Студент: Ивенкова. Л.В.
Группа: М8О-208Б-19
Номер по списку: 11

Лабораторная работа N8

Тема: Диалекты языка МИКРОЛИСП.

Вариант грамматики: j11
Распечатка файла GrammaFeatures.rtf
>
Особенности диалекта j11:

1. Форма or.
    1.1 Один и более операндов.
2. Форма and.
    2.2 Отсутствует.
3. Форма not.
    3.1 Есть.
4. Оператор сравнения.
    4.1 Есть.
5. Оператор отношения.
    5.2 (<= x y)

6. Легкий let.
    6.2 Отсутствует.
7. Числовой if.
    7.2 Отсутствует.
8. Строковый if.
    8.1 Есть.

9. Строковый COND.
    9.2 Отсутствует.

10. Числовой COND.
    10.1 Составная ветвь ELSE и несколько составных клауз.

Скриншот запуска Pars08 для zeller.ss
>

```
parsifal@DESKTOP-3G70RV4:~/SP/Lab8$ ./Pars08
Input gramma name>j11
Gramma:j11.txt
Source>zeller
Source:zeller.ss
   1|;zeller.ss
   2|(define (day-of-week)
   3| (zeller dd
   4|      (cond ((not(or (not(<= mm 3)) (= mm 3)))(+ mm 10))
   5|            (else (- mm 2)))
   6|      (remainder (cond((not(or (not(<= mm 3)) (= mm 3)))(- yyyy 1))
   7|                      (else yyyy))
   8|               100)
   9|      (quotient (cond((not(or (not(<= mm 3)) (= mm 3)))(- yyyy 1))
  10|                     (else yyyy))
  11|              100)
  12| )
  13|)
  14|(define (zeller d m y c)
  15| (neg-to-pos (remainder (+ d y
  16|                           (quotient (-(* 26 m)2) 10)
  17|                           (quotient y 4)
  18|                           (quotient c 4)
  19|                           (* 2(- c))
  20|                       )
  21|          7)
  22| )
  23|)
  24|(define (neg-to-pos d)
  25| (cond((not(or (not(<= d 0)) (= d 0)))(+ d 7))
  26|      (else d)
  27| )
  28|)
  29|(define (birthday dw)
  30|;                  ^{0,...,6}
  31| (display "Your were born on ")
  32| (display
  33|     (if(= dw 1)"Monday "
  34|     (if(= dw 2)"Tuesday "
  35|     (if(= dw 3)"Wednesday "
  36|     (if(= dw 4)"Thursday "
  37|     (if(= dw 5)"Friday "
  38|     (if(= dw 6)"Saturday "
  39|     "Sunday " )))))))
  40|  (display dd)(display ".")
  41|  (display mm)(display ".")
  42| yyyy
  43|)
  44|(define dd 4)
  45|(define mm 6)
  46|(define yyyy 2001)
  47|(birthday (day-of-week))
  48|
-------------------
Good source!
```

Скриншот запуска Pars08 для golden21.ss
>

```
parsifal@DESKTOP-3G70RV4:~/SP/Lab8$ ./Pars08
Input gramma name>j11
Gramma:j11.txt
Source>golden21
Source:golden21.ss
   1|;golden21
   2|(define a 2)(define b 3)
   3|(define (fun x)
   4| (set! x (- x (/ 11 12)))
   5| (- x (expt(- x 2)3)(atan x) 1)
   6|)
   7|(define (golden-section-search a b)
   8| (let(
   9|       (xmin(cond((not(or(not(<= a b)) (= a b)))(golden-start a b))
  10|                 (else(golden-start b a ))))
  11|     )
  12|      (newline)
  13|      xmin
  14| )
  15|)
  16|(define (golden-start a b)
  17| (set! total-iterations 0)
  18| (let(
  19|       (xa (+ a (* mphi(- b a))))
  20|       (xb (+ b (-(* mphi(- b a)))))
  21|     )
  22|      (try a b xa (fun xa) xb (fun xb))
  23| )
  24|)
  25|(define mphi (* (- 3(sqrt 5))(/ 2.0e0)))
  26|(define (try a b xa ya xb yb)
  27| (cond((close-enough? a b)
  28|       (* (+ a b)0.5e0))
  29|       (else
  30|              (display "+")
  31|              (set! total-iterations (+ total-iterations 1))
  32|              (cond((not(or (not(<= ya yb)) (= ya yb)))(set! b xb)
  33|                       (set! xb xa)
  34|                       (set! yb ya)
  35|                       (set! xa (+ a (* mphi(- b a))))
  36|                       (try a b xa (fun xa) xb yb)
  37|                  )
  38|                  (else  (set! a xa)
  39|                       (set! xa xb)
  40|                       (set! ya yb)
  41|                       (set! xb (- b (* mphi(- b a))))
  42|                       (try a b xa ya xb (fun xb))
  43|                  )
  44|             );cond...
  45|       );else...
  46| );cond...
  47|)
  48|(define (close-enough? x y)
  49|  (not(or (not(<=(abs (- x y))tolerance)) (=(abs (- x y))tolerance))))
  50|(define tolerance 0.001e0)
  51|(define total-iterations 0)
  52|(define xmin 0)
```

```
53|(set! xmin(golden-section-search a b))
54|  (display"Interval=\t[")
55|  (display a)
56|  (display" , ")
57|  (display b)
58|  (display"]\n")
59|  (display"Total number of iteranions=")
60|total-iterations
61|  (display"xmin=\t\t")
62|xmin
63|  (display"f(xmin)=\t")
64|(fun xmin)
65|
_____
Good source!
```

Скриншот запуска Pars08 для coin21.ss
>

```
parsifal@DESKTOP-3G70RV4:~/SP/Lab8$ ./Pars08
Input gramma name>j11
Gramma:j11.txt
Source>coin21
Source:coin21.ss
   1|(define VARIANT 11)
   2|(define LAST-DIGIT-OF-GROUP-NUMBER 8)
   3|(define KINDS-OF-COINS 5)
   4|
   5|(define (first-denomination kinds-of-coins)
   6|    (cond
   7|        ((= kinds-of-coins 1) 1)
   8|        ((= kinds-of-coins 2) 2)
   9|        ((= kinds-of-coins 3) 3)
  10|        ((= kinds-of-coins 4) 10)
  11|        ((= kinds-of-coins 5) 15)
  12|        (else 0)
  13|)
  14|)
  15|
  16|(define (count-change amount)
  17|        (display"_____\n amount: ")
  18|        (display amount)
  19|        (newline)
  20|        (display"KINDS-OF-COINS: ")
  21|        (display KINDS-OF-COINS)
  22|        (newline)
  23|        (let((largest-coin (first-denomination KINDS-OF-COINS)))
  24|            (display"largest-coin: ")
  25|            (display largest-coin)
  26|            (newline)
  27|            (cond((not (or (<= amount 0)
  28|                    (<= KINDS-OF-COINS 0)
  29|                    (<= largest-coin 0)))
  30|                    (display"List of coin denominations: ")
  31|                    (denomination-list KINDS-OF-COINS)
  32|                    (display"count-change= ")
  33|                    (cc amount KINDS-OF-COINS))
  34|                (else
  35|                    (display"Improper parameter value!\ncount-change= ")-1)
  36|            )
  37|        )
  38|)
  39|
  40|(define (pier? x? y?) (not(or x? y?)))
  41|
  42|(define (cc amount kinds-of-coins)
```

```scheme
43|    (cond((= amount 0) 1)
44|         ((pier?(not(or (not(<= amount 0)) (= amount 0))) (= kinds-of-coins 0))
45|                (+ (cc amount (- kinds-of-coins 1))
46|                   (cc (- amount (first-denomination kinds-of-coins)) kinds-of-coins)))
47|         (else 0)
48|)
49|)
50|
51|(define (denomination-list  kinds-of-coins)
52|        (cond((= kinds-of-coins 0) (newline) 0)
53|              (else (display(first-denomination kinds-of-coins))
54|                    (display" ")
55|                    (denomination-list (- kinds-of-coins 1)))
56|        )
57|)
58|
59|
60|(define GR-AMOUNT
61|   (remainder (+ (* 100 LAST-DIGIT-OF-GROUP-NUMBER) VARIANT) 231))
62|
63|(display"Variant ")
64|(display VARIANT)
65|(newline)
66|(newline)
67|(display (count-change 100)) (newline)
68|(display (count-change GR-AMOUNT)) (newline)
69|(set! KINDS-OF-COINS 13)
70|(display (count-change 100)) (newline)
71|(display"(c) Ivenkova L.V. 2021\n")
72|
```

Good source!

Source>