

Завдання 2. Динамічна генерація рядків

Версія 6 березня 2023 р.

В цьому завданні ми напишемо невеличку бібліотеку для конвертування інформації, що зберігається в `std::map` у строку для подальшого запису в файл в одному чи декількох форматах: XML, JSON, YAML. Ця процедура відома як серіалізація. Синтаксис кожного формату можна знайти в інтернеті, але для довідки, можна користуватись відповідним `example.*` файлом в цій директорії. Ключ (`key_type`) для нашої `std::map` мусить бути об'єктом типу `std::string`. Значення може бути одне з трьох:

1. аналогічний об'єкт `std::map`,
2. об'єкт `std::vector<std::string>`,
3. об'єкт `std::string`.

Файл заголовка `serializer.h` має в собі всі декларації, які необхідні для роботи. Для виконання завдання необхідно імплементувати по три функції для кожного формату для кожного з трьох типів даних в файлі `serializer.cpp`. Шаблоні функції, задані в `serializer.h` викличуть необхідну функцію для кожного типу даних.

Файл `main.cpp` дає приклад використання цієї бібліотеки. Його можна використовувати як перший етап верифікації роботи програми при її розробці – вона повинна писати текст, аналогічний тому, що знаходиться в файлах `example.*`.¹

Додатковий бал

- Зробіть імплементацію для всіх трьох форматів даних.

¹Зауважте, що порядок виводу елементів на екран, тобто порядок їх додавання у вихідну строку, відрізняється від порядку їх додавання в `std::map`, який співпадає з тим, як вони записані в `example.*`. Це через те, що `std::map` сортує свої пари використовуючи оператор `<` своїх ключів.

- Запишіть кожен результат у файл зі своїм розширенням.
- Напишіть будь-які функції для автоматизованого тестування. Тобто, такі, що передають заздалегідь визначений об'єкт типу `std::map` з різною глибиною вкладення у серіалізатор, отримують рядок, і перевіряють з еталоном, який відповідає переданим даним.