościanu i dwóch cylindrów na jego końcach (1).

Kod okładziny powinien wyglądać następująco:

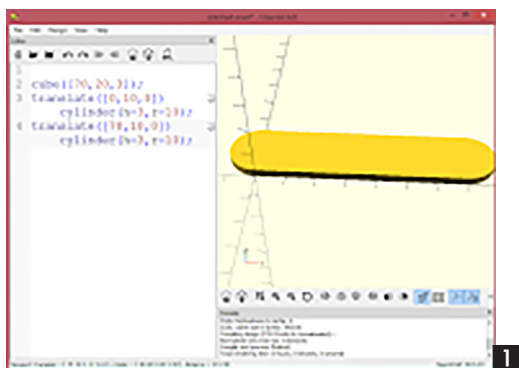
```
//okładzina
cube([70,23,3]);
translate([0,11.5,0]) cylinder(h=3,r=11.5);
translate([70,11.5,0]) cylinder(h=3,r=11.5);
```

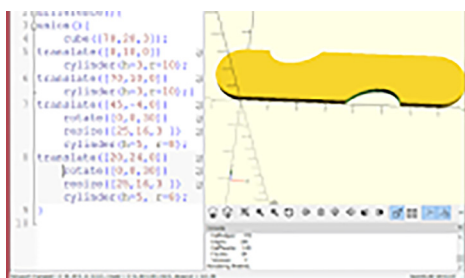
W kolejnym kroku należy scalić bryły tworzące okładzinę, funkcją `union()`, co pozwoli następnie wyciąć na brzegach zagłębienia przeznaczone do otwierania kluczy.

Po zmianach kod będzie wyglądał następująco:

```
union() {
//okładzina
cube([70,23,3]);
translate([0,11.5,0]) cylinder(h=3,r=11.5);
translate([70,11.5,0]) cylinder(h=3,r=11.5);}
```

Teraz należy po bokach wyciąć zagłębienia, które ułatwią otwieranie kluczy ze scyzoryka. Do tego celu trzeba użyć funkcji `resize([x,y,z])` na cylindrze, tworząc owal, który powinien być przekreślony względem krawędzi o  $30^\circ$  (2):





2

```
difference() {
union() {
//okładzina
cube([70,23,3]);
translate([0,11.5,0]) cylinder(h=3,r=11.5);
translate([70,11.5,0]) cylinder(h=3,r=11.5);}
translate([45,4,0]) rotate([0,0,30])
resize([25,16,3]) cylinder(h=5, r=8);
translate([20,24,0]) rotate([0,0,30])
resize([25,16,3]) cylinder(h=5, r=8);
}
```

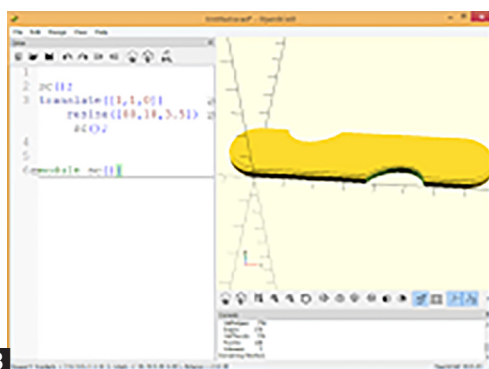
Mając gotową okładzinę, warto zdefiniować ją jako moduł. Upraszczca to dalsze projektowanie. Warto wyrobić sobie nawyk definiowania powtarzających się elementów projektu jako moduły. W tym przypadku moduł należy nazwać `sc()`:

```
sc(); //wywołanie modułu
module sc() { //definicja modułu
difference() {
union() {
//okładzina
cube([70,23,3]);
translate([0,11.5,0]) cylinder(h=3,r=11.5);
translate([70,11.5,0]) cylinder(h=3,r=11.5);}
translate([45,4,0]) rotate([0,0,30])
resize([25,16,3]) cylinder(h=5, r=8);
translate([20,24,0]) rotate([0,0,30])
resize([25,16,3]) cylinder(h=5, r=8);
}}
```

Dobrze, gdyby krawędzie scyzoryka były lekko zaokrąglone. W naszym przypadku wystarczy nałożyć na bazową okładzinę jej zmniejszoną wersję, co w zupełności zniweluje ostrość krawędzi. W tym celu należy wywołać moduł `sc()` z funkcją `resize()`, definiując rozmiar mniejszy o 1 mm z każdej strony oraz o 0,5 mm wyższy, np.: (3)

```
sc(); //wywołanie modułu
translate([0.5,1,0]) resize([91,21,3.5]) sc();
//ponowne wywołanie oraz skalowanie modułu
module sc(){...} //moduł kropki oznaczają kod modułu napisany wcześniej.
```

Nadszedł czas na otwory montażowe oraz doprecyzowanie dokładności okrągłych elementów. Używając funk-

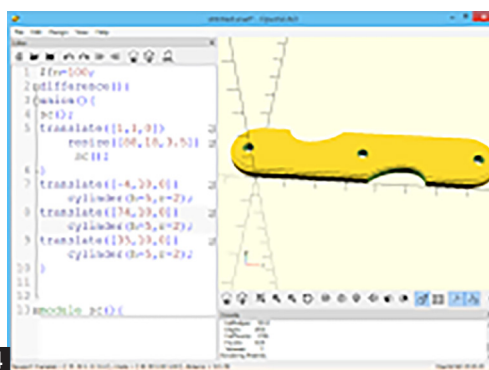


3

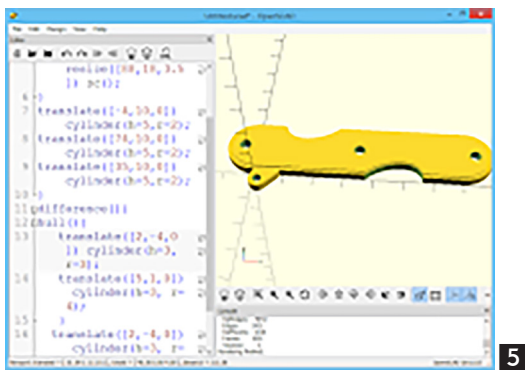
cji `$fn=wartość`, można ją zdefiniować na początku programu dla wszystkich elementów lub dla każdego osobno jako parametr po przecinku. W tym pierwszym przypadku kompilacja podglądu lub eksportu wielokrotnie się wydłuży, szczególnie dla bardziej skomplikowanych projektów – należy o tym pamiętać lub dodawać opcję na samym końcu projektowania. Otwory montażowe muszą się znaleźć na końcu okładziny, zgodnie z wymiarami klucza, oraz dokładnie na środku, jako stoper podczas ich „składania” (4):

```
$fn=100; //dokładność dla całego projektu
difference() {
union() {
sc();
translate([0.5,1,0]) resize([91,21,3.5]) sc();
}
translate([4,11.5,0]) cylinder(h=5,r=2); //otwory
montażowe
translate([74,11.5,0]) cylinder(h=5,r=2);
translate([35,11.5,0]) cylinder(h=5,r=2); //
stoper
}
module sc(){...} //moduł kropki oznaczają kod
modułu napisany wcześniej.
```

Warto pamiętać o uchwycie na smycz lub jakiś łańcuszek – można taki uchwyt szybko wkomponować, używając



4



5

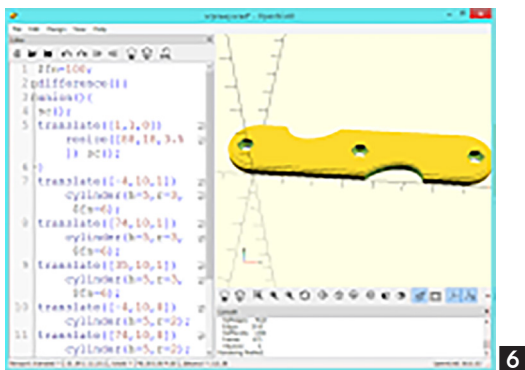
jąc funkcji `hull()`. Pozwala ona łączyć ze sobą dwa różnej wielkości cylindry, przesunięte względem siebie w jednolitą bryłę – wystarczy umieścić cylindry w wybranych miejscach.

Przykład kodu, który należy dodać do poprzedniego w celu uzyskania uchwytu (5):

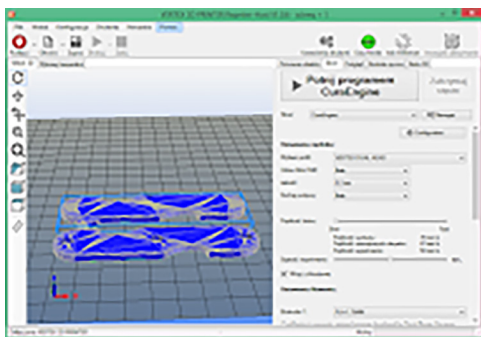
```
difference() {
hull() { //łączy dwa cylindry w jedną bryłę
translate([2,4,0]) cylinder(h=3, r=3);
translate([5,1,0]) cylinder(h=3, r=4); }
translate([2,4,0]) cylinder(h=3, r=1.5); //
wycina otworek na smycz
}
```

W ten sposób została utworzona ostateczna wersja jednej z okładzin. Należy jeszcze wykonać drugą. Nie musi mieć uchwytu, ale koniecznie należy ją odwrócić o 180° oraz wyciąć otwory montażowego w kształcie sześciokąta, które ułatwią montaż nakrętek (6):

```
$fn=100;
rotate([180,0,0])//odwrócenie elementu o 180°
difference() {
union() {
sc();
translate([0.5,1,0.5]) resize([91,21,3.5])
sc();// należy przesunąć w dół o 0,5 mm,
po odwróceniu będzie na górze
```



6



7

```
}
translate([4,11.5,0.5]) cylinder(h=5,r=2);
translate([74,11.5,0.5]) cylinder(h=5,r=2);
translate([35,11.5,0.5]) cylinder(h=5,r=2);
translate([4,11.5,0.5]) cylinder(h=2,r=3,
$fn=6); //wycięcie otworu w kształcie sześciokąta
translate([74,11.5,0.5]) cylinder(h=2,r=3,
$fn=6); //wycięcie otworu w kształcie sześciokąta
translate([35,11.5,0.5]) cylinder(h=2,r=3,
$fn=6); //wycięcie otworu w kształcie sześciokąta
}
module sc(){...} //moduł kropki oznaczają kod
modułu.
```

Każdą z okładzin należy renderować (F6), a następnie wyeksportować jako pliki \*.STL. Teraz pozostał import do Repetier hosta – trzeba ustawić wypełnienie na poziomie 50%, grubość warstwy na 0,2 mm, pociąć i zapisać na karcie pamięci (7). Wydruk obu elementów trwa ok. 30 minut. W międzyczasie można spróbować wykorzystać kilka specjalnych funkcji programu OpenSCAD w celu upiększenia okładzin, co jest istotne i ważne, jeśli nasz przedmiot ma być prezentem.

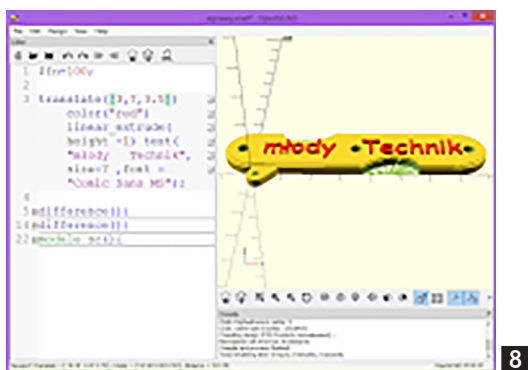
Ciekawą opcją jest możliwość dodania napisów, które mogą być wypukłe lub wklęsłe. Do tego celu używa się komendy `text(„młody Technik”, font = „Comic Sans MS”)`; oraz oczywiście `translate([xyz])`, w celu dokładnej lokalizacji.

W przypadku szczyryka na klucze kod będzie wyglądał następująco (8):

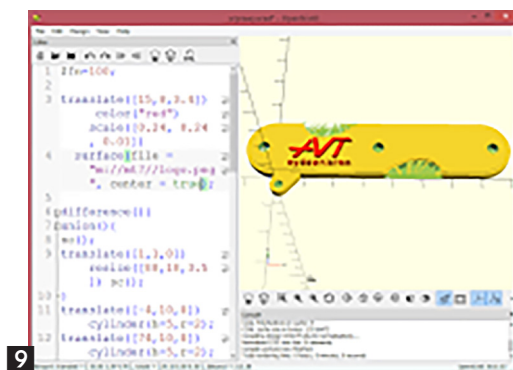
```
translate([3,7,4]) color(„red”) linear_
extrude(height = 1) text(„młody Technik”, size=7
,font =
„Comic Sans MS”);
```

Uważny Czytelnik dostrzeże w tym kodzie nową komendę `color(„red”)`. Pozwala ona ustawić kolor bryły lub brył (gdy użyto „union”). To ważne w przypadku projektów złożonych lub wymagających dokładnego określenia położenia względem siebie.

Dodatkowo pojawiła się komenda `linear_extrude(height = 1)`, która odpowiada za stopień wypukłości (wy-



8



9

miar w osi Z), opcja size pozwala ustalić wielkość czcionki, a opcja font podać, jaka czcionka ma być użyta. Można także ustalać położenie względem punktu zaczepienia, ustalać podkreślenie, styl (np. kursywa), a nawet odległości pomiędzy literami.

## Tworzenie elementów na bazie obrazków graficznych

Kolejną superopcją jest możliwość tworzenia obiektów na bazie płaskich plików graficznych w formacie \*.png. Jako przykład importu obrazka może posłużyć logo wydawnictwa AVT, które zostanie umieszczone na jednej ze ścianek okładziny. Wystarczy w tym celu podać komendę:

```
translate([15,8,3.4]) color('red') scale([0.24, 0.24, 0.01]) surface(file = \"m://logo.png\", center = true);
```

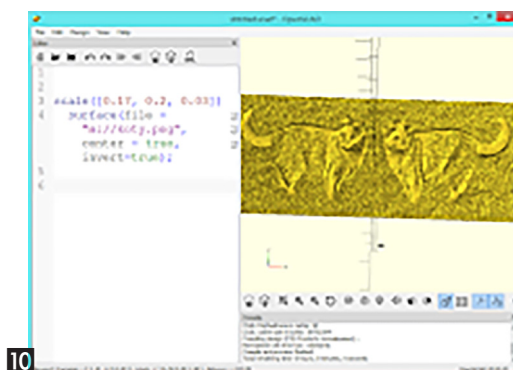
Ważne, aby poprawnie wskazać lokalizację pliku, zapis ścieżki „c:\” musi mieć postać „c://” (9).

Rozmiar w osi XYZ definiuje się w parametrach scale([0.24, 0.24, 0.01]). Wielkość należy dobrać doświadczalnie, obserwując podgląd. Najlepszy efekt uzyskuje się dla obrazków jednobarwnych (monochromatycznych), jednak można także importować grafiki kolorowe. Ta ostatnia opcja umożliwia tworzenie przestrzennych obrazków, w których po wydrukowaniu światło wydobywa odcienie – w ten sposób można tworzyć ciekawe witraże lub lampy. Przykładem niech będzie obrazek domowego zwierzaka [koty.png]. Można z niego zrobić osłonkę na świeczkę, idealną na prezent dla hodowcy kotów. Po zaciągnięciu obrazka do programu należy dobrać jego skalę, można także go obciąć, obrócić, połączyć z innymi obiektami.

Podstawowy kod (10):

```
scale([0.17, 0.2, 0.03]) surface(file = \"m://koty.png\", center = true, invert=true);
```

Po wydrukowaniu i umieszczeniu pod światło można dostrzec obrazek, który jest malowany światłem, na bazie grubości warstw plastiku i przenikalności filamentu.



10

Ważne, aby pamiętać o ustawieniu wysokiego poziomu wypełnienia. Zaleca się stosować wypełnienie powyżej 85%, bo niższa wartość może skutkować przebijaniem struktury wypełnienia w obrębie ciemnych części obrazka, psując cały efekt. Obrazek przedstawia widok wydrukowanego zdjęcia oglądanego pod światło (11).

OpenSCAD umożliwia także import plików typu CAD, np. \*.dxf (format Autocada), (12) do czego służy komenda:

```
linear_extrude(height=2) import(\"C://rekin.dxf\");
```

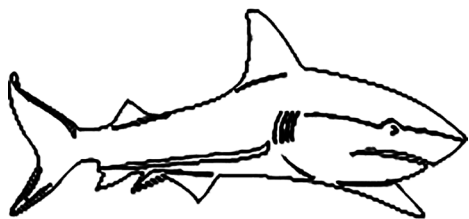
Funkcja linear\_extrude() umożliwia ustalenie grubości (w osi Z) parametrem height=, skrócenie projektu w stopniach parametrem twist=, dokładności skręcanego obiektu (poprzez liczbę kroków) parametrem slices= (13).

## Montaż „scyzoryka”

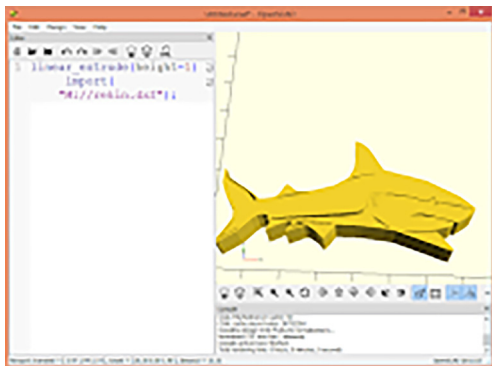
Udało się wydrukować elementy scyzoryka, trzeba go jeszcze jedynie złożyć. Potrzebne do tego będą trzy sru-



11



12



13

by M3 16 mm wraz z nakrętkami oraz śrubokręt. Należy wkręcić śruby w prawą okładzinę, (14) następnie nałożyć klucze, (15) na klucze drugą okładzinę, na koniec wystarczy nałożyć nakrętki i wszystko dokładnie skrócić (16). Skrócone śruby można zabezpieczyć przed odkręcaniem specjalnym klejem lub niewielką ilością bezbarwnego lakieru do paznokci, można także użyć nakrętek z wkładką teflonową. Jak widać na ilustracjach, warto także zaprojektować konektory do kluczy, ponieważ każdy producent ma swój standard otworu do zawieszania.

To oczywiście tylko luźny projekt i propozycja wykorzystania drukarki 3D. Każdy sam podczas projektowania i realizacji może wprowadzać usprawnienia, do czego serdecznie zapraszam. Jeżeli ktoś uzna, że pragnie się pochwalić swoimi modernizacjami z Czytelnikami, zapraszam do przesyłania zdjęć i ewentualnych projektów – najciekawsze zostaną zaprezentowane.

Na zakończenie tego odcinka pragnę zaprezentować jeszcze kilka ciekawych uwag.

## Bryły o podstawie trójkąta

Narysowanie graniastosłupa lub ostrosłupa o podstawie trójkąta równobocznego jest dość trudne, na szczęście można skorzystać z pewnych właściwości służących do ustalania dokładności brył kolistych \$fn=;\$. Ustalając wartość na poziomie 3, zawsze uzyskuje się trójkąt równoboczny wpisany w okrąg (17):

```
cylinder(r=5, h=7, $fn=3); //rysuje
graniastosłup o podstawie trójkąta równobocznego
translate([10,0,0]) cylinder(r1=5,r2=0, h=7,
$fn=3); //czworoscian foremny.
```



14

15



16

## Owal

Narysowanie owalu także jest trudne, jednak korzystając z funkcji zniekształcających cylinder, można uprościć to działanie i to na dwa sposoby (18).

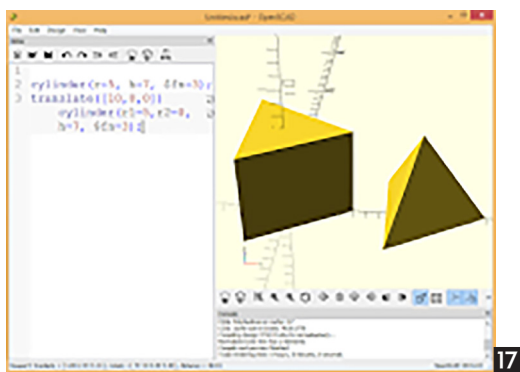
```
$fn=100;
scale([1, 3, 1]) cylinder(r=5, h=1); //skala
umożliwia skalowanie w każdym z wymiarów
resize([10,30,1]) cylinder(r=5,h=1); //umożliwia
dopasowanie bryły do konkretnych potrzeb
```

Korzystając z tej funkcji, w łatwy sposób można zbudować kadłub łodzi (19).

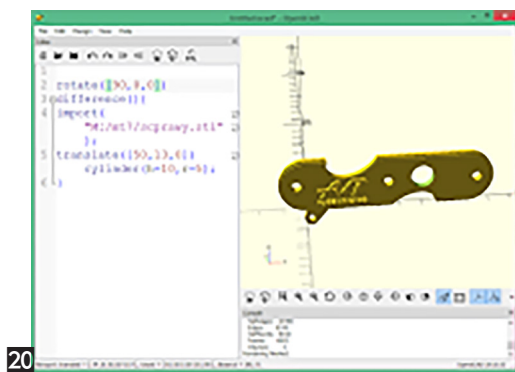
```
$fn=100;
difference() {
resize([12,30,10]) cylinder(r1=10, r2=12,h=10);
//zdeformowany stożek ścięty
translate([0,0,1]) resize([10,28,10])
cylinder(r1=10, r2=12,h=10);
translate([20,10,0]) rotate([5,0,0]) cube(50);
}
translate([3.2,9,1]) rotate([5,0,0])
cube([6,1,9]);
translate([3.5,9.3,5]) rotate([5,0,0])
cube([7,1,5]);
```

## Praca z plikami stl

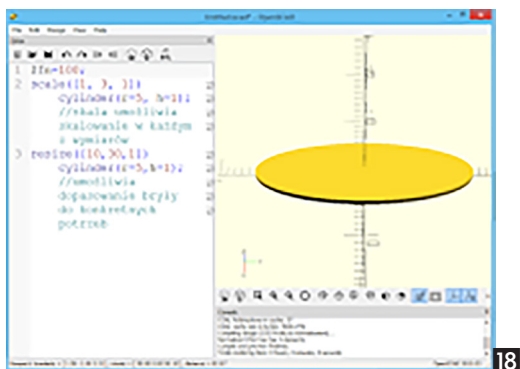
Ciekawą funkcjonalnością aplikacji OpenSCAD jest możliwość pracy z plikami stl, dzięki czemu można dokonywać zmian i korekt w plikach, które są nieodpłatnie udostępniane w Internecie. Jest wiele miejsc w Sieci, z których można pobrać gotowe projekty, niestety bardzo często są one dostosowane do akcesoriów i narzędzi występujących w kraju pochodzenia. W tej sytuacji pozostaje dokonać zmian lub poprawek. Po zaimportowaniu pliku \*.stl do aplikacji (wystarczy przeciągnąć plik myszką na pole komend) można go traktować jak jeden obiekt, co pozwala na obracanie, cięcie lub dodawanie elementów (20).



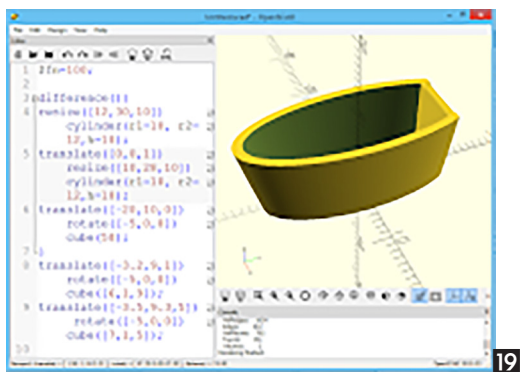
17



20



18



19

Często się zdarza, że pobrane pliki są generowane w złej płaszczyźnie, co praktycznie uniemożliwia poprawny wydruk. Taki plik można wciągnąć do OpenSCAD, a następnie funkcją `rotate([XYZ])` obrócić tak, aby wydruk był możliwy.

Inną ciekawą opcją pracy z plikami \*.stl jest tworzenie rysunków złożeniowych. Można zaciągnąć wszystkie poszczególne elementy projektu, nadać im kolory, a następnie przesunąć funkcją `translate([XYZ])`, aż do uzyskania całego urządzenia. Jako przykład posłuży rysunek złożonego szczyryka na klucze (21):

```

translate([0,0,5]) color("blue") import("M:/scprawy.stl");
rotate([180,0,0]) color("blue") import("M:/sclwy.stl");
translate([45,20,5]) rotate([0,180,10])
color("silver") import("M:/yale_klucz.stl");
translate([25,3,5]) rotate([180,0,10])
color("silver") import("M:/yale_klucz.stl");

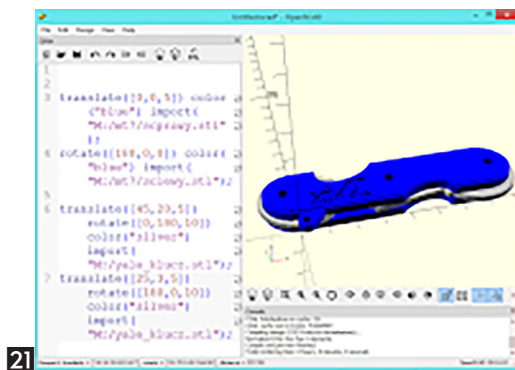
```

## Podsumowanie

OpenSCAD to wdzięczne środowisko programistyczne. Zachęcam do korzystania z jego możliwości. Zapraszam do dzielenia się projektami lub publikowania swoich pomysłów. W połowie roku, w Warszawie, odbędą się zawody w druku 3D, skierowane dla szkół oraz osób indywidualnych. Zapowiada się idealna okazja, by każdy „drukarni” i projektant mógł pochwalić się swoim projektem oraz własnymi umiejętnościami.

Przyszłość druku 3D rysuje się w różowych kolorach – wkracza on z impetem we wszystkie dziedziny nauki i rozrywki. W następnym odcinku zostanie przedstawiona rozbudowa drukarki VERTEX K8400 o drugą głowicę oraz krótki kurs drukowania z użyciem obu głowic. ■

*Jarosław Kita*



21