Министерство Просвещения, Культуры и Исследований Республики Молдова

Технический университет Молдовы

Факультет Вычислительной Техники, Информатики и Микроэлектроники

Департамент Микроэлектроники и Биоинженерии

**Отчет**

**По предмету Programarea Declarativă**

**По лабораторной работе №6**

**Тема:** «Тестирование модуля date\_utils»

Выполнил Cuțar Alexandr (TI-218)

Проверил A.Leah

Кишинев 2023

**Цель:** Создать тестовый модуль с использованием библиотеки unittest для проверки правильности работы функций модуля date\_utils. Протестировать модуль с использованием тестового модуля и удостовериться, что все тесты проходят успешно.

1. Создайте тестовый модуль (unittest) для вашего модуля реализованного в 5 лабораторной date\_utils.
2. Напишите несколько тестовых кейсов для функций, которые требуется протестировать.
3. Запустите тестовый модуль для проверки правильности работы функций в модуле date\_utils.
4. Удостоверьтесь, что все тесты проходят успешно.

**Код test\_date\_utils.py**

# test\_date\_utils.py  
  
import unittest  
from date\_utils import is\_leap\_year, days\_between\_dates, get\_current\_date, add\_days\_to\_date, has\_date\_passed  
  
class TestDateUtils(unittest.TestCase):  
  
 def test\_is\_leap\_year(self):  
 self.assertTrue(is\_leap\_year(2020))  
 self.assertFalse(is\_leap\_year(2019))  
 self.assertTrue(is\_leap\_year(2000))  
 self.assertFalse(is\_leap\_year(1900))  
  
 def test\_days\_between\_dates(self):  
 self.assertEqual(days\_between\_dates("2020-01-01", "2020-01-02"), 1)  
 self.assertEqual(days\_between\_dates("2020-01-02", "2020-01-01"), -1)  
 self.assertEqual(days\_between\_dates("2020-01-01", "2020-01-01"), 0)  
  
 def test\_get\_current\_date(self):  
 # Since this test will be run at an unknown time, we just check if it returns a string  
 self.assertIsInstance(get\_current\_date(), str)  
  
 def test\_add\_days\_to\_date(self):  
 self.assertEqual(add\_days\_to\_date("2020-01-01", 1), "2020-01-02")  
 self.assertEqual(add\_days\_to\_date("2020-01-01", -1), "2019-12-31")  
  
 def test\_has\_date\_passed(self):  
 self.assertTrue(has\_date\_passed("2000-01-01"))  
 self.assertFalse(has\_date\_passed("2999-01-01"))  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 unittest.main()

**Результат выполнения тестов:**

