Лабораторна робота №9

Тема: Doxygen.

Мета: Навчитися працювати з системою Doxygen.

Індивідуальне завдання

- 1. Створити документацію Doxygen.
- 2. Додати можливість генерації Doxygen документації в Makefile.

Хід роботи

1. Створив документацію Doxygen. Вміст показано на рисунку.

```
1 ## Common Project info
 2 PROJECT NAME = "lab05"
 3 PROJECT BRIEF = "Демонстрація документування коду засобами `doxygen`"
 4 PROJECT_NUMBER = 0.1
 6 ## output options
7 OUTPUT_DIRECTORY = ./dist
 8 OUTPUT_LANGUAGE = Ukrainian
9 DOXYFILE_ENCODING = UTF-8
11 ## input options
12 INPUT_ENCODING = UTF-8
13 RECURSIVE = YES
14 FILE_PATTERNS = *.c *.md
15 ##########
16 #USE_MDFILE_AS_MAINPAGE = README.md
17 ###########
18
19
20
21 ## processing options
22 EXTRACT ALL = YES
23 EXTRACT PRIVATE = YES
24 EXTRACT_STATIC = YES
25 JAVADOC AUTOBRIEF = YES
26 OPTIMIZE OUTPUT FOR C = YES
```

Рисунок 1 – Перша половина Doxygen

```
27
28 ## Draw calls graph + object relations (for cpp)
29 HAVE DOT = YES
30 DOT_PATH = /usr/local/bin/dot
31 UML_LOOK = YES
32 TEMPLATE_RELATIONS = YES
33 CALL_GRAPH = YES
35 ## Customized generate options
36 GENERATE_HTML = YES
37 GENERATE_LATEX = YES
38 # COMPACT_LATEX = YES
39
40 ## Default generate options
41 GENERATE_DOCSET = NO
42 GENERATE_HTMLHELP = NO
43 GENERATE_CHI = NO
44 GENERATE_QHP = NO
45 GENERATE_ECLIPSEHELP = NO
46 GENERATE_RTF = NO
47 GENERATE_MAN = NO
48 GENERATE_XML = NO
49 GENERATE_DOCBOOK = NO
50 GENERATE_AUTOGEN_DEF = NO
51 GENERATE_PERLMOD = NO
```

Рисунок 2 – Друга половина Doxygen

2. Написав коментарі у коді програми.

```
1 #include <stdio.h>
 3 * @file main.c виконати завдання 1
 5 * порівняти суму лівої і правої половини 6-значного числа
 6 */
7
 9 * виконання завдання за допомогою циклу
10 * @param num - б-знане число;
11 * @param lsum - сума лівої частини 6-значного числа;
12 * @param гsum - сума правої частини 6-значного числа;
13 */
14
15 /**
16 ^{\circ} Функція main: Порівнює сумму лівої і правої частини 6-значного числа. 17 */
18 int main()
19 {
           int num;
int lsum = 0, rsum = 0;
20
21
22
           printf("Enter a 6-digit number : ");
scanf("%d", &num);
printf("\n");
23
24
25
26
27
           for (int i = 0, _num = num; i < 6; i++)</pre>
28
29
                    if (i < 3) (lsum += _num % 10);</pre>
30
                           (rsum += _num % 10);
31
32
                    _num /= 10;
           }
33
34
            if (lsum == rsum)
35
36
           {
37
                    printf("\n\t\t\t\t
                                          TICKET %d IS HAPPY ! ! \n", num);
38
           }
39
           else
40
           {
                    printf("\n\t\t\t TICKET %d IS UNHAPPY :(\n", num);
41
43
44
           return 0;
45 3
```

Рисунок 3 – Код програми

3. Додав можливість генерації Doxygen документації в Makefile. Для цього додав ціль format у Makefile. На рисунку показано вміст Makefile.

```
1 targets = main.bin
3 C_OPTS = -std=gnu11 -g -Wall -Wextra -Wformat-security -Wfloat-equal -Wshadow -Wconversion -Wlogical-not-parentheses -Wnull-dereference
5 all: clean prep compile run
6
7 clean:
          rm -rf dist
9 ргер:
          mkdir dist
10
11 format:
          doxygen Doxyfile
13 compile: main.bin
14
15 main.bin: src/main.c
          $(CC) $(C_OPTS) $< -o ./dist/$@
17 run: clean prep compile
18
        ./dist/main.bin
19 format: # TODO
21 tidy: # TODO
```

Рисунок 4 – Makefile

4. На рисунку показано результат роботи.

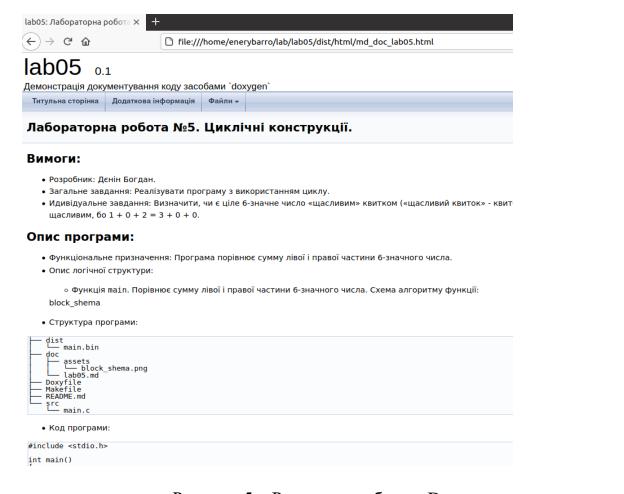


Рисунок 5 – Результат роботи с Doxygen

Висновок: Навчився працювати з системою Doxygen.