Value
{
  Operand (for ASM)
  name
  type 类型: array / bool / func / int / Lable / P* / struct / void
  use list 被谁 use 了
  (HM) renamingMachine static
}

User : operands (use了哪些 value)

instruction : 属于哪个 Basicblock
{
  alloc (name, type, block) eg. %name.alloc = Alloc type*
  binary (op, value1, value2, block) eg. %op = op type val1, type val2
  bitcast (value, type, block) e.g. %name-BC = bitcast rty val to type
  branch (block, dest-block) e.g. br lable %lable-name
          (block, codi, Lblock, Rblock) e.g. br ty codi, lable %l , lable, %r
  call (func, block) e.g. %l = call (ty val1, ...)
  getelementptr (type, typ*, blocks) e.g. %... = getelementptr inbounds typ*, type %--, i32 1
  global (name, type) eg @name-glo = global type*
  icmp (op, value1, value2, block) e.g icmp op type val1, type val2.
  Load (name, addr, block) e.g. %name = load type, type* %@...
  Ret (retval, block) e.g. ret void / ret type %...
  Store (value, addr, block) e.g. store type val , type* addr
  (Trunc)
  (Zext)
}

Constant
{
  bool (val)
  int (val)
  Null
  String (String)
}

BasicBlock
{
  包含若干 instructions
  最后一条跳转指令.
  属于哪个 function
}

Function
{
  包含若干 Basicblock
  返回的地址 (一个 value)
  定义 isBuiltin
  声明 is Used
}

Module
{
  包含若干 function
  String 常量 (int / bool 不用存)
  变量定义
  全局附初值(一个操作对应一个 func)
  包含若干 Class
}