

Commit: 328376e

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-11-05 14:53:58

Virtual machine for single main method

Summary: The VM is working well for input, output and sequential statements.

Issues: Arrays not yet implemented, notices unitary operators were never implemented and no function calls work.

Commit: cd470a6

Author: EliasMera

Commit: EliasMera

Date: 2017-11-05 14:32:28

Adds WriteLine function

Commit: d57a23c

Author: EliasMera

Commit: EliasMera

Date: 2017-11-05 14:25:06

Adds ss_exp to Write function

Commit: cb57bd2

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-11-04 18:23:53

Make switch for VM

Summary: Implicit parent constructor calls removed because they are unnecessary.

Commit: 61b48b2

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-11-04 14:46:26

Read constants and classes from .obj file

Commit: b3eeae4

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-11-04 13:22:57

Create temp.obj file for virtual machine

Commit: 61d3e78

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-11-04 12:04:29

Constants are in their own table

Summary: Each class had control over their constants, this made some of them collide. Therefore, all constants in a single file are taken into account.

Commit: ab72f16

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-11-04 00:35:48

Methods and classes have size variables

Summary: These variables will be used to know how much memory to allocate when a class or function is created.

Commit: 7cfb3f9

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-11-04 00:15:20

Add main method search and pointer

Summary: Files must now have at least one class and must always have at least one main method. If not, compilation fails.

Commit: 26bc31d

Author: EliasMera

Commit: EliasMera

Date: 2017-11-03 21:14:56

Adds arrays functionality

Commit: 34ad6a5

Author: EliasMera

Commit: EliasMera

Date: 2017-11-03 14:53:00

Adds read functions for each type and test inside complex_class.mom

Commit: bda5764

Author: EliasMera

Commit: EliasMera

Date: 2017-11-02 00:14:04

Adds new dim field on arguments and lets arrays of n-dimensions declaration

Commit: 66ef711

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-30 16:54:56

Implicit constructor call implemented

Summary: Tests updated, but checked visually.

Commit: 7901ce9

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-30 16:03:44

Remove constructor parameters

Issues: The implicit parent constructor call is not yet implemented.

Commit: 0c4e85d

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-30 09:41:14

Implemented return statements anywhere

Commit: e142ad4

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-30 09:06:24

Constructors working, except for parent

Commit: b000818

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-29 20:40:16

Function and constructor calls had bugs.

Summary: I tested and fixed for: 1) single function calls. 2) functions called from parent. 3) fuctions called by reference. 4) constructors with and without arguments.

Issues: Still have to check parent() call and check for return statementes in blocks, which are not accounted for.

Commit: 66e9949

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-28 11:17:53

Commit: ond temp commit

Author:

Commit:

Date:

Issues: Testing required.

Commit: 3e26753

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-27 20:51:06

Function calls working as expected

Summary: Functions are now supported.

Issues: Constructors still need some work done.

Commit: fd10465

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-27 20:19:44

Commit: p commit for security

Author:

Commit:

Date:

Issues: Functions are not inserting anything to the operand stack. Recommendation from teacher still to be implemented.

Commit: f06880a

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-26 16:10:07

Simple inheritance for functions achieved

Summary: Compiler looks for methods in ancestors if a function is not found locally.

Issues: It will choose only the first one it finds with that name, since we use function names as keys. Issues is pending and not mandatory to fix.

Commit: 17c09b9

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-26 15:00:45

Interfaces working on compilation

Issues: Types of arguments that are interfaces still not supported or checked.

Commit: 5f6c79d

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-25 22:45:31

Constructors working

Summary: Initial implementation of constructors working, but still issues to be fixed after some queries have been asked to professor.

Commit: 9cd7d21

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-24 15:59:10

Quad generation for constructors implemented

Commit: e2f1726

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-24 15:46:28

Actions for method definitions enabled

Summary: Everything works fine for this aspect, but constructors are still not considered.

Commit: eac2f81

Author: EliasMera

Commit: EliasMera

Date: 2017-10-17 17:44:21

Supports read, generates corresponding Quadrupole

Commit: 40e6374

Author: EliasMera

Commit: EliasMera

Date: 2017-10-17 17:43:34

Supports read, generates corresponding Quadrupole

Commit: bdcdda3

Author: EliasMera

Commit: EliasMera

Date: 2017-10-17 17:27:10

Supports write, generates corresponding Quadrupole

Commit: 96e2cb4

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-15 15:08:48

pre-condition loop statements

Summary: the compiler produces quadruples correctly for while statements.

Commit: 6874958

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-15 12:07:30

Add two-way conditional statement.

Summary: Add test for if-else statements. Passes.

Commit: ac636a4

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-15 11:15:07

Add one-way conditional statement

Summary: The single if statement is implemented and generates the appropriate quadruples. Also, the Operation Enum was added for extra quadruple operations like GOTO. The enums now use explicitly defined numbers instead of the auto() method.

Commit: 6a14a70

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-15 11:14:38

Modify log pdf to new version

Commit: 765c355

Author: EliasMera

Commit: EliasMera

Date: 2017-10-14 19:52:42

Appends types of arguments of each method in its corresponding list

Commit: dc00845

Author: EliasMera

Commit: EliasMera

Date: 2017-10-14 19:40:19

Supports array assignation

Commit: 6f24035

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-14 14:14:40

Small Error in test, undefined variable caught

Commit: 5e734bf

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-14 14:09:30

Working assignations

Summary: Assignations working for definitions and normal assignations.

Issues: 1) the assignations cannot be made to array elements
2) the assignations do not support constructors yet.

Commit: bca6f2e

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-14 13:37:25

Add argument type list

Summary: As specified in lecture, the class object has now a list of the types of its arguments per method.

Issues: Filling of the list not yet implemented.

Commit: 0dc00fb

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-14 10:32:30

Make Unit Tests pass with quadrupoles

Commit: e0b2bc7

Author: EliasMera

Commit: EliasMera

Date: 2017-10-13 22:45:11

Add check validation if a variable is an array in case of trying to access is with `[]` and if the index is inside its bounds

Commit: 98bec53

Author: EliasMera

Commit: EliasMera

Date: 2017-10-13 22:14:32

Add permit of exp like `arr[1] + arr[2]`

Commit: 8e57ce4

Author: EliasMera

Commit: EliasMera

Date: 2017-10-13 21:38:01

Add read fields from classes and add them to global variables with address

Commit: 74d6f1f

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-13 10:26:36

Add add_argument to method class for fields

Commit: 0b57346

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-12 23:17:48

Generate quadrupoles for expressions

Summary: The quadrupoles are generated with their relative virtual addresses as seen in class.

Issues: Not all unit tests pass because there is still issues if the types are not atomic.

Commit: e599056

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-11 22:05:00

Add addresses for local and const elements

Summary: The addresses work, but no check is done to see if they were already defined.

Commit: 3ec54af

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-11 20:35:35

quadrupole generation for expressions

Summary: The listeners generate global quadrupoles, but are not numeric. The expressions test was generated to test such cases.

Commit: d0d12b4

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-09 23:37:15

Change grammar to accomodate expression rules

Commit: 63b9960

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-09 20:10:25

Add counters and memory segment limits

Summary: The segments are divide into global, local, temp and constants. Each one of the latter divided into Int, Real, Text, and Boolean.

Commit: ce6d2f7

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-07 17:26:47

Add script to generate automatically the logs

Summary: Generates automatically the logs of the

Commit: yect based on the commits made by the

Author:

Commit:

Date:

developers.

Commit: identifies the first line of the commit

Author:

Commit:

Date:

and is divided into two sections: 'summary' and 'issues' which it will detect and put in their own section.

Commit: 95f46e3

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-07 10:18:24

Unit test enhanced

Commit: 08d5314

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-07 01:05:35

Improve unit tests, they now check structure

Commit: f1bd53e

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-06 12:09:48

Enable arguments for specifications

Summary: The specifications can gather their arguments and methods and the first two tests were fixed to take into account this.

Commit: 65150b2

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-06 09:59:57

Edit tests

Commit: adfbb80

Author: EliasMera

Commit: EliasMera

Date: 2017-10-05 21:15:11

Adds insertion of local variables of a method

Commit: d317de8

Author: EliasMera

Commit: EliasMera

Date: 2017-10-05 18:20:20

Fix grammar

Commit: 6d911aa

Author: EliasMera

Commit: EliasMera

Date: 2017-10-05 16:11:16

Adds tests for Windows

Commit: 5cc8b3a

Author: EliasMera

Commit: EliasMera

Date: 2017-10-05 11:10:28

Fixed /r error in grammar

Commit: 5ba7799

Author: EliasMera

Commit: Marco A. Peyrot

Date: 2017-10-05 10:39:41

Assignment of arguments to their corresponding methods

Summary: The issue was fixed by changing the grammar to several function args in the method definition.

Commit: 94b21a2

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-04 19:37:22

Check for method and class uniqueness

Summary: If a class, specification or method is redefined, an Error is thrown.

Commit: f5266e0

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-04 17:20:51

Eliminate multiple constructor support

Summary: To ease initial development, the language will not support multiple constructors or methods with the same names. This language policy may change in the future.

Commit: 2de8e5f

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-04 15:52:22

Read arguments

Summary: The parser is capable of creating the tables for classes, specifications and enumerations with their corresponding methods and arguments. However, local variables declared within a method are not yet supported.

Issues: Even though it works correctly, the key for the methods is only their name. This will not enable us to define methods with the same name, but different argument numbers or types. This should be changed.

Commit: 2c0f9e0

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-03 17:30:39

Add Class, Method and Enumeration classes

Summary: They will be subject to change.

Commit: cb8f722

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-01 21:33:09

Add enumeration initial implementation

Commit: 2bbd535

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-01 19:27:04

Add enumeration class

Summary: This class is responsible for holding the metadata of an enumeration in the MoM language.

Commit: c0573c3

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-01 19:06:48

Add semantic table for supported types

Commit: 204573f

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-10-01 14:28:51

Fix grammar and add additional test

Summary: The statute rule permitted the assignation of expressions or constructors to variables without a specified type. The new tests checks that this is no longer the case.

Commit: 3fea706

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-09-24 17:56:02

Expression grammar mistake

Summary: The expressions had brackets instead of parenthesis. Issue fixed. Test improved to check this issue.

Commit: 0307003

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-09-24 17:44:58

Changes done to grammar

Summary: The grammar now supports array syntax and individual functions. boolean constants TRUE and FALSE added to the language.

Commit: dda939a

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-09-24 15:44:49

Restructure project

Summary: The project structure and file hierarchy was changed to comply with the Hitchhiker's Guide to Python. First Unit test working.

Commit: abda926

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-09-24 14:34:14

Scripts folder created Sumamry: To automatize the project testing and development this folder was created. It will hold all the scripts generated for the project. It contains the run_tests.py script which will run all the unit tests developed for the project.

Commit: 075533d

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-09-24 14:32:09

moved test files to resources folder

Summary: The resources folder was created to have more control over where everything is stored. The initial test files were moved to this location.

Commit: 3cd0cac

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-09-24 14:31:21

update removed files (moved to other folder)

Commit: 47a5377

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-09-24 14:29:33

Generate Python 3 parser

Summary: The parser was rebuilt to work on python 3. The grammar string rule was fixed, because it had a slight syntax error.

Commit: dfb0615

Author: Marco A. Peyrot

Commit: Marco A. Peyrot

Date: 2017-09-24 14:09:47

Moved source files to source folder

Commit: 09645d7

Author: Marco Peyrot

Commit: Marco Peyrot

Date: 2017-09-22 23:07:29

Grammar changed, test files added

Summary: Rules updated to be more flexible (need to be checked more carefully). Three test files added. All pass.

Commit: e591498

Author: Marco Peyrot

Commit: Marco Peyrot

Date: 2017-09-22 22:33:29

Grammar modified

Summary: Several issues were uncovered in the grammar, these are described below: 1) Nothing keyword was not properly spelled. 2) Whitespace skipping was not specified, new rule: WS : [\t\n]+ -> skip; 3) IS_A and OF_TYPE tokens joined by underscore '_' 4) Made it possible to define several classes in program : (class_rule)+ EOF 5) Added semi-colon to function definitions to make code consistent.

Commit: a1d719c

Author: Marco Peyrot

Commit: Marco Peyrot

Date: 2017-09-22 22:20:52

Add Simple ErrorListener messages

Summary: The messages are simple enough to help us find bugs in the grammar. The main method reads from the command line and prints 'OK' if the program is good or an error message if not.

Commit: 2f8fe30

Author: Marco Peyrot

Commit: Marco Peyrot

Date: 2017-09-22 17:26:21

Permit EOF file recognition

Summary: The Error handler and Main function are working.
EOF recognition was added to the grammar.

Issues: Missing reading file code. Only working with Python 2.7

Commit: 13abc2c

Author: Marco Peyrot

Commit: Marco Peyrot

Date: 2017-09-22 15:14:29

Moved grammar files to project directory

Summary: Will avoid managing independent copies

Commit: 2358dad

Author: Marco Peyrot

Commit: Marco Peyrot

Date: 2017-09-22 15:09:34

Add generated parser/lexer files

Summary: Files for the parser and lexer were generated cleanly. The code for their generation was: antlr4 -Dlanguage=Python3 MyGrammar.g4

Commit: 76a0a64

Author: Marco Peyrot

Commit: Marco Peyrot

Date: 2017-09-22 15:02:39

Warnings in grammar removed

Summary: Return and Boolean lexemas were not defined. the arguments rule is optional and it contained an empty rule. This was eliminated to avoid ambiguity.

Commit: f1dc4d8

Author: Marco Peyrot

Commit: Marco Peyrot

Date: 2017-09-22 14:43:09

Fix compile errors

Summary: The 'Type' token is a keyword in Python, therefore it cannot be used in our language. The lexema remained, but the rule changed to 'super_type'. The 'Property' token is another keyword in Python 3.x.x therefore its rule name was changed to 'field' as well as its lexema.

Commit: 81d886e

Author: Marco Peyrot

Commit: Marco Peyrot

Date: 2017-09-22 14:25:03

Add IDE and extra files to .gitignore

Summary: The recommended files were taken from: <https://gist.github.com/fjcaetano/4069311>

Commit: d70508b

Author: Marco Peyrot

Commit: Marco Peyrot

Date: 2017-09-22 14:20:25

Remove DS.Store file tracking

Commit: bbb66fe

Author: Marco Peyrot

Commit: Marco Peyrot

Date: 2017-09-22 13:40:42

Remove useless code

Summary: The methods for error recognition in the grammar file are made for Java. The ones for Python will be in the Main file.

Commit: cf97175

Author: Marco Peyrot

Commit: Marco Peyrot

Date: 2017-09-22 10:25:09

Project proposal added

Summary: The proposal's diagrams are not corrected. These will be in their own document.

Commit: 0da7117

Author: Marco Peyrot

Commit: Marco Peyrot

Date: 2017-09-22 10:10:54

final review of grammar

Summary: The final corrections of the grammar were made. There were some minor issues, and some of them helped to identify other errors in the syntax diagrams. All of them were corrected. An additional diagram for the methods inside a specification was added. They are methods without a body that serve as the methods defined in a Java interface.

Issues: The diagrams and grammar are not entirely reviewed for ambiguity and the grammar has not yet been compiled.

Commit: 1bf15c0

Author: Marco Peyrot

Commit: Marco Peyrot

Date: 2017-09-22 01:17:56

1/2 revision of grammar

Commit: e95f9cd

Author: EliasMera

Commit: EliasMera

Date: 2017-09-19 17:16:56

Moves grammar

Commit: 5d4fd73

Author: EliasMera

Commit: GitHub

Date: 2017-09-19 17:10:53

Adds grammar

Commit: 5a57c0a
Author: Marco Peyrot Commit: GitHub Date: 2017-09-18 08:53:35
<div></div>

Commit: tial commit
Author: Commit: Date:
<div></div>