

Лабораторна робота № 15, 17. Структовані типи даних та модульні тести

1 ВИМОГИ

1.1 Розробник

- Радєвич Владислав Романович;
- студент групи КІТ – 320;
- 09.03.2021 р.

1.2 Загальне завдання

З розділу "Розрахункове завдання / Індивідуальні завдання" взяти прикладну галузь, стосовно номеру варіанту за попередньо-визначеною формулою (номер 23) та створити структуру, що відображає "базовий клас". Та розробити модульні тести.

1.3 Індивідуальне завдання

Розробити програму, умови якої надано у лабораторному практикуму. Мною було взято умови для розробки з розділу на оцінку «добре».

2 ОПИС ПРОГРАМИ

2.1 Функціональне призначення

Програма призначена для сортування структури даних заданих птахів з файлу та виведення результату у файл або на екран.

2.2 Опис логічної структури

2.2.1 Основна функція

`int main`

Призначення: головна функція.

Схема алгоритму функції подана на рис. 1.

Опис роботи: демонструє роботу заданої структури та методів оперування ним, викликаючи функція для роботи з структурою.

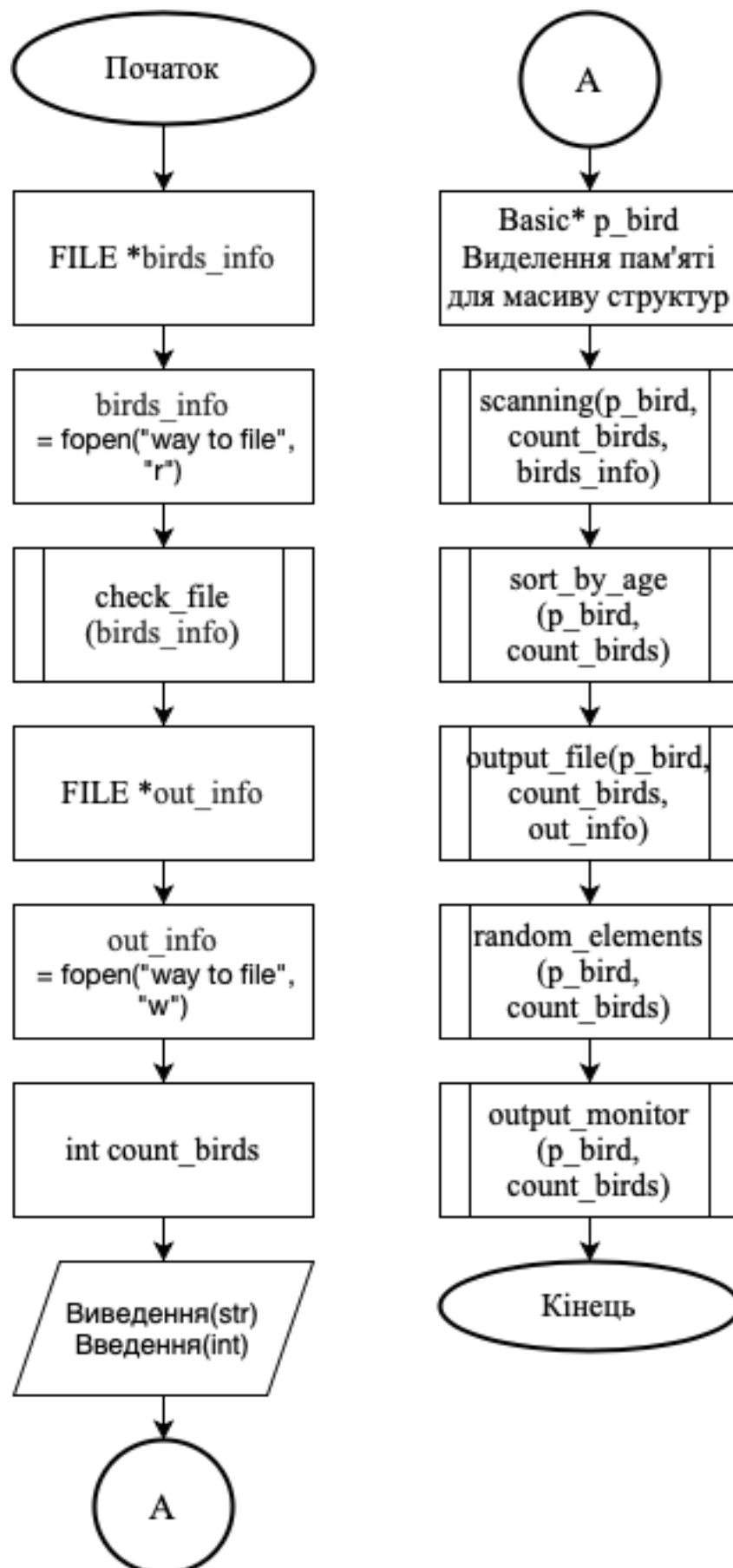


Рисунок 1 – Схема алгоритму функції main

2.2.2 Функція перевірки файлу

```
void check_file (FILE *fmatrix);
```

Призначення: перевірка місцезнаходження файлу по вказаному шляху.

Схема алгоритму функції подана на рис. 2

Опис роботи: функція перевіряє знаходження файлу з даними для структурами по заданій директорії.

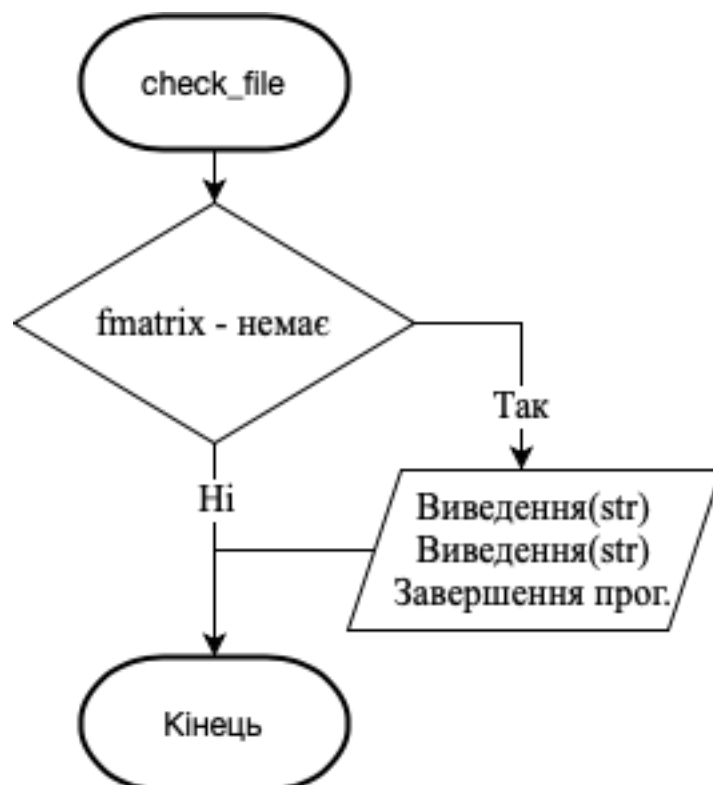


Рисунок 2 – Схема алгоритму функції `check_file`

2.2.3 Функція сканування даних

```
void scanning (Basic* p_bird, int count_birds, FILE *birds_info);
```

Призначення: сканування данні з файлу, в якому заздалегідь написані данні для структури.

Схема алгоритму функції подана на рис. 3.

Опис роботи: функція сканує та переносить данні з заданого файлу в задану структуру за допомогою циклу.

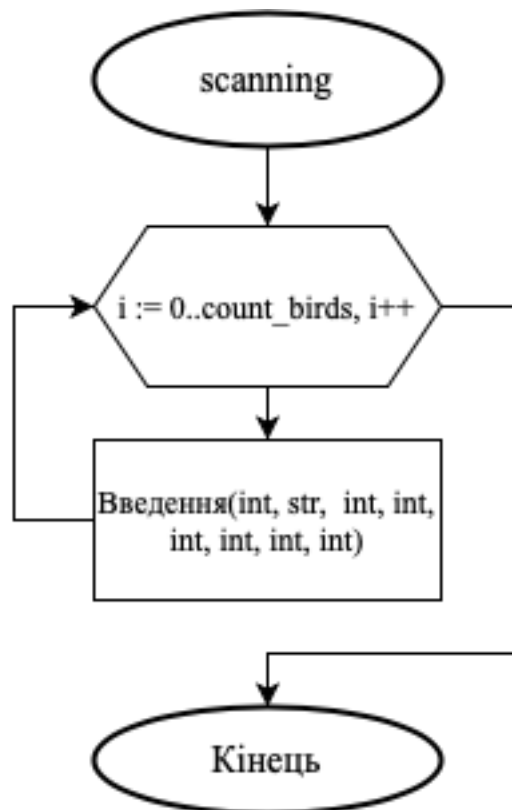


Рисунок 3 – Схема scanning

2.2.4 Функція сортування

```
void sort_by_age (Basic* p_bird, int count_birds);
```

Призначення: сортування масиву з структурами у порядку зростання.

Схема алгоритму функції подана на рис. 4

Опис роботи: сортування масиву структур за певним критерієм, в даному випадку - за віком. Сортування відбувається методом «бульбашки».

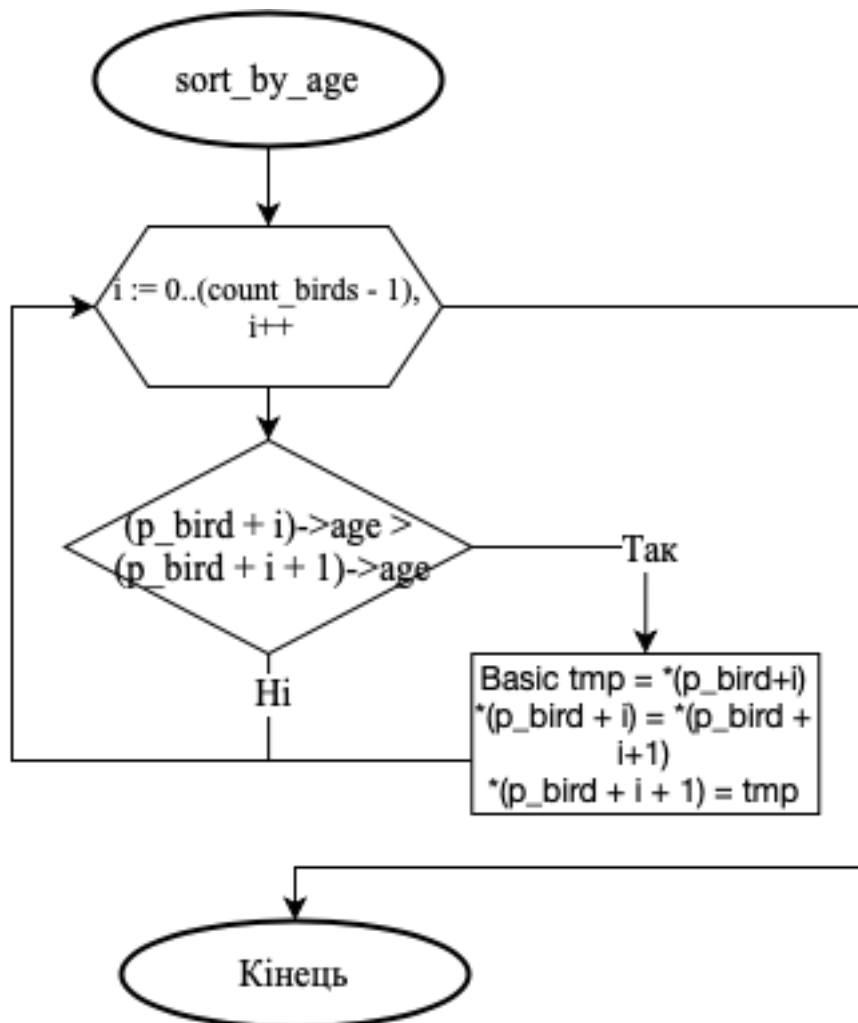


Рисунок 4 – схема алгоритму функції sort_by_age

2.2.5 Функція виводу у файл

```
void output_file (Basic* p_bird, int count_birds, FILE* out_info);
```

Призначення: вивід даних у файл за певної директорією.

Схема алгоритму функції подана на рис. 5

Опис роботи: оброблення усіх полів структури та вивід усіх даних, що містяться в усіх структурах в певний файл.

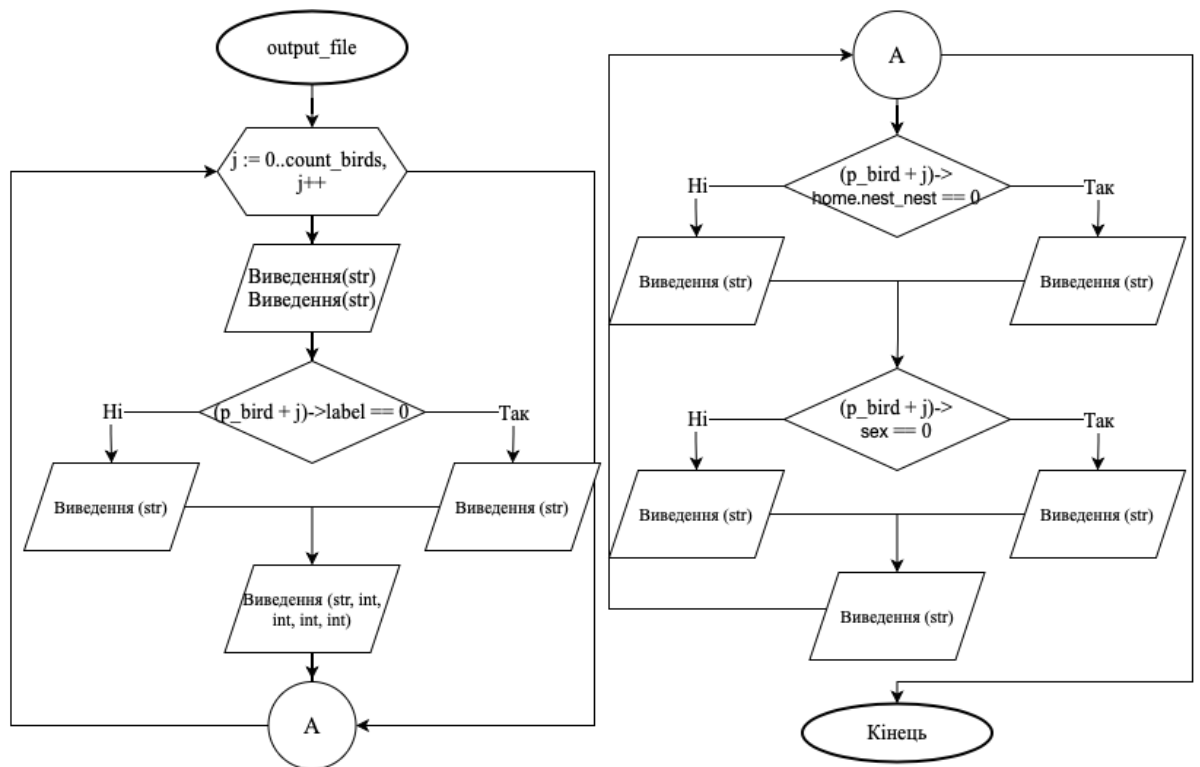


Рисунок 5 – схема алгоритму функції output_file

2.2.6 Функція створення даних про птиць

```
void random_elements (Basic* p_bird, int count_birds);
```

Призначення: створення випадкових даних (з певним обмеженням) для елементів прикладної галузі структури.

Схема алгоритму функції подана на рис. 6

Опис роботи: присвоєння випадкових чисел для елементів прикладної галузі масиву структури у діапазоні від 23 до 230.



Рисунок 6 – схема алгоритму функції random_elements

2.2.7 Функція даних виводу на екран

```
void output_monitor (Basic* p_bird, int count_birds);
```

Призначення: вивід окремих даних кожної структури з масиву структур на екран.

Схема алгоритму функції подана на рис. 7

Опис роботи: оброблення та вивід усіх даних кожної структури з зазначеного масиву структур.

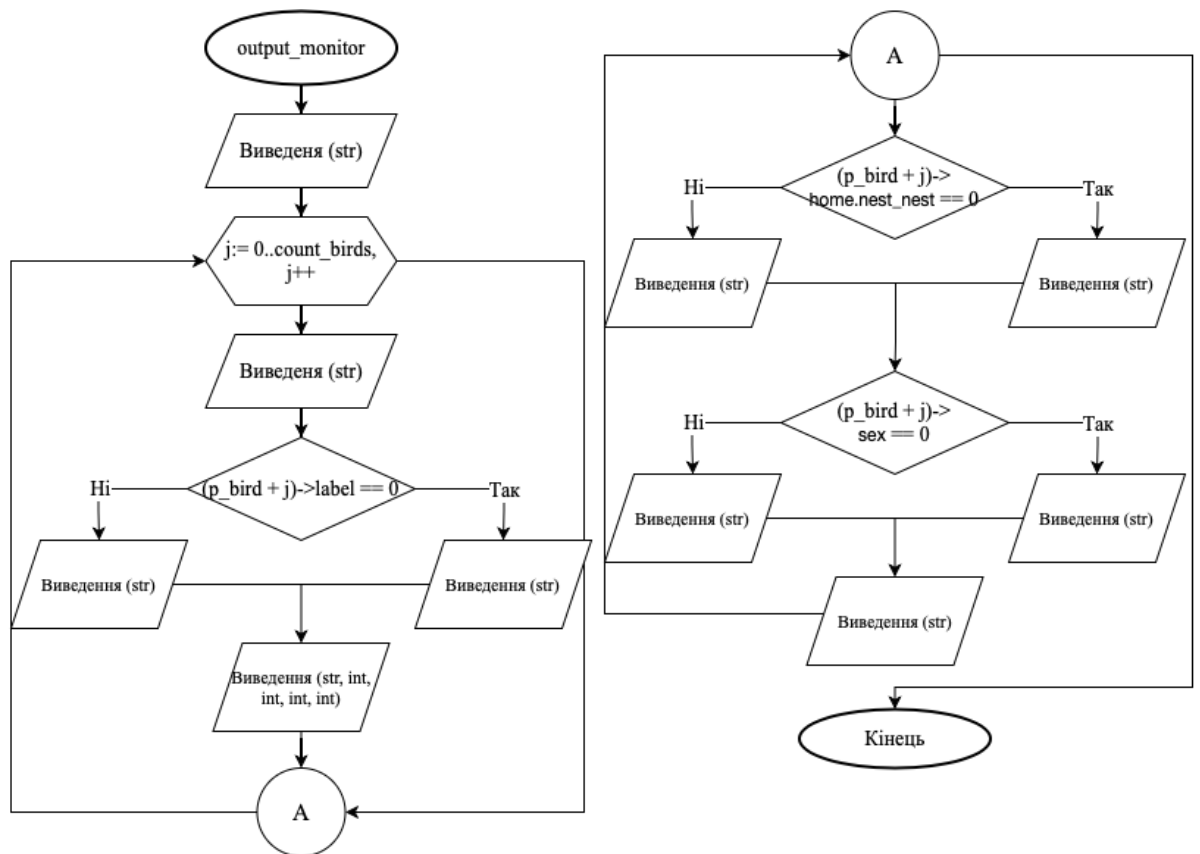


Рисунок 7 – схема алгоритму функції output_monitor

2.3 Структура проекту

```

.
├── Doxyfile
├── Makefile
├── README.md
├── doc
│   ├── Radievych15_17. pdf
│   ├── assets
│   │   ├── flowchart\ output_file.png
│   │   ├── flowchart\ sort_by_age.png
│   │   ├── flowchart_check_file.png
│   │   ├── flowchart_main.png
│   │   ├── flowchart_output_monitor.png
│   │   ├── flowchart_random_elements.png
│   │   └── flowchart_scanning.png
│   └── ~$dievych15_17.docx
├── src
│   ├── lib.c
│   ├── lib.h
│   └── main.c
└── test
    └── test.c

```

3 ВАРІАНТИ ВИКОРИСТАННЯ

Цю програму можна використовувати за для перепису усіх зареєстрованих птахів в окремий файл на комп'ютері, заздалегідь давши про них певну інформацію.

Результат роботи з doxygen продемонстровано на рисунку 8, рисунку 9 та рисунку 10 тавиконання модульних тестів на рисунку 11 та 12.

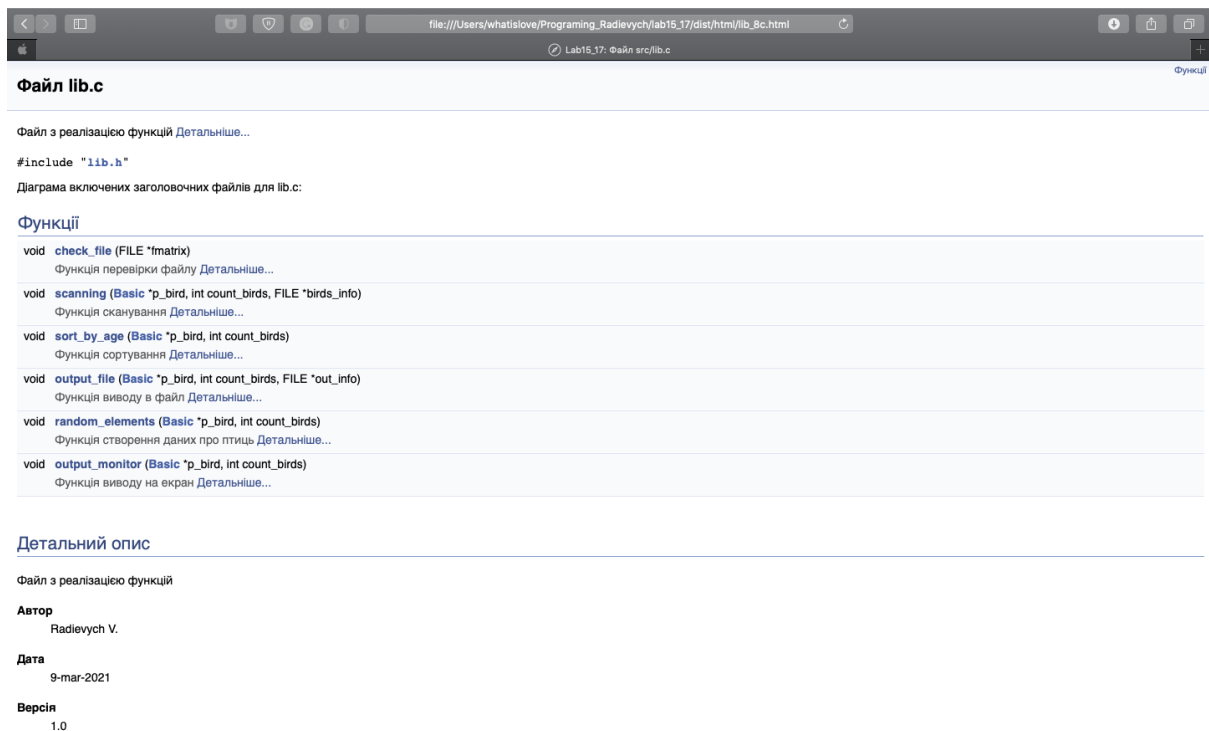


Рисунок 8 – робота з doxygen

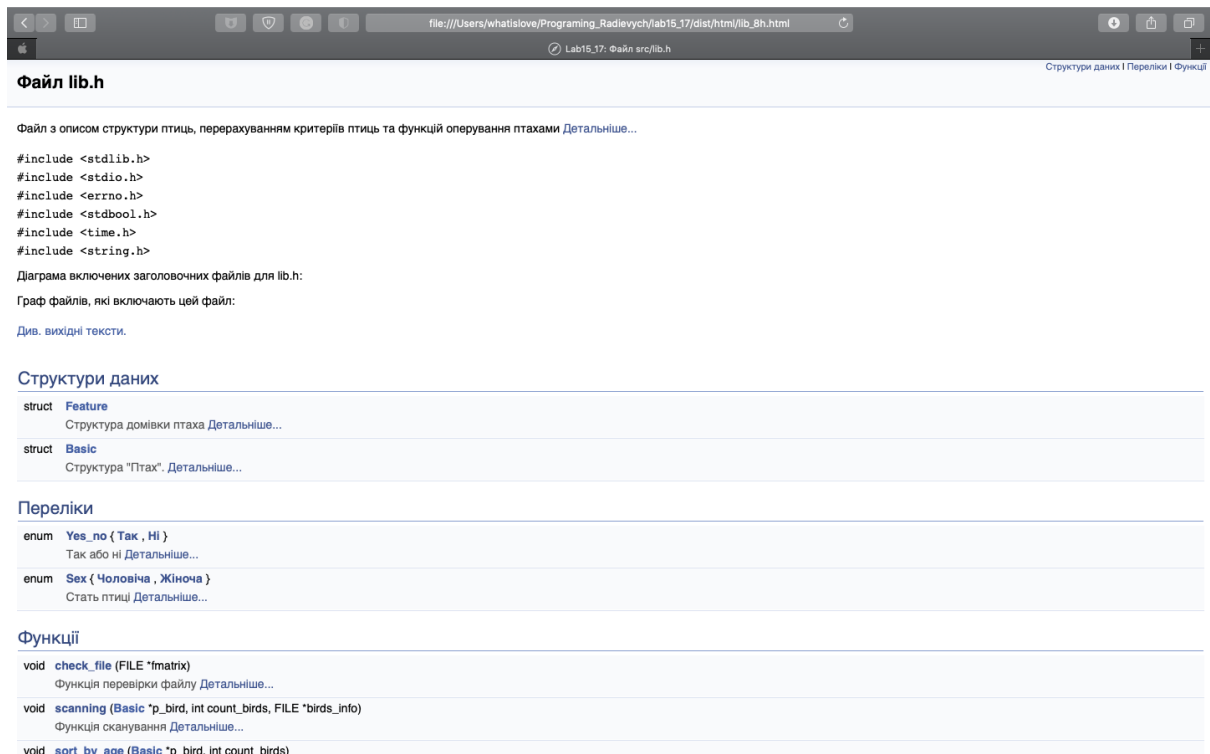


Рисунок 9 – робота з doxygen

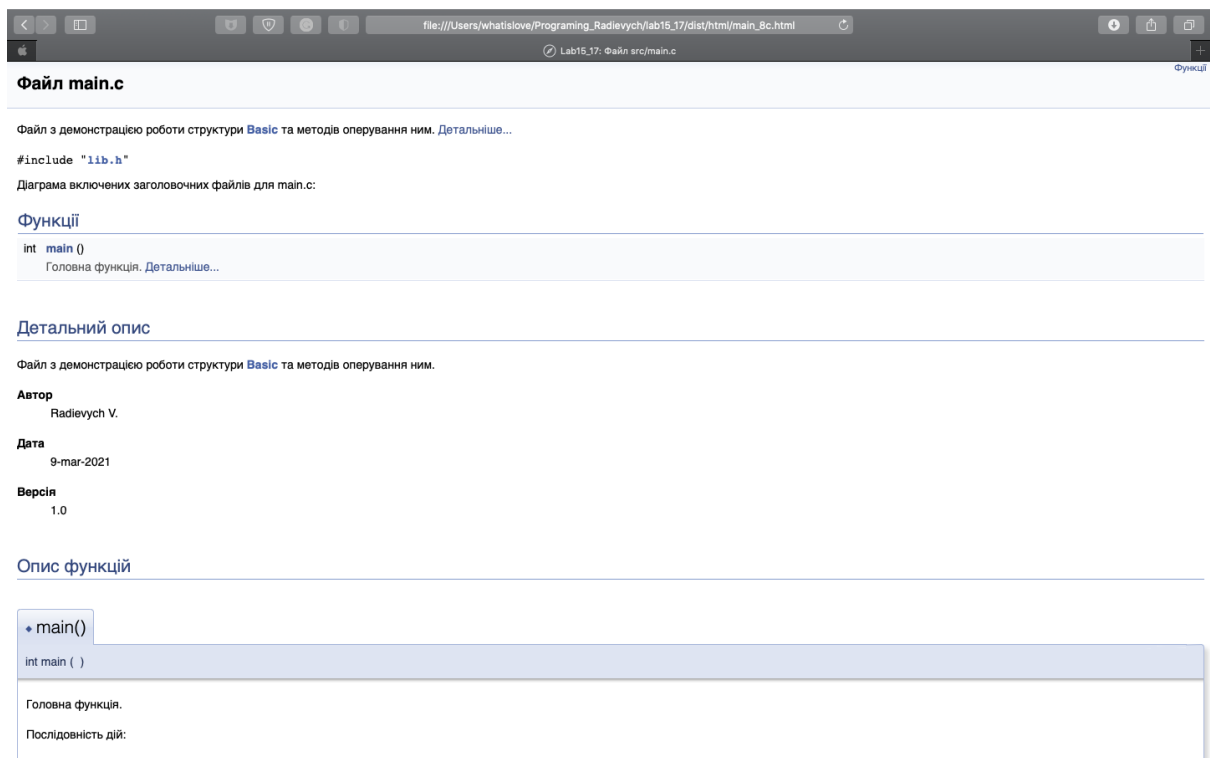


Рисунок 10 – робота з doxygen

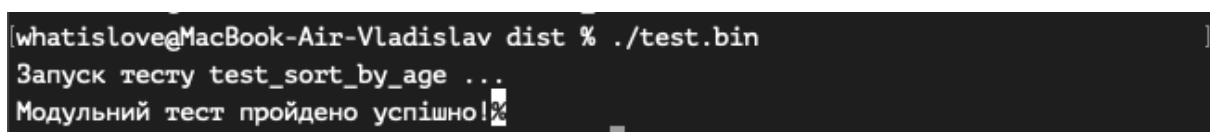


Рисунок 11 – робота з модульними тестами

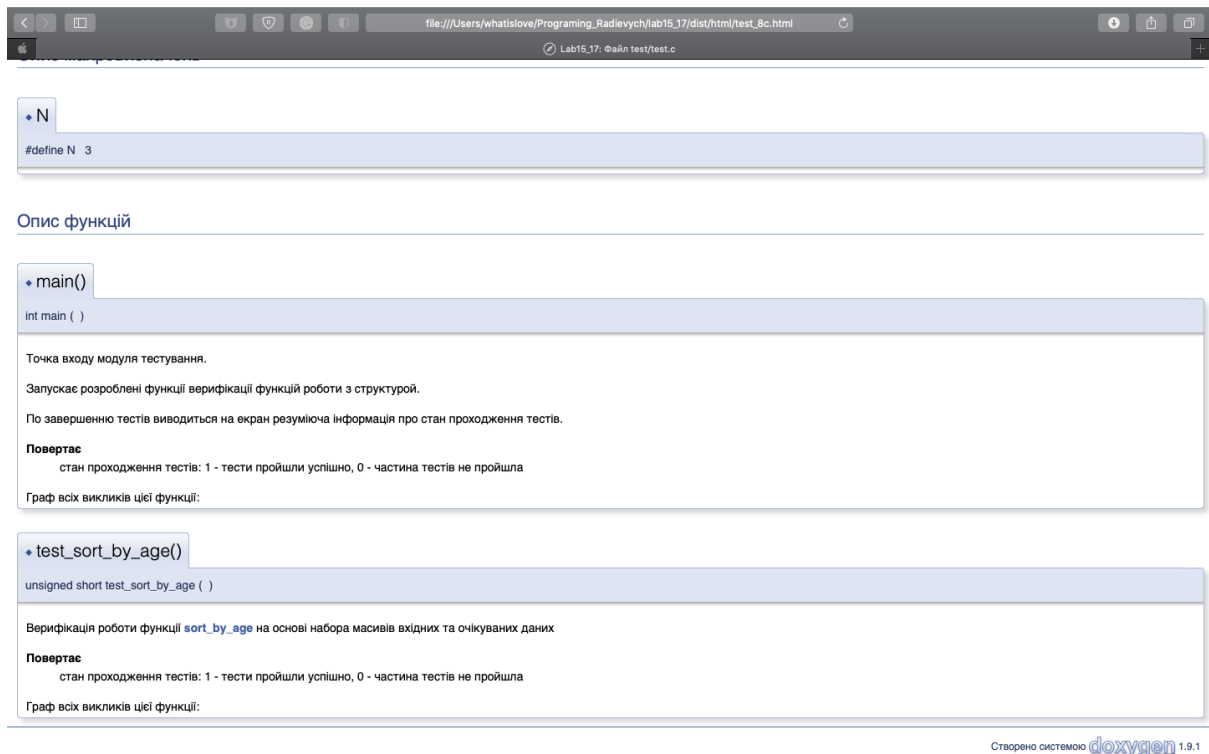


Рисунок 12 – робота з модульними тестами

4 ВИСНОВОК

При виконанні даної лабораторної роботи я закріпив набуті мною навички, створення та взаємодію з структурами, а також розробки модульних тестів.

Посилання на GitHub, де знаходяться усі програми:
https://github.com/KotKHPI/Programming_Radievych