

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Факультет “Информатика и системы управления”
Кафедра “Системы обработки информации и управления”



Дисциплина “Парадигмы и конструкции языков программирования”

Отчет по РК2

Выполнил:

Студент группы ИУ5-36Б

Шах А. М.

Преподаватель:

Гапанюк Ю. Е.

Москва 2025

Условия рубежного контроля №2 по курсу ПиК ЯП

Рубежный контроль представляет собой разработку тестов на языке Python.

- 1) Проведите рефакторинг текста программы рубежного контроля №1 таким образом, чтобы он был пригоден для модульного тестирования.
- 2) Для текста программы рубежного контроля №1 создайте модульные тесты с применением TDD - фреймворка (3 теста).

Код программы

refactored_program.py

```
def calculate_sum(a, b):
    """Возвращает сумму двух чисел."""
    return a + b

def calculate_product(a, b):
    """Возвращает произведение двух чисел."""
    return a * b

def main():
    """Основная функция программы."""
    try:
        a = int(input("Введите a: "))
        b = int(input("Введите b: "))

        sum_result = calculate_sum(a, b)
        prod_result = calculate_product(a, b)

        print(f"Сумма: {sum_result}")
        print(f"Произведение: {prod_result}")
    except ValueError:
        print("Ошибка: введите целые числа.")

if __name__ == "__main__":
    main()
```

test_math_operations.py

```
import unittest
from refactored_program import calculate_sum, calculate_product

class TestMathOperations(unittest.TestCase):

    def test_sum_positive(self):
        self.assertEqual(calculate_sum(5, 3), 8)

    def test_sum_negative(self):
```

```
self.assertEqual(calculate_sum(-5, -3), -8)

def test_product_positive(self):
    self.assertEqual(calculate_product(4, 7), 28)

def test_product_with_zero(self):
    self.assertEqual(calculate_product(5, 0), 0)

def test_product_negative(self):
    self.assertEqual(calculate_product(-4, 5), -20)

if __name__ == "__main__":
    unittest.main()
```

Результат

Ведите a: 3
Ведите b: 4
Сумма: 7
Произведение: 12

Ran 5 tests in 0.000s

OK