**Студент: Епанешников В. С.**
**Группа: М8О-206Б-19**
**Номер по списку: 9**

**Лабораторная работа N8**

**Тема: Диалекты языка МИКРОЛИСП..**


**Вариант грамматики: n09**
**Распечатка файла GrammaFeatures.rtf**
**>**
**Особенности диалекта n09.**

**1. Форма or.**
   **1.2 Отсутствует.**
**2. Форма and.**
   **2.2 Отсутствует.**
**3. Форма not.**
   **3.1 Есть.**
   **(not x? )**
**4. Оператор сравнения.**
   **4.1 Есть.**
   **(=  x y)**
**5. Оператор отношения.**
   **5.1 (<  x y)**
**6. Легкий let.**
   **6.1 Есть.**
   **(let() e pi)**
**7. Числовой if.**
   **7.2 Отсутствует.**
**8. Строковый if.**
   **8.2 Отсутствует.**
**9. Строковый COND.**
   **9.2 Отсутствует.**
**10. Числовой COND.**
   **10.9 Простая ветвь ELSE, одна простая клауза.**
   **(cond(x? 1)(else 2))**

**Сриншот запуска Pars08 для zeller.ss**

\>

```
Source>zeller
Source:zeller.ss
   1|;zeller.ss
   2|(define (day-of-week)
   3| (zeller dd
   4|       (cond((< mm 3)(+ mm 10))(else (- mm 2)))
   5|       (remainder (cond((< mm 3)(- yyyy 1)) (else yyyy)) 100)
   6|       (quotient (cond((< mm 3)(- yyyy 1)) (else yyyy)) 100)
   7| )
   8|)
   9|(define (zeller d m y c)
  10| (neg-to-pos (remainder (+ d y
  11|                           (quotient (-(* 26 m)2) 10)
  12|                           (quotient y 4)
  13|                           (quotient c 4)
  14|                           (* 2(- c))
  15|                        )
  16|             7)
  17| )
  18|)
  19|(define (neg-to-pos d)
  20| (cond((< d 0)(+ d 7))
  21|       (else d)
  22| )
  23|)
  24|
  25|
  26|
  27|(define (birthday dw)
  28|;                  ^{0,...,6}
  29| (display "Your were born on ")
  30|   (display
  31|   (cond((= dw 1)"Monday ")
  32|        ((= dw 2)"Tuesday ")
  33|        ((= dw 3)"Wednesday ")
  34|        ((= dw 4)"Thursday ")
  35|        ((= dw 5)"Friday ")
  36|        ((= dw 6)"Saturday ")
  37|        (else "Sunday ") ))
  38|  (display dd)(display ".")
  39|  (display mm)(display ".")
  40| yyyy
  41|)
  42|(define dd 31)
  43|(define mm 12)
  44|(define yyyy 2021)
  45|(birthday (day-of-week))
  46|

Good source!
```

## Сриншот запуска Pars08 для golden21.ss

>

```
Source>golden21
Source:golden21.ss
    1|; golden21
    2|; Епанешников М80-206Б-19
    3|; [5, 7] 5,712
    4|; e^(-z) +sin(z)
    5|(define a 5)(define b 7)
    6|(define (fun x)
    7|  (set! x (- x (/ 109 110)))
    8|  (+(exp(- x))(sin x))
    9|)
   10|(define (golden-section-search a bz)
   11| (let(
   12|      (xmin(cond((< a b)(golden-start a b)) (else (golden-start b a ))))
   13|      )
   14|      (newline)
   15|      xmin
   16| )
   17|)
   18|(define (golden-start a b)
   19| (set! total-iterations 0)
   20| (let(
   21|      (xa (+ a (* mphi(- b a))))
   22|      (xb (+ b (-(* mphi(- b a)))))
   23|      )
   24|      (try a b xa (fun xa) xb (fun xb))
   25| )
   26|)
   27|(define mphi (* (- 3(sqrt 5))(/ 2.0e0)))
   28|(define (try a b xa ya xb yb)
   29| (cond((close-enough? a b)
   30|      (* (+ a b)0.5e0))
   31|      (else (let() (display "+")
   32|             (set! total-iterations (+ total-iterations 1))
   33|             (cond((< ya yb)(let() (set! b xb)
   34|                         (set! xb xa)
   35|                         (set! yb ya)
   36|                         (set! xa (+ a (* mphi(- b a))))
   37|                         (try a b xa (fun xa) xb yb))
   38|                      )
   39|                  (else  (let() (set! a xa)
   40|                         (set! xa xb)
   41|                         (set! ya yb)
   42|                         (set! xb (- b (* mphi(- b a))))
   43|                         (try a b xa ya xb (fun xb)))
   44|                      )
   45|             );cond...
   46|      );let...
   47| ));cond...
   48|)
   49|(define (close-enough? x y)
   50|  (<(abs (- x y))tolerance))
   51|(define tolerance 0.001e0)
   52|(define total-iterations 0)
   53|(define xmin 0)
   54|(set! xmin(golden-section-search a b))
```

```
55|   (display"Interval=\t[")
56|   (display a)
57|   (display" , ")
58|   (display b)
59|   (display"]\n")
60|   (display"Total number of iterations=")
61|total-iterations
62|   (display"xmin=\t\t")
63|xmin
64|   (display"f(xmin)=\t")
65|(fun xmin)
66|
_____
Good source!

_____
Source>
```

**Сриншот запуска Pars08 для coin21.ss**

>

```
Source:coin21.ss
 1|; coin21
 2|; Епанешников М80-206Б-19
 3|
 4|(define VARIANT 9)
 5|(define LAST-DIGIT-OF-GROUP-NUMBER 6)
 6|(define KINDS-OF-COINS 7)
 7|
 8|(define (first-denomination kinds-of-coins)
 9|   (cond((= kinds-of-coins 1) 1)
10|       (else (cond((= kinds-of-coins 2) 2)
11|       (else (cond((= kinds-of-coins 3) 3)
12|       (else (cond((= kinds-of-coins 4) 5)
13|       (else (cond((= kinds-of-coins 5) 10)
14|       (else (cond((= kinds-of-coins 6) 15)
15|       (else (cond((= kinds-of-coins 7) 20)
16|       (else 0))))))))))))))))
17|)
18|
19|
20|(define (AND3? x? y? z?)
21|    ( = 1 (cond(x? (cond(y? (cond(z? 1) (else 0))) (else 0))) (else 0)))
22|)
23|
24|(define (AND2? x? y?)
25|   ( = 1 (cond(x? (cond(y? 1) (else 0))) (else 0)))
26|)
27|
28|(define (MORE? x y)
29|    (AND2? (not (< x y)) (not (= x y)))
30|)
31|
32|
33|(define (count-change amount)
34|   (display "_____")
35|   (newline)
36|   (display " amount: ")
37|   (newline)
38|   (display "KINDS-OF-COINS: ")
39|   (display KINDS-OF-COINS)
40|   (newline)
41|   (let(
42|        (largest-coin (first-denomination KINDS-OF-COINS))
43|        )
44|       (display "largest-coin: ")
45|       (display largest-coin)
46|       (newline)
47|       (cond((AND3? (MORE? amount 0) (MORE? KINDS-OF-COINS 0) (MORE? largest-coin 0))
48|          (let()
49|             (display "List of coin denominations: ")
50|             (denomination-list KINDS-OF-COINS)
51|             (display "count-change= ")
52|             (cc amount KINDS-OF-COINS)
```

```scheme
53|            ))
54|          (else (let()
55|            (display "Improrer parametr value!")
56|            (newline)
57|            (display "count-change =") -1))
58|      )
59|   )
60|)
61|
62|(define (pier? x? y?)
63|   (not (OR? x? y?))
64|)
65|
66|
67|(define (OR? x? y?)
68|     (not(AND2? (not x?) (not y?)))
69|)
70|
71|
72|(define (cc amount kinds-of-coins)
73|   (cond( (= amount 0) 1)
74|       (else (cond((pier? (< amount 0) (= kinds-of-coins 0))
75|          (+ (cc amount (- kinds-of-coins 1))
76|             (cc (- amount (first-denomination kinds-of-coins)) kinds-of-coins)))
77|          (else 0))))
78|)
79|
80|(define (denomination-list kinds-of-coins)
81|   (cond((= kinds-of-coins 0) (let() (newline) 0))
82|      (else (let()
83|        (display (first-denomination kinds-of-coins))
84|        (display " ")
85|        (denomination-list (- kinds-of-coins 1))
86|      )))
87|)
88|
89|
90|(define (GR-AMOUNT)
91|   (remainder (+ (* 100 LAST-DIGIT-OF-GROUP-NUMBER) VARIANT) 231)
92|)
93|
94|(display "Variant ")
95|(display VARIANT)
96|(newline)
97|(newline)
98|(display (count-change 100))
99|(newline)
100|(display (count-change (GR-AMOUNT)))
101|(newline)
102|(set! KINDS-OF-COINS 13)
103|(display (count-change 100))
104|(newline)
105|(display "(c) Epaneshnikov V.S. 2021")
106|(newline)
107|
```

Good source!

Source>