JavaLite

Frances Cao, Hongfei Chen, Mateo Maturana, Ian Chen

Team Members

Frances Cao

System Architect

Hongfei Chen

Manager

Mateo Maturana

Language Guru

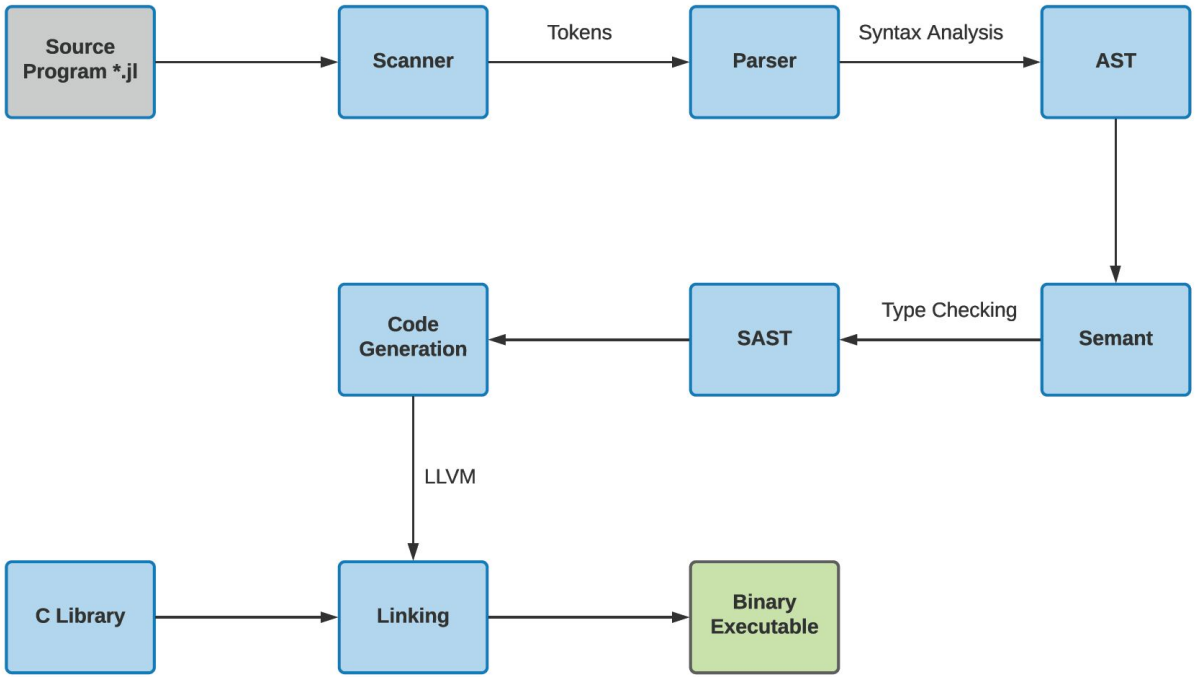
Ian Chen

Tester

Overview

- Although Java is popular for beginners, it may be difficult to learn, understand, and remember the syntax at first
- JavaLite aims to provide similar but simplified syntax -- easier for beginners
    - Functional Programming
    - String Built-Ins
    - Simplified Class Structure

Architectural Design



Variable Declarations

- In MicroC, variables must first be declared before being assigned a value
- In JavaLite, variables are assigned a value when declared
    - More intuitive for beginners

```
int x = 2;    string str = “hello”  ;  bool[] b = [false, true];
```

No Main Method

Java

```
public class HelloWorld {
        public static void main(String[] args) {
                System.out.print(“Hello, World!”);
        }
}
```

JavaLite

```
print(“Hello, World!”);
```

Data Types and Operations

Primitive Types

Data Type	Operation Supported
bool	Logic operators
int	Arithmetics, Comparisons
double	Arithmetics, Comparisons
string	Comparisons

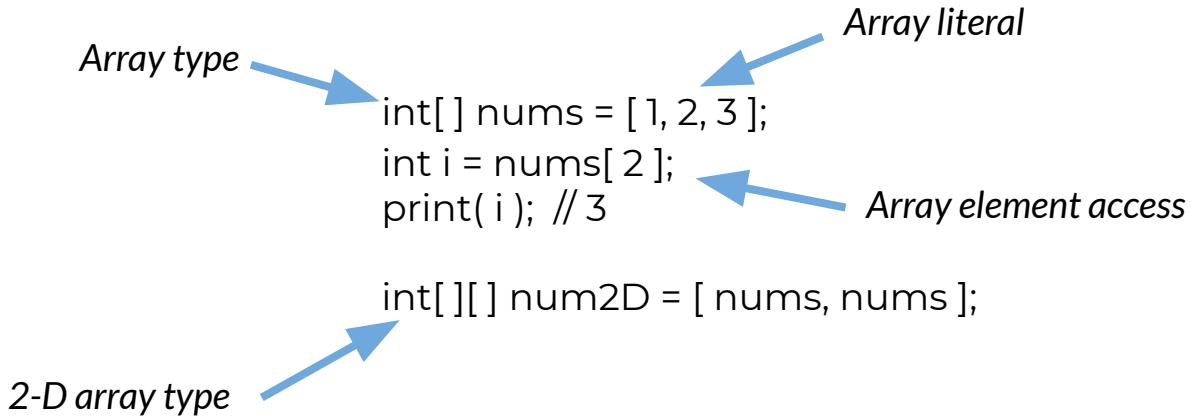
Array

An array can hold elements of all data types, including the primitives types, array types and object types defined by classes. Arrays support element accessing operation.

Class

A class declaration may have any number of fields of any type and it allows recursive class declarations. A constructor is automatically generated with the class name as its function name and the fields as its arguments.

Array

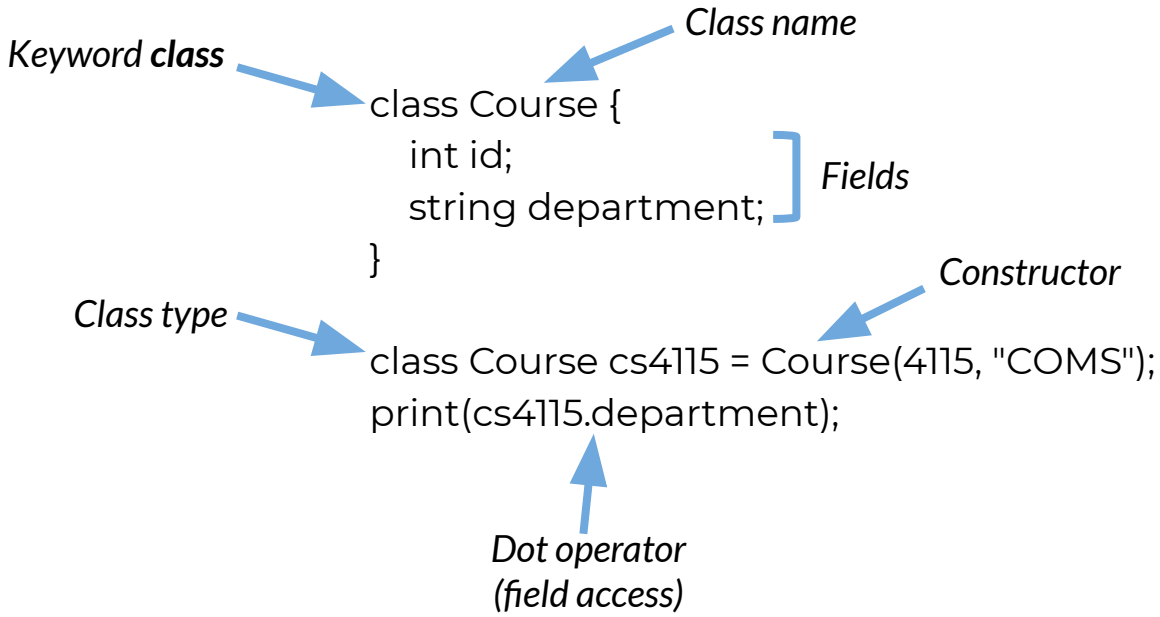

Array type
Array literal
int[ ] nums = [1, 2, 3];
int i = nums[2];
print(i); // 3
Array element access
int[ ][ ] num2D = [nums, nums];
2-D array type

Array

```
… …
define i32 @main() {
entry:
 %arr = alloca i32*, i32 4
 %arrptr = bitcast i32** %arr to i32*
 %arrelt = getelementptr i32, i32* %arrptr, i32 0
 store i32 1, i32* %arrelt
 %arrelt1 = getelementptr i32, i32* %arrptr, i32 1
 store i32 2, i32* %arrelt1
 %arrelt2 = getelementptr i32, i32* %arrptr, i32 2
 store i32 3, i32* %arrelt2
 … …
 %nums = alloca i32*
 store i32* %arrptr, i32** %nums

 %nums3 = load i32*, i32** %nums
 %acceltptr = getelementptr i32, i32* %nums3, i32 2
 %accelt = load i32, i32* %acceltptr
 %i = alloca i32
 store i32 %accelt, i32* %i
 ret i32 0
}
```

```
int[ ] nums = [ 1, 2, 3 ];
int i = nums[ 2 ];
```

Class


Keyword class
Class name
class Course {
int id;
string department;
}
Fields
Class type
Constructor
class Course cs4115 = Course(4115, "COMS");
print(cs4115.department);
Dot operator
(field access)

```
class Course {
   int id;
   string department;
}
class Course cs4115 = Course(4115, "COMS");
```

Class

```
... ...
%Course = type { i32, i8* }
... ...
define %Course* @Course(i32 %id, i8* %department) {
entry:
 ... ...
 %malloccall = tail call i8* @malloc(i32 ptrtoint (%Course* getelementptr (%Course, %Course* null, i32 1) to i32))
 %constrObj = bitcast i8* %malloccall to %Course*
 %id3 = load i32, i32* %id1
 %id4 = getelementptr inbounds %Course, %Course* %constrObj, i32 0, i32 0
 store i32 %id3, i32* %id4
 %department5 = load i8*, i8** %department2
 %department6 = getelementptr inbounds %Course, %Course* %constrObj, i32 0, i32 1
 store i8* %department5, i8** %department6
 ... ...
 ret %Course* %obj7
}

define i32 @main() {
entry:
 %Course_result = call %Course* @Course(i32 4115, i8* getelementptr inbounds ([5 x i8], [5 x i8]* @str, i32 0, i32 0))
 ... ...
}
```

Built-in Functions - Print

```
int i = 3;
int j = 9;
print(i + j);
print(i / j);
```

```
string s = “Hello World!”;
print(s);
```

```
print(true && false);
print(!false);
print("hello" + " world");
print(1 + 2 * 3 + 4);
```

```
double a = 3.14159267;
double b = -2.71828;
print(a + b);
print(a == b);
```

```
int fib(int x)
{
  if (x < 2) return 1;
  return fib(x-1) + fib(x-2);
}
print(fib(0));
print(fib(3));
```

```
string[] strArr = ["hello", "world"];
print(strArr[0]);
```

Built-in Functions - String/Arrays

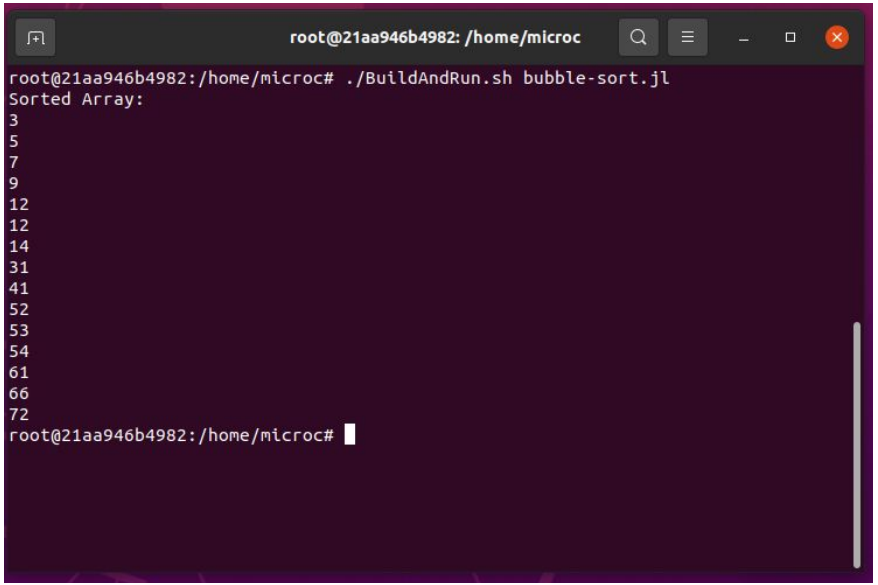
- reverse
- upper
- lower
- len
- substring
- indexOf
- concat

- length

```
string s = "Hello World!!";
string t = "FooFooDooDooBar";

int si = indexOf(s,"e");
print(si);

int ti = indexOf(t,"D");
print(ti);

int sn = indexOf(s,"D");
print(sn);

string c = concat(s,t);
print(c);

s = substring(s,0,6);
c = concat(s,t);
c = concat(c,"!!!!!");
print(c);
```

Testing - Tools

- BuildAndRun
- TestAll
- PerformanceBench



./BuildAndRun.sh bubble-sort.jl

Testing - Representative Programs

GCD

```
int gcd(int a, int b) {
  while (a != b)
    if (a > b) a = a - b;
    else b = b - a;
  return a;
}

print(gcd(14,21));
print(gcd(8,36));
print(gcd(99,121));
```

Bubble Sort (Array)

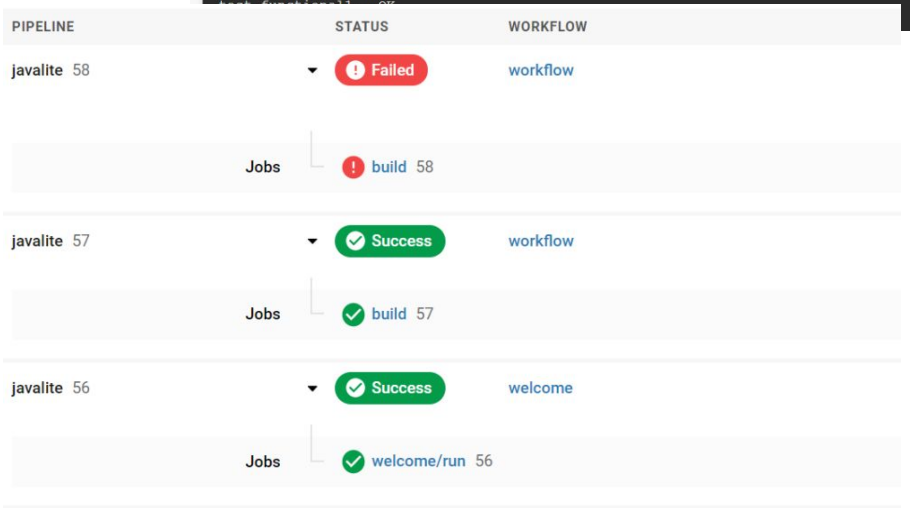
```
int[] sortingArr =
[52,14,72,5,66,7,12,31,9,3,54,41,53,12,61];
int i = 0;
int j = 0;
int tmpForSwap = 0;
int length = 15;

for (i = 0 ; i < length-1 ; i = i + 1) {
    for (j =0; j< length-i-1;j = j + 1){
          if (sortingArr[j] > sortingArr[j+1]){
                tmpForSwap = sortingArr[j];
                sortingArr[j] = sortingArr[j+1];
                sortingArr[j+1] = tmpForSwap;
          }
    }
}

for (i = 0 ; i < length; i = i + 1) { print(sortingArr[i]);}
```

Class

```
class Person {
    string name;
    int age;
    string phrase;
}

void sayhi(class Person p) {
    string n = concat(p.name, " say:");
    print(n);
    string s = p.phrase;
    for (int i = 0; i < p.age; i = i + 1) {
        s = concat(s, p.phrase);
    }
    print(s);
}


class Person alice = Person("Alice", 3, "hey");
sayhi(alice);
```

Testing - CI/CD

```
version: 2.0
jobs:
 build:
   docker:
     - image: columbiasedwards/plt
       options : --rm -it -w=/home/microc
   steps:
     - checkout
     - run:
         ...
```



Testing - Failure Cases

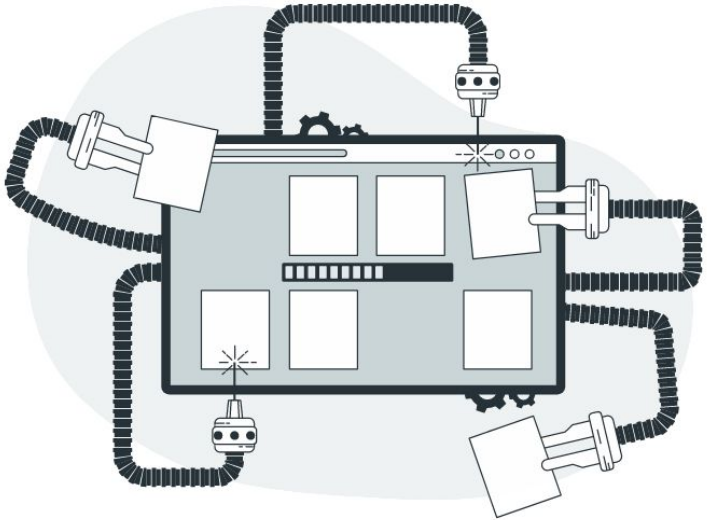
test-functional2.jl

```
int x = 0;
print(x);

int update(int x)
{
    x = x + 1;
    return x;
}

x = update(x);
print(x);
```

fail-functional1.jl

```
int x = 0;


void update()
{
    x = x + 1;
}

update();
print(x);
```

DEMO



Future Work

- Array built-ins (pop/append)
- Class inheritance
- 2D Array Improvements
- Generic

Questions?