**Звіт-ретроспектива для 3-ї лабораторної роботи.**

# CppVerbalExpressions

Дана бібліотека дає змогу конструювати складні регулярні вирази. Вона базується на оригінальної бібліотеці JavaScript [Verbal Expressions](http://verbalexpressions.github.io/JSVerbalExpressions/). Ця бібліотека дуже корисна у випадках, коли з якоюсь строкою необхідно виконати певні дії, пов’язані з її структурою.

Приклади використання:

1. Визначити, чи строка підпадає під шаблон. (Перевірка на валідність посилання, присутності певних послідовностей в строці/тексті).
2. Заміна частини строки. (Дозволяє знаходити частину строки та замінити на іншу вказану строку за бажанням користувача).

Бібліотека здалася мені найзручнішою у використанні, особливо якщо брати до уваги те, що я хотів отримати засіб для знаходження й заміни конструкцій всередині тексту.

Бібліотека являє собою один заголовочний файл, тому просто підключивши його ви отримуєте повний доступ до інструментарію. Сама бібліотека окрім вбудованих бібліотек С++ не використовує жодної сторонньої, що теж зручно. Використання бібліотеки зводиться до кількох функцій, а саме

replace(const std::string& source, const std::string& value);

test(const std::string& value).

Загалом бібліотека є розширенням можливостей **Regular expressions library.**

Дана бібліотека зручніша тим, що принцип створення цих самих регулярних виразів по суті являє собою послідовне додавання шаблонів, що в результаті призводить до зручності й простоти генерації великих виразів-трафаретів. Ця зручність виявляється в тому, що для послідовного додавання нового зразка достатньо просто виклику цього зразка через оператор **.**

Окремим пунктом вважаю за потрібне виділити спеціальні символи, що використовувалися мною для створення шаблонів:

1. word Перевіряє чи є строка словом, що складається з символів (a-z, A-Z, 0-9 або \_)
2. digit Перевіряє чи є строка цифрою (0-9)

Так само наведу кілька зручних та корисних Правил, що лоступні в ції бібліотеці:

1. startOfLine – Видає істину для присутності регулярного виразу лише тоді, коли останній з’явився у строці з початку.
2. endOfLine – --//-- для кінця строки.
3. then і find – саме це Правило відповідає за пошук входження виразу.
4. maybe – вираз може бути як присутній, так і відсутній.
5. anything(But) – будь-які символи будь-яку кількість разів (крім цих відповідно).
6. something(But) – будь-які символи (крім цих відповідно) як мінімум один раз.
7. not – не цей вираз.
8. range – будь-який символ у вказаних (можливо неодноразово) межах (за **ascii** table).

Цикли:

1. multiple – перевіряє на входження попередньої групи символів кілька разів.

Окремим пунктом можна виділити саме функцію test(const std::string& value). Вона є перевіркою на те чи є строка регулярним виразом (чи підпадає під шаблон). Таким чином тести можна проводити на місці. Також можна виводи шаблон, що був використаний для пошуку співпадінь.

Взаємодія між різними методами, спеціальними символами, циклами та правилами є дуже зрозумілою та зручною, документація бібліотеки в JavaScript ідентична до С++, тому її достатньо для використання даної бібліотеки в своїх програмах.

Ця бібліотека виконує конкретне завдання й потреби в тісній взаємодії з іншим кодом майже немає. Проте з виразами можна працювати як зі строками, тобто вивід і аналіз їх є доволі простим та не потребує додаткових функцій.

В цілому, бібліотека надає зручний функціонал для роботи з пошуком та заміною створених власноруч регулярних виразів. Створення останніх також забезпечене можливостями бібліотеки. Зі свої завданням CppVerbalExpressions впоралася на відмінно. Негативних аспектів під час роботи не було помічено. Можливе розширення функціоналу за бажанням програміста. Зручний інтерфейс не створюватиме перешкод.

В майбутньому, якщо стикнуся з відповідною задачею, однозначно звернуся до даної бібліотеки через її дуже зручну імплементацію в будь-якому проєкті.

Джерела:

[Список доступних для використання в лабораторній бібліотек](https://github.com/kzhereb/knu-is-ooop2018/blob/master/LibrariesFrameworks.md)

[Бібліотека на JavaScript](http://verbalexpressions.github.io/JSVerbalExpressions/)

[GitHub сторінка бібліотеки з прикладами, кодом та описом](https://github.com/VerbalExpressions/CppVerbalExpressions)