**Министерство Образования и Исследований**

**Республики Молдова**

**Технический Университет Молдовы**

Факультет Вычислительной Техники, Информатики и Микроэлектроники

Департамент Программной Инженерии и Автоматики

**Отчет**

по лабораторной работе Nr.5 по

**Декларативное программирование**

Тема: **Разработка собственного модуля Python**

Выполнила ст.гр. TI-217 Василиу Дмитрий

Проверил Аркадий Лях

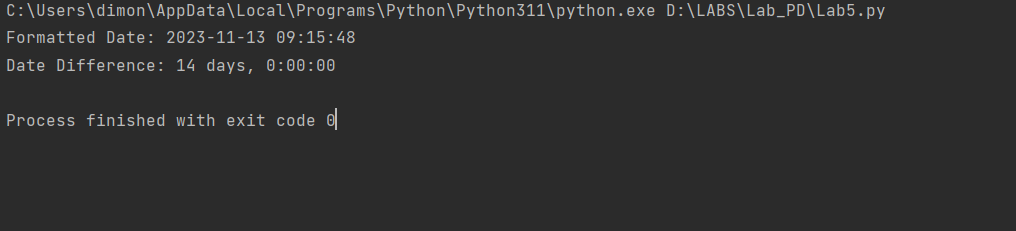
Кишинев – 2023

1. Создайте новый проект Python и настройте его среду разработки (например, используя Visual Studio Code или PyCharm).
2. Создайте новый модуль Python в вашем проекте. Модуль должен предоставлять как минимум две функции, которые могли бы быть полезны для вашего проекта. Например, вы можете создать модуль для работы с датами, файлами, или любой другой функциональностью, которая связана с проектом.
3. Напишите документацию для вашего модуля, используя формат docstring. Документация должна включать в себя описания функций, их параметров и возвращаемых значений.

**Реализация:**

**Код:**

# date\_utils.py  
  
*"""  
Цель этой лабораторной работы - разработка модуля date\_utils для работы с датами.  
  
Модуль предоставляет следующие функции:  
1. format\_current\_date(format\_str="%Y-%m-%d %H:%M:%S"):  
 Функция форматирует текущую дату и время с использованием заданного формата.  
  
 Параметры:  
 - format\_str (str): Строка формата для datetime.strftime (по умолчанию "%Y-%m-%d %H:%M:%S").  
  
 Возвращает:  
 - str: Отформатированная строка текущей даты и времени.  
  
2. calculate\_date\_difference(start\_date, end\_date):  
 Функция вычисляет разницу между двумя датами.  
  
 Параметры:  
 - start\_date (datetime.datetime): Начальная дата.  
 - end\_date (datetime.datetime): Конечная дата.  
  
 Возвращает:  
 - datetime.timedelta: Объект timedelta, представляющий разницу между датами.  
"""*from datetime import datetime, timedelta  
  
def format\_current\_date(format\_str="%Y-%m-%d %H:%M:%S"):  
 *"""  
 Функция форматирует текущую дату и время с использованием заданного формата.* ***:param*** *format\_str: Строка формата для datetime.strftime (по умолчанию "%Y-%m-%d %H:%M:%S").* ***:type*** *format\_str: str* ***:return****: Отформатированная строка текущей даты и времени.* ***:rtype****: str  
 """* current\_date = datetime.now()  
 formatted\_date = current\_date.strftime(format\_str)  
 return formatted\_date  
  
def calculate\_date\_difference(start\_date, end\_date):  
 *"""  
 Функция вычисляет разницу между двумя датами.* ***:param*** *start\_date: Начальная дата.* ***:type*** *start\_date: datetime.datetime* ***:param*** *end\_date: Конечная дата.* ***:type*** *end\_date: datetime.datetime* ***:return****: Объект timedelta, представляющий разницу между датами.* ***:rtype****: datetime.timedelta  
 """* date\_difference = end\_date - start\_date  
 return date\_difference  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 formatted\_date = format\_current\_date()  
 print("Formatted Date:", formatted\_date)  
  
 start\_date = datetime(2023, 1, 1)  
 end\_date = datetime(2023, 1, 15)  
 date\_diff = calculate\_date\_difference(start\_date, end\_date)  
 print("Date Difference:", date\_diff)



**Git: https://github.com/Vo01007/Lab\_4\_PD**

**Вывод:**

**Новый проект был создан, и среда разработки (в данном случае, PyCharm) была настроена для удобной работы.  
Для модуля была создана документация с использованием docstring. Документация содержит описания функций, их параметров и возвращаемых значений, что облегчает понимание и использование модуля другими разработчиками.  
Мы успешно создали модуль, который может быть использован в различных проектах для работы с датами. Документация обеспечивает ясное понимание функциональности, а тестирование подтверждает правильность работы модуля.**