

МІНІСТЕРСТВО ОСВІТИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені Ігоря Сікорського»
НАВЧАЛЬНО-НАУКОВИЙ КОМПЛЕКС
«ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ»
Кафедра Системного проектування

Контрольна робота №6
з дисципліни: “Проектування інформаційних систем”
на тему “Система автоматичного створення довідника користувача
та оформлення коду за допомогою Coding Convention”

Виконав:
Студент 4 курсу
групи ДА-71
Задорожний А.С.

Мета роботи: за допомогою системи генерації довідника користувача створити документ у форматі PDF і HTML для архітектурної програмної моделі.

Завдання:

1. Для кожного з класів API в код програми додати теги з описом керівництва користувача для архітектурної програмної моделі.
2. Обрати Coding Convention. Оформити код програми згідно Coding Convention.
3. Встановити налаштування системи автоматичного створення керівництва користувача
4. Згенерувати HTML документацію керівництва користувача.
5. Згенерувати PDF документацію керівництва користувача.

Хід виконання лабораторної роботи:

1. Розробимо тестовий приклад програми з використанням PEP Code convention та з коментарями для автоматичної документації:

```
class Matrix:
    """
    A simple matrix class
    """
    def __init__(self, rows, columns):
        """
        Construct a new 'Matrix' object.

        :param rows: Number of rows in a matrix
        :param columns: Number of columns in a matrix
        :return: returns nothing
        """
        self.rows = rows
        self.columns = columns
        self.data = [[0 for i in range(columns)] for j in range(rows)]

    def __add__(self, other):
        """
        Create a new 'Matrix' object with matrix addition of matrices with
        same dimensions

        :param other: A second addition term matrix
        :return: A new matrix
        """
        if self.rows == other.rows and self.columns == other.columns:
            result_matrix = Matrix(self.rows, self.columns)
            result_matrix.data = [[self.data[j][i] + other.data[j][i] for i
in range(self.columns)] for j in range(self.rows)]
            return result_matrix
        else:
            return None

    def __sub__(self, other):
        """
```

```

        Create a new 'Matrix' object with matrix subtraction of matrices with
        same dimensions

        :param other: A subtrahend matrix
        :return: A new matrix
        """
        if self.rows == other.rows and self.columns == other.columns:
            result_matrix = Matrix(self.rows, self.columns)
            result_matrix.data = [[self.data[j][i] - other.data[j][i] for i
in range(self.columns)] for j in range(self.rows)]
            return result_matrix
        return None

    def transpose(self):
        """
        Create a new 'Matrix' object using matrix transposition

        :return: A new transposed matrix
        """
        result_matrix = Matrix(self.rows, self.columns)
        result_matrix.data = [[self.data[i][j] for i in range(self.rows)] for
j in
                                range(self.columns)]
        return result_matrix

    def __getitem__(self, pos):
        """
        Create a new 'Matrix' object with matrix subtraction of matrices with
        same dimensions

        :param pos: A tuple in the form (rows, columns), which contains
        element indices
        :return: An element of matrix
        """
        return self.data[pos[0]][pos[1]]

```

2. Сгенеруємо автоматично документацію за допомогою pyDoc:0

Classes

[builtins.object](#)

[Matrix](#)

```
class Matrix(builtins.object)
```

```
    Matrix(rows, columns)
```

```
    A simple matrix class
```

Methods defined here:

```
    __add__(self, other)
```

```
        Create a new 'Matrix' object with matrix addition of matrices with same dimensions
```

```
        :param other: A second addition term matrix
```

```
        :return: A new matrix
```

```
    __getitem__(self, pos)
```

```
        Create a new 'Matrix' object with matrix subtraction of matrices with same dimensions
```

```
        :param pos: A tuple in the form (rows, columns), which contains element indices
```

```
        :return: An element of matrix
```

```
    __init__(self, rows, columns)
```

```
        Construct a new 'Matrix' object.
```

```
        :param rows: Number of rows in a matrix
```

```
        :param columns: Number of columns in a matrix
```

```
        :return: returns nothing
```

```
    __sub__(self, other)
```

```
        Create a new 'Matrix' object with matrix subtraction of matrices with same dimensions
```

```
        :param other: A subtrahend matrix
```

```
        :return: A new matrix
```

```
    transpose(self)
```

```
        Create a new 'Matrix' object using matrix transposition
```

```
        :return: A new transposed matrix
```

Висновки: в даній лабораторній роботі було розглянуто методи створення автоматичної документації, зокрема з використанням ruDoc для мови програмування Python, також код програми був оформлений згідно з конвенцією PEP 8.