

Бгуир, Кафедра Интеллектуальных Информационных Технологий

Отчёт
по лабораторной работе №2

Выполнил: Якимович Илья Викторович

Группа: 121703

Проверил: Никифоров Сергей Александрович

Минск 2022

Цель: Получить навыки проведения объектно-ориентированного анализа предметной области.

Задание:

2.6 Модель XML-документа

При выборе этого варианта студенту необходимо разработать объектную модель для представления XML-документа. Для знакомства с XML рекомендуется прочитать главы 1 и 2 из книги Э. Рея «Изучаем XML». Глава 1 из этой книги является обзорной и прояснит назначение XML, а в главе 2 описан синтаксис XML-разметки.

Разработанная объектная модель должна обеспечивать представление XML-документа в виде дерева и поддерживать следующие понятия XML-разметки:

- документ
- инструкция обработки
- тэг
- комментарий
- CDATA
- атрибут

Описание:

Написан класс `XMLDocument` реализующий структуру xml-документа.

У класса есть конструктор который принимает в себя имя файла .

Функция `print(string filename)` выводит полученный и распаршенный xml-документ в файл название которого , совпадает с переданным аргументом метода

Функция `parse()` метод отвечающий за парсинг xml-документа.

Функция `get_first_child()` метод отвечающий за получение корневого элемента.

Функция `set_attribute_value(const string &attributeName, const string &attributeValue)` устанавливает значение в атрибут имя которого совпадает с соответствующим переданным в метод аргументом.

Функция `get_attribute_value(const string &attributeName)` получаем значение атрибута.

Функция `attribute_size()` количество атрибутов в элементе.

Функция `get_attribute(int index)` получаем атрибут.

Функция `print(ofstream &outputStream, int level)` выводит элемент.

Функция `set_next_sibling(XMLElement *nextSibling)` меняем значение соседу данного элемента.

Функция `set_first_child(XMLElement *firstChild);` меняем значение для первого “сына”.

Функция `add_attribute(XMLAttribute xmlAttribute);` добавить новый атрибут.

Функция `set_pc_data(string pcData)`; установить текстовое значение в элемент.

Функция `has_attributes()`; существует ли атрибут.

Функция `get_name()`; получаем имя.

Функция `get_value()`; получаем значение.

Функция `set_value(string value)`; устанавливаем значение.

Функция `to_string()`; переводим атрибут в тип данных `string`.

Вывод:

В ходе работы были изучены некоторые основы ООП. Были использованы инкапсуляция, понятия класса, конструктора, деструктора, перегрузки.