Пшеничников Артём 467205

Лабораторная работа 1.

Численное интегрирование.

Вариант $467205 \% 30 = 15$.

$$15. \int_0^1 \cos x^2 dx$$

Метод прямоугольников:

$$\int_a^b f(x)\,dx \approx h \sum_{i=0}^{n-1} f(x_i)$$

Для $f(x)$: $\int_0^1 \cos x^2 dx \approx h \sum_{i=0}^{n-1} f(x_i)$

$h = \frac{1-0}{n} = \frac{1}{n}$; $x_i = 0 + (i+0,5)h = (i+\frac{1}{2})h$

```
func1 = lambda x: cos(x ** 2)

a = 0
b = 1
epsilon = 1e-5
```

```
def rectangles(func: Callable[[float], float], start: float, end: float, n: int) -> float:  3 usages  new *
    step = (end - start) / n
    half_step = step / 2
    x = start + half_step
    area = 0.0
    while x < end:
        area += func(x) * step
        x += step
    return area

def calculate(func):  1 usage  new *
    n = 1000
    i_prev = 0
    i = rectangles(func1, a, b, n)
    while abs(i - i_prev) > epsilon:
        n *= 2
        i_prev = i
        i = rectangles(func1, a, b, n)
    return i
```

```
print("Прямоугольники:")
print(f"Площадь = {calculate(rectangles):.10f}")
```

Получаем:

```
Прямоугольники:
Площадь = 0.9045242554
```

Метод трапеций:

$$\int_a^b f(x)\,dx \approx h\left(\frac{1}{2}\left(f(x_0)+f(x_n)\right)+\sum_{i=0}^{n-1} f(x_i)\right)$$

$$h = \frac{b-a}{n} = \frac{1}{n}, \quad x_i = a+ih = ih$$

```python
def trapezia(func: Callable[[float], float], start: float, end: float, n: int) -> float:
    h = (end - start) / n

    step = (end - start) / (n + 1 - 1)
    x = [start + i * step for i in range(n + 1)]

    integral = h / 2 * (func(x[0]) + 2 * sum([func(q) for q in x[1:-1]]) + func(x[-1]))
    return integral
```

```python
print("Трапеции:")
print(f"Площадь = {calculate(trapezia):.10f}")
```

Получаем:

```
Трапеции:
Площадь = 0.9045242554
```

Метод Симпсона:

$$\int_a^b f(x)\,dx \approx \frac{h}{3}\left(f(x_0)+f(x_n)+2\sum_{j=1}^{m-1} f(x_{2j})+4\sum_{j=1}^{m} f(x_{2j-1})\right)$$

$$h = \frac{b-a}{n} = \frac{1}{n}, \quad x_i = a+ih = ih$$

```python
def simpson(func: Callable[[float], float], start: float, end: float, n: int) -> float:
    h = (end - start) / n

    step = (end - start) / (n + 1 - 1)
    x = [start + i * step for i in range(n + 1)]

    integral = h/3 * (func(x[0]) + 4 * sum([func(q) for q in x[1:-1:2]]) +
                      2 * sum([func(q) for q in x[2:-2:2]]) + func(x[-1]))
    return integral
```

```python
print("Метод Симпсона:")
print(f"Площадь = {calculate(simpson):.10f}")
```

Получаем:

```
Метод Симпсона:
Площадь = 0.9045242554
```

Вывод: все методы при малом ε дают исчерпывающую точность.