

Тема: Doxygen.

Мета: Навчитись працювати з системою Doxygen.

Індивідуальне завдання

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

Хід роботи:

- 1) Створили документацію Doxygen. Вміст показано на рисунку.

```
1  ## Common Project info
2  PROJECT_NAME = "lab08"
3  PROJECT_BRIEF = "Демонстрація документування коду засобами `doxygen`"
4  PROJECT_NUMBER = 0.1
5
6  ## output options
7  OUTPUT_DIRECTORY = ./dist
8  OUTPUT_LANGUAGE = Ukrainian
9  DOXYFILE_ENCODING = UTF-8
10
11 ## input options
12 INPUT_ENCODING = UTF-8
13 RECURSIVE = YES
14 FILE_PATTERNS = *.c *.h *.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
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
34
35 ## Customized generate options
```

Рисунок 1 – Перша частина Doxygen

```

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 частина Doxygen

2) Написали коментарі у коді програми.

```

1  #include <stdio.h>
2  /**
3   * @file main.c
4   * порівняти суму лівої і правої половини 6-значного числа
5   */
6
7  /**
8   * виконання завдання за допомогою циклу
9   * @param num - 6-значне число;
10  * @param num - сума лівої частини 6-значного числа;
11  * @param num - сума правої частини 6-значного числа;
12  */
13
14  /**
15   * Функція main: Порівнює суму лівої і правої частини 6-значного числа.
16   */
17
18  int main()
19  {
20      int num;
21      int lsum = 0, rsum = 0;
22
23      printf("Enter a 6-digit number : ");
24      scanf_s("%d", &num);
25      printf("\n");
26
27      for (int i = 0, num1 = num; i < 6; i++)
28      {
29          if (i < 3) (lsum += num1 % 10);
30          else (rsum += num1 % 10);
31
32          num1 /= 10;
33      }
34
35      if (lsum == rsum)
36      {
37          printf("\n\t\t\t\t\t TICKET %d IS HAPPY ! ! !\n", num);

```

Рисунок 3 – Код програми

3) Додали можливість генерації Doxygen документації в Makefile. Для цього додали ціль format у Makefile. На рисунку показано вміст Makefile.

```
1 targets = main.bin
2 CC = gcc
3 C_OPTS = -std-gnu11 -g -Wall -Wextra -Werror -Wformat-security -Wfloat-equal -Wshadow -Wconversion -Wlogical-not-parentheses -Wnull-dereference
4
5 all: clean prep compile run
6
7 clean:
8     rm -rf dist
9 prep:
10     mkdir dist
11 format:
12     doxygen Doxyfile
13 compile: main.bin
14
15 main.bin: src/main.c
16     $(CC) $(C_OPTS) $< -o ./dist/$@
17 run: clean prep compile
18     ./dist/main.bin
19 format: #TODO
20
21 tidy: # TODO
```

Рисунок 4 – Makefile

4) На рисунку показано результат роботи.

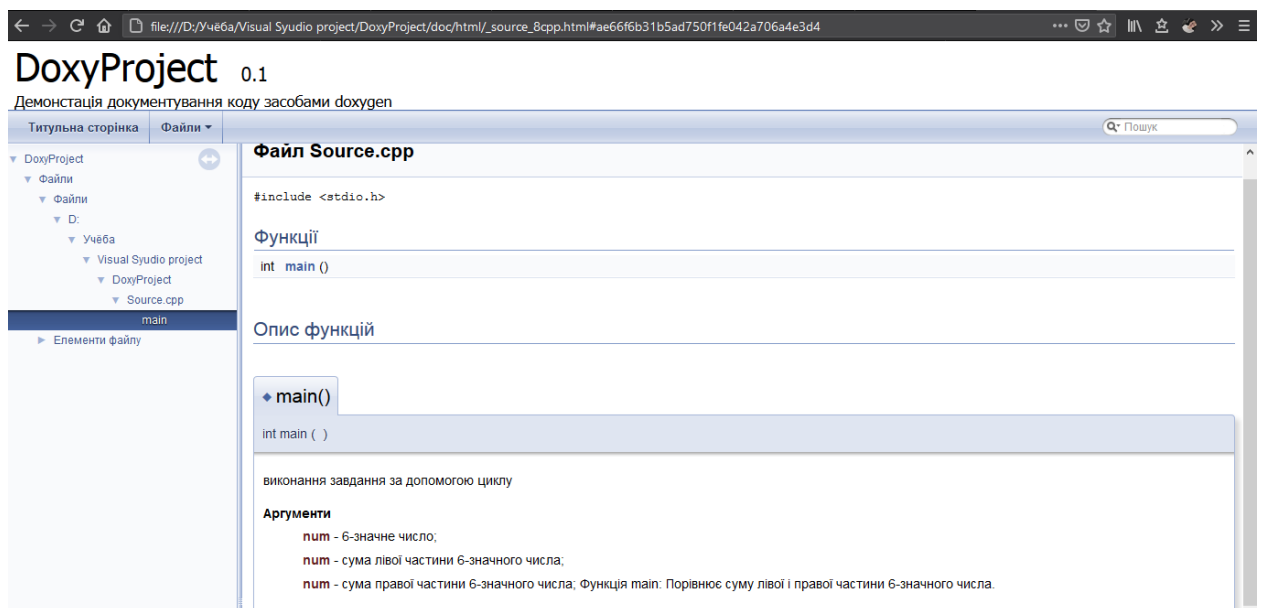


Рисунок 5 – Результат роботи програми

Висновок: Навчилися працювати с системою Doxygen.

github: <https://github.com/Nikita-Stetsenko/repository>

