**การออกแบบภาษา Hunnaeee**

|  |  |  |
| --- | --- | --- |
| **Token** | **Regular Expression** | **ตัวอย่าง** |
| **ค่าคงที่** | | |
| จำนวนเต็มฐาน 10 | [-]?[0-9]+ | -1 , -500 , 0 , 10 , 1000 |
| จำนวนเต็มฐาน 16 | 0[xX][0-9a-fA-F]+ | 0x0000 , 0xAAA , 0XE4 |
| **ตัวแปรพื้นฐาน** | | |
| จำนวนเต็มคิดเครื่องหมายขนาด 64 บิต | [a-z]{2} | ab , bz , er |
| **นิพจน์คำนวณจำนวนเต็ม** | | |
| ติดลบ | “-” | -aa , -5 , -0 |
| คูณ | “\*” | aa \* bb , 0 \* 5 , 10 \* qq |
| หารเอาส่วน | “/” | aa / bb , 11 / 5 , 10 / qq |
| หารเอาเศษ | “%” | aa % bb , 11 % 5 , 10 % qq |
| บวก | “+” | aa + bb , 11 + 5 , 10 + qq |
| ลบ | “-” | aa - bb , 11 - 5 , 10 - qq |
| วงเล็บ | “(” , “)” | (5\*8)+(3-1) |
| **ประโยคคำสั่งที่ทำตามลำดับ** | | |
| ให้ค่าแก่ตัวแปร (assignment) | “=” | aa = 5 |
| แสดงค่าตัวแปรโดดแบบฐาน 10 | “print” | print 5 |
| แสดงค่าตัวแปรโดดแบบฐาน 16 | “print” | print hex(78) |
| แสดงสายอักษรที่กำหนด (ยาวไม่เกิน 255 ตัว) | “print” | print “…” |
| **ประโยคคาสั่งตัดสินใจ** | | |
| เท่ากัน | “==” | 0 == 0 |
| มากกว่าหรือเท่ากับ | “>=” | 5 >= 5 , 5 >= 0 |
| น้อยกว่าหรือเท่ากับ | “<=” | 1 <= 10 , 10 <= 10 |
| มากกว่า | “>” | 5 > 0 |
| น้อยกว่า | “<” | 1 < 10 |
| ประโยคเงื่อนไข แบบที่ 1 | “if” | if pm == 0 {  pr[id] = fm  id = id + 1  } |
| ประโยคเงื่อนไข แบบที่ 2 | “else” | else fm % st == 0 {  pm = 1  } |
| **ประโยคคาสั่งวนซ้ำ** | | |
| For loop | “for” | for fm : to {  pm = 0; st = 2  } |