1. Задание (в программе)

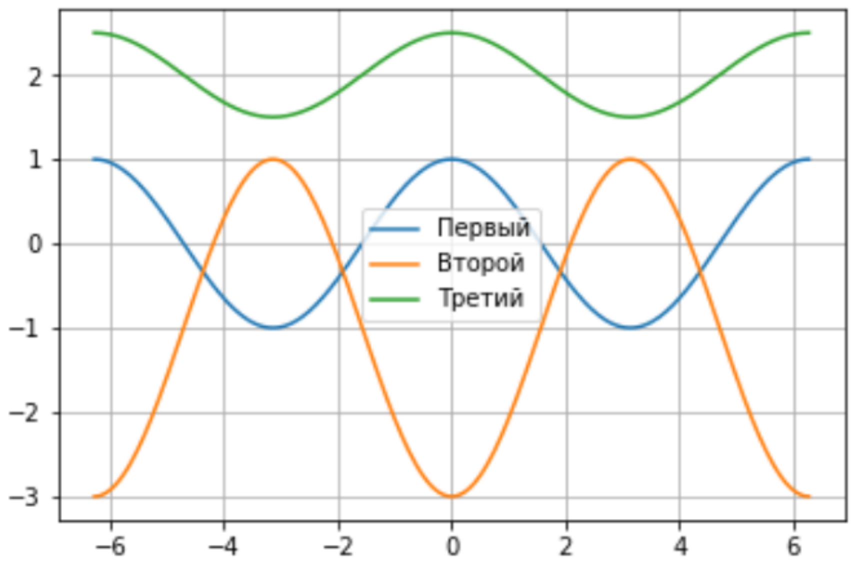
Нарисуйте график функции:

$$y(x) = k \cdot \cos(x - a) + b$$

для некоторых (2-3 различных) значений параметров k, a, b

In [22]:

```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt

# y(x) = k•cos(x - a) + b
x = np.linspace(-2 * np.pi, 2 * np.pi, 201)

def drawcos(k, a, b,label):
    plt.plot(x, k * np.cos(x - a) + b,label=label)
    plt.grid(True)
    plt.legend()
drawcos(1, 0, 0,'Первый')
drawcos(2, np.pi, -1,'Второй')
drawcos(0.5, 0, 2,'Третий')
```



2. Задание
Докажите, что при ортогональном преобразовании сохраняется расстояние между точками.

Условия ортогонального преобразования:

$$\begin{matrix} X = a_{11}x + a_{12}y + a_{13} \\ Y = a_{21}x + a_{22}y + a_{23} \end{matrix} \quad (1) \qquad \begin{matrix} a_{11}^2 + a_{21}^2 = 1 \\ a_{12}^2 + a_{22}^2 = 1 \end{matrix} \quad (2) \qquad a_{11}a_{12} + a_{21}a_{22} = 0 \quad (3)$$

Возьмём две произвольные точки в координатах x,y: (x_1, y_1) и (x_2, y_2). Тогда расстояние между этими точками будет равно: $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

Применим ортагональное преобразование плоскости (1) к координатам (x,y).

После преобразования получаем точки (X_1, Y_1) и (X_2, Y_2), соответсующие (x_1, y_1) и (x_2, y_2)

Запишем расстояние между точками в новых координатах:

$\sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}$, подставляем (1) $\Longrightarrow$

$$\sqrt{(a_{11}x_2 + a_{12}y_2 + a_{13} - (a_{11}x_1 + a_{12}y_1 + a_{13})\ )^2 + (a_{21}x_2 + a_{22}y_2 + a_{23} - (a_{21}x_1 + a_{22}y_1 + a_{23}))^2} \Longrightarrow$$

$$\sqrt{(a_{11}x_2 + a_{12}y_2 + a_{13} - a_{11}x_1 - a_{12}y_1 - a_{13}\ )^2 + (a_{21}x_2 + a_{22}y_2 + a_{23} - a_{21}x_1 - a_{22}y_1 - a_{23})^2} \Longrightarrow$$

$$\sqrt{(a_{11}x_2 + a_{12}y_2 - a_{11}x_1 - a_{12}y_1\ )^2 + (a_{21}x_2 + a_{22}y_2 - a_{21}x_1 - a_{22}y_1)^2} \Longrightarrow$$

$$\sqrt{(a_{11}(x_2 - x_1) + a_{12}(y_2 - y_1))^2 + (a_{21}(x_2 - x_1) + a_{22}(y_2 - y_1))^2} \Longrightarrow$$

$$\sqrt{\begin{matrix} a_{11}^2(x_2 - x_1)^2 + 2a_{12}(y_2 - y_1)a_{11}(x_2 - x_1) + a_{12}^2(y_2 - y_1)^2 + \\ a_{21}^2(x_2 - x_1)^2 + 2a_{21}(y_2 - y_1)a_{22}(x_2 - x_1) + a_{22}^2(y_2 - y_1)^2 \end{matrix}}$$

$$\sqrt{\left(a_{11}^2 + a_{21}^2\right)(x_2 - x_1)^2 + 2(a_{12}a_{11} + a_{21}a_{22})(y_2 - y_1)(x_2 - x_1) + \left(a_{12}^2 + a_{22}^2\right)(y_2 - y_1)^2}$$

Применяя условия (2) и (3) получаем

$$\sqrt{1*(x_2 - x_1)^2 + 2*0*(y_2 - y_1)(x_2 - x_1) + 1*(y_2 - y_1)^2} \Longrightarrow \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Таким образом, мы показали, что при ортогональном преобразовании плоскости расстояние между точками не изменяется

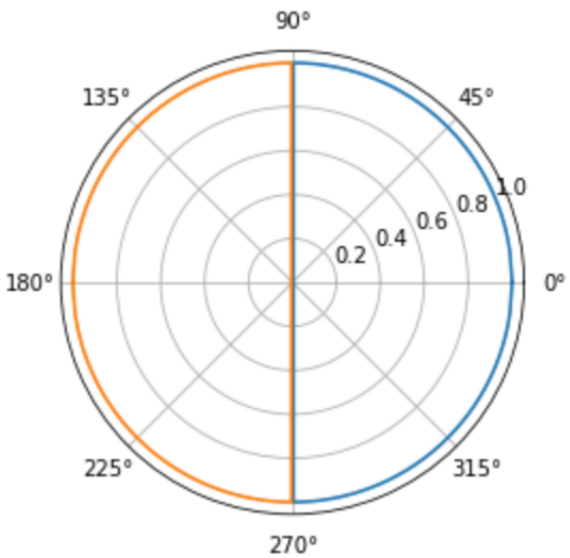
3. Задание (в программе)

1. Напишите код, который будет переводить полярные координаты в декартовы.

```
import math

R = float(input('Введите радиус-вектор R: '))
a = float(input('Введите угол a (в радианах): '))

if R < 0:
    print('Вы ввели недопустимое значение радиус-вектора')
elif a < 0 or a >= 2 * math.pi:
    print('Вы ввели недопустимое значение угла')
else:
    print(f'Полярным координатам ({R}, {a}) '
          f'соответсвуют декартовы координанты (x,y): ({R * math.cos(a)}, {R * math.sin(a)})')
Введите радиус-вектор R: 2
Введите угол a (в радианах): 3.1415
Полярным координатам (2.0, 3.1415) соответсвуют декартовы координанты (x,y): (-1.9999999914153124, 0.0001853071793209805)
```

2. Напишите код, который будет рисовать график окружности в полярных координатах.

```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np

# x ** 2 + y ** 2 = Radius ** 2
# R = sqrt ( x ** 2 + y ** 2) => R=Radius,
# a = arctg (y/x) + n*pi, x!=0, n=+-0,1,2...

def polarcircle(R):
    n = 200
    x = np.linspace(-1, 1, n)
    r = np.full(n, R)
    y = np.sqrt(R ** 2 - np.power(x, 2))
    a = np.arctan(np.divide(y, x))
    # пока что не знаю как красиво исключить x=0

    plt.polar(a, r)
    plt.polar(a + np.pi, r)

polarcircle(1)
```



4. Задание (в программе)

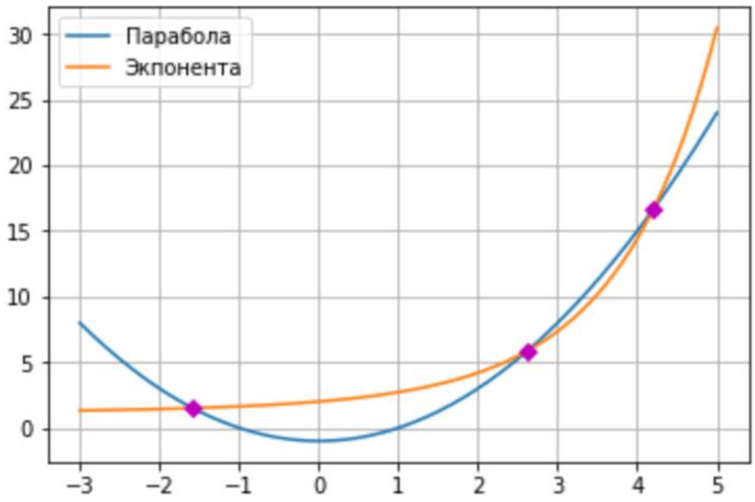
1. Решите систему уравнений:

$$y = x^2 - 1$$

$$\exp(x) + x \cdot (1 - y) = 1$$

```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
from scipy.optimize import fsolve

# y = x2 - 1
# exp(x) + x•(1 - y) = 1 => y = 1 - (1 - exp(x))/x

x = np.linspace(-3, 5, 202)
y1 = np.power(x, 2) - 1
y2 = 1 - np.divide(1 - np.exp(x), x)
plt.plot(x, y1, label='Парабола')
plt.plot(x, y2, label='Экпонента')
plt.legend()
plt.grid(True)

def equations(z):
    x, y = z
    return (np.power(x, 2) - 1 - y, np.exp(x) + np.multiply(x, 1 - y) - 1)

x1, y1 = fsolve(equations, (-2, -2))
x2, y2 = fsolve(equations, (2.5, 2.5))
x3, y3 = fsolve(equations, (4, 4))

plt.plot(x1, y1, 'Dm')
plt.plot(x2, y2, 'Dm')
plt.plot(x3, y3, 'Dm')

print(f'Найдено три решения системы уравнений: ({x1:.3f}, {y1:.3f}), '
      f'({x2:.3f}, {y2:.3f}), ({x3:.3f}, {y3:.3f})')
```

```
Найдено три решения системы уравнений: (-1.582, 1.502), (2.618, 5.855), (4.200, 16.641)
```



Округлил значения решения, т.к. исходный вариант не входил на один лист.

2. Решите систему уравнений и неравенств:

$y = x^2 – 1$

$\exp(x) + x\cdot(1 – y) > 1$ =>

$y = x^2 – 1.$ (1)

$y < 1 - (1- \exp(x)) / x$ (2)

т.к. в предыдущему задании мы уже строили второе уравнение, то визуально легко увидеть, что неравенство соответствует области ниже (не включая) кривой с названием «Экспонента» (2). Таким образом решениями будут пересечения этой области с уравнением кривой (1), т.е. все пары (x,y) удовлетворяющие кривой

$y = x^2 – 1$, для $x \in(-1.582, 2.618)$ и $(4.2, +\infty)$

Найдено три решения системы уравнений: (-1.582, 1.502), (2.618, 5.855), (4.200, 16.641)