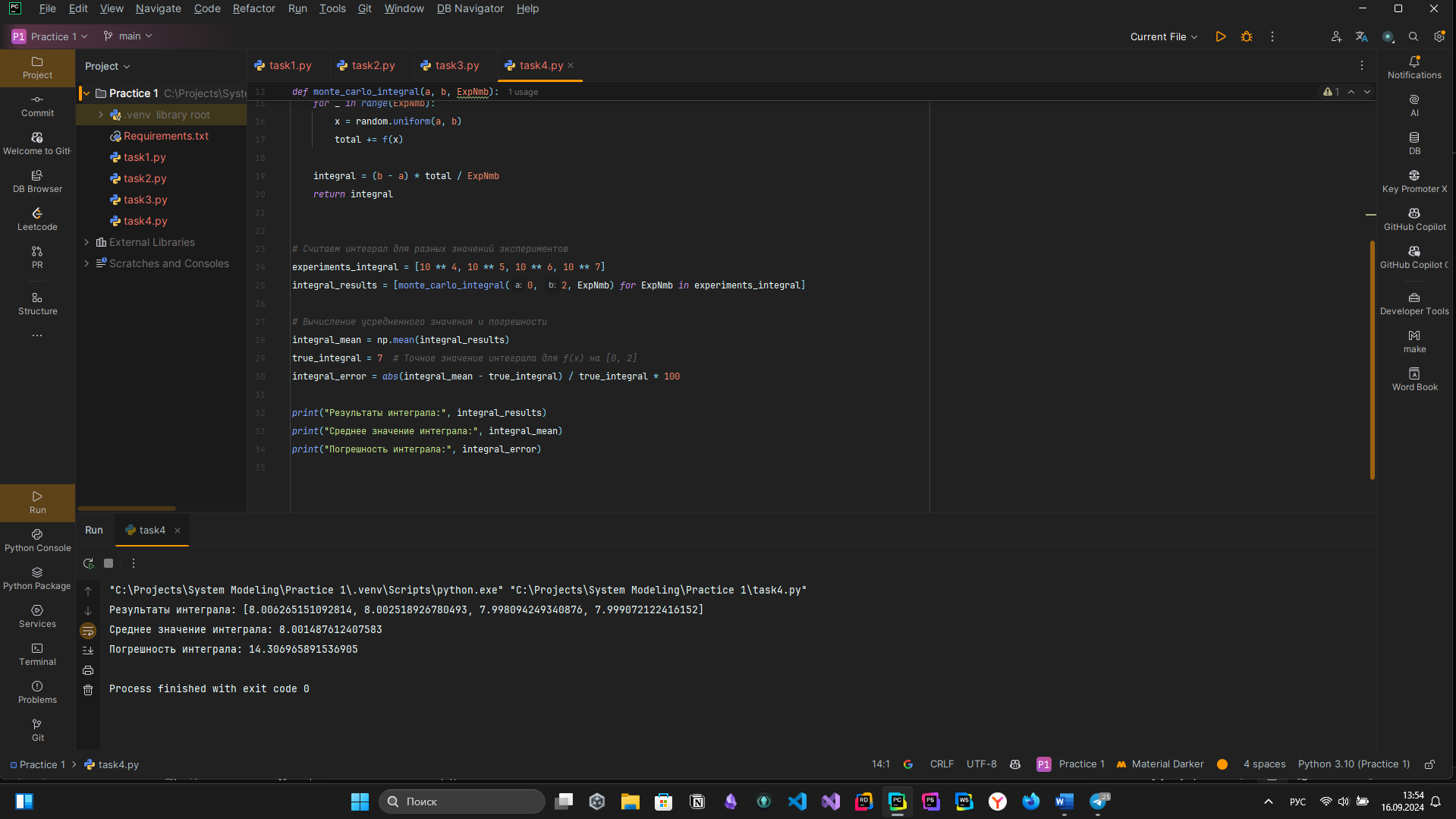
Практика 1

БСБО-09-23

Шутов Кирилл Сергеевич

Пример работы



Листинг кода

|  |
| --- |
| *# Задание 1: Реализация подпрограммы CALC\_PI* import random   def calc\_pi(x0, y0, r0, ExpNmb):  hits = 0   for \_ in range(ExpNmb):  *# Случайные точки в квадрате [-r0, r0] относительно центра окружности (x0, y0)* x = random.uniform(x0 - r0, x0 + r0)  y = random.uniform(y0 - r0, y0 + r0)   *# Проверяем, попала ли точка внутрь окружности* if (x - x0) \*\* 2 + (y - y0) \*\* 2 <= r0 \*\* 2:  hits += 1   *# Приближение числа pi* pi\_approx = 4 \* hits / ExpNmb  return pi\_approx |

|  |
| --- |
| *# Задание 2: Расчеты с разным количеством экспериментов* import task1   def run\_series(x0, y0, r0, experiments):  results = []   for ExpNmb in experiments:  pi\_value = task1.calc\_pi(x0, y0, r0, ExpNmb)  results.append(pi\_value)   return results   *# Число экспериментов для каждой серии* experiments\_1 = [10 \*\* 4, 10 \*\* 5, 10 \*\* 6, 10 \*\* 7, 10 \*\* 8] experiments\_2 = [10 \*\* 4, 10 \*\* 5, 10 \*\* 6, 10 \*\* 7, 10 \*\* 8]  *# Выполнение расчета для всех серий* SERIA\_1 = run\_series(10, 20, 50, experiments\_1) SERIA\_2 = run\_series(10, 20, 50, experiments\_2) |

|  |
| --- |
| *# Задание 3: Расчет погрешности* import numpy as np  from task2 import SERIA\_1, SERIA\_2   def calculate\_error(approx\_pi):  *# Теоретическое значение числа π* pi\_true = np.pi  return abs(approx\_pi - pi\_true) / pi\_true \* 100   *# Погрешности для каждой серии* errors\_1 = [calculate\_error(pi) for pi in SERIA\_1] errors\_2 = [calculate\_error(pi) for pi in SERIA\_2]  *# Усреднение результатов по 5ти сериям* SERIA\_MEAN = np.mean([SERIA\_1, SERIA\_2], axis=0)  *# Погрешности для усредненных значений* errors\_mean = [calculate\_error(pi) for pi in SERIA\_MEAN]  print("Средние значения SERIA:", SERIA\_MEAN) print("Погрешности:", errors\_mean) |

|  |
| --- |
| *# Задание 4: Вычисление определенного интеграла* import random  import numpy as np   def f(x):  return 3 + x   def monte\_carlo\_integral(a, b, ExpNmb):  total = 0   for \_ in range(ExpNmb):  x = random.uniform(a, b)  total += f(x)   integral = (b - a) \* total / ExpNmb  return integral   *# Считаем интеграл для разных значений экспериментов* experiments\_integral = [10 \*\* 4, 10 \*\* 5, 10 \*\* 6, 10 \*\* 7] integral\_results = [monte\_carlo\_integral(0, 2, ExpNmb) for ExpNmb in experiments\_integral]  *# Вычисление усредненного значения и погрешности* integral\_mean = np.mean(integral\_results) true\_integral = 7 *# Точное значение интеграла для f(x) на [0, 2]* integral\_error = abs(integral\_mean - true\_integral) / true\_integral \* 100  print("Результаты интеграла:", integral\_results) print("Среднее значение интеграла:", integral\_mean) print("Погрешность интеграла:", integral\_error) |