MSDScript

Generated by Doxygen 1.10.0

1 MSDScript 1

2 Hierarchical Index 3

2.1 Class Hierarchy . . . 3

3 Class Index 5

3.1 Class List . . . 5

4 File Index 7

4.1 File List . . . 7

5 Class Documentation 9

5.1 _let Class Reference . . . 9

5.1.1 Member Function Documentation . . . 10

5.1.1.1 equals() . . . 10

5.1.1.2 has_variable() . . . 10

5.1.1.3 interp() . . . 10

5.1.1.4 pretty_print_at() . . . 10

5.1.1.5 print() . . . 10

5.1.1.6 subst() . . . 11

5.2 Add Class Reference . . . 11

5.2.1 Constructor & Destructor Documentation . . . 12

5.2.1.1 Add() . . . 12

5.2.2 Member Function Documentation . . . 12

5.2.2.1 equals() . . . 12

5.2.2.2 has_variable() . . . 12

5.2.2.3 interp() . . . 13

5.2.2.4 pretty_print_at() . . . 13

5.2.2.5 print() . . . 13

5.2.2.6 subst() . . . 13

5.3 Expr Class Reference . . . 14

5.3.1 Member Function Documentation . . . 14

5.3.1.1 equals() . . . 14

5.3.1.2 has_variable() . . . 15

5.3.1.3 interp() . . . 15

5.3.1.4 pretty_print() . . . 15

5.3.1.5 pretty_print_at() . . . 15

5.3.1.6 print() . . . 15

5.3.1.7 subst() . . . 16

5.3.1.8 to_pretty_string() . . . 16

5.3.1.9 to_string() . . . 16

5.4 Mult Class Reference . . . 16

5.4.1 Constructor & Destructor Documentation . . . 17

5.4.1.1 Mult() . . . 17

5.4.2 Member Function Documentation . . . 18
5.4.2.1 equals() . . . 18
5.4.2.2 has_variable() . . . 18
5.4.2.3 interp() . . . 18
5.4.2.4 pretty_print_at() . . . 18
5.4.2.5 print() . . . 19
5.4.2.6 subst() . . . 19
5.5 Num Class Reference . . . 19
5.5.1 Constructor & Destructor Documentation . . . 20
5.5.1.1 Num() . . . 20
5.5.2 Member Function Documentation . . . 21
5.5.2.1 equals() . . . 21
5.5.2.2 has_variable() . . . 21
5.5.2.3 interp() . . . 21
5.5.2.4 print() . . . 21
5.5.2.5 subst() . . . 22
5.6 Var Class Reference . . . 22
5.6.1 Constructor & Destructor Documentation . . . 23
5.6.1.1 Var() . . . 23
5.6.2 Member Function Documentation . . . 23
5.6.2.1 equals() . . . 23
5.6.2.2 has_variable() . . . 24
5.6.2.3 interp() . . . 24
5.6.2.4 print() . . . 24
5.6.2.5 subst() . . . 25

6 File Documentation **27**

6.1 /Users/corinnejones/GitHubSchool/Spring2024/CS6015_SoftwareEngineering/MSDScript/MSDScript/cmdline.hpp File Reference . . . 27
6.1.1 Detailed Description . . . 27
6.2 /Users/corinnejones/GitHubSchool/Spring2024/CS6015_SoftwareEngineering/MSDScript/MSDScript/cmdline.hpp 28
6.3 /Users/corinnejones/GitHubSchool/Spring2024/CS6015_SoftwareEngineering/MSDScript/MSDScript/↩ Expr.cpp File Reference . . . 28
6.3.1 Detailed Description . . . 28
6.4 /Users/corinnejones/GitHubSchool/Spring2024/CS6015_SoftwareEngineering/MSDScript/MSDScript/↩ Expr.hpp File Reference . . . 28
6.4.1 Detailed Description . . . 29
6.5 /Users/corinnejones/GitHubSchool/Spring2024/CS6015_SoftwareEngineering/MSDScript/MSDScript/↩ Expr.hpp . . . 29
6.6 /Users/corinnejones/GitHubSchool/Spring2024/CS6015_SoftwareEngineering/MSDScript/MSDScript/main.cpp File Reference . . . 31
6.6.1 Detailed Description . . . 31
6.7 /Users/corinnejones/GitHubSchool/Spring2024/CS6015_SoftwareEngineering/MSDScript/MSDScript/↩ Tests.cpp File Reference . . . 32

6.7.1 Detailed Description 32

Index **33**

Chapter 1

MSDScript

Author

Corinne Jones

Date

01-16-2024

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Expr 14
- Add 11
- Mult 16
- Num 19
- Var 22
- _let 9

Chapter 3

Class Index

3.1 Class List

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

_let 9
Add 11
Expr 14
Mult 16
Num 19
Var 22

Chapter 4

File Index

4.1 File List

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

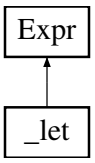
/Users/corinnejones/GitHubSchool/Spring2024/CS6015_SoftwareEngineering/MSDScript/MSDScript/cmdline.hpp
Command line arguments handler for the ExpressionClasses project 27
/Users/corinnejones/GitHubSchool/Spring2024/CS6015_SoftwareEngineering/MSDScript/MSDScript/Expr.cpp
Implementation of expression classes for arithmetic operations 28
/Users/corinnejones/GitHubSchool/Spring2024/CS6015_SoftwareEngineering/MSDScript/MSDScript/Expr.hpp
Declaration of expression classes for arithmetic operations 28
/Users/corinnejones/GitHubSchool/Spring2024/CS6015_SoftwareEngineering/MSDScript/MSDScript/main.cpp
Command line argument handler for test execution 31
/Users/corinnejones/GitHubSchool/Spring2024/CS6015_SoftwareEngineering/MSDScript/MSDScript/Tests.cpp
Tests for the Expression Classes 32

Chapter 5

Class Documentation

5.1 _let Class Reference

Inheritance diagram for _let:



Public Member Functions

- **_let** (Var *lhs, Expr *rhs, Expr *body)
- virtual bool equals (Expr *e)
- virtual int interp ()
- virtual bool has_variable ()
- virtual Expr * subst (string str, Expr *e)
- virtual void print (ostream &ostream)
- void pretty_print_at (ostream &ostream, precedence_t prec)

  Pretty prints the expression at a given precedence.

Public Member Functions inherited from Expr

- string to_string ()

  Converts expression to string representation.
- void pretty_print (ostream &ostream)

  Pretty prints the expression.
- string to_pretty_string ()

  Converts the expression to a pretty string.

Public Attributes

- Var * **lhs**
- Expr * **rhs**
- Expr * **body**

5.1.1 Member Function Documentation

5.1.1.1 equals()

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

Implements Expr.

5.1.1.2 has_variable()

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

Implements Expr.

5.1.1.3 interp()

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

Implements Expr.

5.1.1.4 pretty_print_at()

```
void _let::pretty_print_at (
            ostream & ostream,
            precedence_t prec )  [virtual]
```

Pretty prints the expression at a given precedence.

Parameters

ostream	The output stream to print to.
prec	The precedence context in which to print.

Reimplemented from Expr.

5.1.1.5 print()

```
void _let::print (
            ostream & ostream )  [virtual]
```

Implements Expr.

5.1.1.6 subst()

```
Expr * _let::subst (
            string str,
            Expr * e )  [virtual]
```

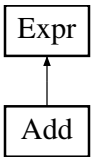
Implements Expr.

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

- /Users/corinnejones/GitHubSchool/Spring2024/CS6015_SoftwareEngineering/MSDScript/MSDScript/Expr.hpp
- /Users/corinnejones/GitHubSchool/Spring2024/CS6015_SoftwareEngineering/MSDScript/MSDScript/Expr.cpp

5.2 Add Class Reference

Inheritance diagram for Add:



Public Member Functions

- Add (Expr *lhs, Expr *rhs)

  Constructs an addition expression.
- virtual bool equals (Expr *e)

  Checks equality of this expression with another expression.
- virtual int interp ()

  Interprets the addition of expressions.
- virtual bool has_variable ()

  Checks if the expression contains a variable.
- virtual Expr * subst (string str, Expr *e)

  Substitutes a variable in the expression with another expression.
- virtual void print (ostream &ostream)

  Prints the addition expression.
- void pretty_print_at (ostream &ostream, precedence_t prec)

  Pretty prints the addition expression with proper precedence.

Public Member Functions inherited from Expr

- string to_string ()

  Converts expression to string representation.
- void pretty_print (ostream &ostream)

  Pretty prints the expression.
- string to_pretty_string ()

  Converts the expression to a pretty string.

Public Attributes

- Expr * **lhs**
- Expr * **rhs**

5.2.1 Constructor & Destructor Documentation

5.2.1.1 Add()

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

Constructs an addition expression.

Parameters

lhs	Left-hand side expression.
rhs	Right-hand side expression.

5.2.2 Member Function Documentation

5.2.2.1 equals()

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

Checks equality of this expression with another expression.

Parameters

e	The expression to compare with.

Returns

True if equal, false otherwise.

Implements Expr.

5.2.2.2 has_variable()

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

Checks if the expression contains a variable.

Returns

True if a variable is present, false otherwise.

Implements Expr.

5.2.2.3 interp()

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

Interprets the addition of expressions.

Returns

The result of the addition.

Implements Expr.

5.2.2.4 pretty_print_at()

```
void Add::pretty_print_at (
            ostream & ostream,
            precedence_t prec )  [virtual]
```

Pretty prints the addition expression with proper precedence.

Parameters

ostream	The output stream.
prec	The precedence level.

Reimplemented from Expr.

5.2.2.5 print()

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

Prints the addition expression.

Parameters

ostream	The output stream.

Implements Expr.

5.2.2.6 subst()

```
Expr * Add::subst (
            string str,
            Expr * e )  [virtual]
```

Substitutes a variable in the expression with another expression.

Parameters

str	The variable to substitute.
e	The expression to substitute with.

Returns

The new expression after substitution.

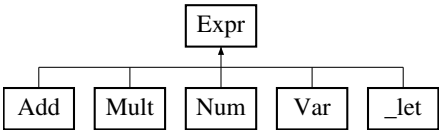
Implements Expr.

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

- /Users/corinnejones/GitHubSchool/Spring2024/CS6015_SoftwareEngineering/MSDScript/MSDScript/Expr.hpp
- /Users/corinnejones/GitHubSchool/Spring2024/CS6015_SoftwareEngineering/MSDScript/MSDScript/Expr.cpp

5.3 Expr Class Reference

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 (string str, Expr *e)=0
- virtual void print (ostream &ostream)=0
- string to_string ()

  Converts expression to string representation.
- virtual void pretty_print_at (ostream &ostream, precedence_t prec)

  Pretty prints the expression at a given precedence.
- void pretty_print (ostream &ostream)

  Pretty prints the expression.
- string to_pretty_string ()

  Converts the expression to a pretty string.

5.3.1 Member Function Documentation

5.3.1.1 equals()

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

Implemented in Add, Mult, Num, and Var.

5.3.1.2 has_variable()

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

Implemented in Add, Mult, Num, and Var.

5.3.1.3 interp()

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

Implemented in Add, Mult, Num, and Var.

5.3.1.4 pretty_print()

```
void Expr::pretty_print (
            ostream & ostream )
```

Pretty prints the expression.

Parameters

ostream	The output stream.

5.3.1.5 pretty_print_at()

```
void Expr::pretty_print_at (
            ostream & ostream,
            precedence_t prec )  [virtual]
```

Pretty prints the expression at a given precedence.

Parameters

ostream	The output stream to print to.
prec	The precedence context in which to print.

Reimplemented in Add, and _let.

5.3.1.6 print()

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

Implemented in Add, Mult, Num, and Var.

5.3.1.7 subst()

```
virtual Expr * Expr::subst (
            string str,
            Expr * e )  [pure virtual]
```

Implemented in Add, Mult, Num, and Var.

5.3.1.8 to_pretty_string()

```
string Expr::to_pretty_string ( )
```

Converts the expression to a pretty string.

Returns

A pretty string representation of the expression.

5.3.1.9 to_string()

```
string Expr::to_string ( )
```

Converts expression to string representation.

Returns

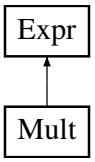
String representation of the expression.

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

- /Users/corinnejones/GitHubSchool/Spring2024/CS6015_SoftwareEngineering/MSDScript/MSDScript/Expr.hpp
- /Users/corinnejones/GitHubSchool/Spring2024/CS6015_SoftwareEngineering/MSDScript/MSDScript/Expr.cpp

5.4 Mult Class Reference

Inheritance diagram for Mult:



Public Member Functions

- Mult (Expr *lhs, Expr *rhs)

  Constructs a multiplication expression.
- virtual bool equals (Expr *e)

  Checks if this expression is equal to another expression.
- virtual int interp ()

  Evaluates the multiplication of the two expressions.
- virtual bool has_variable ()

  Determines if the expression contains a variable.
- virtual Expr * subst (string str, Expr *e)

  Substitutes a variable within the expression.
- virtual void print (ostream &ostream)

  Prints the expression to the provided output stream.
- void pretty_print_at (std::ostream &ostream, precedence_t prec)

  Pretty prints the expression with precedence handling.

Public Member Functions inherited from Expr

- string to_string ()

  Converts expression to string representation.
- virtual void pretty_print_at (ostream &ostream, precedence_t prec)

  Pretty prints the expression at a given precedence.
- void pretty_print (ostream &ostream)

  Pretty prints the expression.
- string to_pretty_string ()

  Converts the expression to a pretty string.

Public Attributes

- Expr * **lhs**
- Expr * **rhs**

5.4.1 Constructor & Destructor Documentation

5.4.1.1 Mult()

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

Constructs a multiplication expression.

Parameters

lhs	Left-hand side expression.
rhs	Right-hand side expression.

5.4.2 Member Function Documentation

5.4.2.1 equals()

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

Checks if this expression is equal to another expression.

Parameters

e	The expression to compare with.

Returns

True if the expressions are equal, false otherwise.

Implements Expr.

5.4.2.2 has_variable()

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

Determines if the expression contains a variable.

Returns

True if a variable is present, false otherwise.

Implements Expr.

5.4.2.3 interp()

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

Evaluates the multiplication of the two expressions.

Returns

The integer result of the multiplication.

Implements Expr.

5.4.2.4 pretty_print_at()

```
void Mult::pretty_print_at (
            std::ostream & ostream,
            precedence_t prec )
```

Pretty prints the expression with precedence handling.

Parameters

ostream	The output stream.
prec	The current precedence level.

5.4.2.5 print()

```
void Mult::print (
            ostream & ostream )  [virtual]
```

Prints the expression to the provided output stream.

Parameters

ostream	The output stream.

Implements Expr.

5.4.2.6 subst()

```
Expr * Mult::subst (
            string str,
            Expr * e )  [virtual]
```

Substitutes a variable within the expression.

Parameters

str	The variable name to replace.
e	The expression to replace it with.

Returns

A new expression with the substitution made.

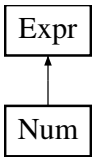
Implements Expr.

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

- /Users/corinnejones/GitHubSchool/Spring2024/CS6015_SoftwareEngineering/MSDScript/MSDScript/Expr.hpp
- /Users/corinnejones/GitHubSchool/Spring2024/CS6015_SoftwareEngineering/MSDScript/MSDScript/Expr.cpp

5.5 Num Class Reference

Inheritance diagram for Num:



Public Member Functions

- Num (int val)

  Initializes a numeric constant expression.
- virtual bool equals (Expr ∗e)

  Checks if this numeric constant is equal to another expression.
- virtual int interp ()

  Evaluates to its numeric value.
- virtual bool has_variable ()

  Checks if the numeric constant contains a variable.
- virtual Expr ∗ subst (string str, Expr ∗e)

  Substitutes a variable within this numeric expression. Since `Num` does not contain variables, it returns itself.
- virtual void print (ostream &ostream)

  Prints the numeric value to the specified output stream.

Public Member Functions inherited from Expr

- string to_string ()

  Converts expression to string representation.
- virtual void pretty_print_at (ostream &ostream, precedence_t prec)

  Pretty prints the expression at a given precedence.
- void pretty_print (ostream &ostream)

  Pretty prints the expression.
- string to_pretty_string ()

  Converts the expression to a pretty string.

Public Attributes

- int **val**

5.5.1 Constructor & Destructor Documentation

5.5.1.1 Num()

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

Initializes a numeric constant expression.

Parameters

val	The numeric value of the expression.

5.5.2 Member Function Documentation

5.5.2.1 equals()

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

Checks if this numeric constant is equal to another expression.

Parameters

e	A pointer to the expression to compare with this numeric constant.

Returns

True if the expressions are equal (i.e., if `e` is also a `Num` with the same value), false otherwise.

Implements Expr.

5.5.2.2 has_variable()

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

Checks if the numeric constant contains a variable.

Returns

False, as numeric constants do not contain variables.

Implements Expr.

5.5.2.3 interp()

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

Evaluates to its numeric value.

Returns

The value of the numeric constant.

Implements Expr.

5.5.2.4 print()

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

Prints the numeric value to the specified output stream.

Parameters

ostream	The output stream where the numeric value will be printed.

Implements Expr.

5.5.2.5 subst()

```
Expr * Num::subst (
            string str,
            Expr * e )  [virtual]
```

Substitutes a variable within this numeric expression. Since Num does not contain variables, it returns itself.

Parameters

str	The variable name to look for substitution.
e	The expression to substitute in place of the variable.

Returns

A pointer to this numeric constant, as no substitution occurs.

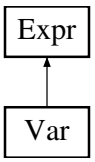
Implements Expr.

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

- /Users/corinnejones/GitHubSchool/Spring2024/CS6015_SoftwareEngineering/MSDScript/MSDScript/Expr.hpp
- /Users/corinnejones/GitHubSchool/Spring2024/CS6015_SoftwareEngineering/MSDScript/MSDScript/Expr.cpp

5.6 Var Class Reference

Inheritance diagram for Var:



Public Member Functions

- Var (string val)

  Constructs a variable expression.
- virtual bool equals (Expr *e)

  Checks if this variable expression is equal to another expression.
- virtual int interp ()

Throws an exception since variables cannot be directly interpreted.

- virtual bool has_variable ()

  Checks if the expression contains a variable.

- virtual Expr * subst (string str, Expr *e)

  Substitutes the variable with another expression if it matches the variable name.

- virtual void print (ostream &ostream)

  Prints the variable's name to the provided output stream.

Public Member Functions inherited from Expr

- string to_string ()

  Converts expression to string representation.

- virtual void pretty_print_at (ostream &ostream, precedence_t prec)

  Pretty prints the expression at a given precedence.

- void pretty_print (ostream &ostream)

  Pretty prints the expression.

- string to_pretty_string ()

  Converts the expression to a pretty string.

Public Attributes

- string **val**

5.6.1 Constructor & Destructor Documentation

5.6.1.1 Var()

```
Var::Var (
            string val )
```

Constructs a variable expression.

Parameters

val	The name of the variable.

5.6.2 Member Function Documentation

5.6.2.1 equals()

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

Checks if this variable expression is equal to another expression.

Parameters

e	A pointer to the expression to compare with this variable expression.

Returns

True if `e` is a `Var` object with the same variable name, false otherwise.

Implements Expr.

5.6.2.2 has_variable()

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

Checks if the expression contains a variable.

Returns

True, as this object represents a variable.

Implements Expr.

5.6.2.3 interp()

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

Throws an exception since variables cannot be directly interpreted.

Exceptions

std::runtime_error	when attempted to interpret a variable.

Implements Expr.

5.6.2.4 print()

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

Prints the variable's name to the provided output stream.

Parameters

ostream	The output stream.

Implements Expr.

5.6.2.5 subst()

```
Expr * Var::subst (
            string str,
            Expr * e )  [virtual]
```

Substitutes the variable with another expression if it matches the variable name.

Parameters

str	The name of the variable to substitute.
e	The expression to substitute in place of the variable.

Returns

The original variable or the substitution.

Implements Expr.

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

- /Users/corinnejones/GitHubSchool/Spring2024/CS6015_SoftwareEngineering/MSDScript/MSDScript/Expr.hpp
- /Users/corinnejones/GitHubSchool/Spring2024/CS6015_SoftwareEngineering/MSDScript/MSDScript/Expr.cpp

Chapter 6

File Documentation

6.1 /Users/corinnejones/GitHubSchool/Spring2024/CS6015_Software↩ Engineering/MSDScript/MSDScript/cmdline.hpp File Reference

Command line arguments handler for the ExpressionClasses project.

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

Functions

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

6.1.1 Detailed Description

Command line arguments handler for the ExpressionClasses project.

Provides a utility function to process and utilize command line arguments passed to an application. This includes parsing, validation, and possibly setting up initial parameters or configurations based on the arguments provided during the execution of the program.

Author

Corinne Jones

Date

1/16/24

6.2 /Users/corinnejones/GitHubSchool/Spring2024/CS6015_Software↩ Engineering/MSDScript/MSDScript/cmdline.hpp

Go to the documentation of this file.

```
00001
00013 #pragma once
00014
00015 #include <stdio.h>
00016 #include <iostream>
00017 #include <string>
00018
00019 using namespace std;
00020
00021 void use_arguments(int argc, char **argv);
```

6.3 /Users/corinnejones/GitHubSchool/Spring2024/CS6015_Software↩ Engineering/MSDScript/MSDScript/Expr.cpp File Reference

Implementation of expression classes for arithmetic operations.

```
#include "Expr.hpp"
```

6.3.1 Detailed Description

Implementation of expression classes for arithmetic operations.

Provides the implementation for various expressions including addition, multiplication, numbers, and variables within an expression evaluation context.

Created by Corinne Jones on 1/16/24.

6.4 /Users/corinnejones/GitHubSchool/Spring2024/CS6015_Software↩ Engineering/MSDScript/MSDScript/Expr.hpp File Reference

Declaration of expression classes for arithmetic operations.

```
#include <stdlib.h>
#include <stdio.h>
#include <string>
#include <stdexcept>
#include <sstream>
```

Classes

- class Expr
- class Add
- class Mult
- class Num
- class Var
- class _let

Enumerations

- enum **precedence_t** { **prec_none** , **prec_add** , **prec_mult** }

6.4.1 Detailed Description

Declaration of expression classes for arithmetic operations.

Defines the abstract base class Expr and its derived classes such as Add, Mult, Num, and Var. These classes are used to construct and evaluate arithmetic expressions involving basic operations and variables. Includes functionality for equality checking, evaluation, variable substitution, and pretty printing of expressions.

Author

Corinne Jones

Date

1/16/24

6.5 /Users/corinnejones/GitHubSchool/Spring2024/CS6015_Software↩ Engineering/MSDScript/MSDScript/Expr.hpp

Go to the documentation of this file.

```
00001
00013 #pragma once
00014
00015 #include <stdlib.h>
00016 #include <stdio.h>
00017 #include <string>
00018 #include <stdexcept>
00019 #include <sstream>
00020
00021 using namespace std;
00022
00023 typedef enum {
00024   prec_none,      // = 0
00025   prec_add,       // = 1
00026   prec_mult       // = 2
00027 } precedence_t;
00028
00029 class Expr {
00030 public:
00031     virtual bool equals (Expr *e)=0;
00032     virtual int interp()=0;
00033     virtual bool has_variable()=0;
00034     virtual Expr* subst(string str, Expr* e)=0;
00035     virtual void print(ostream &ostream)=0;
00036
00037     string to_string();
00038     virtual void pretty_print_at(ostream &ostream, precedence_t prec);
00039     void pretty_print(ostream &ostream);
00040     string to_pretty_string();
00041
00042 };
00043
00044 //======================  ADD  ======================//
00045
00046 class Add : public Expr {
00047
00048 public:
00049     Expr* lhs;
00050     Expr* rhs;
00051
00052     Add(Expr* lhs, Expr* rhs);
00053
00054     virtual bool equals(Expr* e);
```

```
00055
00056     virtual int interp();
00057
00058     virtual bool has_variable();
00059
00060     virtual Expr* subst(string str, Expr* e);
00061
00062     virtual void print(ostream &ostream);
00063
00064     void pretty_print_at(ostream &ostream, precedence_t prec);
00065
00066 };
00067
00068 //======================  MULT  ======================//
00069
00070 class Mult : public Expr {
00071 public:
00072     Expr* lhs;
00073     Expr* rhs;
00074
00075     Mult(Expr* lhs, Expr* rhs);
00076
00077     virtual bool equals(Expr* e);
00078
00079     virtual int interp();
00080
00081     virtual bool has_variable();
00082
00083     virtual Expr* subst(string str, Expr* e);
00084
00085     virtual void print (ostream &ostream);
00086
00087     void pretty_print_at(std::ostream &ostream, precedence_t prec);
00088
00089 };
00090
00091 //======================  NUM  ======================//
00092
00093 class Num : public Expr {
00094 public:
00095     int val;
00096
00097     Num(int val);
00098
00099     virtual bool equals(Expr* e);
00100
00101     virtual int interp();
00102
00103     virtual bool has_variable();
00104
00105     virtual Expr* subst(string str, Expr* e);
00106
00107     virtual void print (ostream &ostream);
00108
00109 };
00110
00111 class Var : public Expr {
00112
00113 public:
00114     string val;
00115
00116     Var (string val);
00117
00118     virtual bool equals(Expr* e);
00119
00120     virtual int interp();
00121
00122     virtual bool has_variable();
00123
00124     virtual Expr* subst(string str, Expr* e);
00125
00126     virtual void print (ostream &ostream);
00127
00128 };
00129
00130 //======================  LET  ======================//
00131
00132 class _let : public Expr {
00133
00134 public:
00135     Var* lhs;
00136     Expr* rhs;
00137     Expr* body;
00138
00139     _let(Var* lhs, Expr* rhs, Expr* body);
00140
00141     virtual bool equals(Expr* e);
```

```
00142
00143     virtual int interp();
00144
00145     virtual bool has_variable();
00146
00147     virtual Expr* subst(string str, Expr* e);
00148
00149     virtual void print(ostream &ostream);
00150
00151     void pretty_print_at(ostream &ostream, precedence_t prec);
00152
00153 };
00154
00155
00156
```

6.6 /Users/corinnejones/GitHubSchool/Spring2024/CS6015_Software↩ Engineering/MSDScript/MSDScript/main.cpp File Reference

Command line argument handler for test execution.

```
#include <iostream>
#include "cmdline.hpp"
#include "Expr.hpp"
#include <string>
#include <cstdlib>
```

Functions

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

6.6.1 Detailed Description

Command line argument handler for test execution.

This file contains the entry point for the application, handling command line arguments to provide help information or to execute tests using the Catch testing framework. It supports '–help' for displaying usage information and '–test' for running tests.

Usage: ./application –help Displays help information. ./application –test Executes all compiled tests.

Author

Corinne Jones

Date

1/16/24

6.7 /Users/corinnejones/GitHubSchool/Spring2024/CS6015_Software↩ Engineering/MSDScript/MSDScript/Tests.cpp File Reference

Tests for the Expression Classes.

```
#include <stdio.h>
#include "Expr.hpp"
#include "catch.h"
```

Functions

- **TEST_CASE** ("TESTING NUM")
- **TEST_CASE** ("TESTING ADD")
- **TEST_CASE** ("TESTING MULT")
- **TEST_CASE** ("TESTING VAR")
- **TEST_CASE** ("Testing to_string()")
- **TEST_CASE** ("Testing Pretty Print")

6.7.1 Detailed Description

Tests for the Expression Classes.

Provides unit tests for evaluating the functionality of arithmetic expressions, including basic operations like addition, multiplication, and variable handling within an expression context. Utilizes the Catch testing framework.

Author

Corinne Jones

Date

1/18/24

Index

/Users/corinnejones/GitHubSchool/Spring2024/CS6015_SoftwareEngineering/MSDScript/MSDScript/Expr.cpp, 28
/Users/corinnejones/GitHubSchool/Spring2024/CS6015_SoftwareEngineering/MSDScript/MSDScript/Expr.hpp, 28
/Users/corinnejones/GitHubSchool/Spring2024/CS6015_SoftwareEngineering/MSDScript/MSDScript/Tests.cpp, 32
/Users/corinnejones/GitHubSchool/Spring2024/CS6015_SoftwareEngineering/MSDScript/MSDScript/cmdline.hpp, 27
/Users/corinnejones/GitHubSchool/Spring2024/CS6015_SoftwareEngineering/MSDScript/MSDScript/main.cpp, 31
_let, 9
    equals, 10
    has_variable, 10
    interp, 10
    pretty_print_at, 10
    print, 10
    subst, 10

Add, 11
    Add, 12
    equals, 12
    has_variable, 12
    interp, 12
    pretty_print_at, 13
    print, 13
    subst, 13

equals
    _let, 10
    Add, 12
    Expr, 14
    Mult, 18
    Num, 21
    Var, 23
Expr, 14
    equals, 14
    has_variable, 14
    interp, 15
    pretty_print, 15
    pretty_print_at, 15
    print, 15
    subst, 15
    to_pretty_string, 16
    to_string, 16

has_variable
    _let, 10
    Add, 12
    Expr, 14
    Mult, 18
    Num, 21
    Var, 24
interp
    _let, 10
    Add, 12
    Expr, 15
    Mult, 18
    Num, 21
    Var, 24

MSDScript, 1
Mult, 16
    equals, 18
    has_variable, 18
    interp, 18
    Mult, 17
    pretty_print_at, 18
    print, 19
    subst, 19

Num, 19
    equals, 21
    has_variable, 21
    interp, 21
    Num, 20
    print, 21
    subst, 22

pretty_print
    Expr, 15
pretty_print_at
    _let, 10
    Add, 13
    Expr, 15
    Mult, 18
print
    _let, 10
    Add, 13
    Expr, 15
    Mult, 19
    Num, 21
    Var, 24

subst
    _let, 10
    Add, 13
    Expr, 15
    Mult, 19
    Num, 22
    Var, 24

to_pretty_string
    Expr, 16
to_string
    Expr, 16

Var, 22
    equals, 23
    has_variable, 24
    interp, 24
    print, 24
    subst, 24
    Var, 23