How we calculate famous constant values?

Application contains some basic constant values like:

- π
- e
- ϕ

Let's start from π.

To get this constant we use Leibniz formula.

$$1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \cdots = \frac{\pi}{4}$$

Write this equation in a more elegant style.

$$\frac{\pi}{4} = \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{2n-1}$$

Proof of this summation you will find here

In [ ]:
```python
import classes.Calculator.Constant as cnst
print('π =', cnst.pi(), '\n')
```

```
π = 3.140592653839794
```

And now let's scratch a graph for this summation we use implemented function in Calculator Constant

In [ ]:
```python
import classes.Calculator.Constant as cnst
import matplotlib.pyplot as plt

res = []
for limits in range(101):
    res.append(4*cnst.calculate_pi(limits))

plt.title("Leibniz summation")
x = list(range(len(res)))
plt.plot(x, res)
plt.xlabel('limit')
plt.ylabel('value of π')
plt.grid(True)
plt.show()
plt.close()
```

As we can see `limit~20` gives to us a nice approximation of the π.

Now it's the for very interesting number e.

Euler's number is very popular in Calculus but also in Physics. The first mentions appeared in 1618, in John Napier's context, when he was working on slide rule. Then Jacob Bernoulli solved most popular limit in Mathematics:

$$\mathrm{e} = \lim_{n\to\infty} \left(1 + \frac{1}{n}\right)^n$$

Calculating this constant from limit is not effective, look at this example:

In [ ]:
```python
import matplotlib.pyplot as plt
import classes.Calculator.Other as oth
n = list(range(101))
res = []
for args in range(1, len(n)+1):
    res.append(oth.power((1+oth.inversion(args)), args))

plt.title("Euler's number calculating from the limit")
plt.xlabel("n")
plt.ylabel("approximation of e")
plt.plot(n, res)
plt.grid(True)
plt.show()
plt.close()
```

In [ ]:
```python
n=100
print("Euler's number for n={} e=".format(n),oth.power(1+oth.inversion(n), n))
```

As we can see to get near approximation of e we have put in the formula $n = 100$ or bigger n, and still we don't get nice approximation as in the previous case, so we use another popular summation.

Formula which we'll use looks:

$$e = \sum_{n=0}^{\infty} \frac{1}{n!} = 1 + \frac{1}{1} + \frac{1}{1 \cdot 2} + \frac{1}{1 \cdot 2 \cdot 3} + \cdots$$

We can calculate e for $limit = 5$.

In [ ]:
```python
import classes.Calculator.Constant as cnst
print('e =',cnst.calculate_e(5))
```

In this way we get good approximation of e, using small limit. Also in this place it's worthy to show chart of this summation like previous example:

In [ ]:
```python
import matplotlib.pyplot as plt
import classes.Calculator.Constant as cnst

n=list(range(101))
res = []
for arg in range(len(n)):
    res.append(cnst.calculate_e(arg))

plt.title("Euler's number calculating from the summation")
plt.xlabel("n")
plt.ylabel("approximation of e")
plt.plot(n, res)
plt.grid(True)
plt.show()
plt.close()
```

Last constant is a ϕ.

This number is more famous for the UI designers or Website developer cause it describes a golden ratio. This constant we can get on two different ways:

- by solving quadratic equation:

$$\phi^2 - \phi - 1 = 0$$

$$\phi = \frac{1 \pm \sqrt{5}}{2}$$

- by summing up the series:

$$\phi = \sum_{n=1}^{\infty} \frac{1}{n}$$

In [ ]:
```python
import matplotlib.pyplot as plt
import numpy as np

phi = np.linspace(-5, 6, 500)
y = phi**2 - phi - 1

plt.plot(phi, y)
plt.title("$f(\\phi) = \\phi^2 - \\phi - 1$", fontsize=13)
plt.xlabel("X")
plt.ylabel("Y")
plt.grid(True)
plt.axhline(y=0, color='black', linewidth=0.5)
plt.axvline(x=0, color='black', linewidth=0.5)
plt.show()

x=list(range(1, 45))
res = []
for arg in range(len(x)):
    res.append(cnst.calculate_phi(arg))

plt.plot(x, res)
plt.title("calculated from sum", fontsize=13)
plt.xlabel("X")
plt.ylabel("Y")
plt.axhline(y=0, color='black', linewidth=0.5)
plt.axvline(x=0, color='black', linewidth=0.5)
plt.grid(True)
plt.show()
```

Bibliography

- Leibniz formula for π
- Euler's number
- Golden ratio

In [ ]: