Semiautodoc

Aleksandra Prohorova, Andrey Barekov

24 April, 2021

Problem

- necessity of documentation
- desire to edit the documentation separately from code
- opportunity of manual fix if something went wrong

Requirements

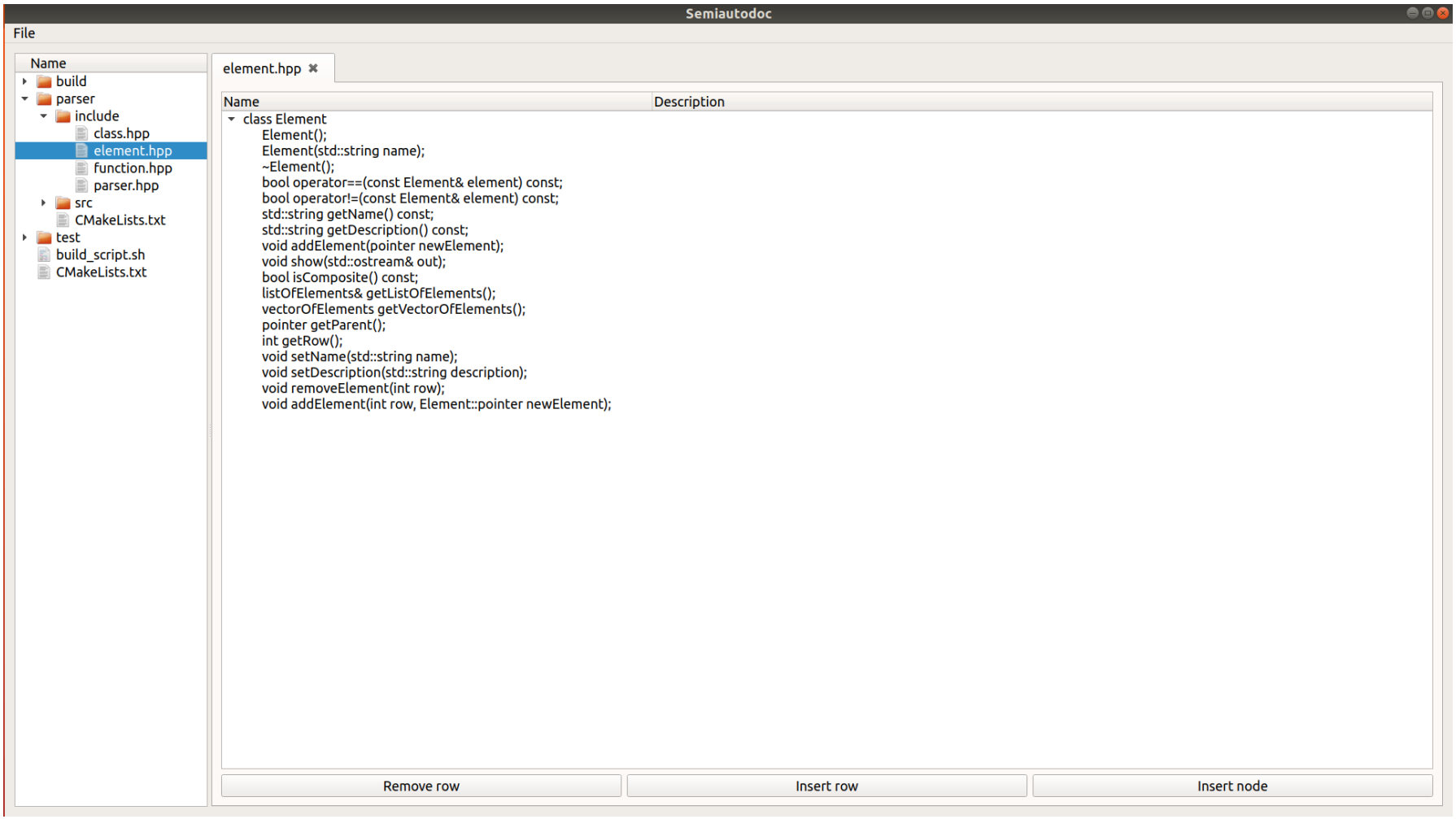
- automatic parsing of classes and function's names from source code
- edit names that were produced as a result of parsing
- add names to the current list if some of them were missed and delete names that seem unimportant, add despription for each element to get most relevant documentation
- get readable markdown file from tree structure of classes and functions

Project structure

- static library for parsing
- unit and functional tests for static library
- GUI

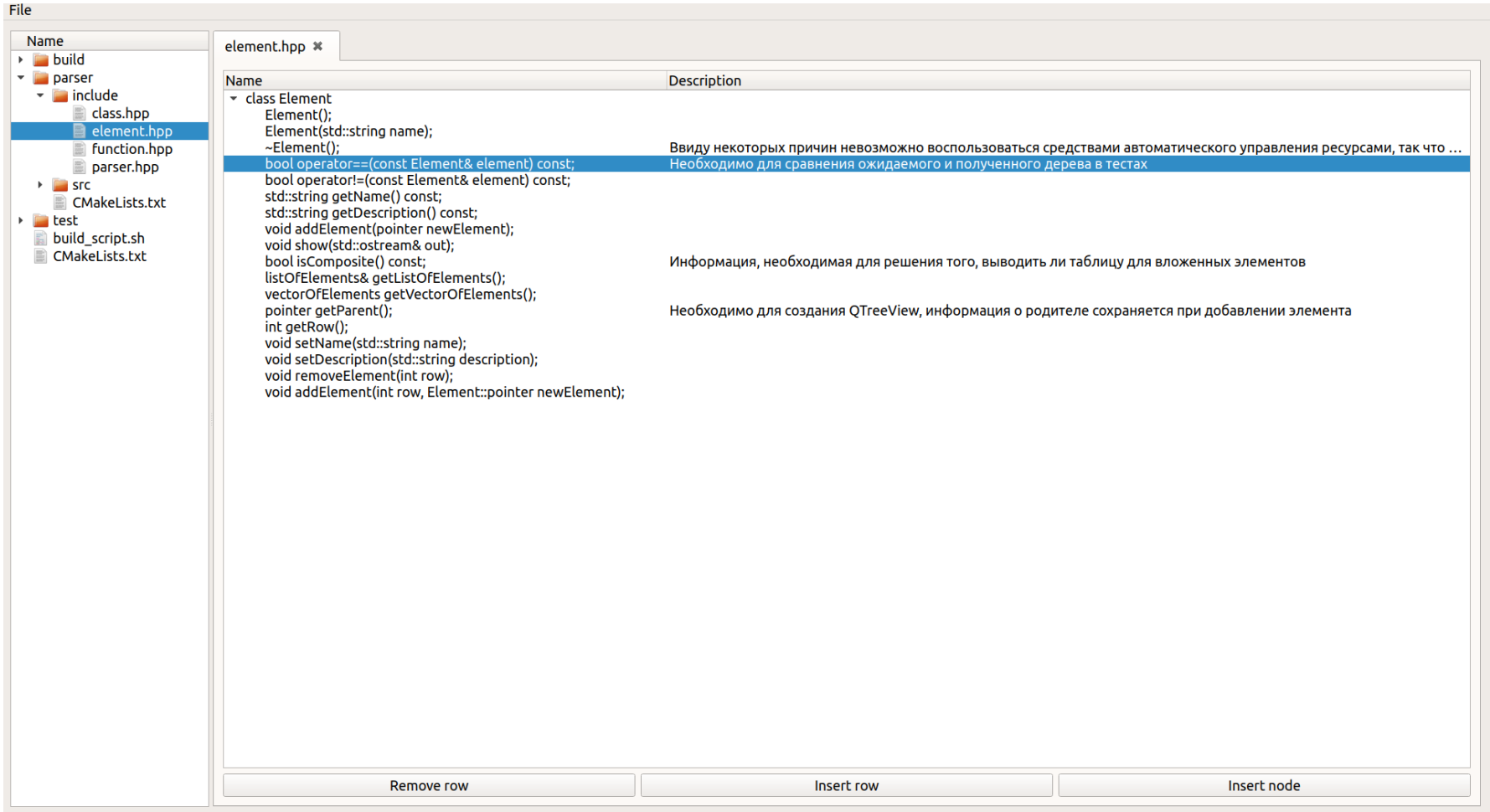
Example

Results of parsing



Example

Edit results of parsing



Example

Markdown file

22 lines (21 sloc) | 1.36 KB Raw Blame

class Element

Name	Description
Element();	
Element(std::string name);	
~Element();	Ввиду некоторых причин невозможно воспользоваться средствами автоматического управления ресурсами, так что освобождать память приходится вручную
bool operator==(const Element& element) const;	Необходимо для сравнения ожидаемого и полученного дерева в тестах
bool operator!=(const Element& element) const;	
std::string getName() const;	
std::string getDescription() const;	
void addElement(pointer newElement);	
void show(std::ostream& out);	
bool isComposite() const;	Информация, необходимая для решения того, выводить ли таблицу для вложенных элементов
listOfElements& getListOfElements();	
vectorOfElements getVectorOfElements();	
pointer getParent();	Необходимо для создания QTreeView, информация о родителе сохраняется при добавлении элемента
int getRow();	
void setName(std::string name);	
void setDescription(std::string description);	
void removeElement(int row);	
void addElement(int row, Element::pointer newElement);	

Source code

https://github.com/aleksandraprohorova/semiautodoc