Lab 8 - Expression Trees

Generated by Doxygen 1.8.5

Thu Oct 24 2013 00:35:25

Contents

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

Class Documentation

3.1 ExprTree< DataType > Class Template Reference

```
#include <ExpressionTree.h>
```

Public Member Functions

- ExprTree ()
- ExprTree (const ExprTree &source)
- ExprTree & operator= (const ExprTree &source)
- ∼ExprTree ()
- void build ()
- void expression () const
- DataType evaluate () const throw (logic_error)
- void clear ()
- void commute ()
- bool isEquivalent (const ExprTree &source) const
- void showStructure () const

3.1.1 Constructor & Destructor Documentation

3.1.1.1 template<typename > ExprTree< typename >::ExprTree ( )

Precondition

New ExprTree Class

Postcondition

root pointing to null

Algorithm:

- point root to null

Exceptional/Error Conditions:

- none

3.1.1.2 template<typename DataType > ExprTree< DataType >::ExprTree (const ExprTree< DataType > & *source*)

Precondition

new ExprTree class

Postcondition

a deep copy of source ExprTree

Parameters

source	Source ExprTree to be deep copied

Algorithm:

- Use overloaded assignment operator to copy the data from the source to this class

3.1.1.3 template<typename DataType > ExprTree< DataType >::∼ExprTree ( )

Precondition

an ExprTreeClass

Postcondition

deallocated ExprTreeClass

Algorithm:

- Use clear member function to deallocate the tree

3.1.2 Member Function Documentation

3.1.2.1 template<typename DataType > void ExprTree< DataType >::build ( )

Precondition

an ExprTreeClass

Postcondition

a built Expression tree

Algorithm:

- Use recBuild member function to make the expression tree

3.1.2.2 template<typename DataType > void ExprTree< DataType >::clear ( )

Precondition

an ExprTreeClass

Postcondition

deallocated ExprTreeClass

Algorithm:

- use the recClear member function to deallocate the exprTree

3.1.2.3 template<typename DataType > void ExprTree< DataType >::commute ( )

Precondition

an ExprTreeClass

Postcondition

commuted Expression Tree

Algorithm:

- use the recCommute member function to commute the exprTree

3.1.2.4 template<typename DataType > DataType ExprTree< DataType >::evaluate ( ) const throw logic_error)

Precondition

a built ExprTreeClass

Postcondition

The mathematical expression is evaluated

Returns

The mathematical evaluation of the expression of the tree

Algorithm:

- use the recEvaluate member function to evaluate the exprTree

Exceptions

A	logic error is thrown if the tree is empty

3.1.2.5 template<typename DataType > void ExprTree< DataType >::expression ( ) const

Precondition

a built ExprTreeClass

Postcondition

The mathematical expression is printed on screen infix notation

Algorithm:

- Use recExpression member function to print the expression

3.1.2.6 template<typename DataType > bool ExprTree< DataType >::isEquivalent (const ExprTree< DataType > & *source*) const

Precondition

an ExprTreeClass

Postcondition

returned boolean if the trees are equivalent

Parameters

source	Source ExprTree to be compared

Returns

This function returns if the trees are equivalent based on commutative properties

Algorithm:

- use the recIsEquivalent member function to check equivalence

3.1.2.7 template<typename DataType > ExprTree< DataType > & ExprTree< DataType >::operator= (const ExprTree< DataType > & *source*)

Precondition

an ExprTreeClass

Postcondition

a deep copy of source ExprTree

Parameters

source	Source ExprTree to be deep copied

Algorithm:

- Use recCopy to make a copy of each node in the ExprTree

3.1.2.8 template<typename DataType > void ExprTree< DataType >::showStructure ( ) const

The documentation for this class was generated from the following files:

- ExpressionTree.h
- ExpressionTree.cpp
- show8.cpp

Chapter 4

File Documentation

4.1 config.h File Reference

Macros

- #define LAB8_TEST1 0
- #define LAB8_TEST2 1
- #define LAB8_TEST3 1

4.1.1 Macro Definition Documentation

4.1.1.1 #define LAB8_TEST1 0

Expression Tree class (Lab 8) configuration file. Activate test #N by defining the corresponding LAB8_TESTN to have the value 1.

4.1.1.2 #define LAB8_TEST2 1

4.1.1.3 #define LAB8_TEST3 1

4.2 ExpressionTree.cpp File Reference

```
#include <iostream>
#include "ExpressionTree.h"
```

4.3 ExpressionTree.h File Reference

```
#include <stdexcept>
#include <iostream>
```

Classes

- class ExprTree< DataType >

4.4 show8.cpp File Reference

4.5 test8.cpp File Reference

```
#include <iostream>
#include <stdexcept>
#include "ExpressionTree.cpp"
#include "config.h"
```

Functions

- template<typename DataType >
  void dummy (ExprTree< DataType > copyTree)
- int main ()

4.5.1 Function Documentation

4.5.1.1 template<typename DataType > void dummy (ExprTree< DataType > *copyTree*)

4.5.1.2 int main ( )

Index