| Домашнее задание |
| --- |

**Результат**

ССЫЛКА на github с выполненным заданием

<https://github.com/makarova1507ana/python312/tree/OOP/lesson31> - последняя практика

**ЗАДАНИЕ (можно прислать один код и за предыдущее дз и за это)**

**Выбрать минимум 1 любую задачу**

**задача**

напиши тестовые сценарии для данной функции и протестируйте ее при помощи pytest

| def calculate\_average(numbers: List[float]) -> float:  """  Вычисляет среднее значение списка чисел.  :param numbers: Список чисел  :return: Среднее значение  """  if not numbers:  raise ValueError("Список чисел не должен быть пустым")  return sum(numbers) / len(numbers) |
| --- |

**задача**

напиши тестовые сценарии для данной функции и протестируйте ее при помощи pytest

| def is\_even(number: int) -> bool:  """  Проверяет, является ли число четным.  :param number: Проверяемое число  :return: True, если число четное, иначе False  """  return number % 2 == 0 |
| --- |

**задача**

напиши тестовые сценарии для данной функции и протестируйте ее при помощи pytest

| def is\_prime(number: int) -> bool:  """  Проверяет, является ли число простым.  :param number: Проверяемое число  :return: True, если число простое, иначе False  """  if number <= 1:  # Числа меньше или равные 1 не являются простыми  return False  elif number == 2:  # 2 - единственное четное простое число  return True  elif number % 2 == 0:  # Все другие четные числа не являются простыми  return False  else:  # Проверяем делители от 3 до квадратного корня из числа  for i in range(3, int(number\*\*0.5) + 1, 2):  if number % i == 0:  return False  return True |
| --- |

**задача**

напиши тестовые сценарии для данной функции и протестируйте ее при помощи pytest

| class Rectangle:  """  Класс Rectangle представляет прямоугольник.  :param width: Ширина прямоугольника  :param height: Высота прямоугольника  """  def \_\_init\_\_(self, width: float, height: float):  self.width = width  self.height = height  def area(self) -> float:  """  Вычисляет площадь прямоугольника.  :return: Площадь прямоугольника  """  return self.width \* self.height  def perimeter(self) -> float:  """  Вычисляет периметр прямоугольника.  :return: Периметр прямоугольника  """  return 2 \* (self.width + self.height)  # Пример использования класса  # Создаем экземпляр прямоугольника  rectangle = Rectangle(width=5, height=10)  # Получаем и выводим площадь и периметр  print(f"Площадь прямоугольника: {rectangle.area()}")  print(f"Периметр прямоугольника: {rectangle.perimeter()}") |
| --- |