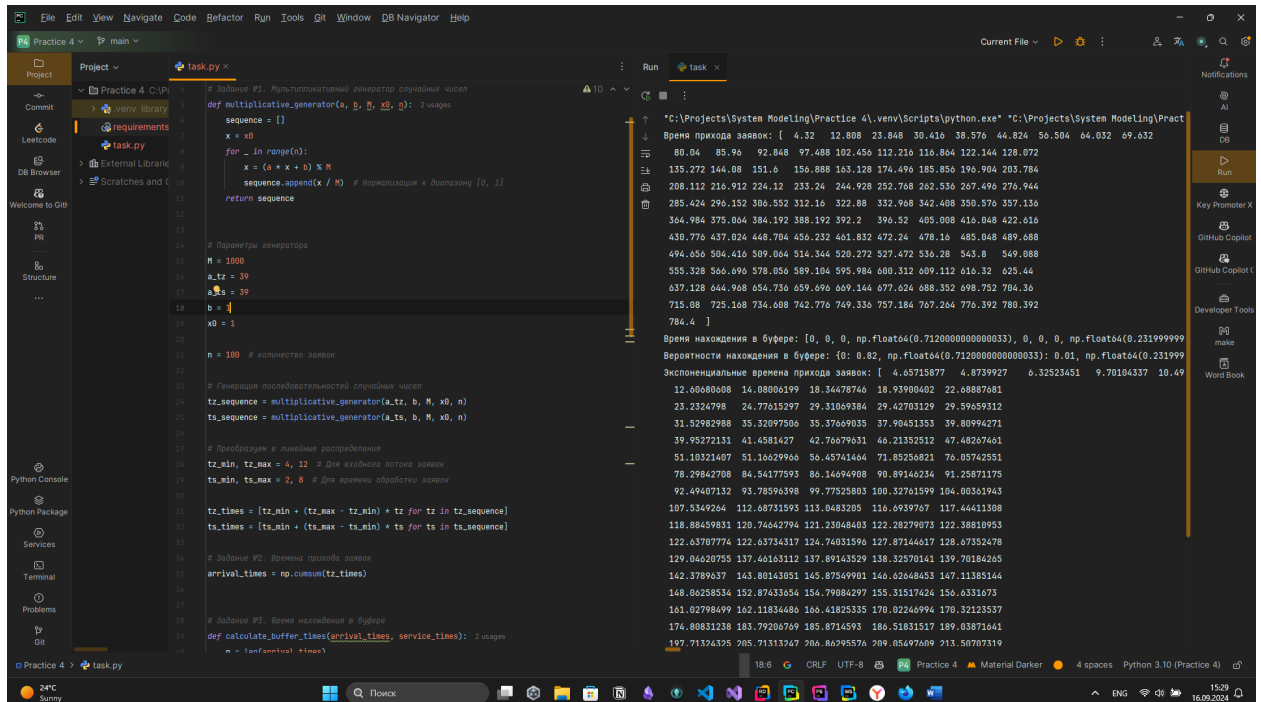


## Пример работы



```
# Задача #1. Рандомизированный генератор случайных чисел
def multiplicative_generator(a, b, M, x0, n): 2 знака
    sequence = []
    x = x0
    for _ in range(n):
        x = (a * x + b) % M
        sequence.append(x / M)  # Нормализация к диапазону [0, 1]
    return sequence

# Параметры генератора
M = 1000
a_tz = 39
a_ts = 39
b = 1
x0 = 1

n = 100  # количество заявок

# Генерация последовательностей случайных чисел
tz_sequence = multiplicative_generator(a_tz, b, M, x0, n)
ts_sequence = multiplicative_generator(a_ts, b, M, x0, n)

# Преобразуем к линейному распределению
tz_min, tz_max = 4, 12  # для входного потока заявок
ts_min, ts_max = 2, 8  # для времени обработки заявок

tz_times = [tz_min + (tz_max - tz_min) * tz for tz in tz_sequence]
ts_times = [ts_min + (ts_max - ts_min) * ts for ts in ts_sequence]

# Задача #2. Времена прихода заявок
arrival_times = np.cumsum(tz_times)

# Задача #3. Времена нахождения в буфере
def calculate_buffer_times(arrival_times, service_times): 2 знака
    n = len(arrival_times)
    buffer_times = [0] * n
    for i in range(1, n):
        buffer_times[i] = max(0, arrival_times[i] - arrival_times[i-1] - service_times[i-1])
    return buffer_times

# Задача #4. Времена нахождения в системе
def calculate_system_times(arrival_times, service_times, buffer_times):
    n = len(arrival_times)
    system_times = [0] * n
    for i in range(1, n):
        system_times[i] = max(0, arrival_times[i] - arrival_times[i-1] - service_times[i-1] + buffer_times[i-1])
    return system_times

# Задача #5. Вероятности нахождения заявок в буфере
from collections import Counter

def program_probabilities(buffer_times): 2 знака
    counter = Counter(buffer_times)
    total = len(buffer_times)
    probabilities = {}
    for k, v in counter.items():
        probabilities[k] = v / total
    return probabilities

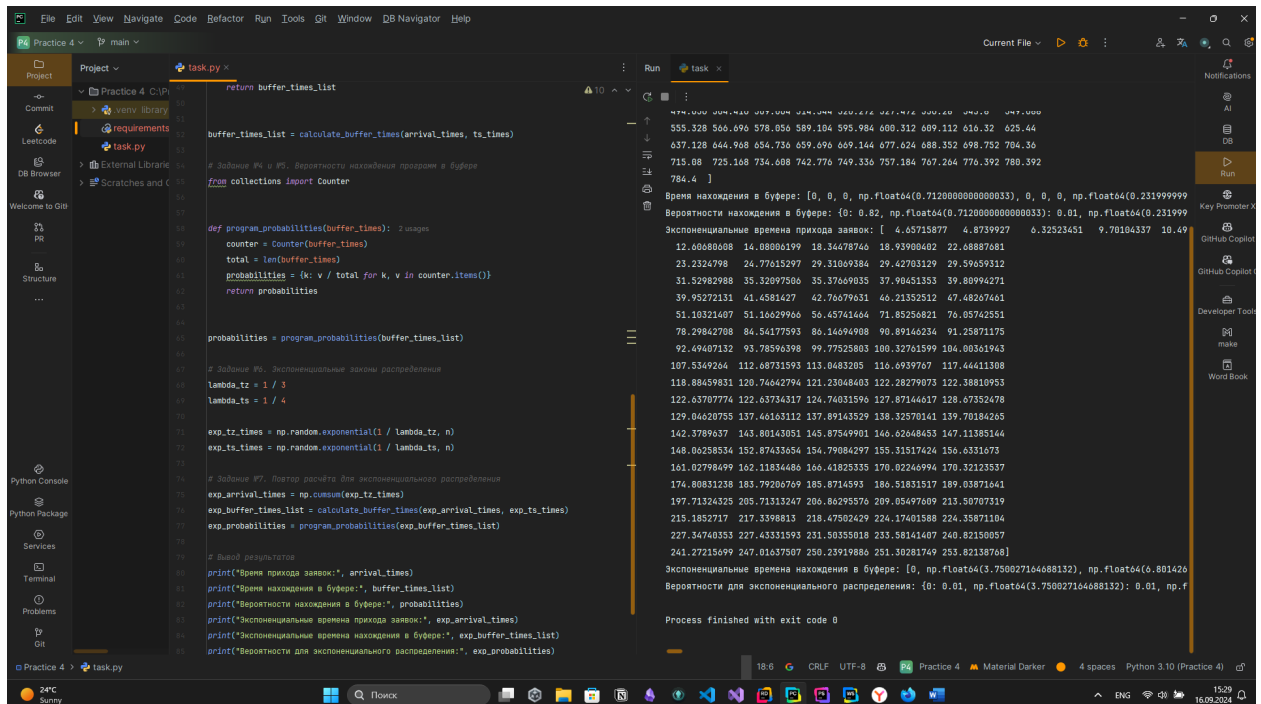
probabilities = program_probabilities(buffer_times_list)

# Задача #6. Экспоненциальные законы распределения
lambda_tz = 1 / 3
lambda_ts = 1 / 4

exp_tz_times = np.random.exponential(1 / lambda_tz, n)
exp_ts_times = np.random.exponential(1 / lambda_ts, n)

# Задача #7. Повтор расчёта для экспоненциального распределения
exp_arrival_times = np.cumsum(exp_tz_times)
exp_buffer_times_list = calculate_buffer_times(exp_arrival_times, exp_ts_times)
exp_probabilities = program_probabilities(exp_buffer_times_list)

# Вывод результатов
print("Времена прихода заявок:", arrival_times)
print("Времена нахождения в буфере:", buffer_times_list)
print("Вероятности нахождения в буфере:", probabilities)
print("Экспоненциальные времена прихода заявок:", exp_arrival_times)
print("Экспоненциальные времена нахождения в буфере:", exp_buffer_times_list)
print("Вероятности для экспоненциального распределения:", exp_probabilities)
```



```
return buffer_times_list

# Задача #4. Времена нахождения в системе
def calculate_system_times(arrival_times, service_times, buffer_times):
    n = len(arrival_times)
    system_times = [0] * n
    for i in range(1, n):
        system_times[i] = max(0, arrival_times[i] - arrival_times[i-1] - service_times[i-1] + buffer_times[i-1])
    return system_times

# Задача #5. Вероятности нахождения заявок в буфере
from collections import Counter

def program_probabilities(buffer_times): 2 знака
    counter = Counter(buffer_times)
    total = len(buffer_times)
    probabilities = {}
    for k, v in counter.items():
        probabilities[k] = v / total
    return probabilities

probabilities = program_probabilities(buffer_times_list)

# Задача #6. Экспоненциальные законы распределения
lambda_tz = 1 / 3
lambda_ts = 1 / 4

exp_tz_times = np.random.exponential(1 / lambda_tz, n)
exp_ts_times = np.random.exponential(1 / lambda_ts, n)

# Задача #7. Повтор расчёта для экспоненциального распределения
exp_arrival_times = np.cumsum(exp_tz_times)
exp_buffer_times_list = calculate_buffer_times(exp_arrival_times, exp_ts_times)
exp_probabilities = program_probabilities(exp_buffer_times_list)

# Вывод результатов
print("Времена прихода заявок:", arrival_times)
print("Времена нахождения в буфере:", buffer_times_list)
print("Вероятности нахождения в буфере:", probabilities)
print("Экспоненциальные времена прихода заявок:", exp_arrival_times)
print("Экспоненциальные времена нахождения в буфере:", exp_buffer_times_list)
print("Вероятности для экспоненциального распределения:", exp_probabilities)
```

## Листинг кода

```
import numpy as np

# Задание №1. Мультипликативный генератор случайных чисел
def multiplicative_generator(a, b, M, x0, n):
    sequence = []
    x = x0
    for _ in range(n):
        x = (a * x + b) % M
        sequence.append(x / M) # Нормализация к диапазону [0, 1]
    return sequence

# Параметры генератора
M = 1000
a_tz = 39
a_ts = 39
b = 1
x0 = 1

n = 100 # количество заявок

# Генерация последовательностей случайных чисел
tz_sequence = multiplicative_generator(a_tz, b, M, x0, n)
ts_sequence = multiplicative_generator(a_ts, b, M, x0, n)

# Преобразуем в линейные распределения
tz_min, tz_max = 4, 12 # Для входного потока заявок
ts_min, ts_max = 2, 8 # Для времени обработки заявок

tz_times = [tz_min + (tz_max - tz_min) * tz for tz in tz_sequence]
ts_times = [ts_min + (ts_max - ts_min) * ts for ts in ts_sequence]

# Задание №2. Времена прихода заявок
arrival_times = np.cumsum(tz_times)

# Задание №3. Время нахождения в буфере
def calculate_buffer_times(arrival_times, service_times):
    n = len(arrival_times)
    buffer_times_list = [0] * n
    finish_time = 0

    for i in range(n):
        if arrival_times[i] < finish_time:
            buffer_times_list[i] = finish_time - arrival_times[i]
            finish_time = max(arrival_times[i], finish_time) + service_times[i]

    return buffer_times_list

buffer_times_list = calculate_buffer_times(arrival_times, ts_times)

# Задание №4 и №5. Вероятности нахождения программ в буфере
from collections import Counter
```

```
def program_probabilities(buffer_times):
    counter = Counter(buffer_times)
    total = len(buffer_times)
    probabilities = {k: v / total for k, v in counter.items()}
    return probabilities

probabilities = program_probabilities(buffer_times_list)

# Задание №6. Экспоненциальные законы распределения
lambda_tz = 1 / 3
lambda_ts = 1 / 4

exp_tz_times = np.random.exponential(1 / lambda_tz, n)
exp_ts_times = np.random.exponential(1 / lambda_ts, n)

# Задание №7. Повтор расчёта для экспоненциального распределения
exp_arrival_times = np.cumsum(exp_tz_times)
exp_buffer_times_list = calculate_buffer_times(exp_arrival_times, exp_ts_times)
exp_probabilities = program_probabilities(exp_buffer_times_list)

# Вывод результатов
print("Время прихода заявок:", arrival_times)
print("Время нахождения в буфере:", buffer_times_list)
print("Вероятности нахождения в буфере:", probabilities)
print("Экспоненциальные времена прихода заявок:", exp_arrival_times)
print("Экспоненциальные времена нахождения в буфере:", exp_buffer_times_list)
print("Вероятности для экспоненциального распределения:", exp_probabilities)
```