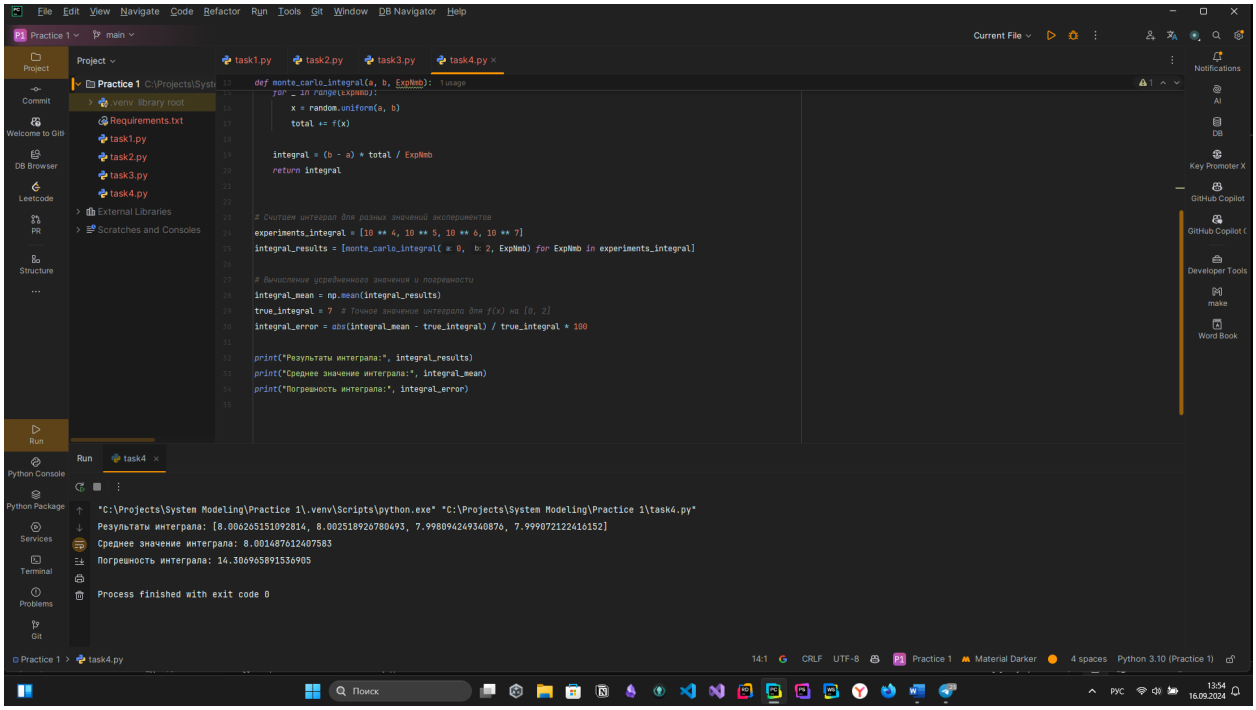
Практика 1

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)
```