msdscript

Generated by Doxygen 1.9.6

Chapter 1

MSDScript-main.cpp

Author

Howard Tung

Date

02-07-2023

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

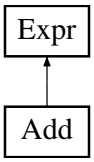
Chapter 5

Class Documentation

5.1 Add Class Reference

`#include <Expr.hpp>`

Inheritance diagram for Add:



Public Member Functions

- Add (Expr *lhs, Expr *rhs)
- bool equals (Expr *expr)

  Check if this class lhs is equals to Expr class rhs provided in parentheses.
- int interp ()

  Add both side.
- bool hasVariable ()

  Check both lhs and rhs Expression class to see if they have variable.
- Expr * subst (std::string s, Expr *expr)

  Replace the lhs variable or rhs that has the string provided in parentheses and replace with the expression that provided.

- virtual bool equals (Expr *expr)=0
- virtual int interp ()=0
- virtual bool hasVariable ()=0
- virtual Expr * subst (std::string s, Expr *expr)=0

Public Attributes

- Expr * lhs

  lhs of Expression
- Expr * rhs

  rhs of Expression

5.1.1 Constructor & Destructor Documentation

5.1.1.1 Add()

```
Add::Add (
            Expr * lhs,
            Expr * rhs )
```

5.1.2 Member Function Documentation

5.1.2.1 equals()

```
bool Add::equals (
            Expr * expr )  [virtual]
```

Check if this class lhs is equals to Expr class rhs provided in parentheses.

Parameters

∗expr	- Provide expression

Returns

true if both equals val, false otherwise.

Implements Expr.

5.1.2.2 hasVariable()

```
bool Add::hasVariable ( )  [virtual]
```

Check both lhs and rhs Expression class to see if they have variable.

Returns

true or false based on if it has variables

Implements Expr.

5.1.2.3 interp()

```
int Add::interp ( )  [virtual]
```

Add both side.

Returns

return the integer added value

Implements Expr.

5.1.2.4 subst()

```
Expr * Add::subst (
            std::string s,
            Expr * expr )  [virtual]
```

Replace the lhs variable or rhs that has the string provided in parentheses and replace with the expression that provided.

Returns

return the expression

Implements Expr.

5.1.3 Member Data Documentation

5.1.3.1 lhs

```
Expr* Add::lhs
```

lhs of Expression

5.1.3.2 rhs

```
Expr* Add::rhs
```

rhs of Expression

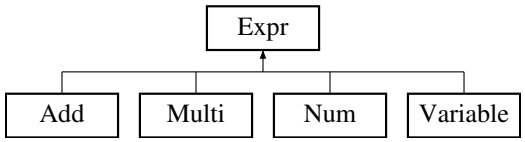
The documentation for this class was generated from the following files:

- /Users/howard/Documents/Github/CS6015/Expr.hpp
- /Users/howard/Documents/Github/CS6015/Expr.cpp

5.2 Expr Class Reference

`#include <Expr.hpp>`

Inheritance diagram for Expr:



Public Member Functions

- virtual bool equals (Expr ∗expr)=0
- virtual int interp ()=0
- virtual bool hasVariable ()=0
- virtual Expr ∗ subst (std::string s, Expr ∗expr)=0

5.2.1 Member Function Documentation

5.2.1.1 equals()

```
virtual bool Expr::equals (
            Expr * expr )  [pure virtual]
```

Implemented in Num, Add, Multi, and Variable.

5.2.1.2 hasVariable()

```
virtual bool Expr::hasVariable ( )  [pure virtual]
```

Implemented in Num, Add, Multi, and Variable.

5.2.1.3 interp()

```
virtual int Expr::interp ( )  [pure virtual]
```

Implemented in Num, Add, Multi, and Variable.

5.2.1.4 subst()

```
virtual Expr * Expr::subst (
            std::string s,
            Expr * expr )  [pure virtual]
```

Implemented in Num, Add, Multi, and Variable.

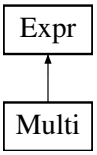
The documentation for this class was generated from the following file:

- /Users/howard/Documents/Github/CS6015/Expr.hpp

5.3 Multi Class Reference

```
#include <Expr.hpp>
```

Inheritance diagram for Multi:



Public Member Functions

- Multi (Expr *lhs, Expr *rhs)
- bool equals (Expr *expr)

  Check if this class lhs is equals to Expr class rhs provided in parentheses.
- int interp ()

  Multiply both side.
- bool hasVariable ()

  Check both lhs and rhs Expression class to see if they have variable.
- Expr * subst (std::string s, Expr *expr)

  Replace the lhs variable or rhs that has the string provided in parentheses and replace with the expression that provided.

- virtual bool equals (Expr *expr)=0
- virtual int interp ()=0
- virtual bool hasVariable ()=0
- virtual Expr * subst (std::string s, Expr *expr)=0

Public Attributes

- Expr * lhs

  lhs of Expression
- Expr * rhs

  rhs of Expression

5.3.1 Constructor & Destructor Documentation

5.3.1.1 Multi()

```
Multi::Multi (
            Expr * lhs,
            Expr * rhs )
```

5.3.2 Member Function Documentation

5.3.2.1 equals()

```
bool Multi::equals (
            Expr * expr )  [virtual]
```

Check if this class lhs is equals to Expr class rhs provided in parentheses.

Parameters

∗expr	- Provide expression

Returns

true if both equals val, false otherwise.

Implements Expr.

5.3.2.2 hasVariable()

```
bool Multi::hasVariable ( )  [virtual]
```

Check both lhs and rhs Expression class to see if they have variable.

Returns

true or false based on if it has variables

Implements Expr.

5.3.2.3 interp()

```
int Multi::interp ( )  [virtual]
```

Multiply both side.

Returns

return the integer multiply value

Implements Expr.

5.3.2.4 subst()

```
Expr * Multi::subst (
            std::string s,
            Expr * expr )  [virtual]
```

Replace the lhs variable or rhs that has the string provided in parentheses and replace with the expression that provided.

Returns

return the expression

Implements Expr.

5.3.3 Member Data Documentation

5.3.3.1 lhs

```
Expr* Multi::lhs
```

lhs of Expression

5.3.3.2 rhs

```
Expr* Multi::rhs
```

rhs of Expression

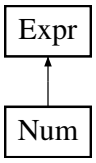
The documentation for this class was generated from the following files:

- /Users/howard/Documents/Github/CS6015/Expr.hpp
- /Users/howard/Documents/Github/CS6015/Expr.cpp

5.4 Num Class Reference

`#include <Expr.hpp>`

Inheritance diagram for Num:



Public Member Functions

- Num (int val)
- bool equals (Expr *expr)

  Check if this class val is equals to Expr class provided in parentheses.
- int interp ()

  return its value
- bool hasVariable ()

  Num has no variable so always return false.
- Expr * subst (std::string s, Expr *expr)

  Directly return the Num expression bc Num class have only val Integer member, so can't be replaced.

- virtual bool equals (Expr *expr)=0
- virtual int interp ()=0
- virtual bool hasVariable ()=0
- virtual Expr * subst (std::string s, Expr *expr)=0

Public Attributes

- int val

5.4.1 Constructor & Destructor Documentation

5.4.1.1 Num()

```
Num::Num (
            int val )
```

5.4.2 Member Function Documentation

5.4.2.1 equals()

```
bool Num::equals (
            Expr * expr )  [virtual]
```

Check if this class val is equals to Expr class provided in parentheses.

Parameters

∗expr	- Provide expression

Returns

true if both equals val, false otherwise.

Implements Expr.

5.4.2.2 hasVariable()

```
bool Num::hasVariable ( )  [virtual]
```

Num has no variable so always return false.

Returns

always false bc it is an Num class.

Implements Expr.

5.4.2.3 interp()

```
int Num::interp ( )  [virtual]
```

return its value

Returns

return its value

Implements Expr.

5.4.2.4 subst()

```
Expr * Num::subst (
            std::string s,
            Expr * expr )  [virtual]
```

Directly return the Num expression bc Num class have only val Integer member, so can't be replaced.

Returns

return the expression

Implements Expr.

5.4.3 Member Data Documentation

5.4.3.1 val

```
int Num::val
```

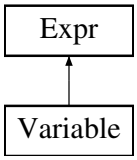
The documentation for this class was generated from the following files:

- /Users/howard/Documents/Github/CS6015/Expr.hpp
- /Users/howard/Documents/Github/CS6015/Expr.cpp

5.5 Variable Class Reference

```
#include <Expr.hpp>
```

Inheritance diagram for Variable:



Public Member Functions

- Variable (std::string string)
- bool equals (Expr ∗expr)

  Check if this class string is equals to Expr class provided in parentheses.
- int interp ()

  Interp cannot be interp so give an error msg.
- bool hasVariable ()

  Bc Variable has variable so always return true.
- Expr ∗ subst (std::string s, Expr ∗expr)

  Replace the variable with the expression that provided.

- virtual bool equals (Expr ∗expr)=0
- virtual int interp ()=0
- virtual bool hasVariable ()=0
- virtual Expr ∗ subst (std::string s, Expr ∗expr)=0

Public Attributes

- std::string string

5.5.1 Constructor & Destructor Documentation

5.5.1.1 Variable()

```
Variable::Variable (
            std::string string )
```

5.5.2 Member Function Documentation

5.5.2.1 equals()

```
bool Variable::equals (
            Expr * expr )  [virtual]
```

Check if this class string is equals to Expr class provided in parentheses.

Parameters

∗expr	- Provide expression

Returns

true if both equals string, false otherwise.

Implements Expr.

5.5.2.2 hasVariable()

```
bool Variable::hasVariable ( )  [virtual]
```

Bc Variable has variable so always return true.

Returns

always true bc it is an Variable class.

Implements Expr.

5.5.2.3 interp()

```
int Variable::interp ( )  [virtual]
```

Interp cannot be interp so give an error msg.

Returns

Bc interp will return an interger

Implements Expr.

5.5.2.4 subst()

```
Expr * Variable::subst (
            std::string s,
            Expr * expr )  [virtual]
```

Replace the variable with the expression that provided.

Returns

return the expression

Implements Expr.

5.5.3 Member Data Documentation

5.5.3.1 string

```
std::string Variable::string
```

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

- /Users/howard/Documents/Github/CS6015/Expr.hpp
- /Users/howard/Documents/Github/CS6015/Expr.cpp

Chapter 6

File Documentation

6.1 /Users/howard/Documents/Github/CS6015/cmdline.cpp File Reference

Getting the input argument and deal with it.

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

Macros

- #define CATCH_CONFIG_RUNNER

Functions

- void use_arguments (int argc, char **argv)

  Check input arguments if its –test or –help then doing different stuff accordingly.

6.1.1 Detailed Description

Getting the input argument and deal with it.

Author

Howard Tung

6.1.2 Macro Definition Documentation

6.1.2.1 CATCH_CONFIG_RUNNER

```
#define CATCH_CONFIG_RUNNER
```

6.1.3 Function Documentation

6.1.3.1 use_arguments()

```
void use_arguments (
            int argc,
            char ** argv )
```

Check input arguments if its –test or –help then doing different stuff accordingly.

Parameters

argc	- how many arguments
argv	- array of the argument

Returns

void - no return value

6.2 /Users/howard/Documents/Github/CS6015/cmdline.h File Reference

Functions

- void use_arguments (int, char $**$)

  Check input arguments if its –test or –help then doing different stuff accordingly.

6.2.1 Function Documentation

6.2.1.1 use_arguments()

```
void use_arguments (
            int argc,
            char ** argv )
```

Check input arguments if its –test or –help then doing different stuff accordingly.

Parameters

argc	- how many arguments
argv	- array of the argument

Returns

void - no return value

6.3 /Users/howard/Documents/Github/CS6015/cmdline.h

Go to the documentation of this file.

```
00001 void use_arguments(int, char**);
```

6.4 /Users/howard/Documents/Github/CS6015/Expr.cpp File Reference

contains expression class function implementation

```
#include "Expr.hpp"
#include "catch.h"
```

Functions

- TEST_CASE ("Test for expression")

6.4.1 Detailed Description

contains expression class function implementation

Author

Howard Tung

6.4.2 Function Documentation

6.4.2.1 TEST_CASE()

```
TEST_CASE (
            "Test for expression"  )
```

6.5 /Users/howard/Documents/Github/CS6015/Expr.hpp File Reference

contains expression class definition

```
#include <stdio.h>
#include <stdexcept>
#include <string>
#include <iostream>
```

Classes

- class Expr
- class Num
- class Add
- class Multi
- class Variable

6.5.1 Detailed Description

contains expression class definition

Author

Howard Tung

6.6 /Users/howard/Documents/Github/CS6015/Expr.hpp

Go to the documentation of this file.

```
00001 //
00002 //  Expr.hpp
00003 //  MSDScript
00004 //
00005 //  Created by Howard Tung on 1/12/23.
00006 //
00007
00013 #include <stdio.h>
00014 #include <stdexcept>
00015 #include <string>
00016 #include <iostream>
00017 #pragma once
00018
00019 class Expr {
00020 public:
00021     virtual bool equals(Expr *expr) = 0;
00022     virtual int interp() = 0;
00023     virtual bool hasVariable() = 0;
00024     virtual Expr* subst(std::string s, Expr* expr) = 0;
00025 //    virtual void print(std::ostream &ostream) = 0;
00026 };
00027
00028 class Num: public Expr{
00029 public:
00030     int val;
00031     Num(int val);
00032     bool equals(Expr *expr);
00033     int interp();
00034     bool hasVariable();
00035     Expr* subst(std::string s, Expr* expr);
00036 //    void print(std::ostream &ostream);
00037 };
00038
00039 class Add : public Expr {
```

```
00040 public:
00041     Expr *lhs;
00042     Expr *rhs;
00043 
00044     Add(Expr *lhs,Expr *rhs);
00045     bool equals(Expr *expr);
00046     int interp();
00047     bool hasVariable();
00048     Expr* subst(std::string s, Expr* expr);
00049 //    void print(std::ostream &ostream);
00050 };
00051 
00052 class Multi : public Expr {
00053 public:
00054     Expr *lhs;
00055     Expr *rhs;
00056 
00057     Multi(Expr *lhs, Expr *rhs);
00058     bool equals(Expr *expr);
00059     int interp();
00060     bool hasVariable();
00061     Expr* subst(std::string s, Expr* expr);
00062 //    void print(std::ostream &ostream);
00063 
00064 };
00065 
00066 class Variable : public Expr {
00067 public:
00068     std::string string;
00069 
00070     Variable(std::string string);
00071     bool equals(Expr *expr);
00072     int interp();
00073     bool hasVariable();
00074     Expr* subst(std::string s, Expr* expr);
00075 //    void print(std::ostream &ostream);
00076 
00077 };
```

6.7 /Users/howard/Documents/Github/CS6015/main.cpp File Reference

```
#include "cmdline.h"
```

Functions

- int main (int argc, char **argv)

  Call use_arguments function.

6.7.1 Function Documentation

6.7.1.1 main()

```
int main (
            int argc,
            char ** argv )
```

Call use_arguments function.

Parameters

argc	- how many arguments are there
argv	- array of arguments

Returns

return integer

Index