variable✔

function declarations✔

blocks✔

conditionals statements and expressions✔

while loops->only for loop in Go✔

Concurrency Control:

Go routines

Channels

Lock

Wait groups

|  |  |  |
| --- | --- | --- |
|  | Go | JSONAST |
| Variable  DONE | package main  func main() {  var x = 3  } | {"tag":"blk","body":{"tag":"seq","stmts":[{"tag":"assmt","sym":[{"tag":"nam","sym":"x"}],"expr":[{"tag":"lit","val":3}]}]}}  DONE |
|  | package main  func main() {  var e,f int  } | {"tag":"blk","body":{"tag":"seq","stmts":[{"tag":"assmt","sym":[{"tag":"nam","sym":"e"},{"tag":"nam","sym":"f"}],"expr":[{"tag":"lit","val":0},{"tag":"lit","val":0}]}]}}  DONE  Only return last f |
|  | package main  func main() {  var a, b, c, d= "hi", 1, true, 1.12  }  Some issue on display boolean | {"tag":"blk","body":{"tag":"seq","stmts":[{"tag":"assmt","sym":[{"tag":"nam","sym":"a"},{"tag":"nam","sym":"b"},{"tag":"nam","sym":"c"},{"tag":"nam","sym":"d"}],"expr":[{"tag":"lit","val":"hi"},{"tag":"lit","val":1},{"tag":"lit","val":1.12},{"tag":"lit","val":true}]}]}}  DONE  Only return last d |
|  | package main  func main() {  f, g := "apple", "pine"  } | {"tag":"blk","body":{"tag":"seq","stmts":[{"tag":"assmt","sym":[{"tag":"nam","sym":"f"},{"tag":"nam","sym":"g"}],"expr":[{"tag":"lit","val":"apple"},{"tag":"lit","val":"pine"}]}]}}  DONE  Only return last g |
|  | package main  func main() {  var x = 3+4  } | {"tag":"blk","body":{"tag":"seq","stmts":[{"tag":"assmt","sym":[{"tag":"nam","sym":"x"}],"expr":[{"tag":"binop","sym":"+","frst":{"tag":"lit","val":3},"scnd":{"tag":"lit","val":4}}]}]}}  DONE 7 |
| Conditional  If else if else | package main  func main() {  n := 0  if n > 0 {  n += 1  } else if n == 0 {  n += 7  } else {  n -= 1  }  } | {"tag":"blk","body":{"tag":"seq","stmts":[{"tag":"assmt","sym":[{"tag":"nam","sym":"n"}],"expr":[{"tag":"lit","val":0}]},{"tag":"cond","pred":{"tag":"binop","sym":">","frst":{"tag":"nam","sym":"n"},"scnd":{"tag":"lit","val":0}},"cons":{"tag":"seq","stmts":[{"tag":"assmt","sym":[{"tag":"nam","sym":"n"}],"expr":[{"tag":"binop","sym":"+","frst":{"tag":"nam","sym":"n"},"scnd":{"tag":"lit","val":1}}]}]},"alt":{"tag":"cond","pred":{"tag":"binop","sym":"==","frst":{"tag":"nam","sym":"n"},"scnd":{"tag":"lit","val":0}},"cons":{"tag":"seq","stmts":[{"tag":"assmt","sym":[{"tag":"nam","sym":"n"}],"expr":[{"tag":"binop","sym":"+","frst":{"tag":"nam","sym":"n"},"scnd":{"tag":"lit","val":7}}]}]},"alt":{"tag":"seq","stmts":[{"tag":"assmt","sym":[{"tag":"nam","sym":"n"}],"expr":[{"tag":"binop","sym":"-","frst":{"tag":"nam","sym":"n"},"scnd":{"tag":"lit","val":1}}]}]}}}]}}  DONE |
| Only if | package main  func main() {  n := 0  if n == 0 {  n += 1  }  } | {"tag":"blk","body":{"tag":"seq","stmts":[{"tag":"assmt","sym":[{"tag":"nam","sym":"n"}],"expr":[{"tag":"lit","val":0}]},{"tag":"cond","pred":{"tag":"binop","sym":"==","frst":{"tag":"nam","sym":"n"},"scnd":{"tag":"lit","val":0}},"cons":{"tag":"seq","stmts":[{"tag":"assmt","sym":[{"tag":"nam","sym":"n"}],"expr":[{"tag":"binop","sym":"+","frst":{"tag":"nam","sym":"n"},"scnd":{"tag":"lit","val":1}}]}]}}]}}  DONE |
| If & Else | package main  func main() {  n := 5  if n > 5 {  n += 1  } else {  n -= 1  }  } | {"tag":"blk","body":{"tag":"seq","stmts":[{"tag":"assmt","sym":[{"tag":"nam","sym":"n"}],"expr":[{"tag":"lit","val":0}]},{"tag":"cond","pred":{"tag":"binop","sym":">","frst":{"tag":"nam","sym":"n"},"scnd":{"tag":"lit","val":0}},"cons":{"tag":"seq","stmts":[{"tag":"assmt","sym":[{"tag":"nam","sym":"n"}],"expr":[{"tag":"binop","sym":"+","frst":{"tag":"nam","sym":"n"},"scnd":{"tag":"lit","val":1}}]},{"tag":"assmt","sym":[{"tag":"nam","sym":"n"}],"expr":{"tag":"binop","sym":"+","frst":[{"tag":"nam","sym":"n"}],"scnd":[{"tag":"lit","val":1}]}}]},"alt":{"tag":"seq","stmts":[{"tag":"assmt","sym":[{"tag":"nam","sym":"n"}],"expr":{"tag":"binop","sym":"-","frst":[{"tag":"nam","sym":"n"}],"scnd":[{"tag":"lit","val":1}]}}]}}]}}  DONE |
| Nested if else | package main  func main() {  n := 10  if n > 0 {  n+=3  if n < 15 {  n += 1  } else {  n -= 1  }  } else {  n -= 1  }  } | {"tag":"blk","body":{"tag":"seq","stmts":[{"tag":"assmt","sym":[{"tag":"nam","sym":"n"}],"expr":[{"tag":"lit","val":10}]},{"tag":"cond","pred":{"tag":"binop","sym":">","frst":{"tag":"nam","sym":"n"},"scnd":{"tag":"lit","val":0}},"cons":{"tag":"seq","stmts":[{"tag":"cond","pred":{"tag":"binop","sym":"<","frst":{"tag":"nam","sym":"n"},"scnd":{"tag":"lit","val":15}},"cons":{"tag":"seq","stmts":[{"tag":"assmt","sym":[{"tag":"nam","sym":"n"}],"expr":[{"tag":"lit","val":1}]}]},"alt":{"tag":"seq","stmts":[{"tag":"assmt","sym":[{"tag":"nam","sym":"n"}],"expr":[{"tag":"lit","val":1}]}]}}]},"alt":{"tag":"seq","stmts":[{"tag":"assmt","sym":[{"tag":"nam","sym":"n"}],"expr":[{"tag":"lit","val":1}]}]}}]}}  DONE |
| For loop  simplest | package main  func main() {  i := 0  for i < 5 {  i = i + 1  }  i  } | {"tag":"blk","body":{"tag":"seq","stmts":[{"tag":"assmt","sym":[{"tag":"nam","sym":"i"}],"expr":[{"tag":"lit","val":0}]},{"tag":"for","pred":{"tag":"binop","sym":"<","frst":{"tag":"nam","sym":"i"},"scnd":{"tag":"lit","val":5}},"body":{"tag":"blk","body":{"tag":"seq","stmts":[{"tag":"assmt","sym":[{"tag":"nam","sym":"i"}],"expr":[{"tag":"binop","sym":"+","frst":{"tag":"nam","sym":"i"},"scnd":{"tag":"lit","val":1}}]}]}}},{"tag":"nam","sym":"i"}]}}  DONE |
| nested | package main  func main() {  i := 0  for i < 5 {  j:=0  for j<3{  j+=1  }  i = i + j  }  i  } | {"tag":"blk","body":{"tag":"seq","stmts":[{"tag":"assmt","sym":[{"tag":"nam","sym":"i"}],"expr":[{"tag":"lit","val":0}]},{"tag":"for","pred":{"tag":"binop","sym":"<","frst":{"tag":"nam","sym":"i"},"scnd":{"tag":"lit","val":5}},"body":{"tag":"blk","body":{"tag":"seq","stmts":[{"tag":"assmt","sym":[{"tag":"nam","sym":"j"}],"expr":[{"tag":"lit","val":0}]},{"tag":"for","pred":{"tag":"binop","sym":"<","frst":{"tag":"nam","sym":"j"},"scnd":{"tag":"lit","val":3}},"body":{"tag":"blk","body":{"tag":"seq","stmts":[{"tag":"assmt","sym":[{"tag":"nam","sym":"j"}],"expr":[{"tag":"binop","sym":"+","frst":{"tag":"nam","sym":"j"},"scnd":{"tag":"lit","val":1}}]}]}}},{"tag":"assmt","sym":[{"tag":"nam","sym":"i"}],"expr":[{"tag":"binop","sym":"+","frst":{"tag":"nam","sym":"i"},"scnd":{"tag":"nam","sym":"j"}}]}]}}},{"tag":"nam","sym":"i"}]}}  DONE |
|  | package main  func main() {  x := 0  i := 0  for i < 100 {  j := 0  for j < 100 {  x = x + i + j  j = j + 1  }  i = i + 1  }  x  } | {"tag":"blk","body":{"tag":"seq","stmts":[{"tag":"assmt","sym":[{"tag":"nam","sym":"x"}],"expr":[{"tag":"lit","val":0}]},{"tag":"assmt","sym":[{"tag":"nam","sym":"i"}],"expr":[{"tag":"lit","val":0}]},{"tag":"for","pred":{"tag":"binop","sym":"<","frst":{"tag":"nam","sym":"i"},"scnd":{"tag":"lit","val":100}},"body":{"tag":"blk","body":{"tag":"seq","stmts":[{"tag":"assmt","sym":[{"tag":"nam","sym":"j"}],"expr":[{"tag":"lit","val":0}]},{"tag":"for","pred":{"tag":"binop","sym":"<","frst":{"tag":"nam","sym":"j"},"scnd":{"tag":"lit","val":100}},"body":{"tag":"blk","body":{"tag":"seq","stmts":[{"tag":"assmt","sym":[{"tag":"nam","sym":"x"}],"expr":[{"tag":"binop","sym":"+","frst":{"tag":"binop","sym":"+","frst":{"tag":"nam","sym":"x"},"scnd":{"tag":"nam","sym":"i"}},"scnd":{"tag":"nam","sym":"j"}}]},{"tag":"assmt","sym":[{"tag":"nam","sym":"j"}],"expr":[{"tag":"binop","sym":"+","frst":{"tag":"nam","sym":"j"},"scnd":{"tag":"lit","val":1}}]}]}}},{"tag":"assmt","sym":[{"tag":"nam","sym":"i"}],"expr":[{"tag":"binop","sym":"+","frst":{"tag":"nam","sym":"i"},"scnd":{"tag":"lit","val":1}}]}]}}},{"tag":"nam","sym":"x"}]}}  DONE |
| Block | package main  func main() {  var x int = 10  {  var x int = 20  }  x  } | 10  DONE |
|  | package main  func main() {  var x int = 10  {  x:= 20  }  x  } | 10  DONE |
|  | package main  func main() {  var x int = 10  var y int =0  {  x,y = 20,15  }  y  } | 15  Done |
| function declare& call | package main  func fact(n int) int {      return factIter(n, 1, 1)  }  func factIter(n, i, acc int) int {      if i > n {          return acc      } else {          return factIter(n, i+1, acc\*i)      }  }  func main() {      result := fact(4)  } | Return 24  DONE |
| Simplest fun | package main  func add(a int, b int) int {  return a + b  }  func main() {  result := add(1, 2)  } | {"tag":"blk","body":{"tag":"seq","stmts":[{"tag":"fun","sym":"add","prms":["a","b"],"body":{"tag":"blk","body":{"tag":"seq","stmts":[{"tag":"ret","expr":{"tag":"binop","sym":"+","frst":{"tag":"nam","sym":"a"},"scnd":{"tag":"nam","sym":"b"}}}]}}},{"tag":"decl","sym":[{"tag":"nam","sym":"result"}],"expr":[{"tag":"app","fun":{"tag":"nam","sym":"add"},"args":[{"tag":"lit","val":1},{"tag":"lit","val":2}]}]}]}}  Return 3  DONE |
| Go Routine |  |  |

Standard Project 2: Concurrent virtual machine for Go

The Concurrent virtual machine project applies the notion of a virtual machine (Module 4) to a concurrent programming language. Baseline expectations:

* Web-based implementation based on Source Academy frontend and js-slang (see [GitHubLinks to an external site.](https://github.com/source-academy))
* Implementation of a sublanguage of Go, consistent with a recent language specification
* Sequential language constructs: variable and function declarations, blocks, conditionals statements and expressions, while loops
* Concurrent constructs: Go routines, concurrency control (wait, wait group etc)
* Implementation should use a virtual machine
* Implementation must use a low-level memory model, and all runtime data structures need to be allocated from a single ArrayBuffer.
* Optional components:
  + visualization of heap and runtime stack
  + type checking
  + memory management

|  |  |  |
| --- | --- | --- |
| Switch ??? | package main  func main() {  day := 3  switch day {  case 1:  day=day+9  case 2:  day=day+8  case 3:  day=day+7  default:  day=0  }  } |  |

package main

import (

"fmt"

"time"

)

func say(s string) {

i:=0

for i < 5{

time.Sleep(100)

fmt.Println(s)

i=i+1

}

}

func main() {

go say("world")

say("hello")

}

Channel

package main

import (

"fmt"

)

func worker(id int, jobs <-chan int, results chan<- int) {

j := 0

for j < 2 {

fmt.Println("worker", id, "started job", j)

results <- j \* 2

j += 1

}

}

func main() {

const numJobs = 10

jobs := make(chan int, numJobs)

results := make(chan int, numJobs)

// Start 3 workers.

w := 0

for w < 3 {

go worker(w, jobs, results)

w += 1

}

// Send 5 jobs and then close the channel.

i := 0

for i < 5 {

jobs <- i

i += 1

}

close(jobs)

// Finally, collect all the results of the work.

a := 0

for a < 5 {

<-results

a = a + 1

}

}

WAIT Group

package main

import (

"fmt"

"sync"

)

func worker(id int, wg \*sync.WaitGroup) {

defer wg.Done() // 在函数退出时调用Done来减少计数器

fmt.Println("Worker",id,"starting")

fmt.Println("Worker",id,"done")

}

func main() {

var wg sync.WaitGroup // 创建WaitGroup变量

i := 0

for i < 5 {

wg.Add(1) // 增加计数器

go worker(i, &wg) // 启动goroutine

i += 1

}

wg.Wait() // 等待所有goroutine完成

fmt.Println("All workers done")

}

{"tag":"blk","body":{"tag":"seq","stmts":[{"tag":"fun","sym":"worker","prms":["id","wg"],"body":{"tag":"blk","body":{"tag":"seq","stmts":[{"tag":"deferStmt","sym":"wg"},{"tag":"display","content":[{"tag":"lit","val":"Worker"},{"tag":"nam","sym":"id"},{"tag":"lit","val":"starting"}]},{"tag":"display","content":[{"tag":"lit","val":"Worker"},{"tag":"nam","sym":"id"},{"tag":"lit","val":"done"}]}]}}},{"tag":"waitgroupdecl","sym":"wg"},{"tag":"decl","sym":[{"tag":"nam","sym":"i"}],"expr":[{"tag":"lit","val":0}]},{"tag":"for","pred":{"tag":"binop","sym":"<","frst":{"tag":"nam","sym":"i"},"scnd":{"tag":"lit","val":5}},"body":{"tag":"blk","body":{"tag":"seq","stmts":[{"tag":"addwait","val":{"tag":"lit","val":1}},{"tag":"gostmt","callbody":{"tag":"blk","body":{"tag":"seq","stmts":[{"tag":"app","fun":{"tag":"nam","sym":"worker"},"args":[{"tag":"nam","sym":"i"},{"tag":"nam","sym":"wg"}]}]}}},{"tag":"assmt","sym":[{"tag":"nam","sym":"i"}],"expr":[{"tag":"binop","sym":"+","frst":{"tag":"nam","sym":"i"},"scnd":{"tag":"lit","val":1}}]}]}}},{"tag":"wait"},{"tag":"display","content":[{"tag":"lit","val":"All workers done"}]}]}}