

Объектно ориентированное программирование

Лабораторная работа №3 “Реализация парсера json”

Задание

Разработать на C++ с использованием фреймворка Qt программу для проверки корректности файлов формата json. При разработке следует руководствоваться принципами ООП.

Программа должна предоставлять возможность:

- 1) Выбрать файл, расположенный в файловой системе компьютера (fileDialog).
- 2) Проверить файл на соответствие формату JSON.
- 3) Вывести на экран содержимое файла.
- 4) Вывести на экран информацию об ошибке (место в файле, комментарий и т.п.), в случае если файл не соответствует формату.

Программа должна содержать два модуля:

- Модуль интерфейса
- Модуль бизнес-логики

Модуль интерфейса

Данный модуль отвечает только за отображение интерфейса и передачу команд модулю бизнес-логики. Загруженные из файла данные не должны храниться в этом модуле. Интерфейс должен быть реализован при помощи Qt.

Модуль бизнес-логики

Данный модуль отвечает за основную функциональность системы, он должен обеспечивать:

- чтение файла;
- проверку формата файла;

Использование сторонних библиотек для работы с Json запрещено.

Входные данные

- json-файл.

Выходные данные / результат

- Сообщение - корректный файл был передан, или нет.
- Вывод текста файла на интерфейсе программы.

Пример входных данных

Json файл

```
{
  "strMember": "new str",
  "intMember": 1,
  "objMember": {
    "arrMamber": [1, 2, 3]
  }
}
```

Ожидаемый результат - файл корректен.

Формат Json

JSON (JavaScript Object Notation) - простой формат обмена данными, удобный для чтения и написания как человеком, так и компьютером. Подробнее можно прочитать [тут](#).

JSON используется для обмена структурированными данными и состоит из элементов. Элемент может быть:

- Простым (разных типов)
 - Число - целочисленное или вещественное, разделитель - точка.
 - Строка - "some string" ограничена кавычками.
 - Спец-типы - true, false, null.
- Объектом - т.е. сложным элементом, состоящим из набора полей и их значений. Каждое поле - это пара "имя": значение. Где значение - произвольный элемент. Поля объекта перечисляются через запятую. Объект заключается в {}. Пример:

```
{
  "strMember": "new str",
  "intMember": 1,
  "objMember": {
    "arrMamber": [1, 2, 3]
  }
}
```

- Массивом - набором других элементов, перечисленных через запятую и заключенный в []. Пример:

```
[1, 3, "str", { "member": "new str"}]
```

Несколько важных правил формата:

- JSON может содержать только 1 корневой элемент, все остальные - вложенные в корневой.

- Пробелы и отступы не учитываются.