14. На відрізку $x \in [-2; 1]$ з кроком $h = 0.2$ розв'язати рівняння методом Рунге-Кутта 4-го порядку точності та модифікованим методом Ейлера: $y' = 4 - x^2 - 2yx$; $y(-2) = 0.5$.

Метод Рунге-Кутта IV п.т.

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$k_1 = hf(x_n; y_n)$$

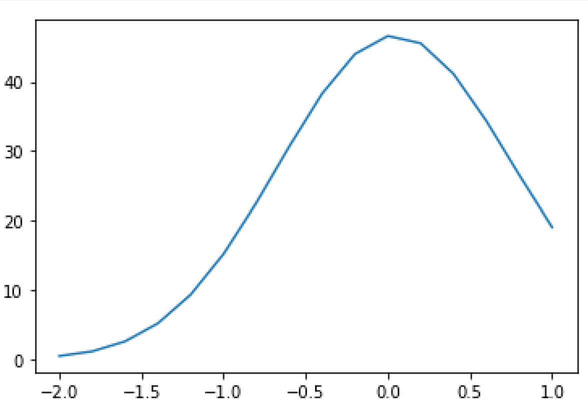
$$k_2 = hf(x_n + \frac{h}{2}; y_n + \frac{k_1}{2})$$

$$k_3 = hf(x_n + \frac{h}{2}; y_n + \frac{k_2}{2})$$

$$k_4 = hf(x_n + h; y_n + k_3)$$

In [2]:
```
import numpy as np
from sympy import *
import matplotlib.pyplot as plt
```

In [9]:
```
def runge_kutta_4(f, x0, y0, h, n):
    x_table, y_table = [x0], [y0]

    for i in range(n):
        x, y = x_table[-1], y_table[-1]

        k1 = h*f(x, y)
        k2 = h*f(x + h/2, y + k1/2)
        k3 = h*f(x + h/2, y + k2/2)
        k4 = h*f(x + h, y + k3)

        x_next = x + h
        y_next = y + 1/6 * (k1 + 2*k2 + 2*k3 + k4)
        x_table.append(x_next)
        y_table.append(y_next)

    return x_table, y_table

f = lambda x, y: 4 - x**2 - 2*x*y

x, y = runge_kutta_4(f, -2, 0.5, 0.2, 15)
plt.plot(x, y)
plt.show()
Matrix([x, y]).T.n(4)
```



Out[9]:

-2.0	0.5
-1.8	1.168
-1.6	2.607
-1.4	5.215
-1.2	9.365
-1.0	15.23
-0.8	22.59
-0.6	30.69
-0.4	38.31
-0.2	44.02
$-2.776 \cdot 10^{-16}$	46.62
0.2	45.57
0.4	41.15
0.6	34.36
0.8	26.58
1.0	19.07

Модифікований метод Ейлера

$$x_{n+\frac{1}{2}} = x_n + \frac{h}{2}$$

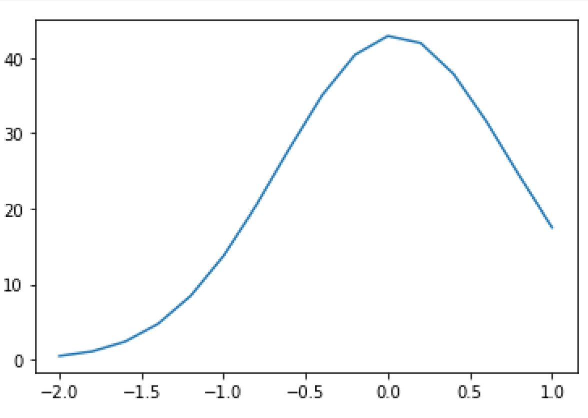
$$y_{n+\frac{1}{2}} = y_n + \frac{h}{2}f(x_n, y_n)$$

$$y_{n+1} = y_n + hf(x_{n+\frac{1}{2}}; y_{n+\frac{1}{2}})$$

$$p = 2$$

$$\Psi(h) = O(h^2)$$

In [8]:
```
def euler(f, x0, y0, h, n):
    x_table, y_table = [x0], [y0]

    for i in range(n):
        x, y = x_table[-1], y_table[-1]
        x_half = x + h/2
        y_half = y + h/2*f(x,y)

        x_next = x + h
        y_next = y + h*f(x_half, y_half)
        x_table.append(x_next)
        y_table.append(y_next)

    return x_table, y_table

f = lambda x, y: 4 - x**2 - 2*x*y

x, y = euler(f, -2, 0.5, 0.2, 15)
plt.plot(x, y)
plt.show()
Matrix([x, y]).T.n(4)
```



Out[8]:

-2.0	0.5
-1.8	1.11
-1.6	2.41
-1.4	4.755
-1.2	8.489
-1.0	13.79
-0.8	20.49
-0.6	27.95
-0.4	35.03
-0.2	40.4
$-2.776 \cdot 10^{-16}$	42.89
0.2	41.96
0.4	37.86
0.6	31.57
0.8	24.39
1.0	17.53

34. На відрізку $x \in [-2; 1]$ з точністю з кроком $h = 0.2$ розв'язати рівняння методом Рунге-Кутта 2-го порядку точності: $y'' = 4 - x^2 - yx$; $y(-2) = 0.5$; $y'(-2) = -1.5$.

Метод Рунге-Кутта II п.т.

$$\alpha_2 = 1/2 \Rightarrow p_2 = 1; \; p_1 = 0; \; \beta_{2_1} = 1/2;$$

$$y_{n+1} = y_n + p_1 k_1(h) + p_2 k_2(h) = y_n + k_2$$

$$k_1 = hf(x_n; y_n)$$

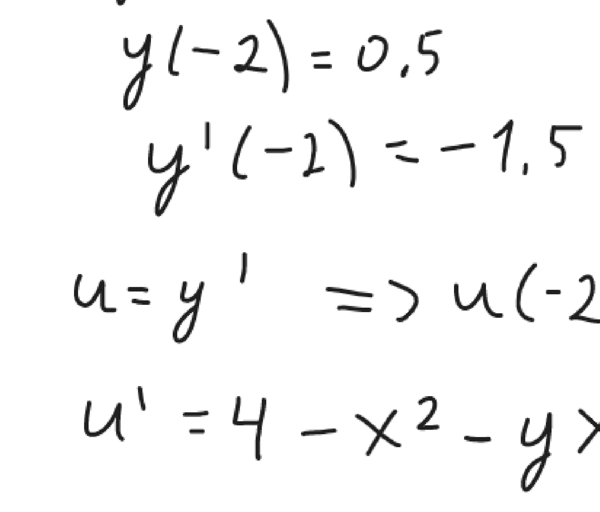
$$k_2 = hf(x_n + \alpha_2 h; y_n + \beta_{2_1} k_1) = hf(x_n + \frac{h}{2}; y_n + \frac{k_1}{2})$$

$$y'' = 4 - x^2 - yx$$
$$y(-2) = 0,5$$
$$y'(-2) = -1,5$$

$$u = y' \Rightarrow u(-2) = -1,5$$
$$u' = 4 - x^2 - yx = f(x, y)$$

$$y' = u$$
$$y(-2) = 0,5$$

In [10]:
```
def runge_kutta_2(f, x0, y0, u0, h, n):
    x_table, y_table, u_table = [x0], [y0], [u0]

    for i in range(n):
        x, y, u = x_table[-1], y_table[-1], u_table[-1]

        k1_u = h*f(x, y)
        k1_y = h*u

        k2_u = h*f(x + h/2, y + k1_y/2)
        k2_y = h*(u + k1_u/2)

        x_next = x + h
        y_next = y + k2_y
        u_next = u + k2_u

        x_table.append(x_next)
        y_table.append(y_next)
        u_table.append(u_next)

    return x_table, y_table

f = lambda x, y: 4 - x**2 - y*x

x, y = runge_kutta_2(f, -2, 0.5, -1.5, 0.2, 15)
plt.plot(x, y)
plt.show()
Matrix([x, y]).T.n(4)
```



Out[10]:

-2.0	0.5
-1.8	0.22
-1.6	-0.01468
-1.4	-0.1936
-1.2	-0.3025
-1.0	-0.3243
-0.8	-0.2396
-0.6	-0.02863
-0.4	0.3272
-0.2	0.8422
$-2.776 \cdot 10^{-16}$	1.523
0.2	2.366
0.4	3.351
0.6	4.438
0.8	5.567
1.0	6.656

In [ ]: