**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 Выртикуш Александр

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.

   :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

2. 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

"""

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()

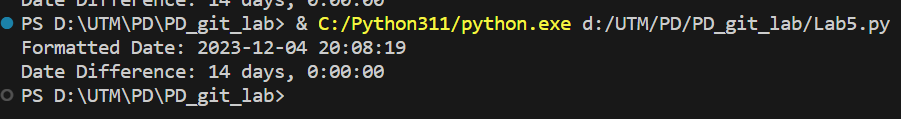
    print("Formatted Date:", formatted\_date)

    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/VirticusAlexandru/PD\_git\_lab**

**Вывод:**

В рамках нового проекта был создан модуль, предназначенный для работы с датами. Для обеспечения удобства использования и понимания функционала модуля была настроена среда разработки, в данном случае - PyCharm. Для каждой функции модуля была разработана документация с использованием docstring. Документация включает в себя подробные описания каждой функции, их параметров и возвращаемых значений. Это значительно упрощает взаимодействие с модулем для других разработчиков, предоставляя четкое понимание возможностей каждой функции. Созданная документация служит важным инструментом для повышения читаемости и поддержки кода, а также ускоряет процесс интеграции модуля в различные проекты. Она становится своеобразным руководством, позволяющим разработчикам быстро освоить функциональность модуля.

В процессе разработки были проведены тесты, которые подтвердили корректную работу каждой функции модуля. Тестирование является важной частью разработки, обеспечивая стабильность и надежность кода. Таким образом, созданный модуль представляет собой надежный инструмент для работы с датами в различных проектах, а его документация обеспечивает ясное понимание и простоту использования для других разработчиков.