

Анимация в Mathematica

на примере курсовой работы по теоретической механике

Юдинцев В. В.
Кафедра теоретической механики
Самарский университет

Стили по умолчанию...

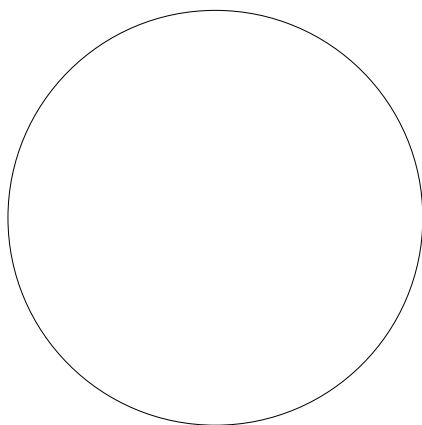
```
In[522]:= SetOptions[Graphics, BaseStyle → {14, FontFamily → "Helvetica"}];
```

Геометрические объекты в Mathematica

Окружность

In[523]:= **Graphics**[**Circle**[{0, 0}, 2]]

Out[523]=

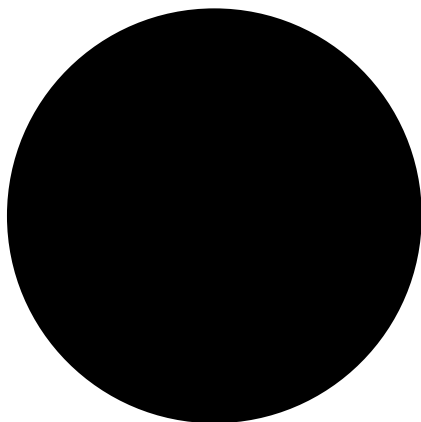


Геометрические объекты в Mathematica

Круг или диск

In[524]:= **Graphics**[**Disk**[{0, 0}, 2]]

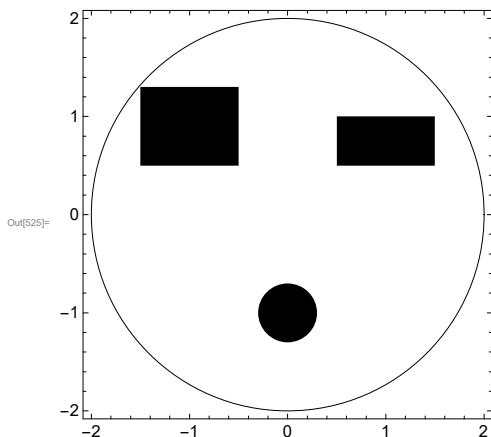
Out[524]=



Несколько объектов

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

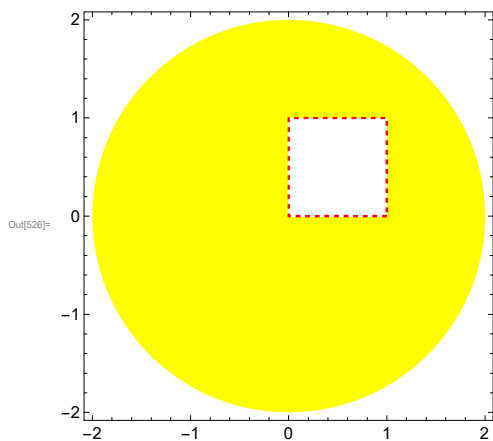
```
In[525]:= Graphics[{
  Circle[{0, 0}, 2],
  Rectangle[{0.5, 0.5}, {1.5, 1}],
  Rectangle[{-1.5, 0.5}, {-0.5, 1.3}],
  Disk[{0, -1}, 0.3]
},
Frame → True]
```



Контур и заливка

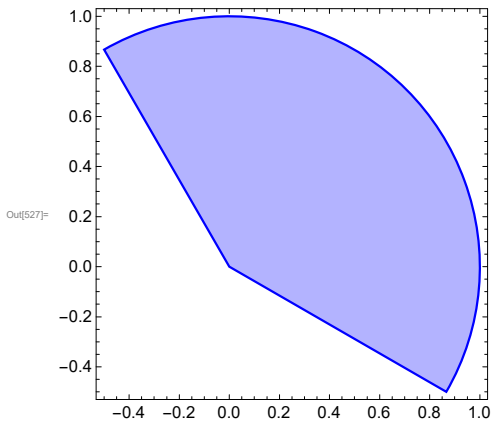
Заливка

```
In[526]:= Graphics[{  
  Yellow, Disk[{0, 0}, 2],  
  White, EdgeForm[{Thick, Red, Dashed}], Rectangle[{0, 0}, {1, 1}]  
},  
Frame → True]
```



Сектор

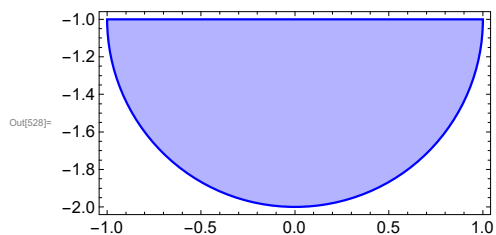
```
In[527]:= Graphics[{
  Lighter[Blue, 0.7],
  EdgeForm[{Thick, Blue}],
  Disk[{0, 0}, 1, {-30°, 120°}],
  Frame → True]
```



Угол отсчитывается от горизонтальной оси против часовой стрелки

Тело 1 механизма

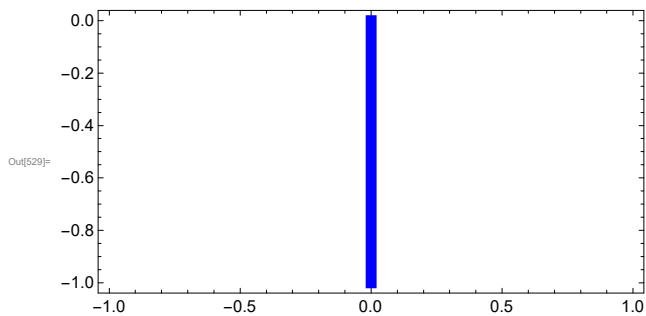
```
In[528]:= Graphics[{  
  Lighter[Blue, 0.7],  
  EdgeForm[{Thick, Blue}],  
  Disk[{0, -1}, 1, {-180°, 0°}],  
  Frame → True]
```



Опора

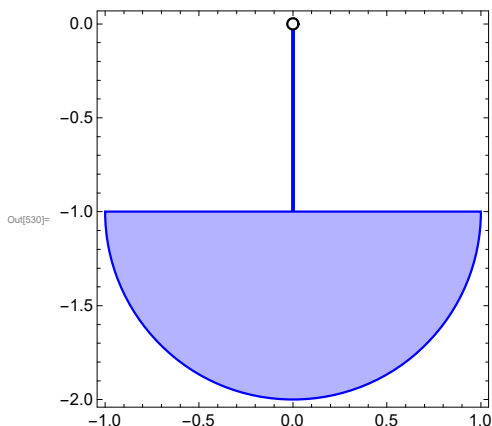
Функция `Line[{{x1,y1},{x2,y2},{x3,y3}}]`

```
In[529]:= Graphics[{  
  Blue, Thickness[0.02],  
  Line[{{0, 0}, {0, -1}}]  
}, Frame → True]
```



Стойка, стержень и диск

```
In[530]:= 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[531]:= 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]
}

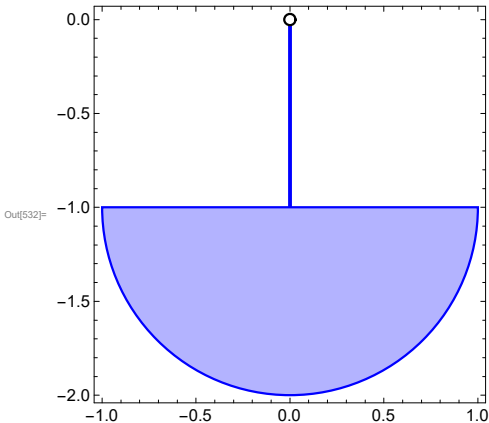
Out[531]= {, Thickness[0.01], Line[{{0, 0}, {0, -1}}], ,
    EdgeForm[{Thickness[Large], }], Disk[{0, -1}, 1, {-180°, 0°}],
    , EdgeForm[{Thickness[Large], }], Disk[{0, 0}, 0.03]}
```

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

Перемещение и поворот

Графический комплекс, состоящий из стойки и пластины

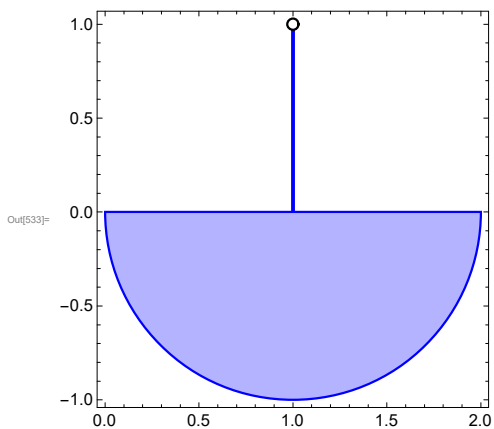
```
In[532]:= Graphics[body1, Frame -> True]
```



Перемещение и поворот

Перенесём объект на 1 и 1 вдоль осей x и y

```
In[533]:= Graphics[
  GeometricTransformation[body1, {RotationMatrix[0], {1, 1}}],
  Frame → True]
```



Матрица поворота

Матрица поворота в плоскости

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

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

```
In[534]:= RotationMatrix[45.0 °] // MatrixForm
```

```
Out[534]/MatrixForm=  $\begin{pmatrix} 0.707107 & -0.707107 \\ 0.707107 & 0.707107 \end{pmatrix}$ 
```

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

```
In[535]:= RotationMatrix[20.0 °] . {1, 0}
```

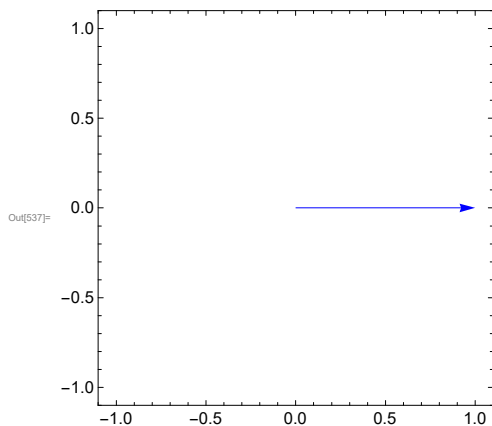
```
Out[535]= {0.939693, 0.34202}
```

Поворот вектора

Исходный вектор

```
In[536]:= r1 = {1.0, 0.0};
```

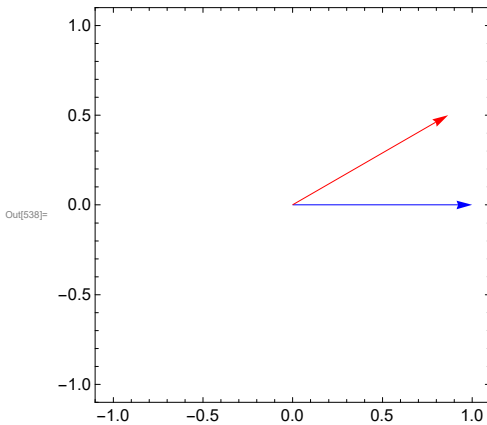
```
Graphics[{  
  Blue, Arrow[{{0, 0}, r1}]  
},  
PlotRange -> {{-1.1, 1.1}, {-1.1, 1.1}}, Frame -> True]
```



Поворот вектора

Повернутый вектор

```
In[538]:= Graphics[{
  Blue, Arrow[{{0, 0}, r1}],
  Red, Arrow[{{0, 0}, RotationMatrix[30 °]. r1]}],
PlotRange -> {{-1.1, 1.1}, {-1.1, 1.1}}, Frame -> True]
```

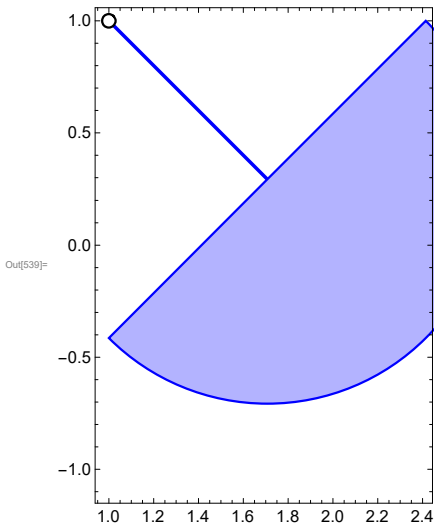


Это “активная” точка зрения на поворот - поворачивается объект.

GeometricTransformation

Использование функции **GeometricTransformation** для поворота и перемещения

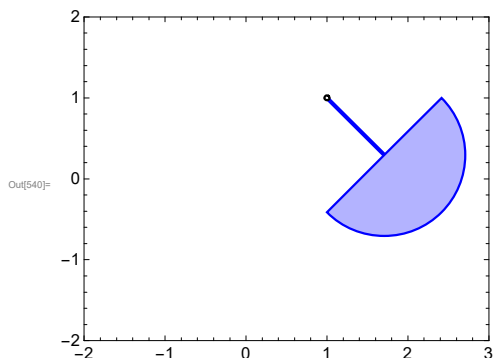
```
In[539]:= Graphics[
  GeometricTransformation[body1, {RotationMatrix[45 °], {1, 1}}],
  Frame → True]
```



Держимся в рамках

Опция **PlotRange** позволяет определить границы область рисунка

```
In[540]:= Graphics[
  GeometricTransformation[body1, {RotationMatrix[45 °], {1, 1}}],
  Frame → True, PlotRange → {{-2, 3}, {-2, 2}}
```



Объект как функция положения

Создадим функцию, для формирования геометрического объекта в заданном положении.

Удалим все определения, связанные с именем **body1**:

```
In[541]:= ClearAll[body1];
```

Определяем функцию от угла поворота с именем **body**

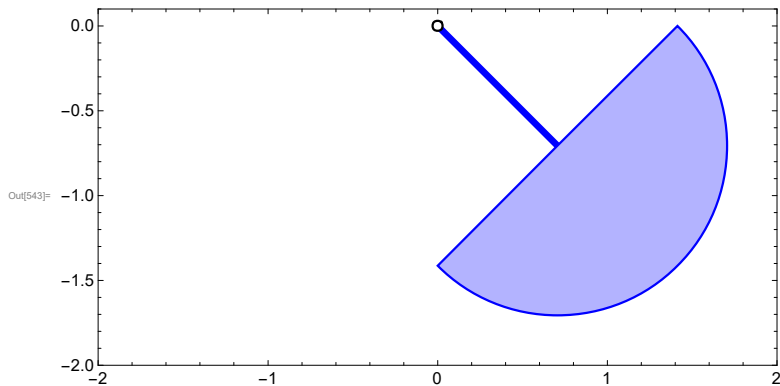
```
In[542]:= body1[φ_] := Module[{b},
  b = {
    (* Стержень *)
    Blue, Thickness[0.01], Line[{{0, 0}, {0, -1}}],
    (* Сектор, пластина *)
    Lighter[Blue, 0.7], EdgeForm[Directive[Thick, Blue]],
    Disk[{0, -1}, 1, {-180°, 0°}],
    (* Стойка *)
    White, EdgeForm[Directive[Thick, Black]],
    Disk[{0, 0}, 0.03]
  };
  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[543]:= Graphics[body1[45 °], Frame → True,  
PlotRange → {{-2, 2}, {-2, 0.1}}, ImageSize → 600]
```



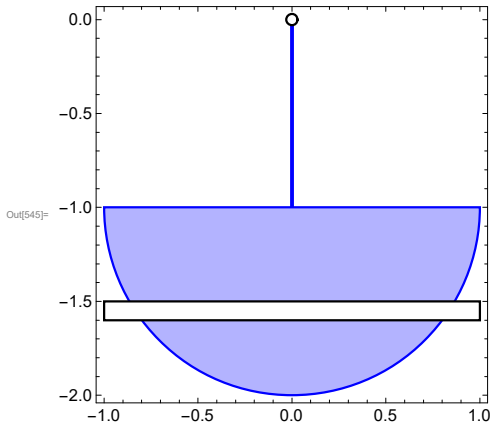
Добавляем канал для шарика

```

In[544]:= 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

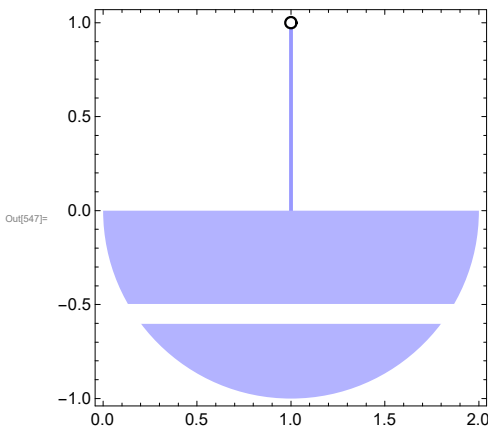
Маскируем (убираем границу у всех объектов), чтобы казалось, что пластина с каналом это единый объект.

```
In[546]:= 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[548]:= Table[i^2, {i, 0, 5}]
```

```
Out[548]:= {0, 1, 4, 9, 16, 25}
```

```
In[549]:= Table[{i, i^2}, {i, 0, 5}]
```

```
Out[549]:= {{0, 0}, {1, 1}, {2, 4}, {3, 9}, {4, 16}, {5, 25}}
```

Пружина в виде ломаной линии

```
In[550]:= Spring[L_, d_, n_] := Table[{i * L/n, (-1)^i d/2}, {i, 0, n}];  
Spring[0.5, 0.01, 5]
```

```
Out[551]:= {{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[552]:= Graphics[  
  Line[Spring[0.5, 0.01, 20]]  
]
```

```
Out[552]:= 
```

Больше “витков”, больше диаметр

```
In[553]:= Graphics[  
  Line[Spring[0.5, 0.03, 50]]  
]
```

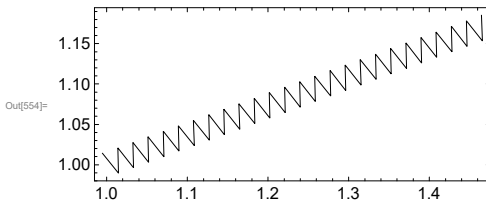
```
Out[553]:= 
```

Перемещение и поворот пружины

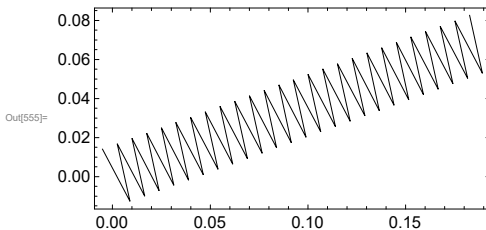
Линию, нарисованную по точкам, тоже можно поворачивать и перемещать, используя функцию **GeometricTransformation**

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

```
In[554]:= Graphics[
  GeometricTransformation[Line[Spring[0.5, 0.03, 50]],
    {RotationMatrix[20°], {1, 1}}],
  Frame -> True
]
```



```
In[555]:= Graphics[
  GeometricTransformation[Line[Spring[0.2, 0.03, 50]],
    {RotationMatrix[20°], {0, 0}}],
  Frame -> True
]
```



Стержень, пластина и пружина

```

In[556]:= body1[φ_] := Module[{b},
  b =
  {
    Lighter[Blue, 0.6],
    Thickness[0.01], Line[{{0, 0}, {0, -1}}],

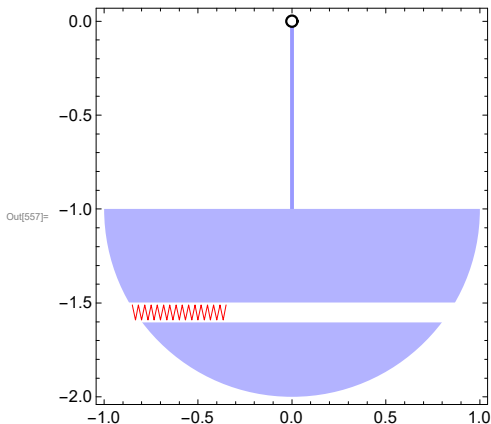
    Lighter[Blue, 0.7], Disk[{0, -1}, 1, {-180°, 0°}],

    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**)

```

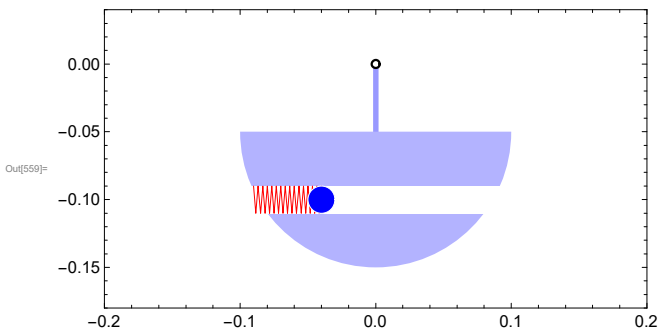
In[558]:= body1[φ_, x_] := Module[{b},
  b =
  {
    (* СТЕРЖЕНЬ *)
    Lighter[Blue, 0.6],
    Thickness[0.01], Line[{{0, 0}, {0, -0.05}}],
    (* ПЛАСТИНА *)
    Lighter[Blue, 0.7], Disk[{0, -0.05}, 0.1, {-180°, 0°}],
    (* ОПОРА *)
    White, EdgeForm[{Thick, Black}], Disk[{0, 0}, 0.003],
    (* 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]

```



Положение шарика относительно неподвижной системы координат $Oxyz$

In[563]:= $\mathbf{r}_M = \text{RotationMatrix}[\varphi[t]] \cdot \rho_M$

$$\text{Out[563]} = \left\{ 2h \sin[\varphi[t]] + \cos[\varphi[t]] \left(-\sqrt{-h^2 + R^2} + l_0 + x[t] \right), \right. \\ \left. -2h \cos[\varphi[t]] + \sin[\varphi[t]] \left(-\sqrt{-h^2 + R^2} + l_0 + x[t] \right) \right\}$$

Вектор абсолютной скорости шарика

In[564]:= $\mathbf{v}_M = \mathbf{D}[\mathbf{r}_M, t] // \text{FullSimplify}$

$$\text{Out[564]} = \left\{ \cos[\varphi[t]] x'[t] + \right. \\ \left(2h \cos[\varphi[t]] + \sqrt{-h^2 + R^2} \sin[\varphi[t]] - \sin[\varphi[t]] (l_0 + x[t]) \right) \varphi'[t], \\ \sin[\varphi[t]] x'[t] + 2h \sin[\varphi[t]] \varphi'[t] + \\ \cos[\varphi[t]] \left(-\sqrt{-h^2 + R^2} + l_0 + x[t] \right) \varphi'[t] \left. \right\}$$

Кинетическая энергия

Кинетическая энергия шарика

In[565]:= $T_2 = \text{FullSimplify}[m_2 \mathbf{v}_M \cdot \mathbf{v}_M / 2]$

$$\text{Out[565]} = \frac{1}{2} m_2 \\ \left(x'[t]^2 + 4h x'[t] \varphi'[t] + \right. \\ \left. \left(3h^2 + R^2 - \left(2\sqrt{-h^2 + R^2} - l_0 - x[t] \right) (l_0 + x[t]) \right) \varphi'[t]^2 \right)$$

Кинетическая энергия первого тела (стержень + сектор)

Стержень длиной h , массой m_{11} ,
вращающийся вокруг оси O . Момент инерции $m_{11} h^2 / 3$

In[566]= $T_{11} = (m_{11} h^2 / 3) * \varphi' [t]^2 / 2;$

Сектор, вращающийся вокруг оси O

In[567]= $J_{12} = \frac{1}{2} * \frac{m_{12} R^2}{2} + m_{12} h^2;$

$$T_{12} = \frac{J_{12} * \varphi' [t]^2}{2}$$

Out[568]= $\frac{1}{2} \left(h^2 m_{12} + \frac{R^2 m_{12}}{4} \right) \varphi' [t]^2$

Кинетическая энергия системы

In[569]= $T_{sys} = FullSimplify [T_{11} + T_{12} + T_2]$

Out[569]=
$$\frac{1}{24} \left(\left(4 h^2 m_{11} + 3 \left(4 h^2 + R^2 \right) m_{12} \right) \varphi' [t]^2 + 12 m_2 \left(x' [t]^2 + 4 h x' [t] \varphi' [t] + \left(3 h^2 + R^2 - \left(2 \sqrt{-h^2 + R^2} - l_0 - x[t] \right) \left(l_0 + x[t] \right) \right) \varphi' [t]^2 \right) \right)$$

Потенциальная энергия

Тело 2: шарик

Нулевой уровень - горизонтальная плоскость,
проходящая через ось вращения O .
Потенциальная энергия шарика (координата y в
неподвижной системе, умноженная на массу шарика и
ускорение свободного падения.

$$\text{In}[570]= \Pi_{g2} = m_2 * g * r_M[[2]]$$

$$\text{Out}[570]= g m_2 \left(-2 h \cos [\varphi [t]] + \sin [\varphi [t]] \left(-\sqrt{-h^2 + R^2} + l_0 + x[t] \right) \right)$$

Потенциальная энергия пружины

$$\text{In}[571]= \Pi_{c2} = \frac{c (x[t] - l_0)^2}{2};$$

Тело 1

Положение центра масс стержня и сектора относительно системы координат $O x_1 y_1 z_1$

$$\text{In}[572]= \rho_1 = \left(m_{11} \left\{ \theta, -\frac{h}{2} \right\} + m_{12} \left\{ \theta, -h - 4 \frac{R}{3\pi} \right\} \right) / (m_{11} + m_{12});$$

Положение центра масс стержня и сектора относительно системы координат $Oxyz$

$$\text{In}[573]= r_1 = \text{RotationMatrix}[\varphi[t]] \cdot \rho_1$$

$$\text{Out}[573]= \left\{ -\frac{\sin [\varphi [t]] \left(-\frac{h m_{11}}{2} + \left(-h - \frac{4 R}{3 \pi} \right) m_{12} \right)}{m_{11} + m_{12}}, \frac{\cos [\varphi [t]] \left(-\frac{h m_{11}}{2} + \left(-h - \frac{4 R}{3 \pi} \right) m_{12} \right)}{m_{11} + m_{12}} \right\}$$

Координата у этого вектора будет *высотой* центра масс системы пластина + стержень относительно нулевого уровня энергии

$$\text{In}[574]= \Pi_{g1} = (m_{11} + m_{12}) * g * r_1[[2]]$$

$$\text{Out}[574]= g \cos [\varphi [t]] \left(-\frac{h m_{11}}{2} + \left(-h - \frac{4 R}{3 \pi} \right) m_{12} \right)$$

Потенциальная энергия системы

$$\text{In}[575]= \Pi_{sys} = \Pi_{g2} + \Pi_{c2} + \Pi_{g1}$$

$$\text{Out}[575]= g \cos [\varphi [t]] \left(-\frac{h m_{11}}{2} + \left(-h - \frac{4 R}{3 \pi} \right) m_{12} \right) + \frac{1}{2} c (-l_0 + x[t])^2 + g m_2 \left(-2 h \cos [\varphi [t]] + \sin [\varphi [t]] \left(-\sqrt{-h^2 + R^2} + l_0 + x[t] \right) \right)$$

Уравнения движения

In[576]= eq1 =

$$\text{FullSimplify}\left[D\left[D\left[T_{\text{sys}}, \varphi'[t]\right], t\right] - D\left[T_{\text{sys}}, \varphi[t]\right] == -D\left[\Pi_{\text{sys}}, \varphi[t]\right]\right]$$

Out[576]= $m_{12} \left(-4 g (3 h \pi + 4 R) \sin[\varphi[t]] - 3 \pi (4 h^2 + R^2) \varphi''[t] \right) +$

$$12 \pi m_2 \left(g \sqrt{-h^2 + R^2} \cos[\varphi[t]] - 2 g h \sin[\varphi[t]] + \right.$$

$$2 \sqrt{-h^2 + R^2} x'[t] \varphi'[t] - 2 h x''[t] - (3 h^2 + R^2) \varphi''[t] -$$

$$(l_0 + x[t])$$

$$\left(g \cos[\varphi[t]] + 2 x'[t] \varphi'[t] + \left(-2 \sqrt{-h^2 + R^2} + l_0 + x[t] \right) \varphi''[t] \right) ==$$

$$2 h \pi m_{11} (3 g \sin[\varphi[t]] + 2 h \varphi''[t])$$

In[577]= eq2 =

$$\text{FullSimplify}\left[D\left[D\left[T_{\text{sys}}, x'[t]\right], t\right] - D\left[T_{\text{sys}}, x[t]\right] == -D\left[\Pi_{\text{sys}}, x[t]\right]\right]$$

Out[577]= $c x[t] +$

$$m_2 \left(g \sin[\varphi[t]] + \left(\sqrt{-h^2 + R^2} - l_0 - x[t] \right) \varphi'[t]^2 + x''[t] + 2 h \varphi''[t] \right) ==$$

$$c l_0$$

Параметры системы

In[578]= params =

$$\{m_{11} \rightarrow 1, m_{12} \rightarrow 1, m_2 \rightarrow 0.3, R \rightarrow 1, l_0 \rightarrow 0.4, g \rightarrow 9.807, c \rightarrow 9.0, h \rightarrow R/2\};$$

Интегрирование


```

In[579]= numEq =
    {eq1, eq2,  $\varphi[0] == 0.3$ ,  $\varphi'[0] == 0.3$ ,  $x[0] == 0$ ,  $x'[0] == 0$ } /. params
Out[579]= 
$$\left\{ -341.77 \sin[\varphi[t]] - 6\pi \varphi''[t] + \right.$$


$$11.3097 \left( 8.49311 \cos[\varphi[t]] - 9.807 \sin[\varphi[t]] + \sqrt{3} x'[t] \varphi'[t] - \right.$$


$$x''[t] - \frac{7 \varphi''[t]}{4} -$$


$$(0.4 + x[t]) (9.807 \cos[\varphi[t]] + 2 x'[t] \varphi'[t] +$$


$$(-1.33205 + x[t]) \varphi''[t]) \Big) = \pi (29.421 \sin[\varphi[t]] + \varphi''[t]),$$


$$9. x[t] +$$


$$0.3 (9.807 \sin[\varphi[t]] + (0.466025 - x[t]) \varphi'[t]^2 + x''[t] + \varphi''[t]) ==$$


$$3.6, \varphi[0] == 0.3, \varphi'[0] == 0.3, x[0] == 0, x'[0] == 0 \Big\}$$

In[580]= sol = NDSolve[numEq, { $\varphi[t]$ ,  $\varphi'[t]$ ,  $x[t]$ ,  $x'[t]$ }, {t, 0, 5}] // Flatten
Out[580]=  $\left\{ \varphi[t] \rightarrow \text{InterpolatingFunction} \left[ \begin{array}{c} \text{Domain: } \{0, 5\} \\ \text{Output: scalar} \end{array} \right] [t], \right.$ 

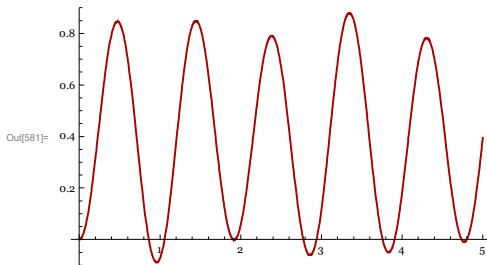
$$\varphi'[t] \rightarrow \text{InterpolatingFunction} \left[ \begin{array}{c} \text{Domain: } \{0, 5\} \\ \text{Output: scalar} \end{array} \right] [t],$$


$$x[t] \rightarrow \text{InterpolatingFunction} \left[ \begin{array}{c} \text{Domain: } \{0, 5\} \\ \text{Output: scalar} \end{array} \right] [t],$$


$$x'[t] \rightarrow \text{InterpolatingFunction} \left[ \begin{array}{c} \text{Domain: } \{0, 5\} \\ \text{Output: scalar} \end{array} \right] [t] \Big\}$$

In[581]= Plot[x[t] /. sol, {t, 0, 5}]

```

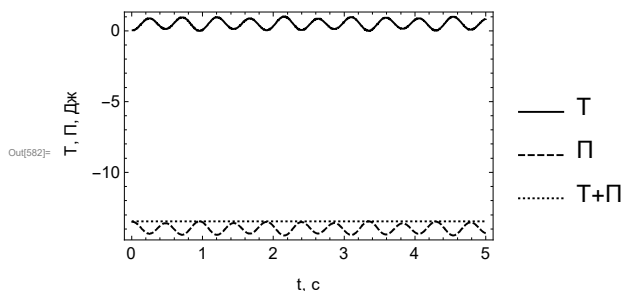


Проверка интеграла энергии

```

In[582]:= Plot[{Tsys, Πsys, Tsys + Πsys} /. params /. sol // Evaluate, {t, 0.0, 5.0},
  Frame → True, BaseStyle → 14, FrameLabel → {"t, c", " T, Π, Дж"},
  PlotLegends → {"T", "Π", "T+Π"}, PlotTheme → "Monochrome"]

```



Рисуем систему

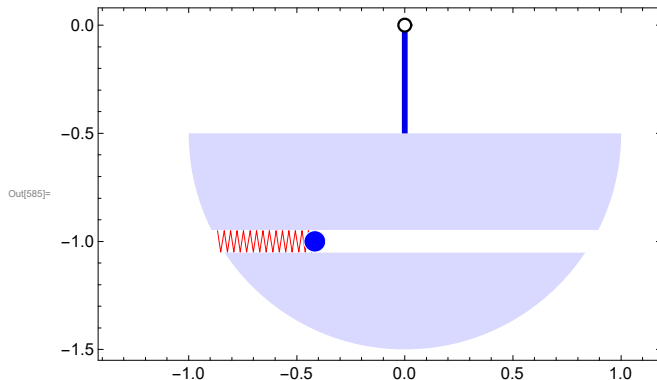
Рисуем систему в соответствии с параметрами в списке
params

```

In[583]:= ClearAll[body1]
body1[ $\varphi_i$ _,  $x_i$ _] := Module[{b, rshar = 0.05, dpru = 0.1},
  b = {(* СТЕРЖЕНЬ *)
    Darker[Blue, 0.1], Thickness[0.01], Line[{0, 0}, {0, -h}],
    (* ПЛАСТИНА *)
    Lighter[Blue, 0.85], Disk[{0, -h}, R, {-180°, 0°}],
    (* ОПОРА *)
    White, EdgeForm[{Thick, Black}], Disk[{0, 0}, 0.03],
    (* КАНАЛ *)
    EdgeForm[{Thin, White}],
    Rectangle[ $\rho_A + \{-0.5, -dpru/2\}$ ,  $\rho_A + \{2R, dpru/2\}$ ],
    (* ПРУЖИНА *)
    Red, Thin, Translate[Line[Spring[x[t] +  $l_0$ , dpru, 30]],  $\rho_A$ ],
    (* ШАРИК *)
    Blue, Disk[ $\rho_M$ , rshar]
  } /. { $\varphi[t] \rightarrow \varphi_i$ ,  $x[t] \rightarrow x_i$ } /. params;
  GeometricTransformation[b, {RotationMatrix[ $\varphi_i$ ], {0, 0}}]
];

```

```
Graphics[body1[0, 0.05], Frame -> True, ImageSize -> 500]
```



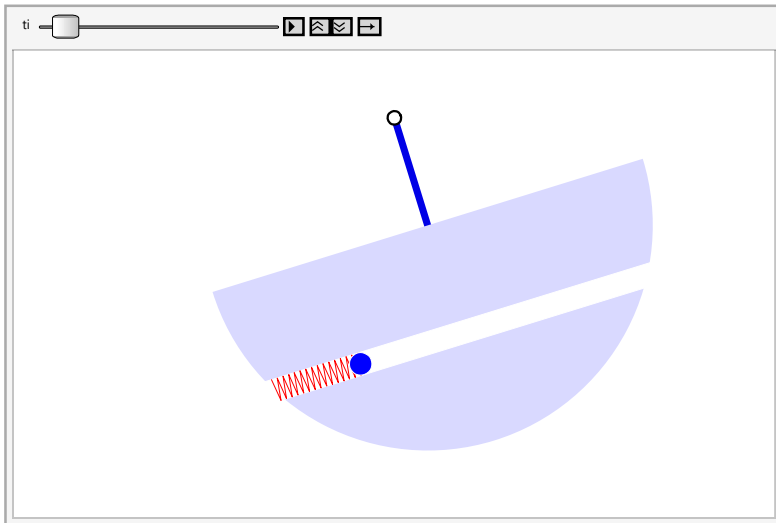
Анимация

```

In[600]:= Animate[
  Graphics[
    body1[ $\varphi[t]$ ,  $x[t]$ ] /. sol /. t  $\rightarrow$  ti,
    PlotRange  $\rightarrow$  {{-1.6, 1.6}, {-1.7, 0.2}}, ImageSize  $\rightarrow$  600
  ],
  {ti, 0, 5, 0.05},
  DisplayAllSteps  $\rightarrow$  True]

```

Out[600]=



Экспорт анимации

Анимацию можно сохранить в файл типа **анимированный gif**

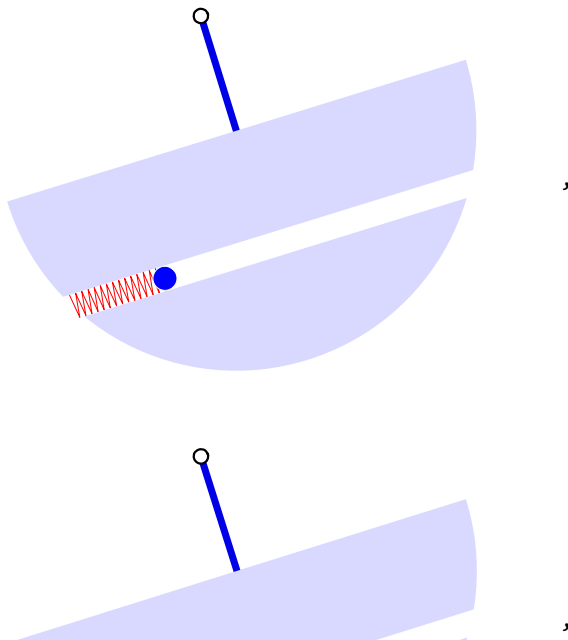
Генерируем кадры: t от 0 до t_{\max} с шагом 0.05 (100 кадров для 5 секунд) при помощи функции **Table**

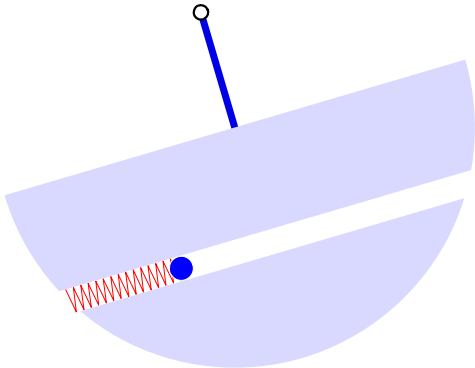
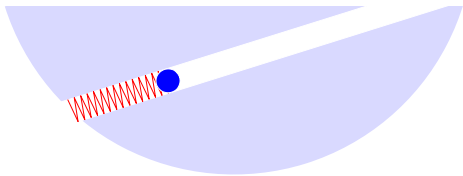
```
In[601]:= aniTable = Table[
  Graphics[
    body1[ $\varphi[t]$ , x[t]] /. sol /. t -> ti,
    PlotRange -> {{-1.5, 1.5}, {-1.5, 0.1}}, ImageSize -> 600
  ],
  {ti, 0, 5, 0.05}];
```

Первые пять кадров

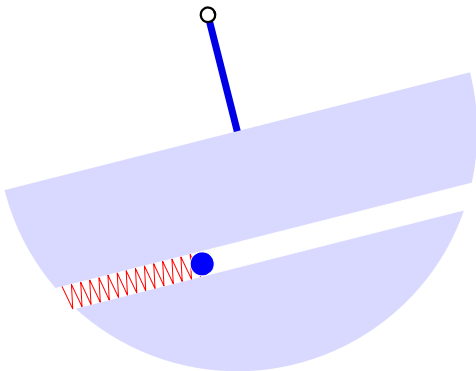
```
In[602]:= aniTable[[1 ;; 4]]
```

Out[602]= {





,



}

Экспорт

Для записи кадров в файл используем функцию **Export**

```
In[603]:= Export["animation.gif", aniTable]
```

```
Out[603]= animation.gif
```

Расширение имени файла (gif, avi) определит формат файла

```
In[604]:= Export["animation.avi", aniTable]
```

```
Out[604]= animation.avi
```

Файл получится большим, т.к. видео не сжато

Экспорт

Чтобы создался файл поменьше попытаемся использовать сжатие (формат MPEG-4)

```
In[518]:= Export["animation.avi", aniTable, "VideoEncoding" → "MPEG-4 Video"]
```

```
Out[518]:= animation.avi
```

Список форматов, поддерживаемых системой (Windows)

```
In[519]:= Internal`$VideoEncodings
```

```
Out[519]:= {Uncompressed}
```

...сжатие не поддерживается -- только Uncompressed (несжатый).