Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Факультет электронно-информационных систем

Кафедра ИИТ

Лабораторная работа №1

По дисциплине «Современные платформы программирования»

Выполнила:

Студентка 3 курса

Группы ПО-11

Лозицкая М.А.

Проверил:

Козик И.Д.

Брест 2025

**Цель работы:** закрепить базовые знания языка программирования Python при решении практических задач

**Вариант 13**

**Задание 1.**Для последовательности из N целых чисел реализовать обработку:

Вывод среднего значения и выборочной дисперсии последовательности.

Код программы:  
def calculate\_mean\_and\_variance(numbers):

"""Вычисляет среднее и выборочную дисперсию последовательности."""

n = len(numbers)

if n == 0:

raise ValueError("Последовательность не должна быть пустой!")

mean = sum(numbers) / n

variance = sum((x - mean) \*\* 2 for x in numbers) / (n - 1) if n > 1 else 0.0

return mean, variance

def input\_sequence():

"""Запрашивает у пользователя последовательность чисел с обработкой ошибок."""

while True:

try:

input\_str = input("Введите последовательность целых чисел через пробел: ").strip()

if not input\_str:

raise ValueError("Ввод не должен быть пустым!")

numbers = list(map(int, input\_str.split()))

return numbers

except ValueError as e:

print(f"Ошибка ввода: {e}. Попробуйте снова.")

except Exception as e:

print(f"Непредвиденная ошибка: {e}. Попробуйте снова.")

if \_\_name\_\_ == "\_\_main\_\_":

try:

numbers = input\_sequence()

mean, variance = calculate\_mean\_and\_variance(numbers)

print(f"Среднее значение: {mean}")

print(f"Выборочная дисперсия: {variance}")

except ValueError as e:

print(f"Ошибка вычислений: {e}")

except ZeroDivisionError:

print("Ошибка: деление на ноль (невозможно вычислить дисперсию для одного элемента).")

except Exception as e:

print(f"Непредвиденная ошибка: {e}")

Результат работы программы:

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

**Задание 2.**

13) Дана строка s, состоящая из слов и пробелов, вернуть длину последнего слова в строке. Слово — это максимальная подстрока, состоящая только из непробельных символов.

Код программы:  
def length\_of\_last\_word(s: str) -> int:  
 words = s.rstrip().split(' ')  
 return len(words[-1])  
while True:  
 # Ввод строки от пользователя  
 s = input("Введите строку (или 'exit' для выхода): ").strip()  
 # Проверка на выход  
 if s.lower() == 'exit':  
 print("Программа завершена.")  
 break  
 # Проверка на пустую строку  
 if not s:  
 print("Ошибка: строка пустая. Попробуйте снова.")  
 continue  
 # Вычисление и вывод результата  
 result = length\_of\_last\_word(s)  
 print(f"Длина последнего слова: {result}\n")

Результат работы программы:

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

**Вывод:** закрепила базовые знания языка программирования Python при решении практических задач