

---

# Графика в Wolfram Mathematica

Юдинцев В. В.

Самарский университет  
Кафедра теоретической механики

Стили по умолчанию...

```
In[1068]:= SetOptions[Graphics, BaseStyle -> {14, FontFamily -> "Helvetica"}];
```

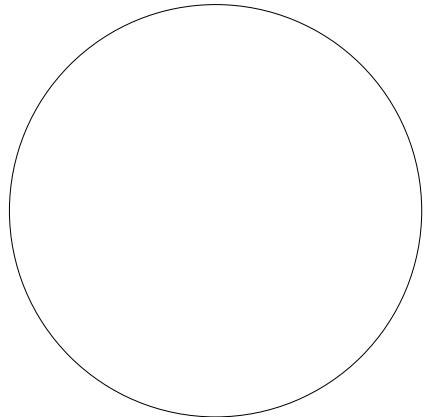
# Геометрические объекты в Mathematica

## Окружность

In[1070]:=

```
Graphics[  
  Circle[{0, 0}, 2]  
]
```

Out[1070]=

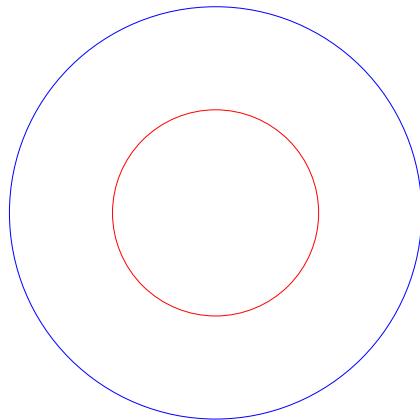


## Graphics

Функция Graphics отображает графические примитивы, передаваемые ей в виде списка. Графические примитивы могут чередоваться описаниями их атрибутов (цвет и толщина линий, тип линий, ...)

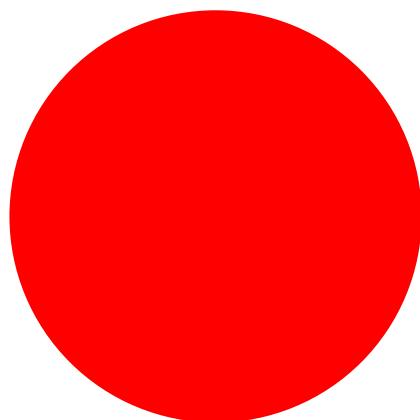
```
In[1071]:= Graphics[{Red, Circle[{0, 0}, 1], Blue, Circle[{0, 0}, 2]}]
```

```
Out[1071]=
```



```
In[1072]:= Graphics[{Red, Disk[{0, 0}, 1]}]
```

```
Out[1072]=
```



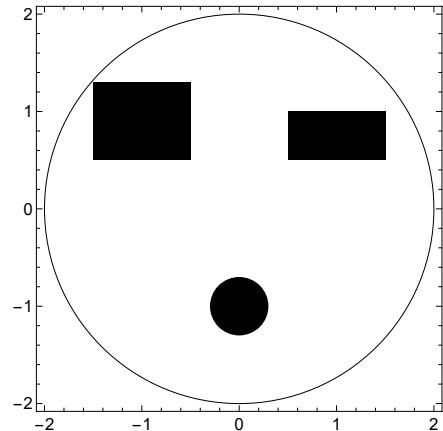
# Несколько объектов

Несколько объектов  $\{g_1, g_2, g_3, \dots, g_n\}$

In[1073]:=

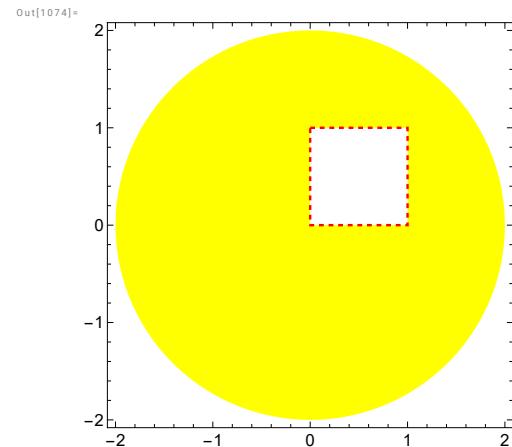
```
Graphics[{  
  Circle[{0, 0}, 2],  
  Rectangle[{0.5, 0.5}, {1.5, 1}],  
  Rectangle[{-1.5, 0.5}, {-0.5, 1.3}],  
  Disk[{0, -1}, 0.3]  
,  
 Frame → True]
```

Out[1073]=



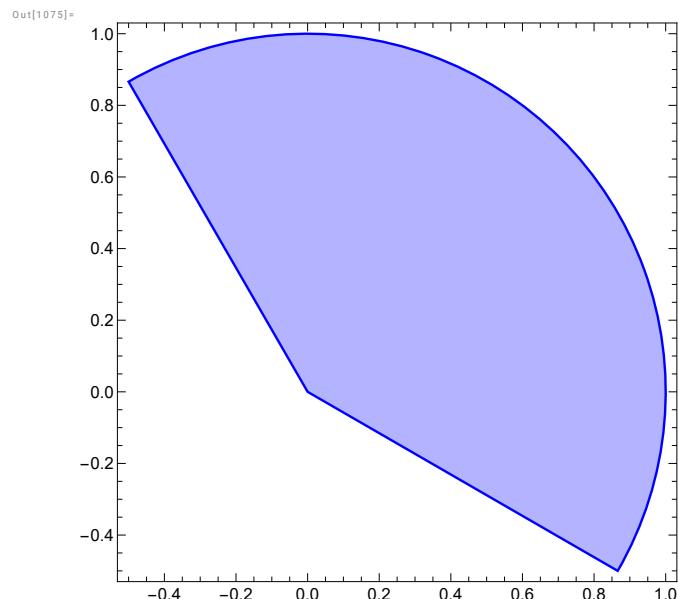
## Контур и заливка

```
In[1074]:= Graphics[{
  Yellow, (* Желтый *)
  Disk[{0, 0}, 2], (* Диск *)
  White, (* Белый *)
  EdgeForm[{Thick, Red, Dashed}], (* Толстая красная рамка пунктиром *)
  Rectangle[{0, 0}, {1, 1}] (* Прямоугольник *)
},
Frame → True]
```



# Сектор

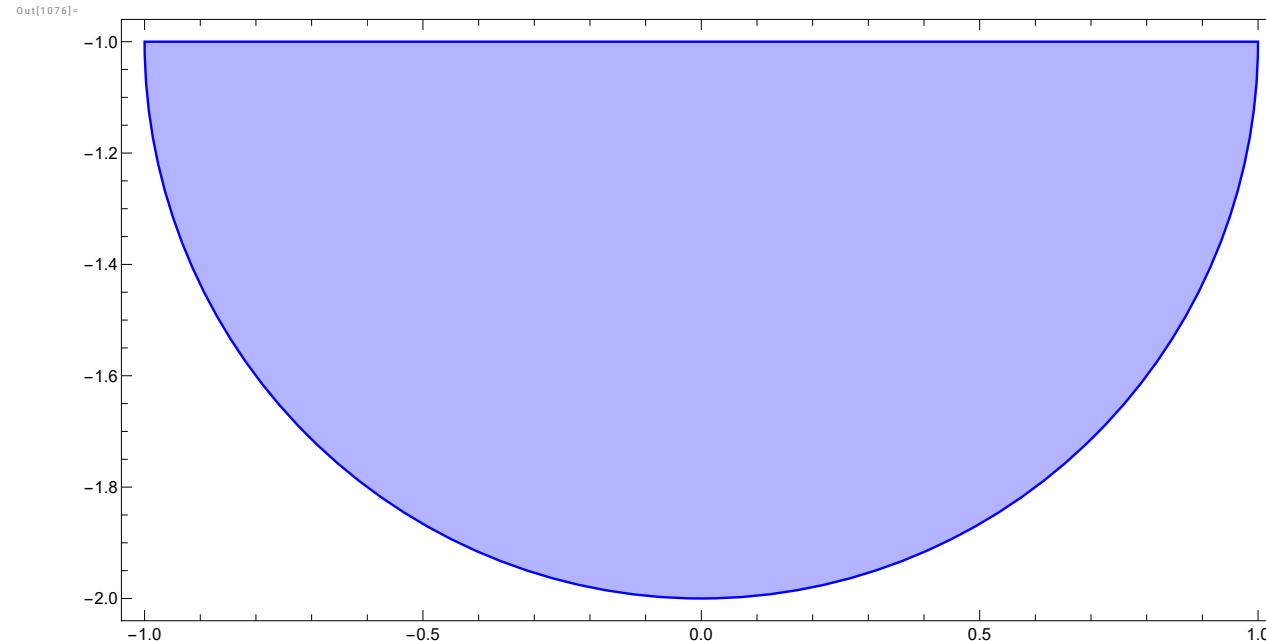
```
In[1075]:= Graphics[
 {
 Lighter[Blue, 0.7], (* Светлее на 70 %. 100 % -- белый *)
 EdgeForm[{Thick, Blue}], (* толстая голубая линия *)
 Disk[{0, 0}, 1, {-30 °, 120 °}]
 },
 Frame → True(* Показать рамку рисунка с надписями осей *),
 ImageSize → 500
 ]
```



Угол отсчитывается от горизонтальной оси против часовой стрелки

## Тело 1 механизма

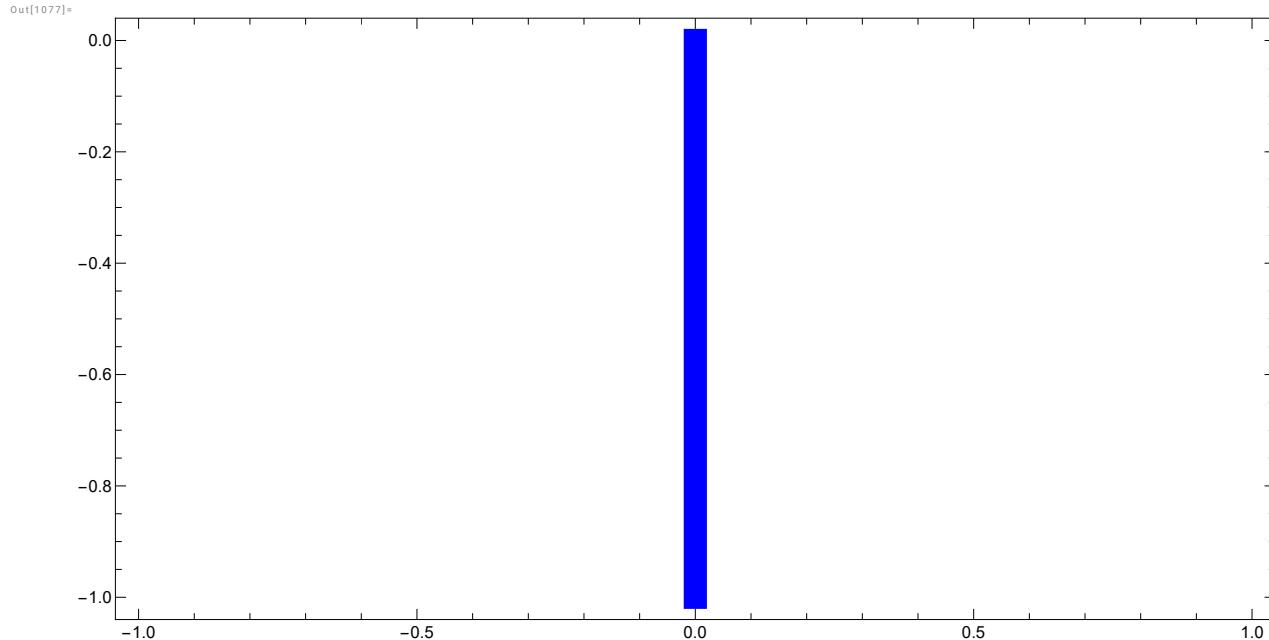
```
In[1076]:= Graphics[{  
    Lighter[Blue, 0.7],  
    EdgeForm[{Thick, Blue}],  
    Disk[{0, -1}, 1, {-180°, 0°}]},  
    Frame → True, ImageSize → 1000]
```



# Опора

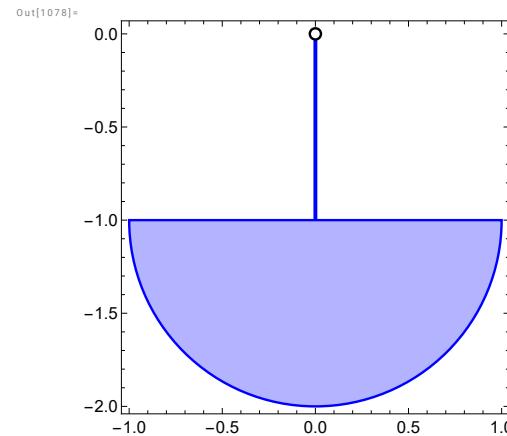
Функция `Line[{{x1,y1},{x2,y2},{x3,y3}}]`

```
In[1077]:= Graphics[  
{  
  Blue,  
  Thickness[0.02],  
  Line[{{0, 0}, {0, -1}}] (* Line[{{x1,y1},{x2,y2},{x3,y3},...}] *)  
,  
 Frame → True, ImageSize → 1000]
```



## Стойка, стержень и диск

```
In[1078]:= Graphics[{
  (* Стержень *)
  Blue,
  Thickness[0.01],
  Line[{{0, 0}, {0, -1}}],
  (* Стойка *)
  White,
  EdgeForm[{Thick, Black}],
  Disk[{0, 0}, 0.03],
  (* Диск *)
  Lighter[Blue, 0.7],
  EdgeForm[{Thick, Blue}],
  Disk[{0, -1}, 1, {-180°, 0°}]
},
Frame → True]
```



## Стойка, стержень и диск

Объединяем в список геометрических объектов, с которым в дальнейшем будем работать, как с единым объектом

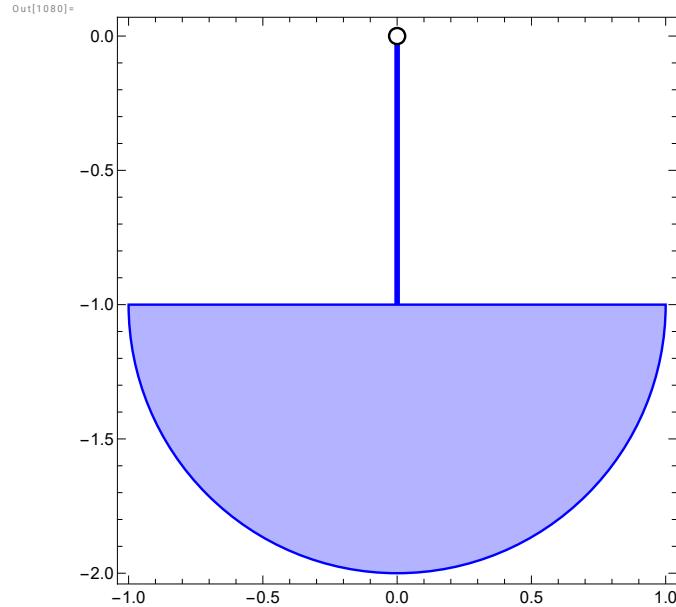
```
In[1079]:= body1 = {  
    Blue, Thickness[0.01], Line[{{0, 0}, {0, -1}}],  
  
    Lighter[Blue, 0.7], EdgeForm[{Thick, Blue}],  
    Disk[{0, -1}, 1, {-180°, 0°}],  
  
    White, EdgeForm[{Thick, Black}],  
    Disk[{0, 0}, 0.03]  
}  
  
Out[1079]= {■, Thickness[0.01], Line[{{0, 0}, {0, -1}}]}, ■, EdgeForm[{Thickness[Large], ■}], Disk[{0, -1}, 1, {-180°, 0°}], □,  
EdgeForm[{Thickness[Large], ■}], Disk[{0, 0}, 0.03]}
```

Чтобы показать эти объекты необходимо использовать функцию **Graphics[]**

## Перемещение и поворот

Графический комплекс, состоящий из стойки и пластины

```
In[1080]:= Graphics[body1, Frame -> True, ImageSize -> 500]
```

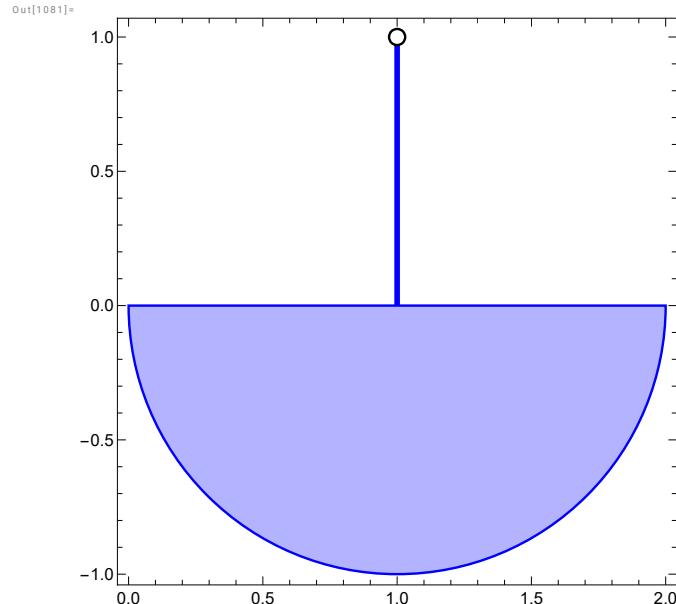


## Перемещение и поворот

Перенесём объект на 1 единицу вдоль осей  $x$  и  $y$ . Используем функцию GeometricTransformation.

Первый аргумент функции - геометрический объект, второй - список, состоящий из двух элементов: матрица поворота и вектор смещения. Таким образом, функция GeometricTransformation определяет поворот и смещение объекта.

```
In[1081]:= Graphics[
  GeometricTransformation[body1, {RotationMatrix[0], {1, 1}}],
  Frame → True, ImageSize → 500]
```



```
GeometricTransformation[ОБЪЕКТ, {МатрицаПоворота, {Смещение}}]
```

## Матрица поворота

---

Матрица поворота в плоскости (вокруг оси z, перпендикулярной плоскости рисунка)

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

Точка с координатами  $\begin{pmatrix} x \\ y \end{pmatrix}$  поворачивается на угол  $\varphi$  вокруг начала координат

```
In[1082]:= RotationMatrix[45.0 °] // MatrixForm
Out[1082]//MatrixForm=

$$\begin{pmatrix} 0.707107 & -0.707107 \\ 0.707107 & 0.707107 \end{pmatrix}$$

```

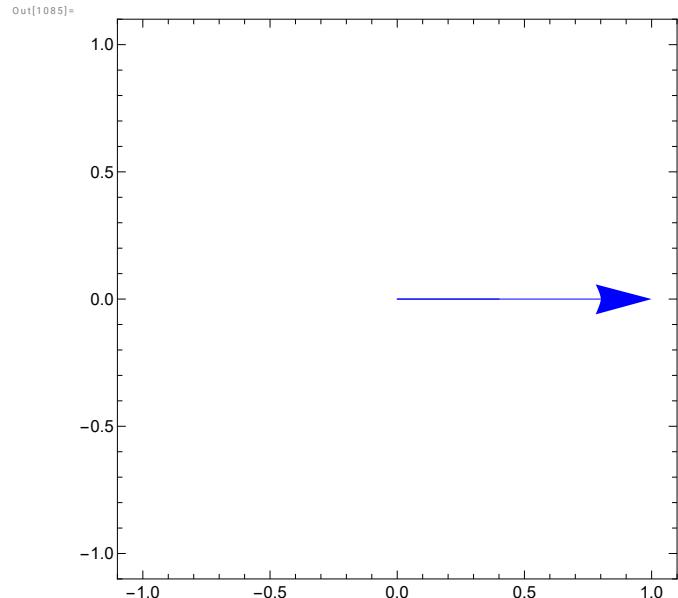
Поворачиваем вектор  $\{1, 0\}$  на 20 градусов против часовой стрелки

```
In[1083]:= RotationMatrix[20.0 °].{1, 0}
Out[1083]= {0.939693, 0.34202}
```

# Поворот вектора

Исходный вектор

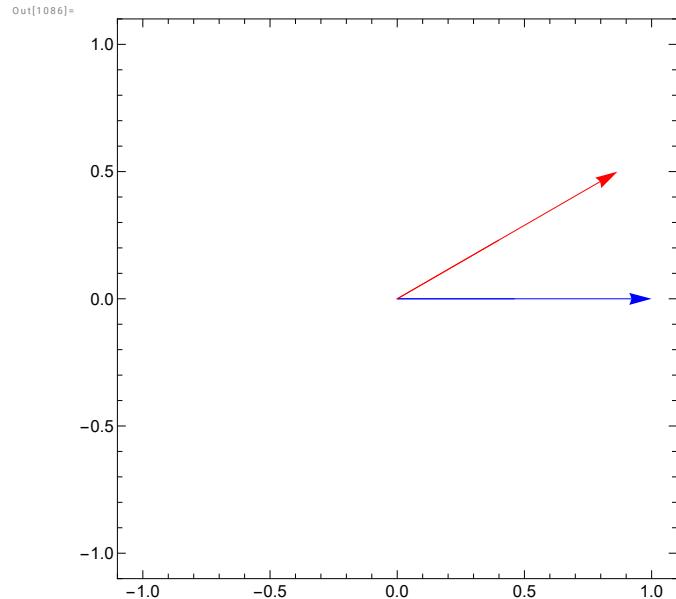
```
In[1084]:= r1 = {1.0, 0.0};  
Graphics[{  
  Blue,  
  Arrowheads[0.1], (* Размер стрелки *)  
  Arrow[{{0, 0}, r1}]  
,  
 PlotRange -> {{-1.1, 1.1}, {-1.1, 1.1}}, Frame -> True, ImageSize -> 500]
```



# Поворот вектора

Повернутый вектор

```
In[1086]:= Graphics[{  
  Blue, Arrow[{{0, 0}, r1}],  
  Red, Arrow[{{0, 0}, RotationMatrix[30 °]. r1}]  
},  
PlotRange -> {{-1.1, 1.1}, {-1.1, 1.1}}, Frame -> True, ImageSize -> 500]
```

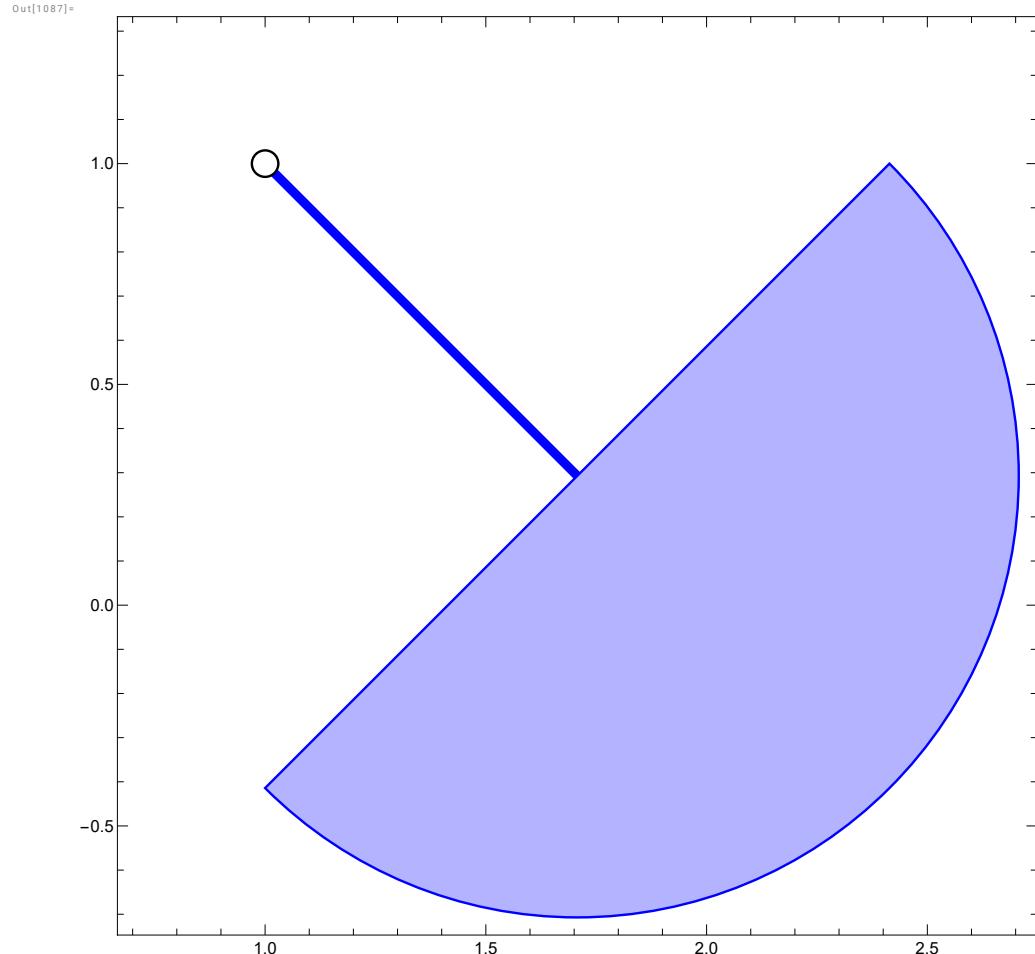


Это “активная” точка зрения на поворот - поворачивается объект.

# GeometricTransformation

Использование функции **GeometricTransformation** для поворота и перемещения

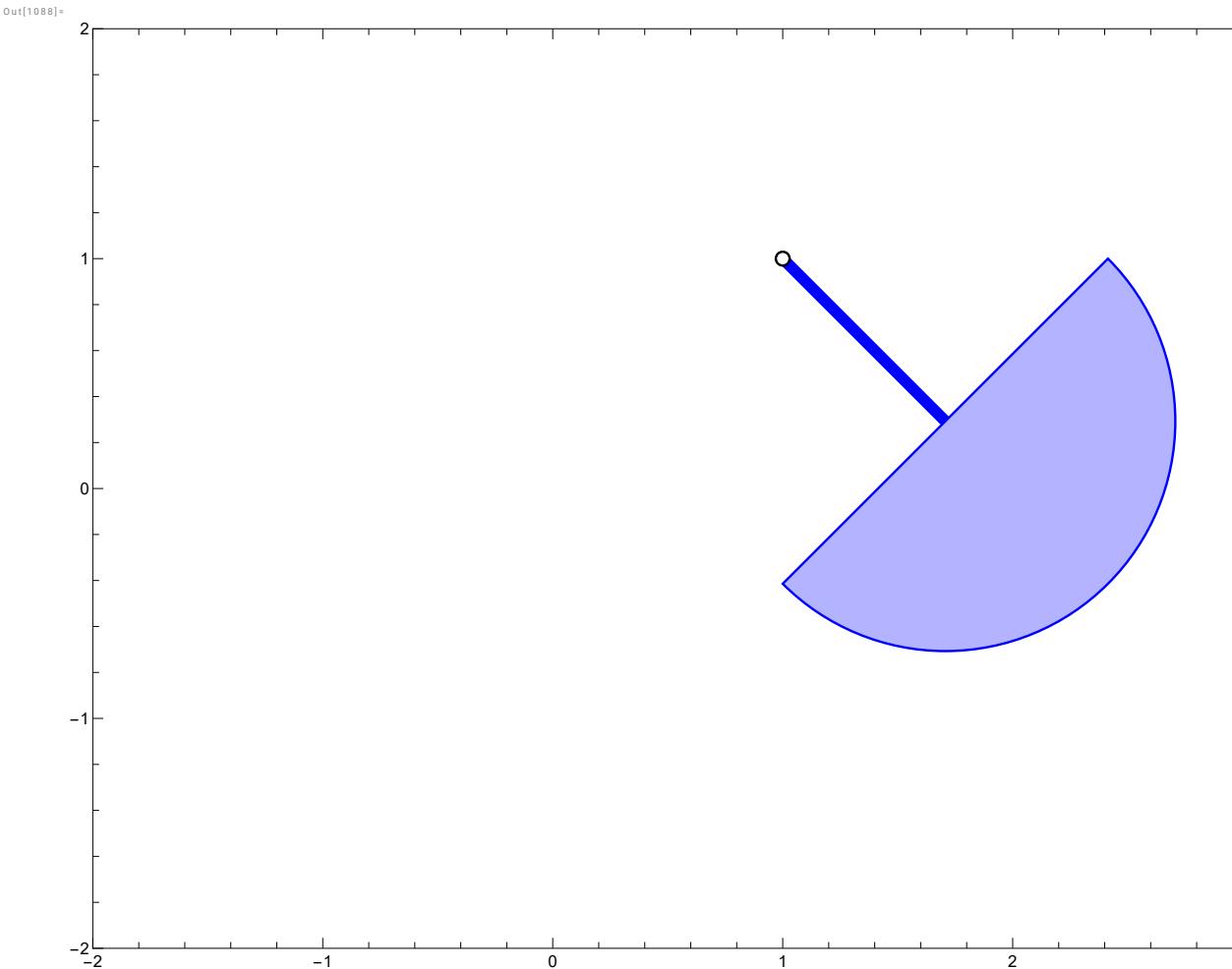
```
In[1087]:= Graphics[
  GeometricTransformation[body1, {RotationMatrix[45 °], {1, 1}}],
  Frame → True, ImageSize → 800]
```



## Держимся в рамках

Опция **PlotRange** позволяет определить границы области рисунка, чтобы при перемещении объекта Mathematica не выполняла автоматический выбор границ по осям координат (для построения анимации это не нужно).

```
In[1088]:= Graphics[
  GeometricTransformation[body1, {RotationMatrix[45 °], {1, 1}}],
  Frame → True, PlotRange → {{-2, 3}, {-2, 2}}, ImageSize → 1000]
```



## Объект как функция положения

Создадим функцию, для формирования геометрического объекта в заданном положении. Удалим все определения, связанные с именем **body1**:

```
In[1089]:= ClearAll[body1];
```

Определяем функцию от угла поворота с именем body

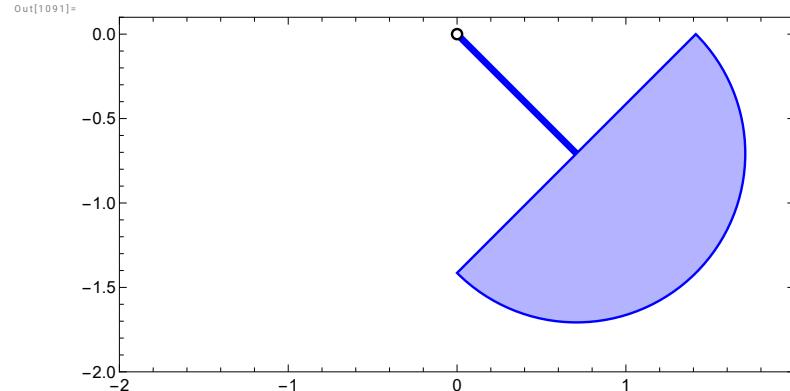
```
In[1090]:= body1[φ_] := Module[{b},
  b = {
    (* Стержень *)
    Blue, Thickness[0.01], Line[{{0, 0}, {0, -1}}],
    (* Сектор, пластина *)
    Lighter[Blue, 0.7], EdgeForm[Directive[Thick, Blue]],
    Disk[{0, -1}, 1, {-180°, 0°}],
    (* Стойка *)
    White, EdgeForm[Directive[Thick, Black]],
    Disk[{0, 0}, 0.03]
  };
  (* Результат работы функции – повернутый и смещенный объект *)
  GeometricTransformation[b, {RotationMatrix[φ], {0, 0}}]
];
```

Функция **Module** позволяет создавать многострочные функции с локальными переменными, как в традиционных языках программирования.

# Поворот

Рисуем, вызывая новую функцию

```
In[1091]:= Graphics[body1[45 °], Frame → True, PlotRange → {{-2, 2}, {-2, 0.1}}, ImageSize → 600]
```

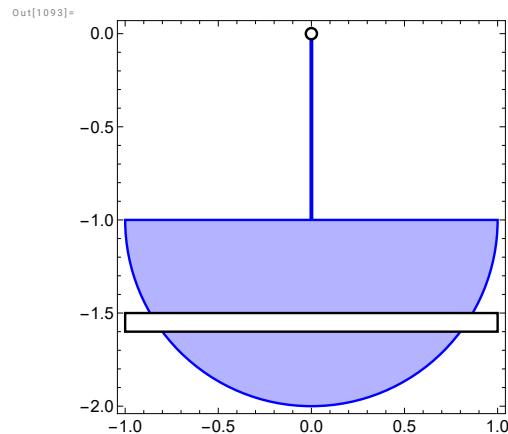


## Добавляем канал для шарика

Нарисуем поверх пластины прямоугольник, который будет изображать канал для шарика.

```
In[1092]:= body1[φ_] := Module[{b},
  b = {
    (* Стержень *)
    Blue, Thickness[0.01], Line[{{0, 0}, {0, -1}}],
    (* Сектор, пластина *)
    Lighter[Blue, 0.7], EdgeForm[Directive[Thick, Blue]],
    Disk[{0, -1}, 1, {-180°, 0°}],
    (* Стойка *)
    White, EdgeForm[Directive[Thick, Black]],
    Disk[{0, 0}, 0.03],
    (* Канал *)
    Rectangle[{-1, -1.6}, {1, -1.5}]
  };
  GeometricTransformation[b, {RotationMatrix[φ], {0, 0}}]
];

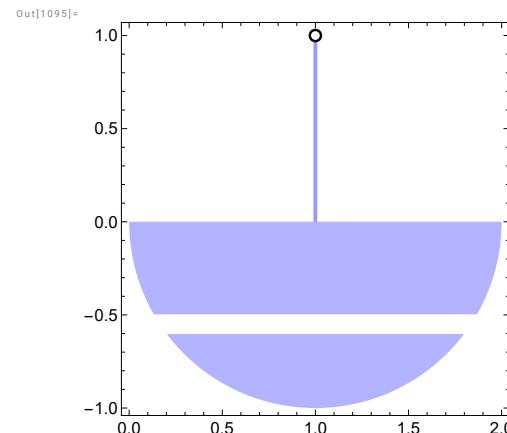
Graphics[body1[0°], Frame → True]
```



## Канал для шарика

Маскируем (убираем границу у всех объектов), чтобы казалось, что пластина с каналом это один объект.

```
In[1094]:= body1[φ_] := Module[{b},
  b = {
    Lighter[Blue, 0.6], Thickness[0.01], Line[{{0, 0}, {0, -1}}],
    Lighter[Blue, 0.7],
    Disk[{0, -1}, 1, {-180 °, 0 °}],
    White, EdgeForm[Directive[Thick, Black]],
    Disk[{0, 0}, 0.03],
    EdgeForm[Directive[Thin, White]],
    Rectangle[{-1, -1.6}, {1, -1.5}]
  };
  GeometricTransformation[b, {RotationMatrix[φ], {1, 1}}]
];
Graphics[body1[0 °], Frame → True]
```



# Пружина

Функция **Table** генерирует списки

**Table[выражение, {i, начальное значение, конечное значение, шаг}]**

```
In[1096]:= Table[i^2, {i, 0, 5}]
Out[1096]= {0, 1, 4, 9, 16, 25}
```

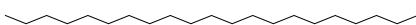
Пружина в виде ломаной линии

```
In[1097]:= Spring[L_, d_, n_] := Table[{i * L/n, (-1)^i d / 2}, {i, 0, n}];
Spring[0.5, 0.01, 5]
Out[1098]= {{0., 0.005}, {0.1, -0.005}, {0.2, 0.005}, {0.3, -0.005}, {0.4, 0.005}, {0.5, -0.005}}
```

Чтобы нарисовать пружину, используем

функцию **Line[{p1, p2, p3,..., pn}]**

```
In[1099]:= Graphics[
  Line[Spring[0.5, 0.01, 20]]
]
Out[1099]=
```



Больше “витков”, больше диаметр

```
In[1100]:= Graphics[
  Line[Spring[0.5, 0.03, 50]]
]
Out[1100]=
```

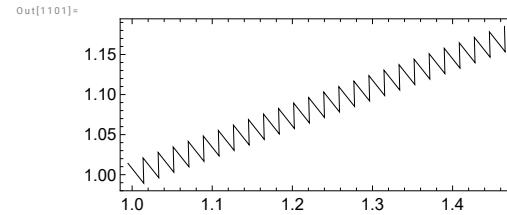


## Перемещение и поворот пружины

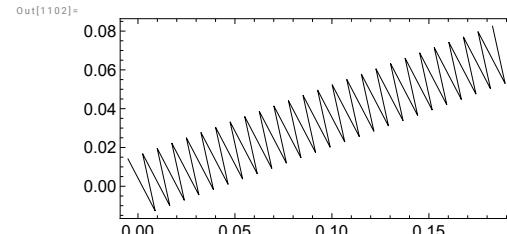
Линию, нарисованную по точкам, тоже можно поворачивать и перемещать, используя функцию **GeometricTransformation**

Поворачиваем на 20 градусов и перемещаем на 1 и 1 по  $x$  и  $y$  соответственно

```
In[1101]:= Graphics[
  GeometricTransformation[Line[Spring[0.5, 0.03, 50]], {RotationMatrix[20 °], {1, 1}}],
  Frame → True
]
```

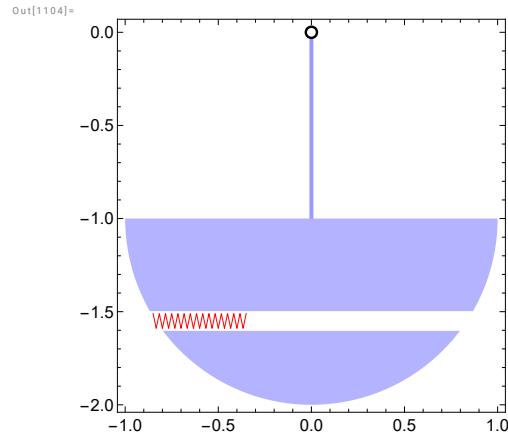


```
In[1102]:= Graphics[
  GeometricTransformation[Line[Spring[0.2, 0.03, 50]], {RotationMatrix[20 °], {0, 0}}],
  Frame → True
]
```



# Стержень, пластина и пружина

```
In[1103]=
body1[φ_] := Module[{b},
  b =
  {
    Lighter[Blue, 0.6], Thickness[0.01], Line[{{0, 0}, {0, -1}}],
    Lighter[Blue, 0.7], Disk[{0, -1}, 1, {-180 °, 0 °}],
    White, EdgeForm[{Thick, Black}],
    Disk[{0, 0}, 0.03],
    EdgeForm[{Thin, White}],
    Rectangle[{-1, -1.6}, {1, -1.5}],
    Red, Thin, Translate[Line[Spring[0.5, 0.08, 30]], {-0.85, -1.55}]}
  ;
  GeometricTransformation[b, {RotationMatrix[φ], {0, 0}}]
];
Graphics[body1[0 °], Frame → True]
```

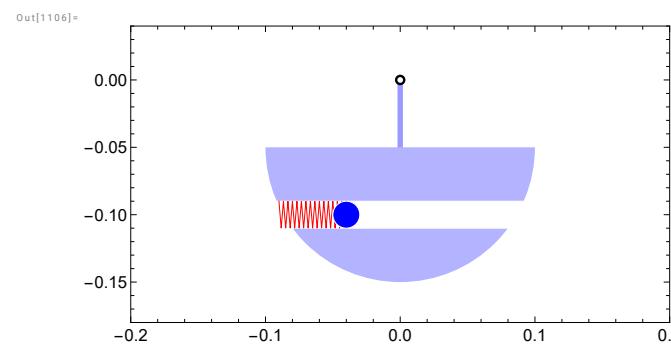


# Шарик

Добавляем в систему шарик (функция **Disk**)

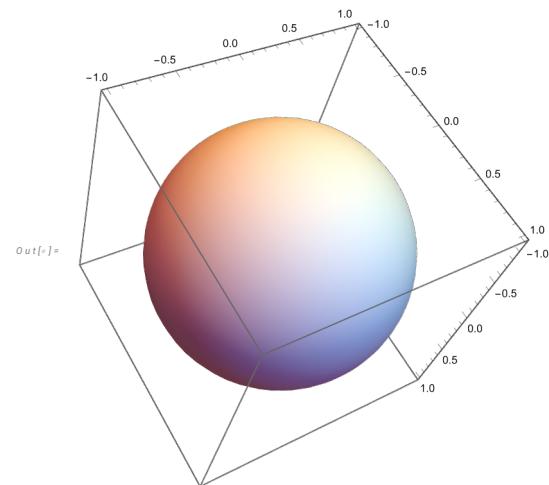
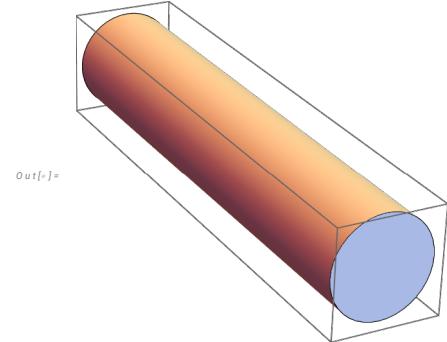
```
In[1105]:= body1[φ_, x_] := Module[{b},
  b =
  {
    (* СТЕРЖЕНЬ *)
    Lighter[Blue, 0.6], Thickness[0.01], Line[{{0, 0}, {0, -0.05}}],
    (* ПЛАСТИНА *)
    Lighter[Blue, 0.7], Disk[{0, -0.05}, 0.1, {-180°, 0°}],
    (* ОПОРА *)
    White, EdgeForm[{Thick, Black}], Disk[{0, 0}, 0.003],
    (* КАНАЛ *)
    EdgeForm[{Thin, White}], Rectangle[{-1, -0.09}, {1, -0.11}],
    (* ПРУЖИНА *)
    Red, Thin, Translate[Line[Spring[x, 0.02, 30]], {-0.09, -0.1}],
    (* ШАРИК *)
    Blue, Disk[{-0.09 + x, -0.1}, 0.01]
  };
  GeometricTransformation[b, {RotationMatrix[φ], {0, 0}}]
];

Graphics[body1[0, 0.05], Frame → True, PlotRange → {{-0.2, 0.2}, {-0.18, 0.04}}, ImageSize → 500]
```



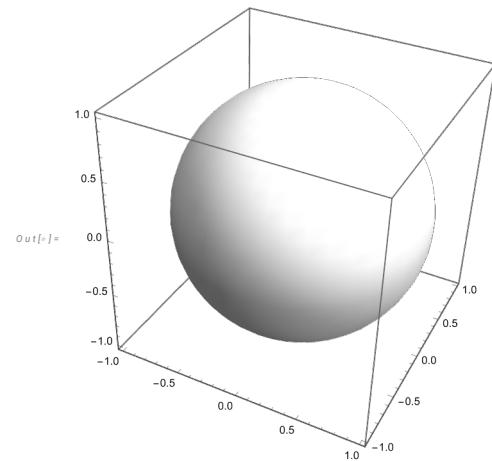
## Переходим в 3D

```
In[1]:= Graphics3D[{Cylinder[{{0, 0, 0}, {0, 0, 1}}, 0.1]}]
Graphics3D[{Sphere[{0, 0, 0}, 1]}, Axes -> True]
```



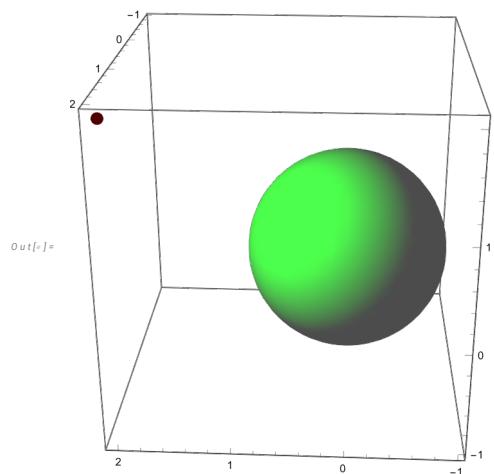
## Освещение

```
Graphics3D[{Sphere[{0, 0, 0}, 1]}, Axes -> True, Lighting -> "Neutral"]
```



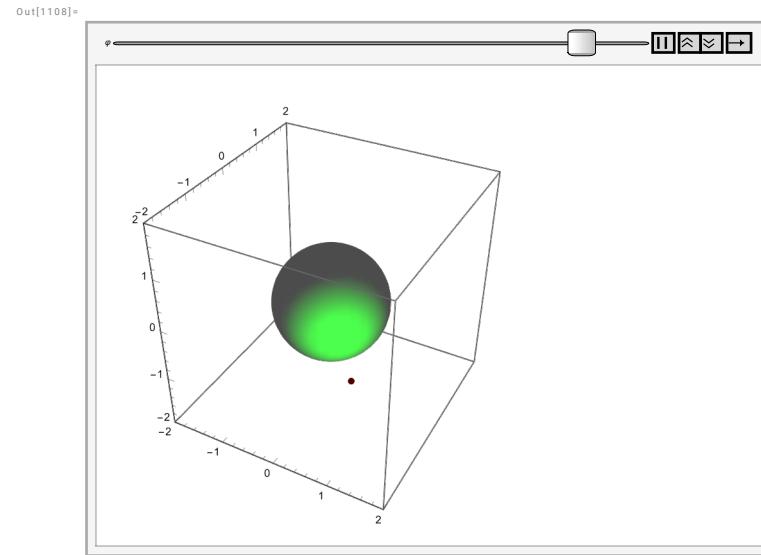
# Освещение

```
In[~]:= lp = 2 {1, 1, 1};  
lightColor = Red;  
Graphics3D[{  
  {White, Sphere[{0, 0, 0}, 1]},  
  {Specularity[0], lightColor, Sphere[lp, 0.05]}  
}, Axes → True, Lighting → {"Point", Green, lp}, {"Ambient", GrayLevel[0.3]}]]
```



# Освещение

```
In[1107]:= lightColor = Red;
Animate[
Graphics3D[{White, Sphere[{0, 0, 0}, 1]},
{lightColor, Sphere[2 {Cos[\varphi], Sin[\varphi], 0}, 0.05]}
], Axes -> True, Lighting -> {"Point", Green, 2 {Cos[\varphi], Sin[\varphi], 0}}, {"Ambient", GrayLevel[0.3]}, PlotRange -> {{-2, 2}, {-2, 2}, {-2, 2}}]
, {\varphi, 0, 2 \pi}]
```



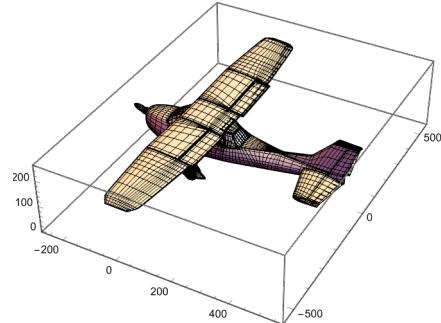
# Импорт 3D объекта

Импорт 3D объекта из файла (.obj)

In[1109]:=

```
SetDirectory[NotebookDirectory[]];  
object = Import["11804_Airplane_v2_12.obj", {"GraphicsComplex"}];  
  
In[1111]:=  
Graphics3D[{object}, Axes -> True]
```

Out[1111]=



## Импорт 3D объекта

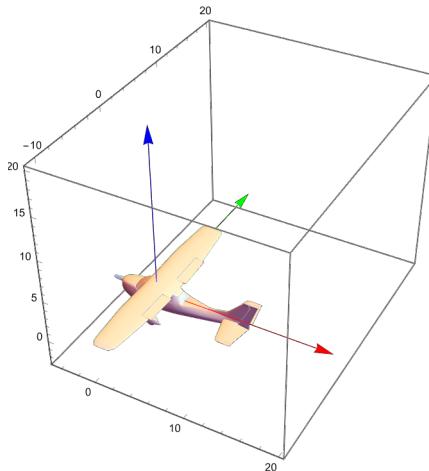
Связанная с объектом система координат (начало координат определяется в файле с моделью)

```
In[1112]:= SK = {Red, Arrow[{{0, 0, 0}, {20, 0, 0}}], Green, Arrow[{{0, 0, 0}, {0, 20, 0}}], Blue, Arrow[{{0, 0, 0}, {0, 0, 20}}]};
```

Масштабирование и перемещение (функции Translate и Scale)

```
In[1113]:= Graphics3D[{SK, EdgeForm[None], Translate[Scale[object, 0.02], {-145, 0, -128}]}, Axes → True]
```

Out[1113]=



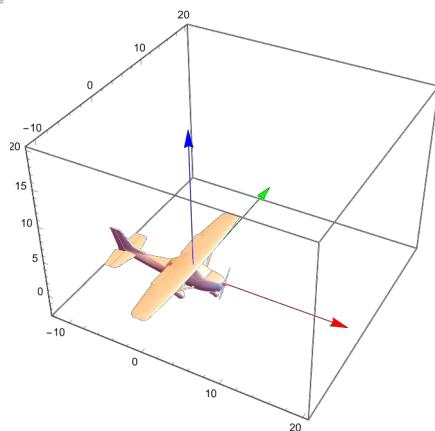
# Импорт 3D объекта

## Поворот

```
In[1114]:= Graphics3D[{EdgeForm[None], SK,
```

```
  Rotate[
    Translate[Scale[object, 0.02], {-145, 0, -128}],
    180°, {0, 0, 1}
  ]
}, Axes → True]
```

```
Out[1114]=
```



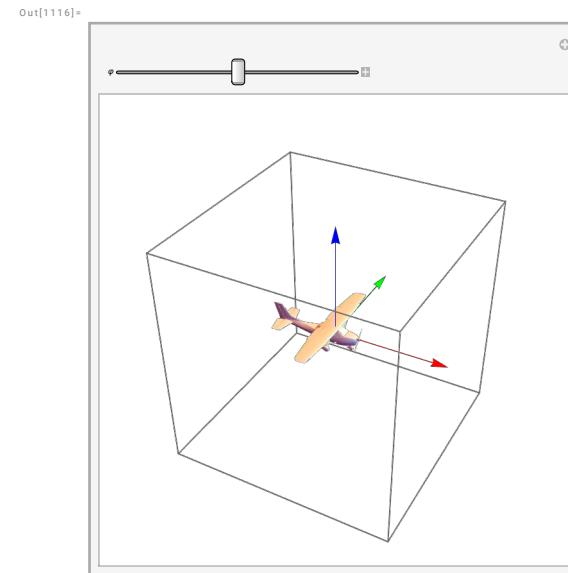
```
In[1115]:=
```

```
plane = {EdgeForm[None], SK, Rotate[Translate[Scale[object, 0.02], {-145, 0, -128}], 180°, {0, 0, 1}]};
```

# Поворот

Поворот вокруг оси z на угол  $\varphi$  (рыскание) - `RotationMatrix[ $\varphi$ , {0,0,1}]`

```
In[1116]:= Manipulate[
 Graphics3D[GeometricTransformation[plane, {RotationMatrix[ $\varphi$ , {0, 0, 1}], {0, 0, 0}}], PlotRange -> {{-20, 20}, {-20, 20}, {-20, 20}}],
 { $\varphi$ , -90°, 90°}
]
```

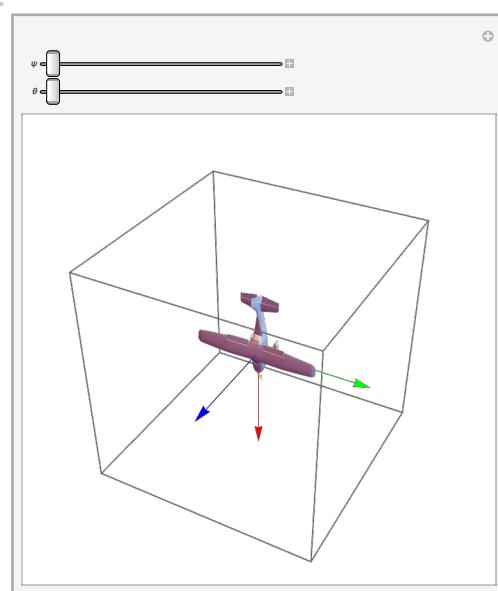


# Два поворота

Перемножив матрицы получим сложный поворот по рысканию и крену

```
In[1117]:= A[ψ_, θ_] := RotationMatrix[ψ, {0, 0, 1}].RotationMatrix[-θ, {0, 1, 0}];

In[1118]:= Manipulate[
  Graphics3D[GeometricTransformation[plane, {A[ψ, θ], {0, 0, 0}}]], PlotRange → {{-20, 20}, {-20, 20}, {-20, 20}}],
  {ψ, -90 °, 90 °},
  {θ, -90 °, 90 °}
]
```



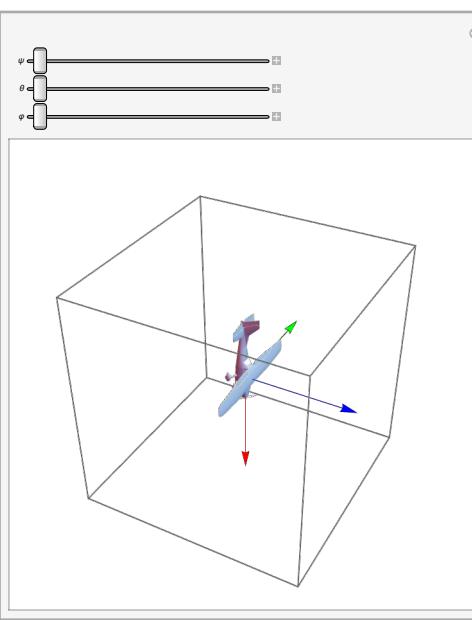
# Три поворота

Рыскание, тангаж и крен

```
In[1119]:= A[\psi_, \theta_, \varphi_] := RotationMatrix[\psi, {0, 0, 1}].RotationMatrix[-\theta, {0, 1, 0}].RotationMatrix[\varphi, {1, 0, 0}];
```

```
In[1120]:= Manipulate[
 Graphics3D[GeometricTransformation[plane, {A[\psi, \theta, \varphi], {0, 0, 0}}]], PlotRange \rightarrow {{-20, 20}, {-20, 20}, {-20, 20}}],
 {\psi, -90 \text{ } \text{ } \text{ } ^\circ, 90 \text{ } \text{ } \text{ } ^\circ},
 {\theta, -90 \text{ } \text{ } \text{ } ^\circ, 90 \text{ } \text{ } \text{ } ^\circ},
 {\varphi, -90 \text{ } \text{ } \text{ } ^\circ, 90 \text{ } \text{ } \text{ } ^\circ}
 ]
```

Out[1120]=



In[1121]:=