

Дисциплина: Объектно-ориентированное программирование

Лабораторная работа №2 «Реализация парсера json»

Задание: разработать на C++ с использованием фреймворка Qt программу для проверки корректности файлов формата json. При разработке следует руководствоваться принципами ООП. Использование сторонних библиотек для работы с json-файлами запрещено.

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 (номер строки, номер символа в строке), описание ошибки, в случае если файл не соответствует формату.

Программа должна содержать два модуля:

1. Модуль интерфейса.
2. Модуль бизнес-логики.

Модуль интерфейса

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

Модуль отвечает за основную функциональность системы, он должен обеспечивать возможность:

1. чтение файла;
2. проверку формата файла;
3. определение наличия ошибки, её места в файле и типа ошибки.

Входные данные

1. json-файл.

Выходные данные / результат

1. Сообщение – корректный файл был передан, или нет.
2. Вывод текста файла на интерфейсе программы.

Пример входных данных

1. Файл корректен

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

Ожидаемый результат - файл корректен.

2. Файл содержит ошибку

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

Ожидаемый результат – «Ошибка обработки массива. Пропущен знак "]" на позиции 5,25»*

* текст ошибки может быть изменён, при условии выполнения требований задания.

Формат Json

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

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

1. Числом – целочисленное или вещественное, разделитель - точка. Реализация более сложных форматов представления чисел, например 1e-2, оценивается бонусным баллом.
2. Строкой – "some string" ограничена кавычками.
3. Специальным типом – true, false, null. Обратите внимание, для специальных типов важен регистр.
4. Объектом – сложным элементом, состоящим из набора полей и их значений. Каждое поле – пара "имя": значение, где ключ – уникальное имя, значение – произвольный элемент. Поля объекта перечисляются через запятую. Объект заключается в {}.

Например:

```
{
  "strMember": "new str",
```

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

Массив – набор элементов, перечисленных через запятую и заключенных в [].

Например:

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

Несколько важных правил формата:

1. JSON должен содержать только 1 корневой элемент, все остальные – вложенные в корневой.
2. Пробелы и отступы не учитываются.