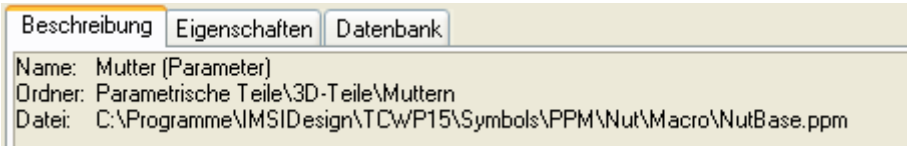
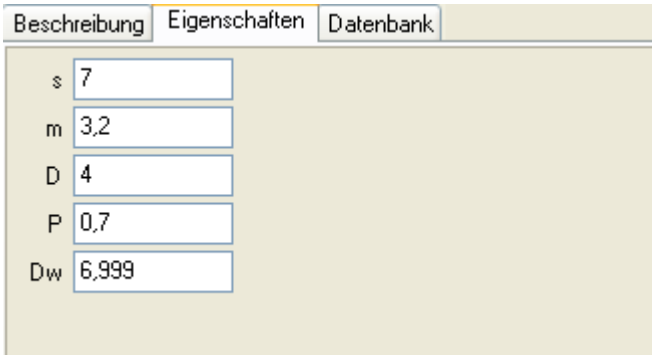
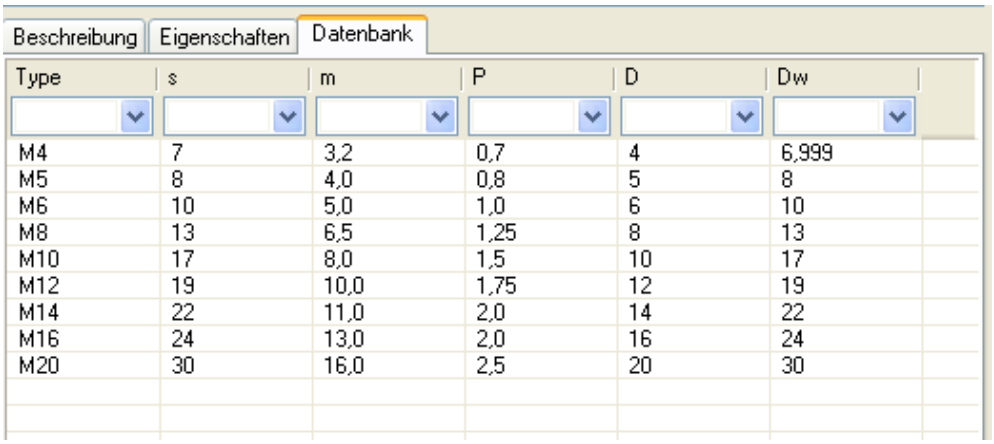
Parametrische Muttern







Type	s	m	P	D	Dw
M4	7	3,2	0,7	4	6,999
M5	8	4,0	0,8	5	8
M6	10	5,0	1,0	6	10
M8	13	6,5	1,25	8	13
M10	17	8,0	1,5	10	17
M12	19	10,0	1,75	12	19
M14	22	11,0	2,0	14	22
M16	24	13,0	2,0	16	24
M20	30	16,0	2,5	20	30

// Nut.ppm

```
Type = Parameter("Muttertyp", "M6",
Set("M4","M5","M6","M8","M10","M12","M14","M16","M20"));


Units(1[mm]);
res9  = IF( Type=="M20",NutBase( 30,  16.0, 2.5,  20,  30 ), 0);
res8  = IF( Type=="M16",NutBase( 24,  13.0, 2.0,  16,  24 ), res9);
res7  = IF( Type=="M14",NutBase( 22,  11.0, 2.0,  14,  22 ), res8);
res6  = IF( Type=="M12",NutBase( 19,  10.0, 1.75, 12,  19 ), res7);
res5  = IF( Type=="M10",NutBase( 17,  8.0,  1.5,  10,  17 ), res6);
res4  = IF( Type=="M8", NutBase( 13,  6.5,  1.25, 8,   13 ), res5);
res3  = IF( Type=="M6", NutBase( 10,  5.0,  1.0,  6,   10  ), res4);
res2  = IF( Type=="M5", NutBase( 8,   4.0,  0.8,  5,   8  ), res3);
res   = IF( Type=="M4", NutBase( 7,   3.2,  0.7,  4,   6.999  ), res2);

Output( res );
```

// Nutbase.ppm

```
//describe base nut
//default values == M4
S = Parameter("s", 7,  LINEAR); //size
M = Parameter("m", 3.2, LINEAR); //thickness
P = Parameter("P", 0.7, LINEAR);  //pitch
D = Parameter("D", 4, LINEAR); //inner diameter
Dw = Parameter("Dw", 6.999, LINEAR);
Input(S,M,P,D,Dw);
Units(1[mm]);
//base hexagon
R = S/(2*cos(30));
a = S*tan(30);
Base2D = Polyline( Point(0, R),
                   Point(S/2, a/2),
                   Point(S/2, a/(-2)),
                   Point(0, R*(-1)),
                   Point(S/(-2), a/(-2)),
                   Point(S/(-2), a/2),
                   Point(0, R)
                  );
Base3D = Thickness(Base2D, M);
//make hole
Hole = Thickness( Circle((D-1.08*P)/2), M );


//inner edges in the hole
ConusInner1 = Cone(D*0.866,D/2,0);
ConusInner2 = RotateX(Move(ConusInner1,0,0,-M),180);


NutBase = BooleanSubtract( Base3D, Hole );
Nut1 = BooleanSubtract(NutBase, ConusInner1);
Nut2 = BooleanSubtract(Nut1, ConusInner2);


//outer edges
tgA = tan(30);
nr = (2*M + Dw*tgA)/(2*tgA);
ConusOuter1 = Cone((M + (Dw/2)*tgA),nr,0);
ConusOuter2 = Move(RotateX(ConusOuter1,180),0,0,M);
Nut3 = BooleanIntersect(Nut2,ConusOuter1);
Nut4 = BooleanIntersect(Nut3,ConusOuter2);
Output(Nut4);
```