Hydrogen

v0.2.1

Generated by Doxygen 1.9.8

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

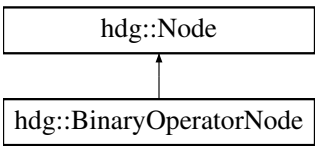
Here is a list of all documented files with brief descriptions:

Chapter 4

Class Documentation

4.1 hdg::BinaryOperatorNode Class Reference

Inheritance diagram for hdg::BinaryOperatorNode:



Public Member Functions

- **BinaryOperatorNode** (const Token &oper, Node *left, Node *right, const Position &position)
- **BinaryOperatorNode** (TokenType oper, Node *left, Node *right, const Position &position, Environment *environment)
- void **setOperator** (const Token &oper)
- void **setLeft** (Node *node)
- void **setRight** (Node *node)
- Token **getOperator** ()
- Node * **getLeft** ()
- Node * **getRight** ()
- std::string toString () override
- Object * interpret () override

Public Member Functions inherited from hdg::Node

- **Node** (const Position &position)
- **Node** (const Position &position, Environment *environment)
- Position * **thisPosition** ()
- Environment * **thisEnvironment** ()

Protected Attributes

- Token **m_oper**
- Node * **m_left**
- Node * **m_right**

Protected Attributes inherited from hdg::Node

- Position **m_position**
- Environment * **m_environment**

4.1.1 Member Function Documentation

4.1.1.1 interpret()

```
Object * hdg::BinaryOperatorNode::interpret ( )  [override], [virtual]
```

Implements hdg::Node.

4.1.1.2 toString()

```
std::string hdg::BinaryOperatorNode::toString ( )  [override], [virtual]
```

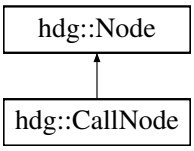
Implements hdg::Node.

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

- include/node/BinaryOperatorNode.h

4.2 hdg::CallNode Class Reference

Inheritance diagram for hdg::CallNode:



Public Member Functions

- **CallNode** (const Position &position, Environment *environment)
- **CallNode** (Node *call, std::vector< Node * >list, TokenType oper, const Position &position, Environment *environment)
- void **addNode** (Node *node)
- void **setOperator** (TokenType oper)
- void **setCall** (Node *call)
- std::string toString () override
- Object * interpret () override

Public Member Functions inherited from hdg::Node

- **Node** (const Position &position)
- **Node** (const Position &position, Environment *environment)
- Position * **thisPosition** ()
- Environment * **thisEnvironment** ()

Protected Attributes

- Node * **m_call**
- std::vector< Node * > **m_list**
- TokenType **m_oper**

Protected Attributes inherited from hdg::Node

- Position **m_position**
- Environment * **m_environment**

4.2.1 Member Function Documentation

4.2.1.1 interpret()

```
Object * hdg::CallNode::interpret ( )  [override], [virtual]
```

Implements hdg::Node.

4.2.1.2 toString()

```
std::string hdg::CallNode::toString ( )  [override], [virtual]
```

Implements hdg::Node.

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

- include/node/CallNode.h

4.3 hdg::Environment Class Reference

Public Member Functions

- **Environment** (std::string name, Environment *parent)
- void **setName** (std::string name)
- void **setParent** (Environment *parent)
- void **setSymbol** (const std::string &name, Object *value)
- void **setLocalSymbol** (const std::string &name, Object *value)
- void **setSymbol** (const std::string &name, const Integer &value)
- void **setSymbol** (std::initializer_list< std::pair< std::string, const Integer & > > list)
- void **setSymbol** (const std::string &name, const Float &value)
- std::string **getName** ()
- Environment * **getParent** ()
- Object * **getSymbol** (const std::string &name)

Protected Attributes

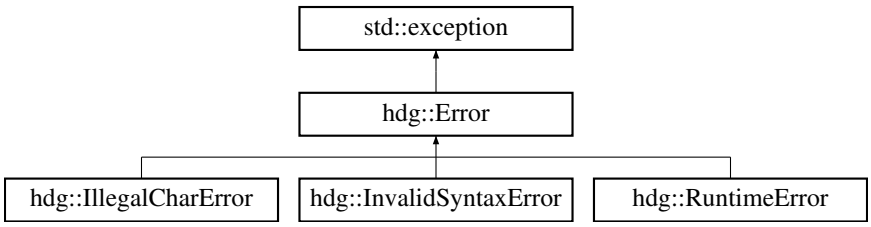
- std::string **m_name**
- Environment * **m_parent**
- SymbolTable **m_symbolTable**

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

- include/basic/Environment.h

4.4 hdg::Error Class Reference

Inheritance diagram for hdg::Error:



Public Member Functions

- **Error** (std::string name, std::string details, const Position &position)
- virtual std::string **toString** ()

Protected Attributes

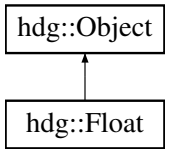
- std::string **m_name**
- std::string **m_details**
- Position **m_position**
- Environment * **m_environment** {}

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

- include/error/Error.h

4.5 hdg::Float Class Reference

Inheritance diagram for hdg::Float:



Public Member Functions

- **Float** (float value)
- **Float** (double value)
- **Float** (double value, const Position &position)
- double **getValue** () const
- Object * plus (Object *other) override
- Object * minus (Object *other) override
- Object * mul (Object *other) override
- Object * div (Object *other) override
- Object * pow (Object *other) override
- Object * equation (Object *other) override
- Object * notEquation (Object *other) override
- Object * greaterThan (Object *other) override
- Object * lessThan (Object *other) override
- Object * greaterThanEquation (Object *other) override
- Object * lessThanEquation (Object *other) override
- Object * andOperator (Object *other) override
- Object * orOperator (Object *other) override
- Object * notOperator () override
- bool isTrue () override
- std::string toString () override
- Object * copy () override

Public Member Functions inherited from hdg::Object

- **Object** (std::string className)
- **Object** (std::string className, const Position &position)
- std::string **getClass** ()
- virtual Object * **parenthesis** (const std::vector< Object * > &list)
- virtual Object * **brackets** (Object *other)
- virtual Object * **braces** (const std::vector< Object * > &list)
- virtual void **illegalOperator** ()

Additional Inherited Members

Protected Attributes inherited from hdg::Object

- std::string **m_class**
- Position **m_position**

4.5.1 Member Function Documentation

4.5.1.1 andOperator()

```
Object * hdg::Float::andOperator (
            Object * other )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.5.1.2 copy()

```
Object * hdg::Float::copy ( )  [override], [virtual]
```

Implements hdg::Object.

4.5.1.3 div()

```
Object * hdg::Float::div (
            Object * other )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.5.1.4 equation()

```
Object * hdg::Float::equation (
            Object * other )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.5.1.5 greaterThan()

```
Object * hdg::Float::greaterThan (
            Object * other )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.5.1.6 greaterThanEquation()

```
Object * hdg::Float::greaterThanEquation (
            Object * other )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.5.1.7 isTrue()

```
bool hdg::Float::isTrue ( )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.5.1.8 lessThan()

```
Object * hdg::Float::lessThan (
            Object * other )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.5.1.9 lessThanEquation()

```
Object * hdg::Float::lessThanEquation (
            Object * other )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.5.1.10 minus()

```
Object * hdg::Float::minus (
            Object * other )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.5.1.11 mul()

```
Object * hdg::Float::mul (
            Object * other )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.5.1.12 notEquation()

```
Object * hdg::Float::notEquation (
            Object * other )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.5.1.13 notOperator()

```
Object * hdg::Float::notOperator ( )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.5.1.14 orOperator()

```
Object * hdg::Float::orOperator (
            Object * other )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.5.1.15 plus()

```
Object * hdg::Float::plus (
            Object * other )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.5.1.16 pow()

```
Object * hdg::Float::pow (
            Object * other )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.5.1.17 toString()

```
std::string hdg::Float::toString ( )  [override], [virtual]
```

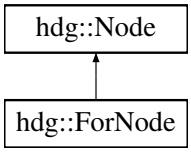
Implements hdg::Object.

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

- include/object/Float.h

4.6 hdg::ForNode Class Reference

Inheritance diagram for hdg::ForNode:



Public Member Functions

- **ForNode** (const Token &index, int to, const Position &position, Environment *parent)
- **ForNode** (const Token &index, int from, int to, int step, Node *expr, const Position &position, Environment *parent)
- void **setIndex** (const Token &index)
- void **setFrom** (int from)
- void **setTo** (int to)
- void **setStep** (int step)
- void **setExpr** (Node *expr)
- std::string toString () override
- Object * interpret () override

Public Member Functions inherited from hdg::Node

- **Node** (const Position &position)
- **Node** (const Position &position, Environment *environment)
- Position * **thisPosition** ()
- Environment * **thisEnvironment** ()

Protected Attributes

- Token **m_index**
- int **m_from**
- int **m_to**
- int **m_step**
- Node * **m_expr**

Protected Attributes inherited from hdg::Node

- Position **m_position**
- Environment * **m_environment**

4.6.1 Member Function Documentation

4.6.1.1 interpret()

```
Object * hdg::ForNode::interpret ( )  [override], [virtual]
```

Implements hdg::Node.

4.6.1.2 toString()

```
std::string hdg::ForNode::toString ( )  [override], [virtual]
```

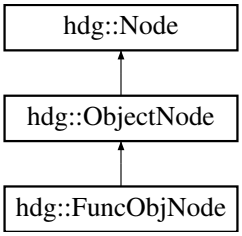
Implements hdg::Node.

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

- include/node/ForNode.h

4.7 hdg::FuncObjNode Class Reference

Inheritance diagram for hdg::FuncObjNode:



Public Member Functions

- **FuncObjNode** (const Position &position, Environment *parent)
- void **setArg** (ObjAssignNode *arg)
- void **setBody** (Node *body)
- void **setName** (const std::string &name)
- std::string toString ()
- Object * interpret ()

Public Member Functions inherited from hdg::ObjectNode

- **ObjectNode** (std::string className, const Position &position, Environment *environment)
- **ObjectNode** (const Position &position, Environment *environment)
- void **setClass** (const std::string &className)
- std::string **getClass** ()

Public Member Functions inherited from hdg::Node

- **Node** (const Position &position)
- **Node** (const Position &position, Environment *environment)
- Position * **thisPosition** ()
- Environment * **thisEnvironment** ()

Protected Attributes

- std::string **m_name**
- std::vector< ObjAssignNode * > **m_args**
- Node * **m_body** {}

Protected Attributes inherited from hdg::ObjectNode

- std::string **m_class**

Protected Attributes inherited from hdg::Node

- Position **m_position**
- Environment * **m_environment**

4.7.1 Member Function Documentation

4.7.1.1 interpret()

```
Object * hdg::FuncObjNode::interpret ( )  [virtual]
```

Implements hdg::ObjectNode.

4.7.1.2 toString()

```
std::string hdg::FuncObjNode::toString ( )  [virtual]
```

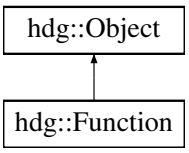
Implements hdg::ObjectNode.

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

- include/node/object_node/FuncObjNode.h

4.8 hdg::Function Class Reference

Inheritance diagram for hdg::Function:



Public Member Functions

- **Function** (std::string name, std::vector< ObjAssignNode * >args, Environment *environment, Node *body, const Position &position)
- Object * parenthesis (const std::vector< Object * > &list) override
- std::string toString () override
- Object * copy () override

Public Member Functions inherited from hdg::Object

- **Object** (std::string className)
- **Object** (std::string className, const Position &position)
- std::string **getClass** ()
- virtual Object * **plus** (Object *other)
- virtual Object * **minus** (Object *other)
- virtual Object * **mul** (Object *other)
- virtual Object * **div** (Object *other)
- virtual Object * **pow** (Object *other)
- virtual Object * **equation** (Object *other)
- virtual Object * **notEquation** (Object *other)
- virtual Object * **greaterThan** (Object *other)
- virtual Object * **lessThan** (Object *other)
- virtual Object * **greaterThanEquation** (Object *other)
- virtual Object * **lessThanEquation** (Object *other)
- virtual Object * **andOperator** (Object *other)
- virtual Object * **orOperator** (Object *other)
- virtual Object * **notOperator** ()
- virtual Object * **brackets** (Object *other)
- virtual Object * **braces** (const std::vector< Object * > &list)
- virtual bool **isTrue** ()
- virtual void **illegalOperator** ()

Protected Attributes

- std::string **m_name**
- std::vector< ObjAssignNode * > **m_args**
- Environment * **m_environment**
- Node * **m_body**

Protected Attributes inherited from hdg::Object

- std::string **m_class**
- Position **m_position**

4.8.1 Member Function Documentation

4.8.1.1 copy()

```
Object * hdg::Function::copy ( )  [override], [virtual]
```

Implements hdg::Object.

4.8.1.2 parenthesis()

```
Object * hdg::Function::parenthesis (
            const std::vector< Object * > & list )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.8.1.3 toString()

```
std::string hdg::Function::toString ( )  [override], [virtual]
```

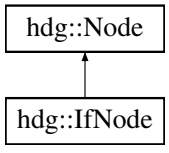
Implements hdg::Object.

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

- include/object/Function.h

4.9 hdg::IfNode Class Reference

Inheritance diagram for hdg::IfNode:



Public Member Functions

- **IfNode** (const Position &position, Environment *parent)
- void **addBranch** (Node *condition, Node *expression)
- std::string toString () override
- Object * interpret () override

Public Member Functions inherited from hdg::Node

- **Node** (const Position &position)
- **Node** (const Position &position, Environment *environment)
- Position * **thisPosition** ()
- Environment * **thisEnvironment** ()

Protected Attributes

- std::vector< Node * > **conditions**
- std::vector< Node * > **expressions**
- Node * **elseExpression**

Protected Attributes inherited from hdg::Node

- Position **m_position**
- Environment * **m_environment**

4.9.1 Member Function Documentation

4.9.1.1 interpret()

```
Object * hdg::IfNode::interpret ( )  [override], [virtual]
```

Implements hdg::Node.

4.9.1.2 toString()

```
std::string hdg::IfNode::toString ( )  [override], [virtual]
```

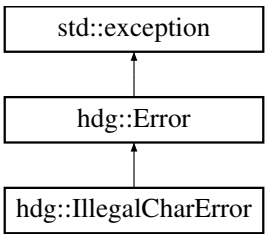
Implements hdg::Node.

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

- include/node/IfNode.h

4.10 hdg::IllegalCharError Class Reference

Inheritance diagram for hdg::IllegalCharError:



Public Member Functions

- **IllegalCharError** (std::string details, const Position &position)

Public Member Functions inherited from hdg::Error

- **Error** (std::string name, std::string details, const Position &position)
- virtual std::string **toString** ()

Additional Inherited Members

Protected Attributes inherited from hdg::Error

- std::string **m_name**
- std::string **m_details**
- Position **m_position**
- Environment * **m_environment** {}

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

- include/error/IllegalCharError.h

4.11 hdg::Indicator Struct Reference

Public Attributes

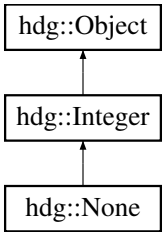
- unsigned long long **index**
- unsigned long long **line**
- unsigned long long **col**

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

- include/basic/Position.h

4.12 hdg::Integer Class Reference

Inheritance diagram for hdg::Integer:



Public Member Functions

- **Integer** (long long value)
- **Integer** (long long value, const Position &position)
- void **setValue** (long long value)
- long long **getValue** () const
- Object * plus (Object *other) override
- Object * minus (Object *other) override
- Object * mul (Object *other) override
- Object * div (Object *other) override
- Object * pow (Object *other) override
- Object * equation (Object *other) override
- Object * notEquation (Object *other) override
- Object * greaterThan (Object *other) override
- Object * lessThan (Object *other) override
- Object * greaterThanEquation (Object *other) override
- Object * lessThanEquation (Object *other) override
- Object * andOperator (Object *other) override
- Object * orOperator (Object *other) override
- Object * notOperator () override
- bool isTrue () override
- std::string toString () override
- Object * copy () override

Public Member Functions inherited from hdg::Object

- **Object** (std::string className)
- **Object** (std::string className, const Position &position)
- std::string **getClass** ()
- virtual Object * **parenthesis** (const std::vector< Object * > &list)
- virtual Object * **brackets** (Object *other)
- virtual Object * **braces** (const std::vector< Object * > &list)
- virtual void **illegalOperator** ()

Additional Inherited Members

Protected Attributes inherited from hdg::Object

- std::string **m_class**
- Position **m_position**

4.12.1 Member Function Documentation

4.12.1.1 andOperator()

```
Object * hdg::Integer::andOperator (
            Object * other )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.12.1.2 copy()

```
Object * hdg::Integer::copy ( )  [override], [virtual]
```

Implements hdg::Object.

4.12.1.3 div()

```
Object * hdg::Integer::div (
            Object * other )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.12.1.4 equation()

```
Object * hdg::Integer::equation (
            Object * other )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.12.1.5 greaterThan()

```
Object * hdg::Integer::greaterThan (
            Object * other )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.12.1.6 greaterThanEquation()

```
Object * hdg::Integer::greaterThanEquation (
            Object * other )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.12.1.7 isTrue()

```
bool hdg::Integer::isTrue ( )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.12.1.8 lessThan()

```
Object * hdg::Integer::lessThan (
            Object * other )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.12.1.9 lessThanEquation()

```
Object * hdg::Integer::lessThanEquation (
            Object * other )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.12.1.10 minus()

```
Object * hdg::Integer::minus (
            Object * other )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.12.1.11 mul()

```
Object * hdg::Integer::mul (
            Object * other )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.12.1.12 notEquation()

```
Object * hdg::Integer::notEquation (
            Object * other )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.12.1.13 notOperator()

```
Object * hdg::Integer::notOperator ( )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.12.1.14 orOperator()

```
Object * hdg::Integer::orOperator (
            Object * other )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.12.1.15 plus()

```
Object * hdg::Integer::plus (
            Object * other )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.12.1.16 pow()

```
Object * hdg::Integer::pow (
            Object * other )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.12.1.17 toString()

```
std::string hdg::Integer::toString ( )  [override], [virtual]
```

Implements hdg::Object.

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

- include/object/Integer.h

4.13 hdg::Interpreter Class Reference

Public Member Functions

- std::string **interpret** (const std::string &fPath, std::string code, Mode mode=release)

Protected Member Functions

- void **init** ()

Protected Attributes

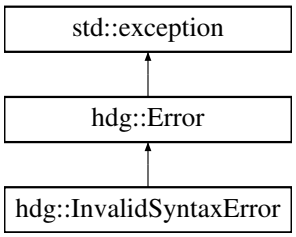
- Environment * **m_globalEnvironment**

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

- include/Interpreter.h

4.14 hdg::InvalidSyntaxError Class Reference

Inheritance diagram for hdg::InvalidSyntaxError:



Public Member Functions

- **InvalidSyntaxError** (std::string details, const Position &position)

Public Member Functions inherited from hdg::Error

- **Error** (std::string name, std::string details, const Position &position)
- virtual std::string **toString** ()

Additional Inherited Members

Protected Attributes inherited from hdg::Error

- std::string **m_name**
- std::string **m_details**
- Position **m_position**
- Environment * **m_environment** {}

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

- include/error/InvalidSyntaxError.h

4.15 hdg::Lexer Class Reference

Public Member Functions

- **Lexer** (std::string fPath, std::string *code)
- std::string **getFilePath** ()
- std::string * **thisText** ()
- std::vector< Token > & **getTokens** ()
- void **run** ()

Protected Member Functions

- void **advance** ()
- void **buildNumber** ()
- void **buildGreaterThan** ()
- void **buildLessThan** ()
- void **buildEquation** ()
- void **buildIdentifier** ()
- void **buildString** ()

Protected Attributes

- std::string **m_fPath**
- std::string * **m_code**
- std::vector< Token > **m_tokens**
- char **m_currentChar**
- Indicator **m_pos**

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

- include/Lexer.h

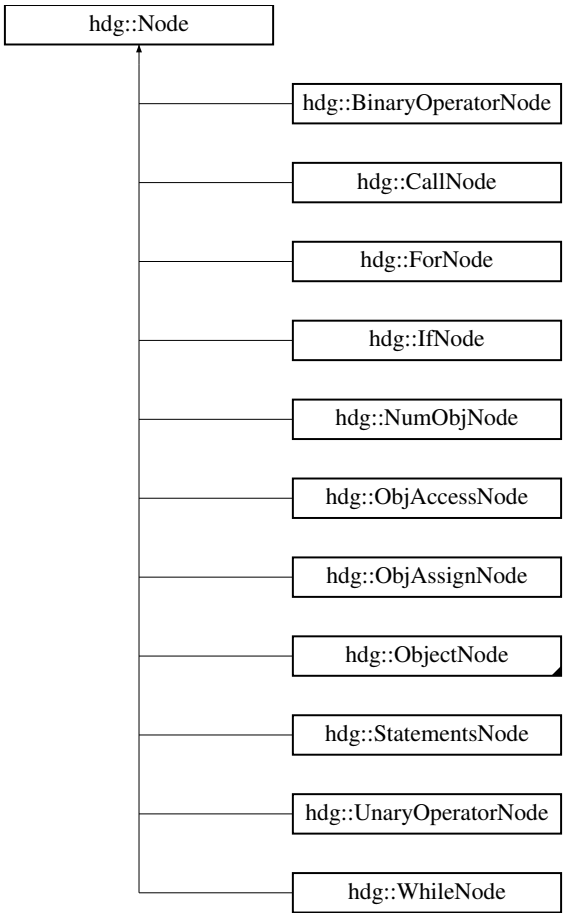
4.16 hdg::List Class Reference

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

- include/object/List.h

4.17 hdg::Node Class Reference

Inheritance diagram for hdg::Node:



Public Member Functions

- **Node** (const Position &position)
- **Node** (const Position &position, Environment *environment)
- Position * **thisPosition** ()
- Environment * **thisEnvironment** ()
- virtual std::string **toString** ()=0
- virtual Object * **interpret** ()=0

Protected Attributes

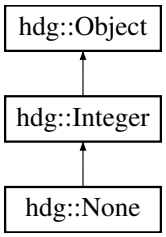
- Position **m_position**
- Environment * **m_environment**

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

- include/node/Node.h

4.18 hdg::None Class Reference

Inheritance diagram for hdg::None:



Public Member Functions

- Object * notOperator () override
- std::string toString () override
- Object * copy () override

Public Member Functions inherited from hdg::Integer

- **Integer** (long long value)
- **Integer** (long long value, const Position &position)
- void **setValue** (long long value)
- long long **getValue** () const
- Object * plus (Object *other) override
- Object * minus (Object *other) override
- Object * mul (Object *other) override
- Object * div (Object *other) override
- Object * pow (Object *other) override
- Object * equation (Object *other) override
- Object * notEquation (Object *other) override
- Object * greaterThan (Object *other) override
- Object * lessThan (Object *other) override
- Object * greaterThanEquation (Object *other) override
- Object * lessThanEquation (Object *other) override
- Object * andOperator (Object *other) override
- Object * orOperator (Object *other) override
- bool isTrue () override

Public Member Functions inherited from hdg::Object

- **Object** (std::string className)
- **Object** (std::string className, const Position &position)
- std::string **getClass** ()
- virtual Object * **parenthesis** (const std::vector< Object * > &list)
- virtual Object * **brackets** (Object *other)
- virtual Object * **braces** (const std::vector< Object * > &list)
- virtual void **illegalOperator** ()

Additional Inherited Members

Protected Attributes inherited from hdg::Object

- std::string **m_class**
- Position **m_position**

4.18.1 Member Function Documentation

4.18.1.1 copy()

```
Object * hdg::None::copy ( )  [override], [virtual]
```

Reimplemented from hdg::Integer.

4.18.1.2 notOperator()

```
Object * hdg::None::notOperator ( )  [override], [virtual]
```

Reimplemented from hdg::Integer.

4.18.1.3 toString()

```
std::string hdg::None::toString ( )  [override], [virtual]
```

Reimplemented from hdg::Integer.

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

- include/object/None.h

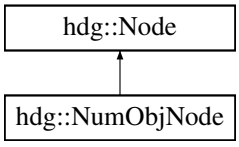
4.19 hdg::NotImplementedError Class Reference

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

- include/error/NotImplementedError.h

4.20 hdg::NumObjNode Class Reference

Inheritance diagram for hdg::NumObjNode:



Public Member Functions

- **NumObjNode** (long long value, const Position &position)
- **NumObjNode** (float value, const Position &position)
- **NumObjNode** (double value, const Position &position)
- std::string toString () override
- Object * interpret () override

Public Member Functions inherited from hdg::Node

- **Node** (const Position &position)
- **Node** (const Position &position, Environment *environment)
- Position * **thisPosition** ()
- Environment * **thisEnvironment** ()

Protected Attributes

- union {
  long long **i**
  double **f**
  } **m_value**

- std::string **m_class**

Protected Attributes inherited from hdg::Node

- Position **m_position**
- Environment * **m_environment**

4.20.1 Member Function Documentation

4.20.1.1 interpret()

```
Object * hdg::NumObjNode::interpret ( )  [override], [virtual]
```

Implements hdg::Node.

4.20.1.2 toString()

```
std::string hdg::NumObjNode::toString ( )  [override], [virtual]
```

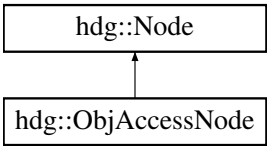
Implements hdg::Node.

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

- include/node/object_node/NumObjNode.h

4.21 hdg::ObjAccessNode Class Reference

Inheritance diagram for hdg::ObjAccessNode:



Public Member Functions

- **ObjAccessNode** (std::string name, const Position &position, Environment ∗environment)
- std::string toString () override
- Object ∗ interpret () override

Public Member Functions inherited from hdg::Node

- **Node** (const Position &position)
- **Node** (const Position &position, Environment ∗environment)
- Position ∗ **thisPosition** ()
- Environment ∗ **thisEnvironment** ()

Protected Attributes

- std::string **m_name**

Protected Attributes inherited from hdg::Node

- Position **m_position**
- Environment ∗ **m_environment**

4.21.1 Member Function Documentation

4.21.1.1 interpret()

```
Object * hdg::ObjAccessNode::interpret ( )  [override], [virtual]
```

Implements hdg::Node.

4.21.1.2 toString()

```
std::string hdg::ObjAccessNode::toString ( )  [override], [virtual]
```

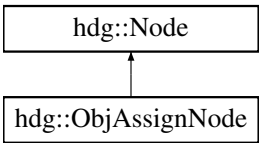
Implements hdg::Node.

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

- include/node/ObjAccessNode.h

4.22 hdg::ObjAssignNode Class Reference

Inheritance diagram for hdg::ObjAssignNode:



Public Member Functions

- **ObjAssignNode** (std::string name, Node ∗obj, const Position &position, Environment ∗environment)
- void **setExpr** (Node ∗expr)
- std::string **getName** ()
- std::string toString () override
- Object ∗ interpret () override

Public Member Functions inherited from hdg::Node

- **Node** (const Position &position)
- **Node** (const Position &position, Environment ∗environment)
- Position ∗ **thisPosition** ()
- Environment ∗ **thisEnvironment** ()

Protected Attributes

- std::string **m_name**
- Node ∗ **m_expr**

Protected Attributes inherited from hdg::Node

- Position **m_position**
- Environment ∗ **m_environment**

4.22.1 Member Function Documentation

4.22.1.1 interpret()

```
Object * hdg::ObjAssignNode::interpret ( )  [override], [virtual]
```

Implements hdg::Node.

4.22.1.2 toString()

```
std::string hdg::ObjAssignNode::toString ( )  [override], [virtual]
```

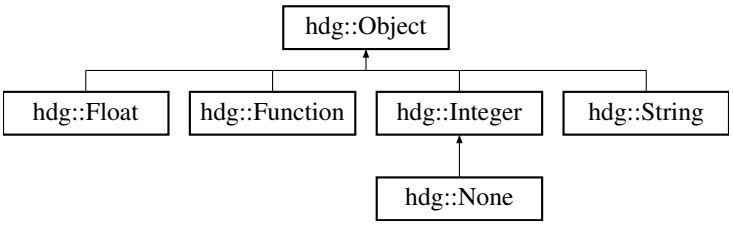
Implements hdg::Node.

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

- include/node/ObjAssignNode.h

4.23 hdg::Object Class Reference

Inheritance diagram for hdg::Object:



Public Member Functions

- **Object** (std::string className)
- **Object** (std::string className, const Position &position)
- std::string **getClass** ()
- virtual Object * **plus** (Object *other)
- virtual Object * **minus** (Object *other)
- virtual Object * **mul** (Object *other)
- virtual Object * **div** (Object *other)
- virtual Object * **pow** (Object *other)
- virtual Object * **equation** (Object *other)
- virtual Object * **notEquation** (Object *other)
- virtual Object * **greaterThan** (Object *other)
- virtual Object * **lessThan** (Object *other)
- virtual Object * **greaterThanEquation** (Object *other)
- virtual Object * **lessThanEquation** (Object *other)
- virtual Object * **andOperator** (Object *other)
- virtual Object * **orOperator** (Object *other)
- virtual Object * **notOperator** ()
- virtual Object * **parenthesis** (const std::vector< Object * > &list)
- virtual Object * **brackets** (Object *other)
- virtual Object * **braces** (const std::vector< Object * > &list)
- virtual bool **isTrue** ()
- virtual void **illegalOperator** ()
- virtual std::string **toString** ()=0
- virtual Object * **copy** ()=0

Protected Attributes

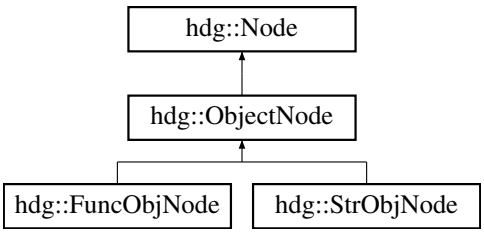
- std::string **m_class**
- Position **m_position**

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

- include/object/Object.h

4.24 hdg::ObjectNode Class Reference

Inheritance diagram for hdg::ObjectNode:



Public Member Functions

- **ObjectNode** (std::string className, const Position &position, Environment *environment)
- **ObjectNode** (const Position &position, Environment *environment)
- void **setClass** (const std::string &className)
- std::string **getClass** ()
- virtual std::string toString ()=0
- virtual Object * interpret ()=0

Public Member Functions inherited from hdg::Node

- **Node** (const Position &position)
- **Node** (const Position &position, Environment *environment)
- Position * **thisPosition** ()
- Environment * **thisEnvironment** ()

Protected Attributes

- std::string **m_class**

Protected Attributes inherited from hdg::Node

- Position **m_position**
- Environment * **m_environment**

4.24.1 Member Function Documentation

4.24.1.1 interpret()

```
virtual Object * hdg::ObjectNode::interpret ( )  [pure virtual]
```

Implements hdg::Node.

4.24.1.2 toString()

```
virtual std::string hdg::ObjectNode::toString ( )  [pure virtual]
```

Implements hdg::Node.

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

- include/node/object_node/ObjectNode.h

4.25 hdg::Parser Class Reference

Public Member Functions

- **Parser** (std::vector< Token > tokens, Environment *environment)
- Node * **run** ()

Protected Member Functions

- void **advance** ()
- void **retreat** ()
- Node * **expr** (Environment *environment)
- Node * **compExpr** (Environment *environment)
- Node * **arithExpr** (Environment *environment)
- Node * **term** (Environment *environment)
- Node * **factor** (Environment *environment)
- Node * **power** (Environment *environment)
- Node * **call** (Environment *environment)
- Node * **atom** (Environment *environment)
- Node * **ifExpr** (Environment *environment)
- Node * **forExpr** (Environment *environment)
- Node * **whileExpr** (Environment *environment)
- Node * **funcExpr** (Environment *environment)
- Node * **statements** (Environment *environment)
- Node * **binaryOperator** (Environment *environment, const std::set< Token, std::less<> > &opers, std::function< Node *(Environment *envir)> funA, std::function< Node *(Environment *envir)> funB=nullptr)
- Node * **unaryOperator** (Environment *environment, const std::set< Token, std::less<> > &opers, std::function< Node *(Environment *envir)> fun)

Protected Attributes

- std::vector< Token > **m_tokens**
- std::vector< Token >::iterator **m_currentToken**
- Environment * **m_environment**

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

- include/Parser.h

4.26 hdg::Position Class Reference

Public Member Functions

- **Position** (std::string fPath, std::string *context, const Indicator &start, const Indicator &end)
- **Position** (std::string fPath, std::string *context, const Indicator &start)
- **Position** (const Position &position)
- void **setStart** (int index, int line, int col)
- void **setStart** (const Indicator &indicator)
- void **setEnd** (int index, int line, int col)
- void **setEnd** (const Indicator &indicator)
- Indicator **getStart** ()
- Indicator **getEnd** ()
- std::string **getFilePath** ()
- std::string * **thisContext** ()
- std::string **arrow** ()

Protected Attributes

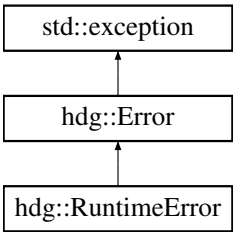
- std::string **m_fPath**
- std::string * **m_context**
- Indicator **m_start** {}
- Indicator **m_end** {}

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

- include/basic/Position.h

4.27 hdg::RuntimeError Class Reference

Inheritance diagram for hdg::RuntimeError:



Public Member Functions

- **RuntimeError** (std::string details, const Position &position)

Public Member Functions inherited from hdg::Error

- **Error** (std::string name, std::string details, const Position &position)
- virtual std::string **toString** ()

Additional Inherited Members

Protected Attributes inherited from hdg::Error

- std::string **m_name**
- std::string **m_details**
- Position **m_position**
- Environment * **m_environment** {}

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

- include/error/RuntimeError.h

4.28 hdg::Shell Class Reference

Public Member Functions

- void **exe** (int argc, char *argv[ ])

Static Public Member Functions

- static std::string **input** ()
- static std::string **input** (const std::string &path)

Protected Attributes

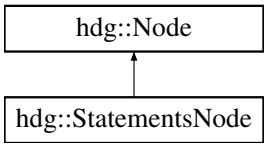
- Mode **m_mode**
- std::string **m_fPath**

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

- include/Shell.h

4.29 hdg::StatementsNode Class Reference

Inheritance diagram for hdg::StatementsNode:



Public Member Functions

- **StatementsNode** (const Position &position, Environment *environment)
- void **append** (Node *node)
- std::string toString () override
- Object * interpret () override

Public Member Functions inherited from hdg::Node

- **Node** (const Position &position)
- **Node** (const Position &position, Environment *environment)
- Position * **thisPosition** ()
- Environment * **thisEnvironment** ()

Protected Attributes

- std::vector< Node * > **m_statements**

Protected Attributes inherited from hdg::Node

- Position **m_position**
- Environment * **m_environment**

4.29.1 Member Function Documentation

4.29.1.1 interpret()

```
Object * hdg::StatementsNode::interpret ( )  [override], [virtual]
```

Implements hdg::Node.

4.29.1.2 toString()

```
std::string hdg::StatementsNode::toString ( )  [override], [virtual]
```

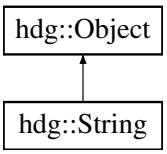
Implements hdg::Node.

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

- include/node/StatementsNode.h

4.30 hdg::String Class Reference

Inheritance diagram for hdg::String:



Public Member Functions

- **String** (std::string value)
- **String** (std::string value, const Position &position)
- std::string **getValue** ()
- Object * plus (Object *other) override
- bool isTrue () override
- std::string toString () override
- Object * copy () override

Public Member Functions inherited from hdg::Object

- **Object** (std::string className)
- **Object** (std::string className, const Position &position)
- std::string **getClass** ()
- virtual Object * **minus** (Object *other)
- virtual Object * **mul** (Object *other)
- virtual Object * **div** (Object *other)
- virtual Object * **pow** (Object *other)
- virtual Object * **equation** (Object *other)
- virtual Object * **notEquation** (Object *other)
- virtual Object * **greaterThan** (Object *other)
- virtual Object * **lessThan** (Object *other)
- virtual Object * **greaterThanEquation** (Object *other)
- virtual Object * **lessThanEquation** (Object *other)
- virtual Object * **andOperator** (Object *other)
- virtual Object * **orOperator** (Object *other)
- virtual Object * **notOperator** ()
- virtual Object * **parenthesis** (const std::vector< Object * > &list)
- virtual Object * **brackets** (Object *other)
- virtual Object * **braces** (const std::vector< Object * > &list)
- virtual void **illegalOperator** ()

Protected Attributes

- std::string **m_value**

Protected Attributes inherited from hdg::Object

- std::string **m_class**
- Position **m_position**

4.30.1 Member Function Documentation

4.30.1.1 copy()

```
Object * hdg::String::copy ( )  [override], [virtual]
```

Implements hdg::Object.

4.30.1.2 isTrue()

```
bool hdg::String::isTrue ( )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.30.1.3 plus()

```
Object * hdg::String::plus (
            Object * other )  [override], [virtual]
```

Reimplemented from hdg::Object.

4.30.1.4 toString()

```
std::string hdg::String::toString ( )  [override], [virtual]
```

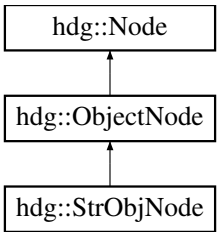
Implements hdg::Object.

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

- include/object/String.h

4.31 hdg::StrObjNode Class Reference

Inheritance diagram for hdg::StrObjNode:



Public Member Functions

- **StrObjNode** (std::string value, const Position &position, Environment *environment)
- void **setValue** (const std::string &value)
- std::string toString () override
- Object * interpret () override

Public Member Functions inherited from hdg::ObjectNode

- **ObjectNode** (std::string className, const Position &position, Environment *environment)
- **ObjectNode** (const Position &position, Environment *environment)
- void **setClass** (const std::string &className)
- std::string **getClass** ()

Public Member Functions inherited from hdg::Node

- **Node** (const Position &position)
- **Node** (const Position &position, Environment *environment)
- Position * **thisPosition** ()
- Environment * **thisEnvironment** ()

Protected Attributes

- std::string **m_value**

Protected Attributes inherited from hdg::ObjectNode

- std::string **m_class**

Protected Attributes inherited from hdg::Node

- Position **m_position**
- Environment * **m_environment**

4.31.1 Member Function Documentation

4.31.1.1 interpret()

```
Object * hdg::StrObjNode::interpret ( )  [override], [virtual]
```

Implements hdg::ObjectNode.

4.31.1.2 toString()

```
std::string hdg::StrObjNode::toString ( )  [override], [virtual]
```

Implements hdg::ObjectNode.

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

- include/node/object_node/StrObjNode.h

4.32 hdg::Token Class Reference

Public Member Functions

- **Token** (TokenType type, std::string value, const Position &position)
- **Token** (TokenType type, std::string value)
- **Token** (TokenType type)
- **Token** (TokenType type, const Position &position)
- **Token** (const Token &tok)
- bool **match** (TokenType type, const std::string &value)
- void **setType** (TokenType type)
- void **setValue** (const std::string &value)
- TokenType **getType** ()
- std::string **getValue** ()
- Position * **thisPosition** ()
- std::string **toString** ()

Protected Attributes

- TokenType **m_type**
- std::string **m_value**
- Position **m_position**

Friends

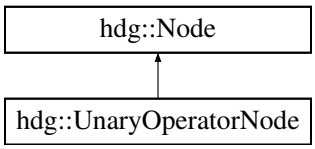
- std::ostream & **operator<<** (std::ostream &out, Token &tok)
- bool **operator<** (const Token &a, const Token &b)

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

- include/basic/Token.h

4.33 hdg::UnaryOperatorNode Class Reference

Inheritance diagram for hdg::UnaryOperatorNode:



Public Member Functions

- **UnaryOperatorNode** (TokenType oper, Node *obj, const Position &position, Environment *environment)
- **UnaryOperatorNode** (Token oper, Node *obj, const Position &position, Environment *environment)
- void **setOperator** (const Token &oper)
- void **setObject** (Node *obj)
- Token **getOperator** ()
- Node * **getObject** ()
- std::string toString () override
- Object * interpret () override

Public Member Functions inherited from hdg::Node

- **Node** (const Position &position)
- **Node** (const Position &position, Environment *environment)
- Position * **thisPosition** ()
- Environment * **thisEnvironment** ()

Protected Attributes

- Token **m_oper**
- Node * **m_obj**

Protected Attributes inherited from hdg::Node

- Position **m_position**
- Environment * **m_environment**

4.33.1 Member Function Documentation

4.33.1.1 interpret()

```
Object * hdg::UnaryOperatorNode::interpret ( )  [override], [virtual]
```

Implements hdg::Node.

4.33.1.2 toString()

```
std::string hdg::UnaryOperatorNode::toString ( )  [override], [virtual]
```

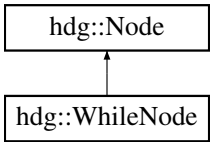
Implements hdg::Node.

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

- include/node/UnaryOperatorNode.h

4.34 hdg::WhileNode Class Reference

Inheritance diagram for hdg::WhileNode:



Public Member Functions

- **WhileNode** (Node ∗condition, Node ∗expression, const Position &position, Environment ∗parent)
- void **setCondition** (Node ∗condition)
- void **setExpression** (Node ∗expression)
- std::string toString () override
- Object ∗ interpret () override

Public Member Functions inherited from hdg::Node

- **Node** (const Position &position)
- **Node** (const Position &position, Environment ∗environment)
- Position ∗ **thisPosition** ()
- Environment ∗ **thisEnvironment** ()

Protected Attributes

- Node ∗ **m_condition**
- Node ∗ **m_expression**

Protected Attributes inherited from hdg::Node

- Position **m_position**
- Environment ∗ **m_environment**

4.34.1 Member Function Documentation

4.34.1.1 interpret()

```
Object * hdg::WhileNode::interpret ( )  [override], [virtual]
```

Implements hdg::Node.

4.34.1.2 toString()

```
std::string hdg::WhileNode::toString ( )  [override], [virtual]
```

Implements hdg::Node.

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

- include/node/WhileNode.h

Chapter 5

File Documentation

5.1 Environment.h

```
00001 //
00002 // Created by Magnesium on 2023/7/27.
00003 //
00004
00005 #ifndef HDG_ENVIRONMENT_H
00006 #define HDG_ENVIRONMENT_H
00007
00008 #include <map>
00009 #include <string>
00010 #include <initializer_list>
00011 #include <stack>
00012 #include "../object/Object.h"
00013 #include "../object/Integer.h"
00014 #include "../object/Float.h"
00015
00016 namespace hdg {
00017     typedef std::map<std::string, Object*> SymbolTable;
00018
00019     class Environment {
00020     protected:
00021         std::string m_name;
00022         Environment* m_parent;
00023         SymbolTable m_symbolTable;
00024
00025
00026     public:
00027         Environment(std::string name, Environment* parent);
00028         ~Environment();
00029
00030         void setName(std::string name);
00031         void setParent(Environment* parent);
00032         void setSymbol(const std::string& name, Object* value);
00033         void setLocalSymbol(const std::string& name, Object* value);
00034
00035         void setSymbol(const std::string& name, const Integer& value);
00036         void setSymbol(std::initializer_list<std::pair<std::string, const Integer&>> list);
00037         void setSymbol(const std::string& name, const Float& value);
00038
00039         std::string getName();
00040         Environment* getParent();
00041         Object* getSymbol(const std::string& name);
00042     };
00043
00044 } // hdg
00045
00046 #endif //HDG_ENVIRONMENT_H
```

5.2 Position.h

```
00001 //
00002 // Created by Magnesium on 2023/7/25.
00003 //
00004
00005 #ifndef HDG_POSITION_H
```

```
00006 #define HDG_POSITION_H
00007
00008 #include <string>
00009
00010 namespace hdg {
00011
00012     struct Indicator{
00013     public:
00014         unsigned long long index;
00015         unsigned long long line;
00016         unsigned long long col;
00017     };
00018
00019     class Position {
00020     protected:
00021         std::string m_fPath;
00022         std::string* m_context;
00023
00024         Indicator m_start{};
00025         Indicator m_end{};
00026
00027     public:
00028         Position();
00029         Position(std::string fPath, std::string* context, const Indicator& start, const Indicator&
      end);
00030         Position(std::string fPath, std::string* context, const Indicator& start);
00031         Position(const Position& position);
00032
00033         void setStart(int index, int line, int col);
00034         void setStart(const Indicator& indicator);
00035         void setEnd(int index, int line, int col);
00036         void setEnd(const Indicator& indicator);
00037
00038         Indicator getStart();
00039         Indicator getEnd();
00040         std::string getFilePath();
00041         std::string* thisContext();
00042
00043         std::string arrow();
00044     };
00045
00046 } // hdg
00047
00048 #endif //HDG_POSITION_H
```

5.3 Token.h

```
00001 //
00002 // Created by Magnesium on 2023/7/11.
00003 //
00004
00005 #ifndef HDG_TOKEN_H
00006 #define HDG_TOKEN_H
00007
00008 #include <string>
00009 #include <map>
00010 #include <utility>
00011 #include <iostream>
00012 #include "Position.h"
00013
00014 namespace hdg {
00015
00016     enum TokenType{
00017         INT,
00018         FLOAT,      // "float"
00019         STRING,
00020         IDENTIFIER,
00021         KEYWORD,
00022
00023         NE,         // "!="
00024         EE,         // "=="
00025         GT,
00026         LT,
00027         GTE,
00028         LTE,
00029
00030         PLUS,
00031         MINUS,
00032         MUL,
00033         DIV,
00034         POW,
00035
00036         LPAREN,     //
```

```
00037         RPAREN,
00038         LBRACKET,    //
00039         RBRACKET,
00040         LBRACE,      //
00041         RBRACE,
00042         EQ,          // "="
00043
00044         COLON,
00045         COMMA,
00046
00047         EF,
00048         EL,          // end of line ";" and "\n"
00049     };
00050
00051     static std::map<TokenType, std::string> tokenTypeName = {
00052             {INT,           "INT"},
00053             {FLOAT,         "FLOAT"},
00054             {STRING,        "STRING"},
00055             {IDENTIFIER,    "IDENTIFIER"},
00056             {KEYWORD,       "KEYWORD"},
00057
00058             {EE,            "EE"},
00059             {GT,            "GT"},
00060             {LT,            "LT"},
00061             {GTE,           "GTE"},
00062             {LTE,           "LTE"},
00063
00064             {PLUS,          "PLUS"},
00065             {MINUS,         "MINUS"},
00066             {MUL,           "MUL"},
00067             {DIV,           "DIV"},
00068             {POW,           "POW"},
00069
00070             {LPAREN,        "LPAREN"},
00071             {RPAREN,        "RPAREN"},
00072             {LBRACKET,      "LBRACKET"},
00073             {RBRACKET,      "RBRACKET"},
00074             {LBRACE,        "LBRACE"},
00075             {RBRACE,        "RBRACE"},
00076             {EQ,            "EQ"},
00077
00078             {COLON,         "COLON"},
00079             {COMMA,         "COMMA"},
00080
00081             {EF,            "EF"},          // EOF   end of file    EOFC++EF
00082             {EL,            "EL"},          // EOL   end of line    ""EL""EF"
00083     };
00084     class Token {
00085     protected:
00086         TokenType m_type;
00087         std::string m_value;
00088
00089         Position m_position;
00090
00091     public:
00092         Token();
00093         Token(TokenType type, std::string value, const Position& position);
00094         Token(TokenType type, std::string value);
00095         Token(TokenType type);
00096         Token(TokenType type, const Position& position);
00097         Token(const Token& tok);
00098         ~Token();
00099
00100         bool match(TokenType type, const std::string& value);
00101         void setType(TokenType type);
00102         void setValue(const std::string& value);
00103         TokenType getType();
00104         std::string getValue();
00105         Position* thisPosition();
00106         std::string toString();
00107
00108         friend std::ostream& operator<<(std::ostream& out, Token& tok);
00109         friend bool operator<(const Token& a, const Token& b);
00110     };
00111
00112     bool operator<(const Token& left, const Token& right);
00113 } // hdg
00114
00115 #endif //HDG_TOKEN_H
```

5.4 Error.h

```
00001 //
```

```
00002 // Created by Magnesium on 2023/7/16.
00003 //
00004
00005 #ifndef HDG_ERROR_H
00006 #define HDG_ERROR_H
00007
00008 #include <iostream>
00009 #include <exception>
00010 #include <utility>
00011 #include <sstream>
00012 #include "../basic/Position.h"
00013 #include "../basic/Environment.h"
00014
00015 namespace hdg {
00016
00017     class Error: public std::exception{
00018     protected:
00019         std::string m_name;
00020         std::string m_details;
00021
00022         Position m_position;
00023         Environment* m_environment{};
00024
00025     public:
00026         Error();
00027         Error(std::string name, std::string details, const Position& position);
00028         virtual std::string toString();
00029     };
00030
00031 } // hdg
00032
00033 #endif //HDG_ERROR_H
```

5.5 IllegalCharError.h

```
00001 //
00002 // Created by Magnesium on 2023/7/16.
00003 //
00004
00005
00006 #ifndef HDG_ILLEGALCHARERROR_H
00007 #define HDG_ILLEGALCHARERROR_H
00008
00009 #include <utility>
00010 #include "Error.h"
00011
00012 namespace hdg {
00013
00014     class IllegalCharError: public Error{
00015     public:
00016         IllegalCharError(std::string details, const Position& position);
00017     };
00018
00019 } // hdg
00020
00021 #endif //HDG_ILLEGALCHARERROR_H
```

5.6 InvalidSyntaxError.h

```
00001 //
00002 // Created by Magnesium on 2023/7/22.
00003 //
00004
00005
00006 #ifndef HDG_INVALIDSYNTAXERROR_H
00007 #define HDG_INVALIDSYNTAXERROR_H
00008
00009 #include <utility>
00010 #include "Error.h"
00011
00012 namespace hdg {
00013
00014     class InvalidSyntaxError: public Error{
00015     public:
00016         InvalidSyntaxError(std::string details, const Position& position);
00017     };
00018
00019 } // hdg
00020
00021 #endif //HDG_INVALIDSYNTAXERROR_H
```

5.7 NotImplementedError.h

```
00001 //
00002 // Created by Magnesium on 2023/8/8.
00003 //
00004
00005 #ifndef HDG_NOTIMPLEMENTEDERROR_H
00006 #define HDG_NOTIMPLEMENTEDERROR_H
00007
00008 #include "RuntimeError.h"
00009
00010 namespace hdg {
00011
00012     class NotImplementedError{
00013     };
00014
00015 } // hdg
00016
00017 #endif //HDG_NOTIMPLEMENTEDERROR_H
```

5.8 RuntimeError.h

```
00001 //
00002 // Created by Magnesium on 2023/7/22.
00003 //
00004
00005
00006 #ifndef HDG_RUNTIMEERROR_H
00007 #define HDG_RUNTIMEERROR_H
00008
00009 #include <utility>
00010 #include "Error.h"
00011
00012 namespace hdg {
00013
00014     class RuntimeError: public Error{
00015     public:
00016         RuntimeError(std::string details, const Position& position);
00017     };
00018
00019 } // hdg
00020
00021 #endif //HDG_RUNTIMEERROR_H
```

5.9 Interpreter.h

```
00001 //
00002 // Created by Magnesium on 2023/7/21.
00003 //
00004
00005
00006 #ifndef HDG_INTERPRETER_H
00007 #define HDG_INTERPRETER_H
00008
00009 #include <string>
00010 #include "Parser.h"
00011 #include "Lexer.h"
00012 #include "node/Node.h"
00013 #include "basic/Environment.h"
00014
00015 namespace hdg {
00016     enum Mode{
00017         debug,
00018         release,
00019     };
00020
00021     class Interpreter {
00022     protected:
00023         Environment* m_globalEnvironment;
00024
00025     public:
00026         Interpreter();
00027         ~Interpreter();
00028
00029         std::string interpret(const std::string& fPath, std::string code, Mode mode = release);
00030
00031     protected:
00032         void init();
00033     };
```

```
00034
00035 } // hdg
00036
00037 #endif //HDG_INTERPRETER_H
```

5.10 Lexer.h

```
00001 //
00002 // Created by Magnesium on 2023/7/11.
00003 //
00004
00005 #ifndef HDG_LEXER_H
00006 #define HDG_LEXER_H
00007
00008 #include <string>
00009 #include <vector>
00010 #include <iostream>
00011 #include <utility>
00012 #include <set>
00013 #include "basic/Token.h"
00014 #include "error/IllegalCharError.h"
00015
00016 namespace hdg {
00017     static std::set<std::string> keywordSet = {
00018             "not",
00019             "and",
00020             "or",
00021
00022             "if",
00023             "elif",
00024             "else",
00025
00026             "for",
00027             "from",
00028             "to",
00029             "step",
00030             "while",
00031
00032             "function",
00033     };
00034
00035     enum LegalChar{
00036         ILLEGAL = 0,
00037         DIGITAL = 1,
00038         LOWERCASE = 2,
00039         UPPERCASE = 4,
00040         UNDERLINE = 8
00041     };
00042
00043     LegalChar whatIsThis(char c);
00044     bool whatIsThis(char c, int target);
00045     std::ostream& operator«(std::ostream& out, std::vector<Token>& tokens);
00046
00047     class Lexer {
00048     protected:
00049         std::string m_fPath;
00050         std::string* m_code;
00051         std::vector<Token> m_tokens;
00052
00053         char m_currentChar;
00054         Indicator m_pos;
00055
00056     public:
00057         explicit Lexer(std::string fPath, std::string* code);
00058         std::string getFilePath();
00059         std::string* thisText();
00060         std::vector<Token>& getTokens();
00061         void run();
00062
00063     protected:
00064         void advance();
00065
00066         void buildNumber();
00067         void buildGreaterThan();
00068         void buildLessThan();
00069         void buildEquation();
00070         void buildIdentifier();
00071         void buildString();
00072     };
00073 } // hdg
00074
00075 #endif //HDG_LEXER_H
```

5.11 BinaryOperatorNode.h

```
00001 //
00002 // Created by Magnesium on 2023/7/19.
00003 //
00004
00005 #ifndef HDG_BINARYOPERATORNODE_H
00006 #define HDG_BINARYOPERATORNODE_H
00007
00008 #include <string>
00009 #include <cmath>
00010 #include <utility>
00011 #include "Node.h"
00012 #include "../basic/Token.h"
00013 #include "../error/RuntimeError.h"
00014
00015 namespace hdg {
00016
00017     class BinaryOperatorNode: public Node{
00018     protected:
00019         Token m_oper;
00020         Node* m_left;
00021         Node* m_right;
00022
00023     public:
00024         BinaryOperatorNode(const Token& oper, Node* left, Node* right, const Position& position);
00025         BinaryOperatorNode(TokenType oper, Node* left, Node* right, const Position& position,
      Environment* environment);
00026         ~BinaryOperatorNode() override;
00027
00028         void setOperator(const Token& oper);
00029         void setLeft(Node* node);
00030         void setRight(Node* node);
00031         Token getOperator();
00032         Node* getLeft();
00033         Node* getRight();
00034
00035         std::string toString() override;
00036         Object* interpret() override;
00037     };
00038
00039 } // hdg
00040
00041 #endif //HDG_BINARYOPERATORNODE_H
```

5.12 CallNode.h

```
00001 //
00002 // Created by Magnesium on 2023/8/11.
00003 //
00004
00005 #ifndef HDG_CALLNODE_H
00006 #define HDG_CALLNODE_H
00007
00008 #include "Node.h"
00009 #include "../basic/Token.h"
00010
00011 namespace hdg {
00012
00013     class CallNode: public Node{
00014     protected:
00015         Node* m_call;
00016         std::vector<Node*> m_list;
00017         TokenType m_oper;
00018
00019     public:
00020         CallNode();
00021         CallNode(const Position& position, Environment* environment);
00022         CallNode(Node* call, std::vector<Node*>list, TokenType oper, const Position& position,
      Environment* environment);
00023         ~CallNode();
00024
00025         void addNode(Node* node);
00026         void setOperator(TokenType oper);
00027         void setCall(Node* call);
00028
00029         std::string toString() override;
00030         Object* interpret() override;
00031     };
00032
00033 } // hdg
00034
00035 #endif //HDG_CALLNODE_H
```

5.13 ForNode.h

```
00001 //
00002 // Created by Magnesium on 2023/8/2.
00003 //
00004
00005 #ifndef HDG_FORNODE_H
00006 #define HDG_FORNODE_H
00007
00008 #include "../basic/Token.h"
00009 #include "Node.h"
00010
00011 namespace hdg {
00012
00013     class ForNode: public Node{
00014     protected:
00015         Token m_index;
00016         int m_from;
00017         int m_to;
00018         int m_step;
00019         Node* m_expr;
00020
00021     public:
00022         ForNode(const Token& index, int to, const Position& position, Environment* parent);
00023         ForNode(const Token& index, int from, int to, int step, Node* expr, const Position& position,
      Environment* parent);
00024         ~ForNode() override;
00025
00026         void setIndex(const Token& index);
00027         void setFrom(int from);
00028         void setTo(int to);
00029         void setStep(int step);
00030         void setExpr(Node* expr);
00031
00032         std::string toString() override;
00033         Object* interpret() override;
00034     };
00035
00036 } // hdg
00037
00038 #endif //HDG_FORNODE_H
```

5.14 IfNode.h

```
00001 //
00002 // Created by Magnesium on 2023/8/1.
00003 //
00004
00005 #ifndef HDG_IFNODE_H
00006 #define HDG_IFNODE_H
00007
00008 #include <vector>
00009 #include "Node.h"
00010 #include "../object/None.h"
00011
00012 namespace hdg {
00013
00014     class IfNode: public Node{
00015     protected:
00016         std::vector<Node*> conditions;
00017         std::vector<Node*> expressions;
00018         Node* elseExpression;
00019
00020     public:
00021         IfNode(const Position& position, Environment* parent);
00022         ~IfNode() override;
00023
00024         void addBranch(Node* condition, Node* expression);
00025
00026         std::string toString() override;
00027         Object* interpret() override;
00028     };
00029
00030 } // hdg
00031
00032 #endif //HDG_IFNODE_H
```

5.15 Node.h

```
00001 //
```

```
00002 // Created by Magnesium on 2023/7/19.
00003 //
00004
00005 #ifndef HDG_NODE_H
00006 #define HDG_NODE_H
00007
00008 #include <string>
00009 #include "../object/Object.h"
00010 #include "../basic/Position.h"
00011 #include "../basic/Environment.h"
00012
00013
00014 namespace hdg {
00015
00016     class Node {
00017     protected:
00018         Position m_position;
00019         Environment* m_environment;
00020
00021     public:
00022         Node();
00023         explicit Node(const Position& position);
00024         Node(const Position& position, Environment* environment);
00025         virtual ~Node() = 0;
00026
00027         Position* thisPosition();
00028         Environment* thisEnvironment();
00029
00030         virtual std::string toString() = 0;
00031         virtual Object* interpret() = 0;
00032     };
00033
00034 } // hdg
00035
00036 #endif //HDG_NODE_H
```

5.16 ObjAccessNode.h

```
00001 //
00002 // Created by Magnesium on 2023/7/27.
00003 //
00004
00005 #ifndef HDG_OBJACCESSNODE_H
00006 #define HDG_OBJACCESSNODE_H
00007
00008 #include "Node.h"
00009
00010 namespace hdg {
00011
00012     class ObjAccessNode: public Node{
00013     protected:
00014         std::string m_name;
00015
00016     public:
00017         ObjAccessNode(std::string name, const Position& position, Environment* environment);
00018         ~ObjAccessNode() override;
00019
00020         std::string toString() override;
00021         Object* interpret() override;
00022     };
00023
00024 } // hdg
00025
00026 #endif //HDG_OBJACCESSNODE_H
```

5.17 ObjAssignNode.h

```
00001 //
00002 // Created by Magnesium on 2023/7/27.
00003 //
00004
00005 #ifndef HDG_OBJASSIGNNODE_H
00006 #define HDG_OBJASSIGNNODE_H
00007
00008 #include "Node.h"
00009 #include "object_node/ObjectNode.h"
00010
00011 namespace hdg {
00012
```

```
00013     class ObjAssignNode: public Node{
00014     protected:
00015         std::string m_name;
00016         Node* m_expr;
00017 
00018     public:
00019         ObjAssignNode();
00020         ObjAssignNode(std::string name, Node* obj, const Position& position, Environment*
      environment);
00021         ~ObjAssignNode() override;
00022 
00023         void setExpr(Node* expr);
00024         std::string getName();
00025 
00026         std::string toString() override;
00027         Object* interpret() override;
00028     };
00029 
00030 } // hdg
00031 
00032 #endif //HDG_OBJASSIGNNODE_H
```

5.18 FuncObjNode.h

```
00001 //
00002 // Created by Magnesium on 2023/8/10.
00003 //
00004 
00005 #ifndef HDG_FUNCOBJNODE_H
00006 #define HDG_FUNCOBJNODE_H
00007 
00008 #include <set>
00009 #include "ObjectNode.h"
00010 #include "../ObjAssignNode.h"
00011 #include "../../object/None.h"
00012 #include "../../object/Function.h"
00013 
00014 namespace hdg {
00015 
00016     class FuncObjNode: public ObjectNode {
00017     protected:
00018         std::string m_name;
00019         std::vector<ObjAssignNode*> m_args;
00020         Node* m_body{};
00021 
00022     public:
00023         FuncObjNode();
00024         FuncObjNode(const Position& position, Environment* parent);
00025         ~FuncObjNode() override;
00026 
00027         void setArg(ObjAssignNode* arg);
00028         void setBody(Node* body);
00029         void setName(const std::string& name);
00030 
00031         std::string toString();
00032         Object* interpret();
00033     };
00034 
00035 } // hdg
00036 
00037 #endif //HDG_FUNCOBJNODE_H
```

5.19 NumObjNode.h

```
00001 //
00002 // Created by Magnesium on 2023/7/25.
00003 //
00004 
00005 #ifndef HDG_NUMOBJNODE_H
00006 #define HDG_NUMOBJNODE_H
00007 
00008 #include "ObjectNode.h"
00009 #include "../../object/Integer.h"
00010 #include "../../object/Float.h"
00011 
00012 namespace hdg {
00013 
00014     class NumObjNode: public Node{
00015     protected:
```

```
00016         union{
00017             long long i;
00018             double f;
00019         }m_value;
00020         std::string m_class;
00021
00022     public:
00023         NumObjNode(long long value, const Position& position);
00024         NumObjNode(float value, const Position& position);
00025         NumObjNode(double value, const Position& position);
00026         ~NumObjNode() override;
00027
00028         std::string toString() override;
00029         Object* interpret() override;
00030     };
00031
00032 } // hdg
00033
00034 #endif //HDG_NUMOBJNODE_H
```

5.20 ObjectNode.h

```
00001 //
00002 // Created by Magnesium on 2023/8/9.
00003 //
00004
00005 #ifndef HDG_OBJECTNODE_H
00006 #define HDG_OBJECTNODE_H
00007
00008 #include "../Node.h"
00009
00010 namespace hdg {
00011
00012     class ObjectNode: public Node{
00013     protected:
00014         std::string m_class;
00015
00016     public:
00017         ObjectNode();
00018         ObjectNode(std::string className, const Position& position, Environment* environment);
00019         ObjectNode(const Position& position, Environment* environment);
00020         ~ObjectNode() override;
00021
00022         void setClass(const std::string& className);
00023         std::string getClass();
00024
00025         virtual std::string toString() = 0;
00026         virtual Object* interpret() = 0;
00027     };
00028
00029 } // hdg
00030
00031 #endif //HDG_OBJECTNODE_H
```

5.21 StrObjNode.h

```
00001 //
00002 // Created by Magnesium on 2023/8/12.
00003 //
00004
00005 #ifndef HDG_STROBJNODE_H
00006 #define HDG_STROBJNODE_H
00007
00008 #include "ObjectNode.h"
00009 #include "../../object/String.h"
00010
00011 namespace hdg {
00012
00013     class StrObjNode: public ObjectNode{
00014     protected:
00015         std::string m_value;
00016
00017     public:
00018         StrObjNode();
00019         StrObjNode(std::string value, const Position& position, Environment* environment);
00020         ~StrObjNode() override;
00021
00022         void setValue(const std::string& value);
00023
```

```
00024         std::string toString() override;
00025         Object* interpret() override;
00026     };
00027
00028 } // hdg
00029
00030 #endif //HDG_STROBJNODE_H
```

5.22 StatementsNode.h

```
00001 //
00002 // Created by Magnesium on 2023/8/14.
00003 //
00004
00005 #ifndef HDG_STATEMENTSNODE_H
00006 #define HDG_STATEMENTSNODE_H
00007
00008 #include <vector>
00009 #include "Node.h"
00010
00011 namespace hdg {
00012
00013     class StatementsNode: public Node{
00014     protected:
00015         std::vector<Node*> m_statements;
00016
00017     public:
00018         StatementsNode();
00019         StatementsNode(const Position& position, Environment* environment);
00020         ~StatementsNode() override;
00021
00022         void append(Node* node);
00023
00024         std::string toString() override;
00025         Object* interpret() override;
00026     };
00027
00028 } // hdg
00029
00030 #endif //HDG_STATEMENTSNODE_H
```

5.23 UnaryOperatorNode.h

```
00001 //
00002 // Created by Magnesium on 2023/7/22.
00003 //
00004
00005
00006 #ifndef HDG_UNARYOPERATORNODE_H
00007 #define HDG_UNARYOPERATORNODE_H
00008
00009 #include "Node.h"
00010 #include "../basic/Token.h"
00011 #include "../object/Integer.h"
00012 #include "../object/Float.h"
00013
00014 namespace hdg {
00015
00016     class UnaryOperatorNode: public Node{
00017     protected:
00018         Token m_oper;
00019         Node* m_obj;
00020
00021     public:
00022         UnaryOperatorNode(TokenType oper, Node* obj, const Position& position, Environment*
      environment);
00023         UnaryOperatorNode(Token oper, Node* obj, const Position& position, Environment* environment);
00024         ~UnaryOperatorNode() override;
00025
00026         void setOperator(const Token& oper);
00027         void setObject(Node* obj);
00028
00029         Token getOperator();
00030         Node* getObject();
00031
00032         std::string toString() override;
00033         Object* interpret() override;
00034     };
00035
```

```
00036 } // hdg
00037
00038 #endif //HDG_UNARYOPERATORNODE_H
```

5.24 WhileNode.h

```
00001 //
00002 // Created by Magnesium on 2023/8/2.
00003 //
00004
00005 #ifndef HDG_WHILENODE_H
00006 #define HDG_WHILENODE_H
00007
00008 #include "Node.h"
00009 #include "../object/None.h"
00010
00011 namespace hdg {
00012
00013     class WhileNode: public Node {
00014     protected:
00015         Node* m_condition;
00016         Node* m_expression;
00017
00018     public:
00019         WhileNode(Node* condition, Node* expression, const Position& position, Environment* parent);
00020         ~WhileNode() override;
00021
00022         void setCondition(Node* condition);
00023         void setExpression(Node* expression);
00024
00025         std::string toString() override;
00026         Object* interpret() override;
00027
00028     };
00029
00030 } // hdg
00031
00032 #endif //HDG_WHILENODE_H
```

5.25 Float.h

```
00001 //
00002 // Created by Magnesium on 2023/7/25.
00003 //
00004
00005
00006 #ifndef HDG_FLOAT_H
00007 #define HDG_FLOAT_H
00008
00009 #include <iostream>
00010 #include "Object.h"
00011 #include "Integer.h"
00012
00013 namespace hdg {
00014
00015     class Float: public Object {
00016     private:
00017         double m_value;
00018
00019     public:
00020         explicit Float(float value);
00021         explicit Float(double value);
00022         explicit Float(double value, const Position& position);
00023
00024         [[nodiscard]] double getValue() const;
00025
00026         Object* plus(Object* other) override;
00027         Object* minus(Object* other) override;
00028         Object* mul(Object* other) override;
00029         Object* div(Object* other) override;
00030         Object* pow(Object* other) override;
00031
00032         Object* equation(Object* other) override;
00033         Object* notEquation(Object* other) override;
00034         Object* greaterThan(Object* other) override;
00035         Object* lessThan(Object* other) override;
00036         Object* greaterThanEquation(Object* other) override;
00037         Object* lessThanEquation(Object* other) override;
00038
```

```
00039         Object* andOperator(Object* other) override;
00040         Object* orOperator(Object* other) override;
00041         Object* notOperator() override;
00042 
00043         bool isTrue() override;
00044         std::string toString() override;
00045         Object* copy() override;
00046     };
00047 
00048 } // hdg
00049 
00050 #endif //HDG_FLOAT_H
```

5.26 Function.h

```
00001 //
00002 // Created by Magnesium on 2023/8/4.
00003 //
00004 
00005 #ifndef HDG_FUNCTION_H
00006 #define HDG_FUNCTION_H
00007 
00008 #include <sstream>
00009 #include "../basic/Environment.h"
00010 #include "../node/Node.h"
00011 #include "../node/ObjAssignNode.h"
00012 #include "../object/None.h"
00013 #include "Object.h"
00014 
00015 namespace hdg {
00016 
00017     class Function: public Object{
00018     protected:
00019         std::string m_name;
00020         std::vector<ObjAssignNode*> m_args;
00021         Environment* m_environment;
00022 
00023         Node* m_body;
00024     public:
00025         Function();
00026         Function(std::string name, std::vector<ObjAssignNode*>args, Environment* environment, Node*
      body, const Position& position);
00027         ~Function() override;
00028 
00029         Object* parenthesis(const std::vector<Object*>& list) override;
00030 
00031         std::string toString() override;
00032         Object* copy() override;
00033     };
00034 
00035 } // hdg
00036 
00037 #endif //HDG_FUNCTION_H
```

5.27 Integer.h

```
00001 //
00002 // Created by Magnesium on 2023/7/25.
00003 //
00004 
00005 
00006 #ifndef HDG_INTEGER_H
00007 #define HDG_INTEGER_H
00008 
00009 #include "Object.h"
00010 #include "Float.h"
00011 #include "../error/RuntimeError.h"
00012 
00013 namespace hdg {
00014 
00015     class Integer: public Object{
00016     private:
00017         long long m_value;
00018 
00019     public:
00020         explicit Integer(long long value);
00021         Integer(long long value, const Position& position);
00022 
00023         void setValue(long long value);
```

```
00024         long long getValue() const;
00025
00026         Object* plus(Object* other) override;
00027         Object* minus(Object* other) override;
00028         Object* mul(Object* other) override;
00029         Object* div(Object* other) override;
00030         Object* pow(Object* other) override;
00031
00032         Object* equation(Object* other) override;
00033         Object* notEquation(Object* other) override;
00034         Object* greaterThan(Object* other) override;
00035         Object* lessThan(Object* other) override;
00036         Object* greaterThanEquation(Object* other) override;
00037         Object* lessThanEquation(Object* other) override;
00038
00039         Object* andOperator(Object* other) override;
00040         Object* orOperator(Object* other) override;
00041         Object* notOperator() override;
00042
00043         bool isTrue() override;
00044         std::string toString() override;
00045         Object* copy() override;
00046     };
00047
00048 } // hdg
00049
00050 #endif //HDG_INTEGER_H
```

5.28 List.h

```
00001 //
00002 // Created by Magnesium on 2023/8/4.
00003 //
00004
00005 #ifndef HDG_LIST_H
00006 #define HDG_LIST_H
00007
00008 namespace hdg {
00009
00010     class List {
00011
00012     };
00013
00014 } // hdg
00015
00016 #endif //HDG_LIST_H
```

5.29 None.h

```
00001 //
00002 // Created by Magnesium on 2023/8/2.
00003 //
00004
00005 #ifndef HDG_NONE_H
00006 #define HDG_NONE_H
00007
00008 #include "Object.h"
00009 #include "Integer.h"
00010
00011 namespace hdg {
00012
00013     class None: public Integer{
00014     public:
00015         None();
00016
00017         Object* notOperator() override;
00018
00019         std::string toString() override;
00020         Object* copy() override;
00021     };
00022
00023 } // hdg
00024
00025 #endif //HDG_NONE_H
```

5.30 Object.h

```
00001 //
00002 // Created by Magnesium on 2023/7/25.
00003 //
00004
00005
00006 #ifndef HDG_OBJECT_H
00007 #define HDG_OBJECT_H
00008
00009 #include <string>
00010 #include <utility>
00011 #include <cmath>
00012 #include <functional>
00013 #include "../basic/Position.h"
00014 #include "../error/RuntimeError.h"
00015
00016 namespace hdg {
00017
00018     class Object {
00019     protected:
00020         std::string m_class;
00021         Position m_position;
00022
00023     public:
00024         Object();
00025         explicit Object(std::string className);
00026         Object(std::string className, const Position& position);
00027
00028         virtual ~Object();
00029
00030         std::string getClass();
00031
00032         virtual Object* plus(Object* other);
00033         virtual Object* minus(Object* other);
00034         virtual Object* mul(Object* other);
00035         virtual Object* div(Object* other);
00036         virtual Object* pow(Object* other);
00037
00038         virtual Object* equation(Object* other);
00039         virtual Object* notEquation(Object* other);
00040         virtual Object* greaterThan(Object* other);
00041         virtual Object* lessThan(Object* other);
00042         virtual Object* greaterThanEquation(Object* other);
00043         virtual Object* lessThanEquation(Object* other);
00044
00045         virtual Object* andOperator(Object* other);
00046         virtual Object* orOperator(Object* other);
00047         virtual Object* notOperator();
00048
00049         virtual Object* parenthesis(const std::vector<Object*>& list);
00050         virtual Object* brackets(Object* other);
00051         virtual Object* braces(const std::vector<Object*>& list);
00052
00053         virtual bool isTrue();
00054         virtual void illegalOperator();
00055         virtual std::string toString() = 0;
00056         virtual Object* copy() = 0;
00057     };
00058 } // hdg
00059
00060 #endif //HDG_OBJECT_H
```

5.31 String.h

```
00001 //
00002 // Created by Magnesium on 2023/8/4.
00003 //
00004
00005 #ifndef HDG_STRING_H
00006 #define HDG_STRING_H
00007
00008 #include "Object.h"
00009
00010 namespace hdg {
00011
00012     class String: public Object {
00013     protected:
00014         std::string m_value;
00015
00016     public:
00017         String();
00018         String(std::string value);
```

```
00019         String(std::string value, const Position& position);
00020         ~String() override;
00021
00022         std::string getValue();
00023
00024         Object* plus(Object* other) override;
00025         bool isTrue() override;
00026
00027         std::string toString() override;
00028         Object* copy() override;
00029     };
00030
00031 } // hdg
00032
00033 #endif //HDG_STRING_H
```

5.32 Parser.h

```
00001 //
00002 // Created by Magnesium on 2023/7/16.
00003 //
00004
00005
00006 #ifndef HDG_PARSER_H
00007 #define HDG_PARSER_H
00008
00009 #include <vector>
00010 #include <utility>
00011 #include <functional>
00012 #include <set>
00013 #include <iostream>
00014 #include "basic/Token.h"
00015 #include "basic/Environment.h"
00016 #include "node/Node.h"
00017 #include "node/BinaryOperatorNode.h"
00018 #include "node/UnaryOperatorNode.h"
00019 #include "node/object_node/NumObjNode.h"
00020 #include "node/ObjAssignNode.h"
00021 #include "node/ObjAccessNode.h"
00022 #include "node/object_node/FuncObjNode.h"
00023 #include "node/object_node/StrObjNode.h"
00024 #include "node/IfNode.h"
00025 #include "node/ForNode.h"
00026 #include "node/WhileNode.h"
00027 #include "node/CallNode.h"
00028 #include "node/StatementsNode.h"
00029 #include "error/InvalidSyntaxError.h"
00030
00031 namespace hdg {
00032     class Parser {
00033     protected:
00034         std::vector<Token> m_tokens;
00035         std::vector<Token>::iterator m_currentToken;
00036         Environment* m_environment;
00037
00038     public:
00039         Parser(std::vector<Token> tokens, Environment* environment);
00040         Node* run();
00041
00042     protected:
00043         void advance();
00044         void retreat();
00045
00046         Node* expr(Environment* environment);
00047         Node* compExpr(Environment* environment);
00048         Node* arithExpr(Environment* environment);
00049         Node* term(Environment* environment);
00050         Node* factor(Environment* environment);
00051         Node* power(Environment* environment);
00052         Node* call(Environment* environment);
00053         Node* atom(Environment* environment);
00054         Node* ifExpr(Environment* environment);
00055         Node* forExpr(Environment* environment);
00056         Node* whileExpr(Environment* environment);
00057         Node* funcExpr(Environment* environment);
00058         Node* statements(Environment* environment);
00059
00060         Node* binaryOperator(
00061                 Environment* environment,
00062                 const std::set<Token, std::less<»&opers,
00063                 std::function<Node*(Environment* envir)> funA,
00064                 std::function<Node*(Environment* envir)> funB=nullptr
00065                         );
```

```
00066         Node* unaryOperator(
00067                 Environment* environment,
00068                 const std::set<Token, std::less<»&opers,
00069                 std::function<Node*(Environment* envir)> fun
00070                 );
00071     };
00072
00073 } // hdg
00074
00075 #endif //HDG_PARSER_H
```

5.33 Shell.h

```
00001 //
00002 // Created by Magnesium on 2023/7/14.
00003 //
00004
00005
00006 #ifndef HDG_SHELL_H
00007 #define HDG_SHELL_H
00008
00009 #include <iostream>
00010 #include <sstream>
00011 #include <fstream>
00012 #include "Lexer.h"
00013 #include "Parser.h"
00014 #include "Interpreter.h"
00015 #include "error/IllegalCharError.h"
00016 #include "basic/Environment.h"
00017
00018 namespace hdg {
00019
00020     class Shell {
00021     protected:
00022         Mode m_mode;
00023         std::string m_fPath;
00024
00025     public:
00026         Shell();
00027
00028         [[noreturn]] void exe(int argc, char* argv[]);
00029
00030         static std::string input();
00031         static std::string input(const std::string& path);
00032     };
00033
00034 } // hdg
00035
00036 #endif //HDG_SHELL_H
```

Index