Тема

**Поддържане на HTML**

Програма, реализираща четене и операции върху HTML файлове.

Изготвил: Мария Маргаритова, първи курс, Информационни системи, група 1

**СЪДЪРЖАНИЕ:**

Глава 1. Увод

Глава 2. Преглед на предметната област

Глава 3. Проектиране

Глава 4. Реализация, тестване

Глава 5. Заключение

Използвана литература

**Глава 1. Увод**

Идеята на проекта е да се създаде програма, която реализира четене и операции върху HTML файлове. Възможните операции са: зареждане на HTML файл, създаване на нов HTML файл, записване на файл, избор на елементи, промяна на текст на елемент, промяна на текст на атрибут и добавяне на нов елемент.

Задачите в разработката се делят на няколко етапа:

1. Обмисляне и оформяне на идеята.
2. Събиране на данни.
3. Избор на подходящи технологии.
4. Създаване на моделите.
5. Създаване на помощни класове.
6. Създаване на класа с бизнес логиката.
7. Тестване

**Глава 2. Преглед на предметната област**

2.1. Основни дефиниции, концепции и алгоритми, които ще бъдат използвани

Кодът е разделен ‘.h’ и ‘.cpp‘ файлове, като във файловете с разширение ‘.h’ се намират дефиницията на класа, методите и член-данните му. Променливите са именувани по конвенцията ‘camelCase’, класовете започват с главна буква, а всички функции, променливи и „неймспейсове“ с малка. В класовете имаме ‚private‘ и ‚public‘ част, като в първата слагаме всички променливи и функции, които не искаме да бъдат достъпни за други класове, а във втората функции, които искаме да използваме и в други класове.

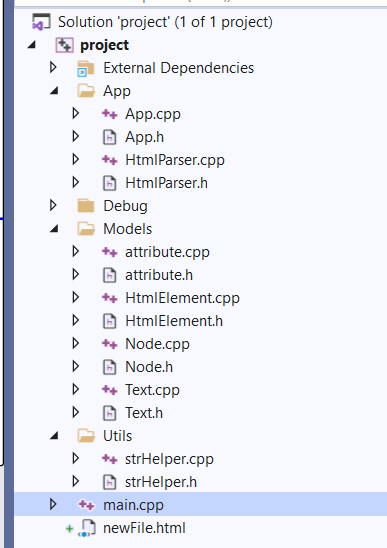
2.2. Дефиниране на проблеми и сложност на поставената задача

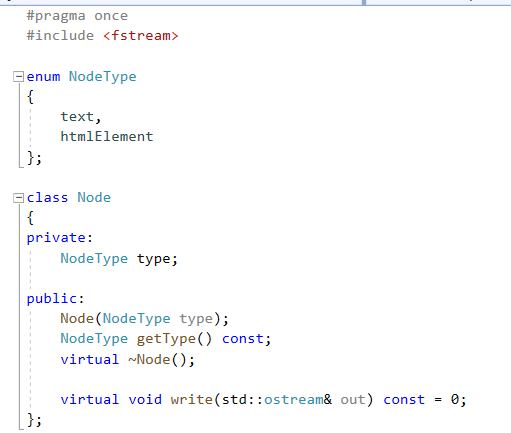
Трябва да се избере по какъв начин точно да бъдат създадени класовете, така че да няма повтарящ се код, така че да се спазят всички изисквания на обектното ориентирано програмиране и да може задачата да бъде разделена на по-малки части. Проблем е как да се организира паметта така, че да няма изтичане на данни при създаването на абстрактния клас и при неговите наследници.

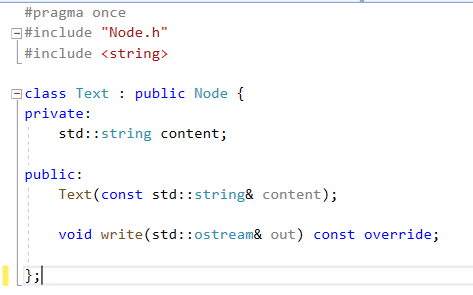
2.3. Подходи, методи за решаване на поставените проблемите

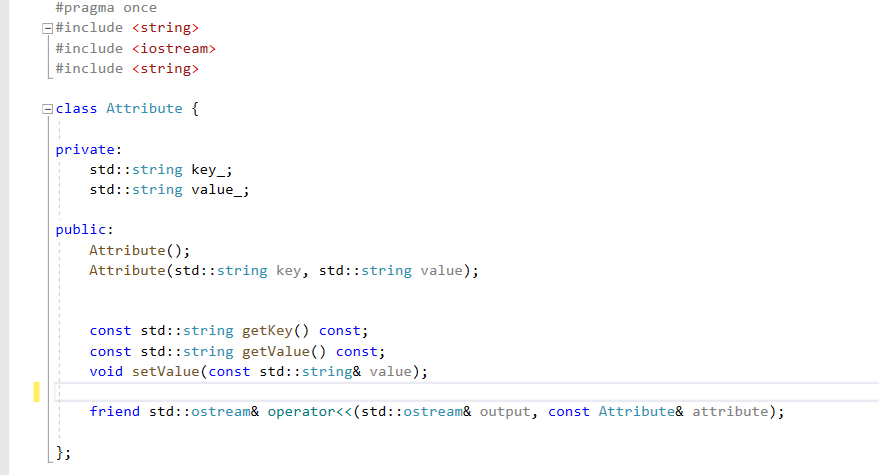
Тъй като се налага да се създаде една ‚pure virtual‘ функция в класа ‘Node’, създаваме виртуален деструктор, за да може когато се наследява този клас да няма изтичане на данни. Най—подходящото разделение на компонентите се състои в това, че проектът е разделен на 3 папки: App, Models, Utils. В Utils имаме помощни функции, които сме ги отделили там, защото не са част от моделите, но се нуждаем от тях в създаването на бизнес логиката. В папка Models създаваме моделите, описваме класовете, кое какво наследява и основните им функционалности. В App е бизнес логиката, която обединява Models и Utils. Четвъртия компонент е класа main, където се стартира програмата и обединява всички класове в едно.

**Глава 3. Проектиране**

****3.1. Обща архитектура – ООП дизайн

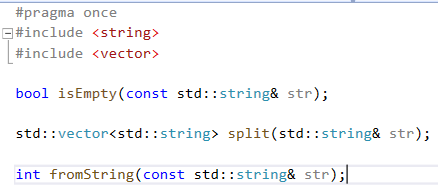
Проектът е разделен на 3 папки: App, Models, Utils. В Utils имаме помощни функции, които сме ги отделили там, защото не са част от моделите, но се нуждаем от тях в създаването на бизнес логиката. В папка Models създаваме моделите, описваме класовете, кое какво наследява и основните им функционалности. В App е бизнес логиката, която обединява Models и Utils. Четвъртия компонент е класа main, където се стартира програмата и обединява всички класове в едно.





Това са класовете от папка ‘Models’. Класът Node се наследява от класовете HtmlElement и Text. Класът Attribute го създаваме за да можем да направим вектор от атрибути в HtmlElement.





В папка App създаваме класа HtmlParser и App, като в HtmlParser се създава логиката за прочитане на html файл и зареждането на информацията. А в App се използват абсолютно всички функционалности, за да се създаде бизнес логиката.

В App създаваме публичната функция run, която ще се извика в main() и ще зареди цялата програма. Можем да получим различни команди от конзолата като четем стринга и го разделяме по такъв начин, че да изведем всяка дума от него във вектор. Първия елемент на вектора може да бъде: "load","new","save","select","select\_one","set","add". Спрямо това, коя от тях е, се извиква функцията която отговаря на нея. Като аргументи подаваме останалите елементи от вектора в зависимост коя е командата.

Глава 4. Реализация

Когато се стартира програмата, това съобщение се показва, като може да се въведе някоя от командите.

HTML PARSER----TRY IT NOW:

Enter a command:

load newFile.html – командата load

The file has been loaded! – показва се, че файла е зареден.

Enter a command:

Имаме възможността да въведем, нови команди.

**Глава 5. Заключение**

Основната идея на проекта е да се реализира програма, която може да реализира различни операции върху Html файлове. Проектът има още много перспективи за надграждане, в най-различни насоки. Могат да се добавят още функционалности, както и да се създаде по-добро графично оформление.