

# *Звіт-ретроспектива до Лабораторної роботи №3*

*Карпишин О., К-28*

Відгук про бібліотеки

Бібліотеки, що використовувалися

- zlib (<http://www.zlib.net>)
- libPNG (<http://www.libpng.org>)
- libHARU (<http://libharu.org>)
- nlohmann/json (<https://github.com/nlohmann/json>)

## **nlohmann/json**

1. Які конкретні задачі планували вирішувати за допомогою цієї бібліотеки?

Дана бібліотека є хорошим засобом для читання і генерування JSON файлів.

2. Чому було обрано саме цю бібліотеку, а не аналоги?

Я обрала цю бібліотеку, адже вона використовує сучасну C++ і являється певною ідеєю імплементації роботи з JSON в стандартну C++.

3. Наскільки просто та зрозуміло було отримати, встановити, налаштувати та почати використовувати цю бібліотеку?

Отримати, встановити і налаштувати і почати бібліотеку було досить легко.

4. Наскільки зрозумілою та корисною була документація бібліотеки?

Документація бібліотеки зручна і зрозуміла.

5. Наскільки було зрозуміло, як саме використовувати бібліотеку, які класи/методи/функції використовувати для вирішення поставлених задач?

Бібліотека імплементує тип даних `json`, який за допомогою фігурних дужок можна ініціалізувати як звичайний масив. Але творець бібліотеки надає можливість створювати змінну типу `json` зразу з масивами.

Приклад =>

```
json j = {
    {"pi", 3.141},
    {"happy", true},
    {"name", "Niels"},
    {"nothing", nullptr},
    {"answer", {
        {"everything", 42}
    }},
    {"list", {1, 0, 2}},
    {"object", {
        {"currency", "USD"},
        {"value", 42.99}
    }}
};
```

6. Наскільки зручно було використовувати бібліотеку, чи не треба було писати багато надлишкового коду?

Бібліотеку використовувати досить зручно, писати багато надлишкового коду не потрібно. Робота бібліотеки нагадує роботу `map`, тобто в поле з іменем `"pi"` значення записуються `j["pi"] = 3.14`, для чого, щоб взяти значення використовуємо метод `.get<type>()`, де замість `type` пишемо потрібний нам тип. Масиви зберігаються у `vector<type>`.

7. Наскільки зрозумілою була поведінка класів/методів/функцій з бібліотеки?

Достатньо зрозуміло.

8. Наскільки зрозумілою була взаємодія між різними класами/методами/функціями цієї бібліотеки, а також взаємодія між бібліотекою та власним кодом?

Якщо добре розібрати з документацією, то проблем не виникає.

9. Чи виникали якісь проблеми з використанням бібліотеки? Чи вдалось їх вирішити, як саме?

Проблем не виникало.

10. Що хорошого можна сказати про цю бібліотеку, які були позитивні аспекти використання бібліотеки?

Бібліотека зрозуміла, проста.

11. Що поганого можна сказати про цю бібліотеку, які були негативні аспекти використання бібліотеки?

На щастя, нічого.

12. Якби довелось вирішувати аналогічну задачу, але вже враховуючи досвід використання в цій лабораторній роботі, що варто було б робити так само, а що змінити? Можливо, використати інші бібліотеки, чи використати інші можливості цієї бібліотеки, чи інакше організувати код, чи ще щось?

Якщо враховувати досвід, то процес написання можна було б пришвидшити і уникнути деяких (хоч і недостатньо значущих для згадування тут) помилок.

## libHARU

1. Які конкретні задачі планували вирішувати за допомогою цієї бібліотеки?

Генерування файлів у форматі PDF.

2. Чому було обрано саме цю бібліотеку, а не аналоги?

Я обрала саме цю бібліотеку, адже вона є безкоштовною, із відкритим вихідним кодом, сумісна з C++

3. Наскільки просто та зрозуміло було отримати, встановити, налаштувати та почати використовувати цю бібліотеку?

Як на мене складно. У статті на Wiki було досить просто: розпакувати, запустити `make` або `make`, однак бібліотека потребує також установки `zlib` і `libhary`, про які у посібнику не написано. До того ж, воно потребує старих версій компілятора. Потрібно було використовувати `CMake`, однак це я дізналася уже з коментарів користувачів, які теж стикнулися з подібною проблемою.

4. Наскільки зрозумілою та корисною була документація бібліотеки?

Не надто корисною. Єдине, що тут є приклади використання тієї чи іншої функції, а також пошук по кодах помилок.

5. Наскільки було зрозуміло, як саме використовувати бібліотеку, які класи/методи/функції використовувати для вирішення поставлених задач?

Для створення нового PDF документа, необхідно мати функцію обробки помилок, яку користувач повинен писати самостійно. Наприклад, для нормального функціонування бібліотеки необхідно таке

```
jmp_buf env;
#ifdef HPDF_DLL
void __stdcall
#else
void
#endif
error_handler(HPDF_STATUS error_no,
               HPDF_STATUS detail_no,
               void *user_data) {
    printf("ERROR: error_no=%04X, detail_no=%u\n", (HPDF_UINT)error_no,
           (HPDF_UINT)detail_no);
    longjmp(env, 1);
}
```

6. Наскільки зручно було використовувати бібліотеку, чи не треба було писати багато надлишкового коду?

Як показує приклад вище, певні проблеми є.

7. Наскільки зрозумілою була поведінка класів/методів/функцій з бібліотеки?

Більш-менш зрозуміло.

8. Наскільки зрозумілою була взаємодія між різними класами/методами/функціями цієї бібліотеки, а також взаємодія між бібліотекою та власним кодом?

Бібліотека не може працювати з типом `string`, тому через це необхідно вручну переводити його в `char* const`.

9. Чи виникали якісь проблеми з використанням бібліотеки? Чи вдалось їх вирішити, як саме?

Виникали помилки, які вказували на неможливість запуску графічного інтерфейсу, а також на зупинку запису тексту у файл.

10. Що хорошого можна сказати про цю бібліотеку, які були позитивні аспекти використання бібліотеки?

Вона безкоштовна😊

11. Що поганого можна сказати про цю бібліотеку, які були негативні аспекти використання бібліотеки?

Занадто багато незрозумілих речей.

12. Якби довелось вирішувати аналогічну задачу, але вже враховуючи досвід використання в цій лабораторній роботі, що варто було б робити так само, а що змінити? Можливо, використати інші бібліотеки, чи використати інші можливості цієї бібліотеки, чи інакше організувати код, чи ще щось?

Можливо використовувати інші мови програмування, оскільки нормальних бібліотек для генерування PDF на C++ я так і не знайшла.