

Uniwersytet Śląski w Katowicach		Wydział Nauk Ścisłych i Technicznych			
		Instytut Fizyki			
		Rok	III	Semestr	V
Kierunek	Informatyka stosowana				
Przedmiot	SiNWO - laboratorium				
Prowadzący	dr Wojciech Gurdziel				
Tytuł ćwiczenia	Automatyzacja generacji dokumentacji			Nr ćwiczenia	IV
Sprawozdanie wykonał: (Imię i Nazwisko)	Jakub Kraus				
Data wykonania ćwiczenia	24.01.2024	Data oddania sprawozdania		01.02.2024	

Spis treści

1	Cel ćwiczenia	3
2	Przebieg ćwiczenia	3
2.1	Kod użyty do dokumentacji	3
2.2	Używanie wizarda doxygen	5
2.3	Dokumentacja	7
3	Wnioski	8
	Bibliografia	8

Spis rysunków

1	Okno wizarda doxygen	5
2	Generacja plików dokumentacji	5
3	Pliki wygenerowane przez doxygen	6
4	Dokumentacja wygenerowana przez doxygen	7
5	Metody klasy Queue w dokumentacji wygenerowanej przez doxygen	7

Listings

1	Kod Python	3
---	----------------------	---

1 Cel ćwiczenia

Celem ćwiczenia jest zapoznanie się z narzędziem doxygen, które służy do automatyzacji generacji dokumentacji. Doxygen jest narzędziem, które pozwala na generowanie dokumentacji w formacie HTML, LaTeX, RTF oraz PostScript.

2 Przebieg ćwiczenia

Doxygen jest narzędziem, które jest dostępne na wielu platformach, w tym również na Windowsie 11, którego użyłem do wykonania tego zadania. Na potrzeby zadania wykorzystałem kod Python do tworzenia kolejek FIFO.

2.1 Kod użyty do dokumentacji

```
class Queue:
    """Klasa reprezentująca kolejkę.

    Kolejka to struktura danych,
    która działa na zasadzie "first in, first out" (FIFO).
    Elementy są dodawane na końcu kolejki
    i usuwane z początku kolejki.

    Attributes:
        items (list): Lista przechowująca elementy kolejki.

    Methods:
        enqueue(item): Dodaje element na koniec kolejki.
        dequeue(): Usuwa i zwraca element z początku kolejki.
        is_empty(): Sprawdza, czy kolejka jest pusta.
        size(): Zwraca liczbę elementów w kolejce.
    """

    def __init__(self):
        """Inicjalizuje pustą kolejkę."""
        self.items = []

    def enqueue(self, item):
        """Dodaje element na koniec kolejki.

        Args:
            item: Element do dodania do kolejki.
        """
        self.items.append(item)

    def dequeue(self):
        """Usuwa i zwraca element z początku kolejki.

        Returns:
            Element z początku kolejki.
        """
```

```

    if not self.is_empty():
        return self.items.pop(0)
    else:
        raise IndexError("Dequeue_from_an_empty_queue")

def is_empty(self):
    """Sprawdza, czy kolejka jest pusta.

    Returns:
        True, jeśli kolejka jest pusta;
        False w przeciwnym razie.
    """
    return len(self.items) == 0

def size(self):
    """Zwraca liczbę elementów w kolejce.

    Returns:
        Liczba elementów w kolejce.
    """
    return len(self.items)

# Przykłady użycia
if __name__ == "__main__":
    my_queue = Queue()

    my_queue.enqueue(1)
    my_queue.enqueue(2)
    my_queue.enqueue(3)

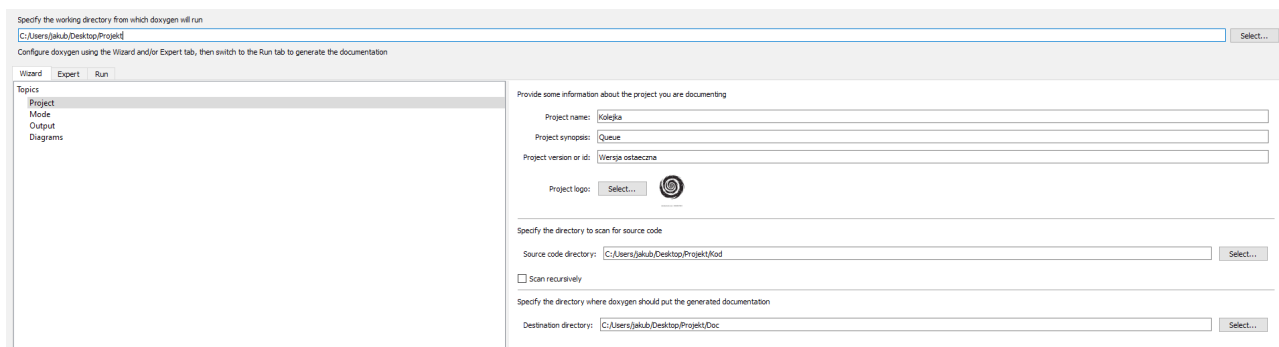
    print("Size_of_the_queue:", my_queue.size())
    print("Dequeue:", my_queue.dequeue())
    print("Is_the_queue_empty?", my_queue.is_empty())

```

Listing 1: Kod Python

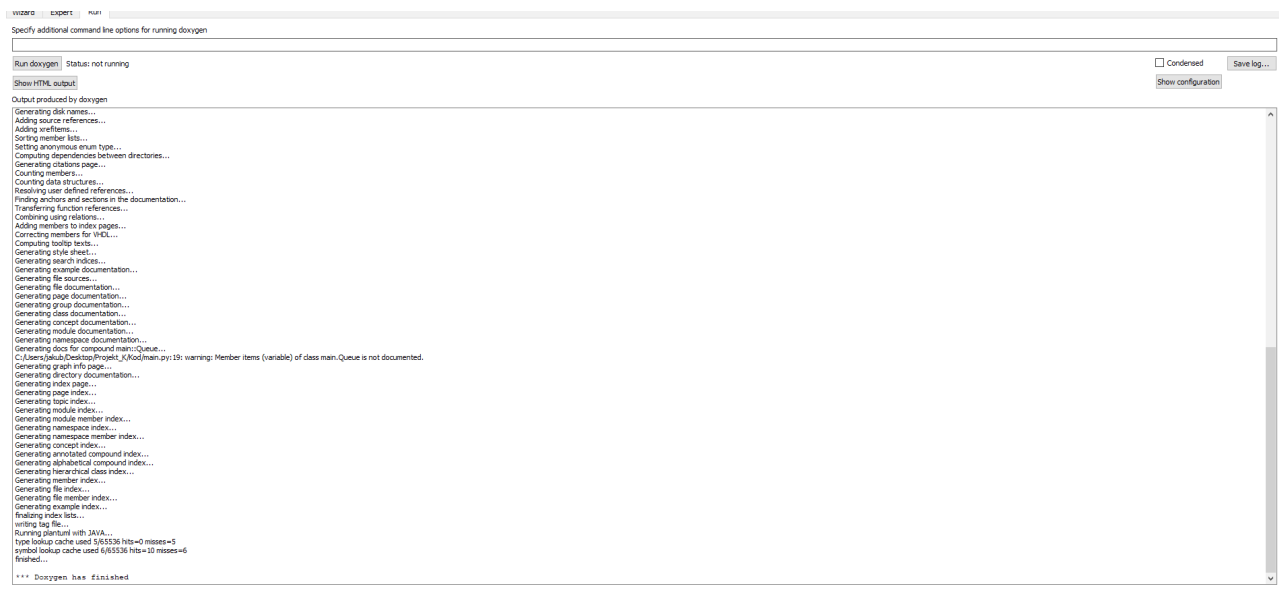
2.2 Używanie wizarda doxygen

Doxygen posiada wbudowany wizar, który pozwala na wygenerowanie pliku konfiguracyjnego oraz na wygenerowanie dokumentacji.



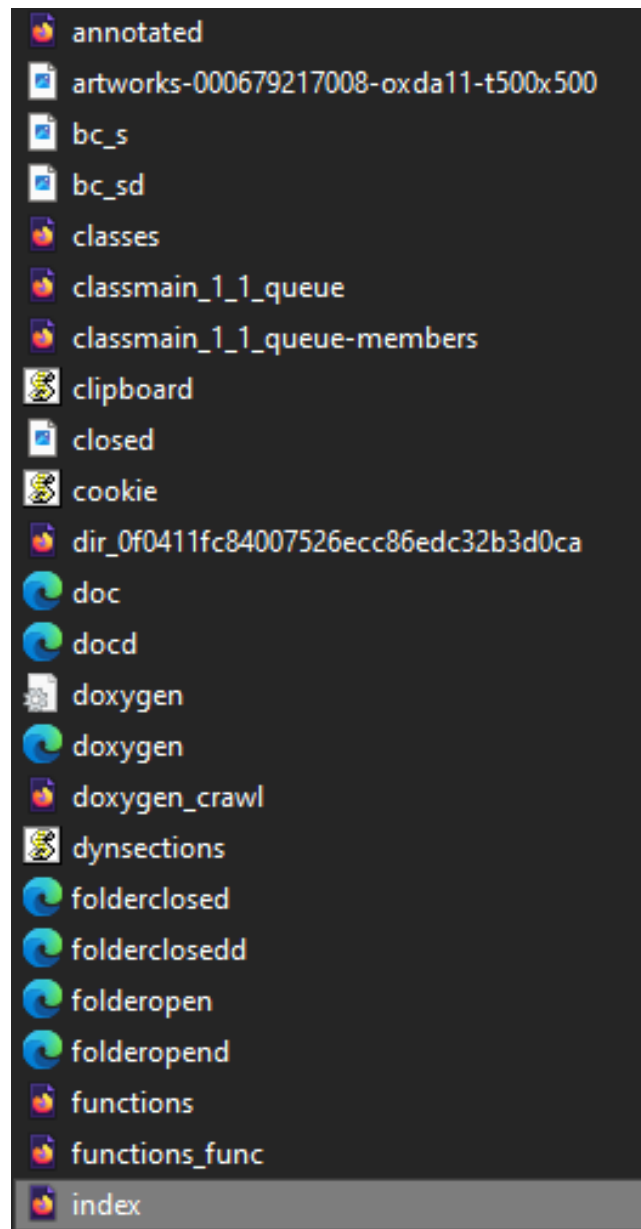
Rysunek 1: Okno wizarda doxygen

Dzięki wizardowi doxygen, można wygenerować plik konfiguracyjny, który jest niezbędny do wygenerowania dokumentacji. Ułatwia to pracę z doxygenem, a co za tym idzie pozwala na krótszy czas pracy potrzebny do stworzenia dokumentacji.



Rysunek 2: Generacja plików dokumentacji

Doxygen szybko generuje pliki dokumentacji, które współpracują z każdym systemem operacyjnym. W tym przypadku są to pliki HTML, które można otworzyć w dowolnej przeglądarce internetowej.



Rysunek 3: Pliki wygenerowane przez doxygen

Aby zobaczyć dokumentację, wystarczy otworzyć plik `index.html`. W tym przypadku dokumentacja została wygenerowana w języku angielskim, jednak doxygen wspiera wiele języków, w tym również polski; który niestety jest aktualnie “unmaintained”.

2.3 Dokumentacja

Jak widać na rysunku 4, dokumentacja została wygenerowana poprawnie, jest czytelna oraz zawiera wszystko co wprowadziliśmy jako input do wizarda.



Rysunek 4: Dokumentacja wygenerowana przez doxygen



Rysunek 5: Metody klasy Queue w dokumentacji wygenerowanej przez doxygen

3 Wnioski

Po zapoznaniu się z Doxygenem, można stwierdzić, że narzędzie to jest skutecznym rozwiązaniem do automatycznej generacji dokumentacji kodu. Jego integracja z kodem źródłowym oraz możliwość generacji różnych formatów dokumentacji sprawiają, że jest to praktyczne narzędzie dla programistów. Bogate opcje konfiguracyjne oraz wsparcie dla diagramów klas dodają elastyczności i usprawniają proces tworzenia czytelnej dokumentacji.

Nie zmienia to jednak faktu, że doxygen posiada wady. Jedną z nich jest fakt braku wsparcia dla języka polskiego, który jednakże może ulec zmianie. Doxygen również, zdaniem wokalnej części programistów, wymaga “zaśmiecenia” kodu źródłowego komentarzami, które są niezbędne do wygenerowania dokumentacji.

Wnioskując, Doxygen jest wartościowym narzędziem dla projektów wymagających zwięzłej i aktualnej dokumentacji kodu, które są w stanie pominąć jego wady.

Bibliografia

- [1] *Doxygen homepage*. URL: <https://www.doxygen.nl/index.html> (term. wiz. 23.01.2024).
- [2] Microsoft. pl-pl. URL: <https://www.microsoft.com/pl-pl/windows> (term. wiz. 23.01.2024).
- [3] *My Project: Internationalization*. URL: <https://www.doxygen.nl/manual/langhowto.html> (term. wiz. 23.01.2024).