MSDScript

Generated by Doxygen 1.9.6

Chapter 1

MSDScript

Author

Laura Zhang

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

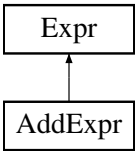
Here is a list of all documented files with brief descriptions:

Chapter 5

Class Documentation

5.1 AddExpr Class Reference

Inheritance diagram for AddExpr:



Public Member Functions

- **AddExpr** (Expr *lhs, Expr *rhs)
- bool equals (Expr *e)

  check if two AddExprs are equal
- int interp ()

  returns an int for the value of an expression
- bool has_variable ()

  returns true if the expression is a variable or contains a variable
- Expr * subst (std::string parameter, Expr *expr)

  everywhere that the expression contains a variable matching the string, the result Expr should have the given replacement*

- virtual bool equals (Expr *e)=0
- virtual int interp ()=0
- virtual bool has_variable ()=0
- virtual Expr * subst (std::string parameter, Expr *expr)=0

Public Attributes

- Expr * **lhs_**
- Expr * **rhs_**

5.1.1 Member Function Documentation

5.1.1.1 equals()

```
bool AddExpr::equals (
            Expr * e )  [virtual]
```

check if two AddExprs are equal

Parameters

e	rhs Expr* to be compared

Returns

true if two AddExprs are equal, null if rhs is NULL

Implements Expr.

5.1.1.2 has_variable()

```
bool AddExpr::has_variable ( )  [virtual]
```

returns true if the expression is a variable or contains a variable

Returns

true if either lhs or rhs is a variable or contains a variable

Implements Expr.

5.1.1.3 interp()

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

returns an int for the value of an expression

Returns

the sum of the subexpression values

Implements Expr.

5.1.1.4 subst()

```
Expr * AddExpr::subst (
            std::string parameter,
            Expr * expr )  [virtual]
```

everywhere that the expression contains a variable matching the string, the result Expr* should have the given replacement

Parameters

parameter	the parameter to be substituted
expr	a new expression

Returns

a new object without changing the current object

Implements Expr.

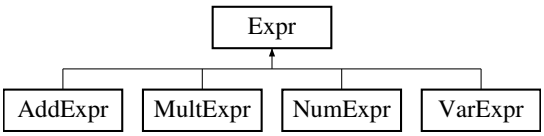
The documentation for this class was generated from the following files:

- /Users/laurazhang/msdscript/expr.h
- /Users/laurazhang/msdscript/expr.cpp

5.2 Expr Class Reference

`#include <expr.h>`

Inheritance diagram for Expr:



Public Member Functions

- virtual bool equals (Expr *e)=0
- virtual int interp ()=0
- virtual bool has_variable ()=0
- virtual Expr * subst (std::string parameter, Expr *expr)=0

5.2.1 Detailed Description

<expr> = <number> | <expr> + <expr> | <expr> * <expr> | <variable>

5.2.2 Member Function Documentation

5.2.2.1 equals()

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

Implemented in NumExpr, AddExpr, MultExpr, and VarExpr.

5.2.2.2 has_variable()

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

Implemented in NumExpr, AddExpr, MultExpr, and VarExpr.

5.2.2.3 interp()

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

Implemented in NumExpr, AddExpr, MultExpr, and VarExpr.

5.2.2.4 subst()

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

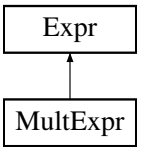
Implemented in NumExpr, AddExpr, MultExpr, and VarExpr.

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

- /Users/laurazhang/msdscript/expr.h

5.3 MultExpr Class Reference

Inheritance diagram for MultExpr:



Public Member Functions

- **MultExpr** (Expr *lhs, Expr *rhs)
- bool equals (Expr *e)

  check if two MultExprs are equal
- int interp ()

  returns an int for the value of an expression
- bool has_variable ()

  returns true if the expression is a variable or contains a variable
- Expr * subst (std::string parameter, Expr *expr)

  everywhere that the expression contains a variable matching the string, the result Expr should have the given replacement*

- virtual bool equals (Expr *e)=0
- virtual int interp ()=0
- virtual bool has_variable ()=0
- virtual Expr * subst (std::string parameter, Expr *expr)=0

Public Attributes

- Expr * **lhs_**
- Expr * **rhs_**

5.3.1 Member Function Documentation

5.3.1.1 equals()

```
bool MultExpr::equals (
            Expr * e )  [virtual]
```

check if two MultExprs are equal

Parameters

e	rhs Expr* to be compared

Returns

true if two MultExprs are equal, null if rhs is NULL

Implements Expr.

5.3.1.2 has_variable()

```
bool MultExpr::has_variable ( )  [virtual]
```

returns true if the expression is a variable or contains a variable

Returns

true if either lhs or rhs is a variable or contains a variable

Implements Expr.

5.3.1.3 interp()

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

returns an int for the value of an expression

Returns

the product of the subexpression values

Implements Expr.

5.3.1.4 subst()

```
Expr * MultExpr::subst (
            std::string parameter,
            Expr * expr )  [virtual]
```

everywhere that the expression contains a variable matching the string, the result Expr* should have the given replacement

Parameters

parameter	the parameter to be substituted
expr	a new expression

Returns

a new object without changing the current object

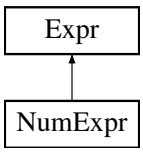
Implements Expr.

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

- /Users/laurazhang/msdscript/expr.h
- /Users/laurazhang/msdscript/expr.cpp

5.4 NumExpr Class Reference

Inheritance diagram for NumExpr:



Public Member Functions

- **NumExpr** (int val)
- bool equals (Expr ∗e)

  check if two NumExprs are equal
- int interp ()

  returns an int for the value of an expression
- bool has_variable ()

  returns true if the expression is a variable or contains a variable
- Expr ∗ subst (std::string parameter, Expr ∗expr)

  everywhere that the expression contains a variable matching the string, the result Expr∗ should have the given replacement

- virtual bool equals (Expr ∗e)=0
- virtual int interp ()=0
- virtual bool has_variable ()=0
- virtual Expr ∗ subst (std::string parameter, Expr ∗expr)=0

Public Attributes

- int **val_**

5.4.1 Member Function Documentation

5.4.1.1 equals()

```
bool NumExpr::equals (
            Expr * e )  [virtual]
```

check if two NumExprs are equal

Parameters

e	rhs Expr∗ to be compared

Returns

true if two NumExprs are equal, null if rhs is NULL

Implements Expr.

5.4.1.2 has_variable()

```
bool NumExpr::has_variable ( )  [virtual]
```

returns true if the expression is a variable or contains a variable

Returns

false

Implements Expr.

5.4.1.3 interp()

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

returns an int for the value of an expression

Returns

the value of a NumExpr

Implements Expr.

5.4.1.4 subst()

```
Expr * NumExpr::subst (
            std::string parameter,
            Expr * expr )  [virtual]
```

everywhere that the expression contains a variable matching the string, the result Expr* should have the given replacement

Parameters

parameter	the parameter to be substituted
expr	a new expression

Returns

a new object without changing the current object

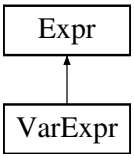
Implements Expr.

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

- /Users/laurazhang/msdscript/expr.h
- /Users/laurazhang/msdscript/expr.cpp

5.5 VarExpr Class Reference

Inheritance diagram for VarExpr:



Public Member Functions

- **VarExpr** (std::string var)
- bool equals (Expr *e)

  check if two VarExprs are equal
- int interp ()

  returns an int for the value of an expression
- bool has_variable ()

  returns true if the expression is a variable or contains a variable
- Expr * subst (std::string parameter, Expr *expr)

  everywhere that the expression contains a variable matching the string, the result Expr should have the given replacement*

- virtual bool equals (Expr *e)=0
- virtual int interp ()=0
- virtual bool has_variable ()=0
- virtual Expr * subst (std::string parameter, Expr *expr)=0

Public Attributes

- std::string **var_**

5.5.1 Member Function Documentation

5.5.1.1 equals()

```
bool VarExpr::equals (
            Expr * e )  [virtual]
```

check if two VarExprs are equal

Parameters

e	rhs Expr* to be compared

Returns

true if two VarExprs are equal, null if rhs is NULL

Implements Expr.

5.5.1.2 has_variable()

```
bool VarExpr::has_variable ( )  [virtual]
```

returns true if the expression is a variable or contains a variable

Returns

true

Implements Expr.

5.5.1.3 interp()

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

returns an int for the value of an expression

Returns

throw an std::runtime_error exception

Implements Expr.

5.5.1.4 subst()

```
Expr * VarExpr::subst (
            std::string parameter,
            Expr * expr )  [virtual]
```

everywhere that the expression contains a variable matching the string, the result Expr* should have the given replacement

Parameters

parameter	the parameter to be substituted
expr	a new expression

Returns

a new object without changing the current object

Implements Expr.

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

- /Users/laurazhang/msdscript/expr.h
- /Users/laurazhang/msdscript/expr.cpp

Chapter 6

File Documentation

6.1 /Users/laurazhang/msdscript/cmdline.h File Reference

Declarations of use_arguments.

Functions

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

6.1.1 Detailed Description

Declarations of use_arguments.

Author

Laura Zhang

6.2 /Users/laurazhang/msdscript/cmdline.h

Go to the documentation of this file.

```
00001
00007 #pragma once
00008
00009 int use_arguments(int argc, char **argv);
```

6.3 /Users/laurazhang/msdscript/expr.cpp File Reference

Definitions of methods in Expr and its child classes.

```
#include "expr.h"
#include <stdexcept>
```

6.3.1 Detailed Description

Definitions of methods in Expr and its child classes.

Author

Laura Zhang

6.4 /Users/laurazhang/msdscript/expr.h File Reference

Declarations of methods in Expr and its child classes.

```
#include <string>
```

Classes

- class Expr
- class NumExpr
- class AddExpr
- class MultExpr
- class VarExpr

6.4.1 Detailed Description

Declarations of methods in Expr and its child classes.

Author

Laura Zhang

6.5 /Users/laurazhang/msdscript/expr.h

Go to the documentation of this file.

```
00001
00007 #pragma once
00008
00009 #include <string>
00010
00017 class Expr {
00018 public:
00019     virtual bool equals(Expr *e) = 0;
00020     virtual int interp() = 0;
00021     virtual bool has_variable() = 0;
00022     virtual Expr* subst(std::string parameter, Expr* expr) = 0;
00023 };
00024
00025 class NumExpr : public Expr {
00026 public:
00027     int val_;
00028
00029     NumExpr(int val);
00030     bool equals(Expr *e);
00031     int interp();
00032     bool has_variable();
00033     Expr* subst(std::string parameter, Expr* expr);
00034 };
```

```
00035
00036 class AddExpr : public Expr {
00037 public:
00038     Expr *lhs_;
00039     Expr *rhs_;
00040
00041     AddExpr(Expr *lhs, Expr *rhs);
00042     bool equals(Expr *e);
00043     int interp();
00044     bool has_variable();
00045     Expr* subst(std::string parameter, Expr* expr);
00046 };
00047
00048 class MultExpr : public Expr {
00049 public:
00050     Expr *lhs_;
00051     Expr *rhs_;
00052
00053     MultExpr(Expr *lhs, Expr *rhs);
00054     bool equals(Expr *e);
00055     int interp();
00056     bool has_variable();
00057     Expr* subst(std::string parameter, Expr* expr);
00058 };
00059
00060 class VarExpr : public Expr {
00061 public:
00062     std::string var_;
00063
00064     VarExpr(std::string var);
00065     bool equals(Expr *e);
00066     int interp();
00067     bool has_variable();
00068     Expr* subst(std::string parameter, Expr* expr);
00069 };
```

Index