msdscript

Generated by Doxygen 1.10.0

Chapter 1

MSDScript

Author

Aiden Pratt

Date

02-06-2024

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

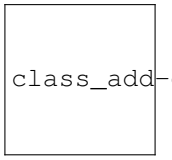
Class Documentation

5.1 Add Class Reference

Add class, child of expression class.

`#include <expr.h>`

Inheritance diagram for Add:



Public Member Functions

- Add (expr *lhs, expr *rhs)
  add constructor implementation
- bool equals (expr *e) override
  checks if add objects are the same
- int interp () override
  interps the add object mathematically
- bool has_variable () override
  checks for a variable in the add object
- expr * subst (std::string string, expr *e) override
  substitutes the passed in string with the passed in expression if applicable
- void print (std::ostream &ostream) override
  prints the add object to the ostream
- void pretty_print (std::ostream &ostream, precedence_t p) override
  prints the add object to the ostream in a pretty way

Public Member Functions inherited from expr

- std::string **to_string** ()
- std::string **to_pp_string** ()
- void **pretty_print_at** (std::ostream &ostream)

Public Attributes

- expr * **lhs**

  left hands side expression of Add class
- expr * **rhs**

  right hands side expression of Add class

5.1.1 Detailed Description

Add class, child of expression class.

5.1.2 Constructor & Destructor Documentation

5.1.2.1 Add()

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

add constructor implementation

Parameters

left	hand side
right	hand side

5.1.3 Member Function Documentation

5.1.3.1 equals()

```
bool Add::equals (
            expr * e )  [override], [virtual]
```

checks if add objects are the same

Parameters

expression	

Returns

true if the add expressions are equal

< object dynamic cast to check if parameter expression object is a add

Implements expr.

5.1.3.2 has_variable()

```
bool Add::has_variable ( )  [override], [virtual]
```

checks for a variable in the add object

Returns

returns true if there is a variable in the add objects lhs or rhs

Implements expr.

5.1.3.3 interp()

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

interps the add object mathematically

Returns

return the mathematical addition equation of of lhs and rhs

Implements expr.

5.1.3.4 pretty_print()

```
void Add::pretty_print (
            std::ostream & ostream,
            precedence_t p )  [override], [virtual]
```

prints the add object to the ostream in a pretty way

Parameters

ostream	@param precedence

Implements expr.

5.1.3.5 print()

```
void Add::print (
            std::ostream & ostream )  [override], [virtual]
```

prints the add object to the ostream

Parameters

ostream	

Implements expr.

5.1.3.6 subst()

```
expr * Add::subst (
            std::string string,
            expr * e )  [override], [virtual]
```

substitutes the passed in string with the passed in expression if applicable

Parameters

string	
expression	

Returns

returns a new add object with the substituted variables if applicable

Implements expr.

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

- /Users/aidenpratt/Documents/Documents - Aiden's MacBook Pro/AP-6015/msdscript/expr.h
- /Users/aidenpratt/Documents/Documents - Aiden's MacBook Pro/AP-6015/msdscript/expr.cpp

5.2 expr Class Reference

expression class

`#include <expr.h>`

Inheritance diagram for expr:

Public Member Functions

- virtual bool equals (expr *e)=0

  Calculates distance between two points.
- virtual int interp ()=0

  definition for interp in expr
- virtual bool has_variable ()=0
- virtual expr * subst (std::string string, expr *e)=0
- virtual void print (std::ostream &)=0
- std::string **to_string** ()
- std::string **to_pp_string** ()
- void **pretty_print_at** (std::ostream &ostream)
- virtual void pretty_print (std::ostream &ostream, precedence_t p)=0

5.2.1 Detailed Description

expression class

5.2.2 Member Function Documentation

5.2.2.1 equals()

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

Calculates distance between two points.

Parameters

expr	*e argument to check equals with

Implemented in Num, Add, Mult, and Var.

5.2.2.2 has_variable()

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

Implemented in Num, Add, Mult, and Var.

5.2.2.3 interp()

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

definition for interp in expr

Returns

0

Implemented in Num, Add, Mult, and Var.

5.2.2.4 pretty_print()

```
virtual void expr::pretty_print (
            std::ostream & ostream,
            precedence_t p )  [pure virtual]
```

Implemented in Add, Mult, Var, and Num.

5.2.2.5 print()

```
virtual void expr::print (
            std::ostream &  )  [pure virtual]
```

Implemented in Num, Add, Mult, and Var.

5.2.2.6 subst()

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

Implemented in Num, Add, Mult, and Var.

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

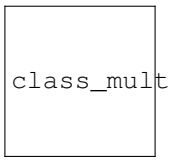
- /Users/aidenpratt/Documents/Documents - Aiden's MacBook Pro/AP-6015/msdscript/expr.h
- /Users/aidenpratt/Documents/Documents - Aiden's MacBook Pro/AP-6015/msdscript/expr.cpp

5.3 Mult Class Reference

Mult class, child of expression class.

`#include <expr.h>`

Inheritance diagram for Mult:



Public Member Functions

- Mult (expr *lhs, expr *rhs)
  mult constructor implementation
- bool equals (expr *e) override
  checks if add objects are the same
- int interp () override
  interps the mult object mathematically
- bool has_variable () override
  checks for a variable in the add object
- expr * subst (std::string string, expr *e) override
  substitutes the passed in string with the passed in expression if applicable
- void print (std::ostream &ostream) override
  prints the add object to the ostream
- void pretty_print (std::ostream &ostream, precedence_t p) override
  prints the add object to the ostream in a pretty way

Public Member Functions inherited from expr

- std::string **to_string** ()
- std::string **to_pp_string** ()
- void **pretty_print_at** (std::ostream &ostream)

Public Attributes

- expr * **lhs**

  left hands side expression of Mult class
- expr * **rhs**

  right hands side expression of Mult class

5.3.1 Detailed Description

Mult class, child of expression class.

5.3.2 Constructor & Destructor Documentation

5.3.2.1 Mult()

```
Mult::Mult (
            expr * lhs,
            expr * rhs )
```

mult constructor implementation

Parameters

left	hand side
right	hand side

5.3.3 Member Function Documentation

5.3.3.1 equals()

```
bool Mult::equals (
            expr * e )  [override], [virtual]
```

checks if add objects are the same

Parameters

expression	

Returns

true if the mult expressions are equal

< object dynamic cast to check if parameter expression object is a mult

Implements expr.

5.3.3.2 has_variable()

```
bool Mult::has_variable ( )  [override], [virtual]
```

checks for a variable in the add object

Returns

returns true if there is a variable in the add objects lhs or rhs

Implements expr.

5.3.3.3 interp()

```
int Mult::interp ( )  [override], [virtual]
```

interps the mult object mathematically

Returns

return the mathematical addition equation of of lhs and rhs

Implements expr.

5.3.3.4 pretty_print()

```
void Mult::pretty_print (
            std::ostream & ostream,
            precedence_t p )  [override], [virtual]
```

prints the add object to the ostream in a pretty way

Parameters

ostream	@param precedence

Implements expr.

5.3.3.5 print()

```
void Mult::print (
```

```
            std::ostream & ostream )  [override], [virtual]
```

prints the add object to the ostream

Parameters

ostream	

Implements expr.

5.3.3.6 subst()

```
expr * Mult::subst (
            std::string string,
            expr * e )  [override], [virtual]
```

substitutes the passed in string with the passed in expression if applicable

Parameters

string	
expression	

Returns

returns a new add object with the substituted variables if applicable

Implements expr.

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

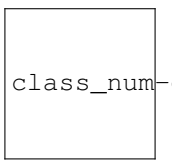
- /Users/aidenpratt/Documents/Documents - Aiden's MacBook Pro/AP-6015/msdscript/expr.h
- /Users/aidenpratt/Documents/Documents - Aiden's MacBook Pro/AP-6015/msdscript/expr.cpp

5.4 Num Class Reference

Num class, child of expression class.

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

Inheritance diagram for Num:



Public Member Functions

- Num (int val)

  num constructor implementation
- bool equals (expr *e) override

  checking equals
- int interp () override

  interps the num value
- bool has_variable () override

  checks if this object is a variable
- expr * subst (std::string string, expr *e) override

  implementation of subst in num
- void print (std::ostream &ostream) override

  prints this value to the provided output stream
- void pretty_print (std::ostream &override, precedence_t p) override

  prints this value to the stream, calls print

Public Member Functions inherited from expr

- std::string **to_string** ()
- std::string **to_pp_string** ()
- void **pretty_print_at** (std::ostream &ostream)

Public Attributes

- int **val**

  int that will be the Num value

5.4.1 Detailed Description

Num class, child of expression class.

5.4.2 Constructor & Destructor Documentation

5.4.2.1 Num()

```
Num::Num (
            int val )  [explicit]
```

num constructor implementation

Parameters

val	

5.4.3 Member Function Documentation

5.4.3.1 equals()

```
bool Num::equals (
            expr * e )  [override], [virtual]
```

checking equals

Parameters

expression	

Returns

true or false if the values are equal to each other

< object dynamic cast to check if parameter expression object is a num

Implements expr.

5.4.3.2 has_variable()

```
bool Num::has_variable ( )  [override], [virtual]
```

checks if this object is a variable

Returns

returns that this num object is not a variable

Implements expr.

5.4.3.3 interp()

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

interps the num value

Returns

the val of this num object

Implements expr.

5.4.3.4 pretty_print()

```
void Num::pretty_print (
            std::ostream & ostream,
            precedence_t p )  [override], [virtual]
```

prints this value to the stream, calls print

Parameters

ostream	
precedence	

Implements expr.

5.4.3.5 print()

```
void Num::print (
            std::ostream & ostream )  [override], [virtual]
```

prints this value to the provided output stream

Parameters

ostream	

Implements expr.

5.4.3.6 subst()

```
expr * Num::subst (
            std::string string,
            expr * e )  [override], [virtual]
```

implementation of subst in num

Returns

returns a new num of the same value

Implements expr.

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

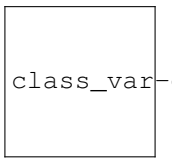
- /Users/aidenpratt/Documents/Documents - Aiden's MacBook Pro/AP-6015/msdscript/expr.h
- /Users/aidenpratt/Documents/Documents - Aiden's MacBook Pro/AP-6015/msdscript/expr.cpp

5.5 Var Class Reference

Var class, child of expression class.

`#include <expr.h>`

Inheritance diagram for Var:



Public Member Functions

- Var (std::string var)

  constructor for var class
- bool equals (expr ∗e) override

  checks if var objects are teh same
- int interp () override

  interp implementation for var, returns a runtime error
- bool has_variable () override

  checks if this object is a variable
- expr ∗ subst (std::string string, expr ∗e) override

  substitute method implementation for var class
- void print (std::ostream &ostream) override

  print implementation for var class, prints var to ostream
- void pretty_print (std::ostream &ostream, precedence_t p) override

  pretty print implementation for var class

Public Member Functions inherited from expr

- std::string **to_string** ()
- std::string **to_pp_string** ()
- void **pretty_print_at** (std::ostream &ostream)

Public Attributes

- std::string **var**

  string that will be the var

5.5.1 Detailed Description

Var class, child of expression class.

5.5.2 Constructor & Destructor Documentation

5.5.2.1 Var()

```
Var::Var (
            std::string var )
```

constructor for var class

Parameters

var	

5.5.3 Member Function Documentation

5.5.3.1 equals()

```
bool Var::equals (
            expr * e )  [override], [virtual]
```

checks if var objects are teh same

Parameters

e	

Returns

returns true if the vars are the equals

< object dynamic cast to check if parameter expression object is a var

Implements expr.

5.5.3.2 has_variable()

```
bool Var::has_variable ( )  [override], [virtual]
```

checks if this object is a variable

Returns

returns that this var object is a variable

Implements expr.

5.5.3.3 interp()

```
int Var::interp ( )  [override], [virtual]
```

interp implementation for var, returns a runtime error

Implements expr.

5.5.3.4 pretty_print()

```
void Var::pretty_print (
            std::ostream & ostream,
            precedence_t p )  [override], [virtual]
```

pretty print implementation for var class

Parameters

ostream	
precedence	

Implements expr.

5.5.3.5 print()

```
void Var::print (
            std::ostream & ostream )  [override], [virtual]
```

print implementation for var class, prints var to ostream

Parameters

ostream	

Implements expr.

5.5.3.6 subst()

```
expr * Var::subst (
            std::string string,
            expr * e )  [override], [virtual]
```

substitute method implementation for var class

Parameters

string	
expression	

Returns

returns a new var object

Implements expr.

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

- /Users/aidenpratt/Documents/Documents - Aiden's MacBook Pro/AP-6015/msdscript/expr.h
- /Users/aidenpratt/Documents/Documents - Aiden's MacBook Pro/AP-6015/msdscript/expr.cpp

Chapter 6

File Documentation

6.1 /Users/aidenpratt/Documents/Documents - Aiden's MacBook Pro/AP-6015/msdscript/cmdline.cpp File Reference

implementation of use_arguments for the command line

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

Functions

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

6.1.1 Detailed Description

implementation of use_arguments for the command line

6.2 /Users/aidenpratt/Documents/Documents - Aiden's MacBook Pro/AP-6015/msdscript/cmdline.hpp File Reference

file with declaration of use_arguments

Functions

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

6.2.1 Detailed Description

file with declaration of use_arguments

Parameters

argc	
argv	

6.3 /Users/aidenpratt/Documents/Documents - Aiden's MacBook Pro/AP-6015/msdscript/cmdline.hpp

Go to the documentation of this file.

```
00001 //
00002 // Created by Aiden Pratt on 1/11/24.
00003 //
00004 #ifndef msdscript_CMDLINE_H
00005 #define msdscript_CMDLINE_H
00006
00007
00015 void use_arguments(int argc, char **argv);
00016
00017
00018
00019 #endif //msdscript_CMDLINE_H
```

6.4 /Users/aidenpratt/Documents/Documents - Aiden's MacBook Pro/AP-6015/msdscript/expr.cpp File Reference

contains expression class and children classes: num, add, mult, var

```
#include "expr.h"
#include "catch.h"
#include <utility>
#include <stdexcept>
#include <iostream>
#include <sstream>
```

6.4.1 Detailed Description

contains expression class and children classes: num, add, mult, var

Author

Aiden Pratt

6.5 /Users/aidenpratt/Documents/Documents - Aiden's MacBook Pro/AP-6015/msdscript/expr.h File Reference

contains declarations for methods of each class

```
#include <string>
#include <sstream>
```

Classes

- class expr

  expression class
- class Num

  Num class, child of expression class.
- class Add

  Add class, child of expression class.
- class Mult

  Mult class, child of expression class.
- class Var

  Var class, child of expression class.

Enumerations

- enum **precedence_t** { **prec_none** = 0 , **prec_add** = 1 , **prec_mult** = 2 , **prec_let** = 3 }

6.5.1 Detailed Description

contains declarations for methods of each class

Author

Aiden Pratt

6.6 /Users/aidenpratt/Documents/Documents - Aiden's MacBook Pro/AP-6015/msdscript/expr.h

Go to the documentation of this file.

```
00001 //
00002 // Created by Aiden Pratt on 1/16/24.
00003 //
00004
00005 #ifndef msdscript_EXPR_H
00006 #define msdscript_EXPR_H
00007 #include <string>
00008 #include <sstream>
00009
00020 typedef enum {
00021     prec_none = 0,
00022     prec_add = 1,
00023     prec_mult = 2,
00024     prec_let = 3
00025 } precedence_t;
00026
00030
      //-----------------------------------------EXPR-----------------------------------------------------------//
00031 class expr {
00032 public:
00033     virtual bool equals(expr *e) = 0;//each subclass must override
00034     virtual int interp() = 0;
00035     virtual bool has_variable() = 0;
00036     virtual expr* subst(std::string string , expr *e) = 0;
00037     virtual void print(std::ostream&) = 0;
00038
00039     std::string to_string(){
00040         std::stringstream st("");
00041         this->print(st);
00042         return st.str();
00043     }
00044
00045     std::string to_pp_string(){
```

```
00046         std::stringstream st("");
00047         this->pretty_print_at(st);
00048         return st.str();
00049     }
00050
00051     void pretty_print_at(std::ostream &ostream){
00052        this->pretty_print(ostream, prec_none);
00053     }
00054
00055     virtual void pretty_print(std::ostream &ostream, precedence_t p) = 0;
00056 };
00057
00058
      //----------------------------------------------NUM-----------------------------------------------//
00059
00063 class Num : public expr {
00064 public:
00065     int val;
00066     explicit Num(int val);
00067     bool equals(expr *e) override;
00068     int interp() override;
00069     bool has_variable() override;
00070     expr* subst(std::string string , expr *e) override;
00071     void print(std::ostream &ostream) override;
00072     void pretty_print(std::ostream &override, precedence_t p) override;
00073
00074 };
00075
00076
      //----------------------------------------------ADD-----------------------------------------------//
00077
00081 class Add : public expr{
00082 public:
00083     expr *lhs;
00084     expr *rhs;
00085     Add(expr *lhs, expr *rhs);
00086     bool equals(expr *e) override;
00087     int interp() override;
00088     bool has_variable() override;
00089     expr* subst(std::string string , expr *e) override;
00090     void print(std::ostream &ostream) override;
00091     void pretty_print(std::ostream &ostream, precedence_t p) override;
00092 };
00093
00094
      //----------------------------------------------MULT----------------------------------------------//
00095
00099 class Mult : public expr {
00100 public:
00101     expr *lhs;
00102     expr *rhs;
00103     Mult(expr *lhs, expr *rhs);
00104     bool equals(expr *e) override;
00105     int interp() override;
00106     bool has_variable() override;
00107     expr* subst(std::string string , expr *e) override;
00108     void print(std::ostream &ostream) override;
00109     void pretty_print(std::ostream &ostream, precedence_t p) override;
00110 };
00111
00112
      //----------------------------------------------VAR-----------------------------------------------//
00113
00117 class Var : public expr{
00118 public:
00119     std::string var;
00120     Var(std::string var);
00121     bool equals(expr *e) override;
00122     int interp() override;
00123     bool has_variable() override;
00124     expr* subst(std::string string , expr *e) override;
00125     void print(std::ostream &ostream) override;
00126     void pretty_print(std::ostream &ostream, precedence_t p) override;
00127 };
00128
00129
00130 #endif //ASSIGNMENT2_EXPR_H
```

6.7 /Users/aidenpratt/Documents/Documents - Aiden's MacBook Pro/AP-6015/msdscript/main.cpp File Reference

main file for calling use_arguments

```
#include <iostream>
#include "cmdline.hpp"
```

Functions

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

6.7.1 Detailed Description

main file for calling use_arguments