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

«СИСТЕМЫ ПРОГРАММИРОВАНИЯ»
Курсовая работа 2021.
Часть 1.

Для заданного в Лабораторной №8 диалекта языка МИКРОЛИСП разработать синтаксически управляемый транслятор на язык C++, применяя методику из Лабораторной №9, Правила TranslationRules21.rtf и TextLayout.txt .

Работоспособность транслятора проверить на трёх контрольных задачах из Лабораторной №8.

Перечень документов в отчете.
Вариант грамматики: j11

ДОРОЛНИТЕЛЬНОЕ ЗАДАНИЕ:
После оператора присваивания поместите комментарий с именем прототипа переменной в Микролиспе, если имена различаются.
Например, f__a_E = 5./*f-a!*/
Продукции могут получать имена ТОЛЬКО из атрибута name и НЕ могут извлекать их из кода на C++.
Продемонстрируйте на контрольной задаче golden21.

- **скриншот трансляции**

```
parsifal@DESKTOP-3G70RV4:~/SP/curs1$ g++ Mlispgen.cpp -o Mlispgen
parsifal@DESKTOP-3G70RV4:~/SP/curs1$ ./Mlispgen
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|
```

_____

Code:

```
/*  ILV   */
#include "mlisp.h"
extern double a/*2*/ ;
        extern double b/*2*/ ;
        double fun/*3*/ (double x);
        double golden__section__search/*7*/ (double a, double b);
        double golden__start/*16*/ (double a, double b);
        extern double mphi/*25*/ ;
        double __ILV__try/*26*/ (double a, double b
         , double xa, double ya
         , double xb, double yb);
        bool close__enough_Q/*48*/ (double x, double y);
         extern double tolerance/*50*/ ;
        extern double total__iterations/*51*/ ;
        extern double xmin/*52*/ ;
        //________________
double a/*2*/ = 2.;
```

```
double b/*2*/ = 3.;

double fun/*3*/ (double x){
 x = (x - (11. / 12.));
        return
(x - expt((x - 2.)
         , 3.)
          - atan(x) - 1.);
         }

double golden__section__search/*7*/ (double a, double b){
 {
double xmin((((!( (!( a <= b )) || ( a == b ) ))
        ? (golden__start(a
         , b)
         )
        : (golden__start(b
         , a)
         )));
        newline();
        return
xmin;
        }
}

double golden__start/*16*/ (double a, double b){
 total__iterations = 0./*total-iterations*/;
        {
double xa((a + (mphi * (b - a)))),
        xb((b + (- (mphi * (b - a)))));
        return
__ILV__try(a
         , b
         , xa
         , fun(xa)
         , xb
         , fun(xb))
         ;
        }
}
```

```
double mphi/*25*/ = ((3. - sqrt(5.)) * (1. / 2.0e0));

double __ILV__try/*26*/ (double a, double b
         , double xa, double ya
         , double xb, double yb){
 return
(close__enough_Q(a, b)
        ? (((a + b) * 0.5e0))
        : (display("+"),
        total__iterations = (total__iterations + 1.)/*total-iterations*/,
        ((!( (!( ya <= yb )) || ( ya == yb ) ))
        ? (b = xb,
        xb = xa,
        yb = ya,
        xa = (a + (mphi * (b - a))),
        __ILV__try(a
         , b
         , xa
         , fun(xa)
         , xb
         , yb)
         )
        : (a = xa,
        xa = xb,
        ya = yb,
        xb = (b - (mphi * (b - a))),
        __ILV__try(a
         , b
         , xa
         , ya
         , xb
         , fun(xb))
         ))));
         }

bool close__enough_Q/*48*/ (double x, double y){
 return
 (!( (!( abs((x - y)) <= tolerance )) || ( abs((x - y)) == tolerance ) ));
         }

double tolerance/*50*/ = 0.001e0;
```

```
double total__iterations/*51*/ = 0.;

double xmin/*52*/ = 0.;
        int main(){
 xmin = golden__section__search(a
         , b)
         ;
        display("Interval=\t[");
        display(a);
        display(" , ");
        display(b);
        display("]\n");
        display("Total number of iteranions=")
        display(total__iterations);
         newline();
         display("xmin=\t\t");
        display(xmin);
         newline();
         display("f(xmin)=\t");
        display(fun(xmin));
         newline();
         std::cin.get();
         return 0;
         }

Code is saved to file golden21.cpp !
```

- **Скриншот запуска в C++**

```
parsifal@DESKTOP-3G70RV4:~/SP/curs1$ g++ golden21.cpp -o golden21
parsifal@DESKTOP-3G70RV4:~/SP/curs1$ ./golden21
++++++++++++++
Interval=       [2 , 3]
Total number of iteranions=15
xmin=           2.434675016371661
f(xmin)=        -0.3583063254111947
```

- **Распечатка изменённых продукций:**

```
int tCG::p49(){ //    SET -> HSET E )
    if (S1->name == decor(S1->name))
        S1->obj += S2->obj;
```

```
    else S1->obj += S2->obj + "/*" + S1->name + "*/";
    return 0;
}

int tCG::p50(){ //    HSET -> ( set! $id
    S1->obj = decor(S3->name) + " = ";
    S1->name = S3->name;
    return 0;
}
```

Контрольная задача №1 – zeller.

Полный скриншот трансляции без трассировки (крупный белый шрифт на ярком черном фоне).

```
parsifal@DESKTOP-3G70RV4:~/SP/curs1$ ./Mlispgen
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|
```

_____

Code:

```
/*  ILV   */
#include "mlisp.h"
double day__of__week/*2*/ ();
        double zeller/*14*/ (double d, double m
         , double y, double c);
        double neg__to__pos/*24*/ (double d);
        double birthday/*29*/ (double dw);
        extern double dd/*44*/ ;
        extern double mm/*45*/ ;
        extern double yyyy/*46*/ ;
        //_____________
double day__of__week/*2*/ (){
 return
zeller(dd
         , (((!( (!( mm <= 3. )) || ( mm == 3. ) ))
        ? ((mm + 10.))
        : ((mm - 2.)))
         , remainder(((!( (!( mm <= 3. )) || ( mm == 3. ) ))
        ? ((yyyy - 1.))
        : (yyyy))
         , 100.)

         , quotient(((!( (!( mm <= 3. )) || ( mm == 3. ) ))
        ? ((yyyy - 1.))
        : (yyyy))
         , 100.)
         )
         ;
```

```
         }

double zeller/*14*/ (double d, double m
         , double y, double c){
 return
neg__to__pos(remainder((d + y + quotient(((26. * m) - 2.)
         , 10.)
          + quotient(y
         , 4.)
          + quotient(c
         , 4.)
          + (2. * (- c)))
         , 7.)
         );
         }

double neg__to__pos/*24*/ (double d){
 return
((!( (!( d <= 0. )) || ( d == 0. ) ))
        ? ((d + 7.))
        : (d));
         }

double birthday/*29*/ (double dw){
 display("Your were born on ");
        display((( dw == 1. )
        ? "Monday "
        : (( dw == 2. )
        ? "Tuesday "
        : (( dw == 3. )
        ? "Wednesday "
        : (( dw == 4. )
        ? "Thursday "
        : (( dw == 5. )
        ? "Friday "
        : (( dw == 6. )
        ? "Saturday "
        : "Sunday ")))))));
        display(dd);
        display(".");
```

```
        display(mm);
        display(".");
        return
yyyy;
         }

double dd/*44*/ = 4.;

double mm/*45*/ = 6.;

double yyyy/*46*/ = 2001.;
        int main(){
 display(birthday(day__of__week()));
         newline();
         std::cin.get();
         return 0;
         }

Code is saved to file zeller.cpp !
```

Распечатка файла zeller.cpp .

```
>
/*  ILV   */
#include "mlisp.h"
double day__of__week/*2*/ ();
double zeller/*14*/ (double d, double m
      , double y, double c);
double neg__to__pos/*24*/ (double d);
double birthday/*29*/ (double dw);
extern double dd/*44*/ ;
extern double mm/*45*/ ;
extern double yyyy/*46*/ ;
//____________________
double day__of__week/*2*/ (){
 return
zeller(dd
      , ((!( (!( mm <= 3. )) || ( mm == 3. ) ))
      ? ((mm + 10.))
      : ((mm - 2.)))
       , remainder(((!( (!( mm <= 3. )) || ( mm == 3. ) ))
      ? ((yyyy - 1.))
      : (yyyy))
```

```
       , 100.)

       , quotient(((!( (!( mm <= 3. )) || ( mm == 3. ) ))
      ? ((yyyy - 1.))
      : (yyyy))
       , 100.)
       )
       ;
       }

double zeller/*14*/ (double d, double m
       , double y, double c){
 return
neg__to__pos(remainder((d + y + quotient(((26. * m) - 2.)
       , 10.)
        + quotient(y
       , 4.)
        + quotient(c
       , 4.)
        + (2. * (- c)))
       , 7.)
       );
       }

double neg__to__pos/*24*/ (double d){
 return
((!( (!( d <= 0. )) || ( d == 0. ) ))
      ? ((d + 7.))
      : (d));
       }

double birthday/*29*/ (double dw){
 display("Your were born on ");
      display((( dw == 1. )
      ? "Monday "
      : (( dw == 2. )
      ? "Tuesday "
      : (( dw == 3. )
      ? "Wednesday "
      : (( dw == 4. )
      ? "Thursday "
      : (( dw == 5. )
      ? "Friday "
      : (( dw == 6. )
```

```
      ? "Saturday "
      : "Sunday ")))))));
      display(dd);
      display(".");
      display(mm);
      display(".");
      return
yyyy;
       }

double dd/*44*/ = 4.;

double mm/*45*/ = 6.;

double yyyy/*46*/ = 2001.;
      int main(){
 display(birthday(day__of__week()));
       newline();
       std::cin.get();
       return 0;
       }
```

Скриншот запуска задачи на C++.
>

```
parsifal@DESKTOP-3G70RV4:~/SP/curs1$ g++ zeller.cpp -o zeller
parsifal@DESKTOP-3G70RV4:~/SP/curs1$ ./zeller
Your were born on Monday 4.6.2001
```

Контрольная задача №2 – golden21.

Полный скриншот трансляции без трассировки (крупный белый шрифт на ярком черном фоне).

>

```
parsifal@DESKTOP-3G70RV4:~/SP/curs1$ ./Mlispgen
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|
```

Code:

```
/*  ILV   */
#include "mlisp.h"
extern double a/*2*/ ;
        extern double b/*2*/ ;
        double fun/*3*/ (double x);
        double golden__section__search/*7*/ (double a, double b);
        double golden__start/*16*/ (double a, double b);
        extern double mphi/*25*/ ;
        double __ILV__try/*26*/ (double a, double b
         , double xa, double ya
         , double xb, double yb);
```

```
        bool close__enough_Q/*48*/ (double x, double y);
         extern double tolerance/*50*/ ;
        extern double total__iterations/*51*/ ;
        extern double xmin/*52*/ ;
        //_________________
double a/*2*/ = 2.;

double b/*2*/ = 3.;

double fun/*3*/ (double x){
 x = (x - (11. / 12.));
        return
(x - expt((x - 2.)
         , 3.)
          - atan(x) - 1.);
         }

double golden__section__search/*7*/ (double a, double b){
 {
double xmin((((!( (!( a <= b )) || ( a == b ) ))
        ? (golden__start(a
         , b)
         )
        : (golden__start(b
         , a)
         )));
        newline();
        return
xmin;
        }
}

double golden__start/*16*/ (double a, double b){
 total__iterations = 0.;
        {
double xa((a + (mphi * (b - a)))),
        xb((b + (- (mphi * (b - a)))));
        return
__ILV__try(a
         , b
         , xa
         , fun(xa)
```

```
          , xb
          , fun(xb))
          ;
        }
}

double mphi/*25*/ = ((3. - sqrt(5.)) * (1. / 2.0e0));

double __ILV__try/*26*/ (double a, double b
          , double xa, double ya
          , double xb, double yb){
 return
(close__enough_Q(a, b)
        ? (((a + b) * 0.5e0))
        : (display("+"),
        total__iterations = (total__iterations + 1.),
        ((!( (!( ya <= yb )) || ( ya == yb ) ))
        ? (b = xb,
        xb = xa,
        yb = ya,
        xa = (a + (mphi * (b - a))),
        __ILV__try(a
          , b
          , xa
          , fun(xa)
          , xb
          , yb)
          )
        : (a = xa,
        xa = xb,
        ya = yb,
        xb = (b - (mphi * (b - a))),
        __ILV__try(a
          , b
          , xa
          , ya
          , xb
          , fun(xb))
          ))));
          }

bool close__enough_Q/*48*/ (double x, double y){
```

```
 return
 (!( (!( abs((x - y)) <= tolerance )) || ( abs((x - y)) == tolerance ) ));
         }

double tolerance/*50*/ = 0.001e0;

double total__iterations/*51*/ = 0.;

double xmin/*52*/ = 0.;
        int main(){
 xmin = golden__section__search(a
         , b)
         ;
        display("Interval=\t[");
        display(a);
        display(" , ");
        display(b);
        display("]\n");
        display("Total number of iteranions=");
        display(total__iterations);
         newline();
         display("xmin=\t\t");
        display(xmin);
         newline();
         display("f(xmin)=\t");
        display(fun(xmin));
         newline();
         std::cin.get();
         return 0;
         }

Code is saved to file golden21.cpp !
```

Распечатка файла golden21.cpp .

>

```
/*  ILV  */
#include "mlisp.h"
extern double a/*2*/ ;
      extern double b/*2*/ ;
      double fun/*3*/ (double x);
      double golden__section__search/*7*/ (double a, double
b);
      double golden__start/*16*/ (double a, double b);
      extern double mphi/*25*/ ;
      double __ILV__try/*26*/ (double a, double b
       , double xa, double ya
       , double xb, double yb);
```

```
      bool close__enough_Q/*48*/ (double x, double y);
       extern double tolerance/*50*/ ;
      extern double total__iterations/*51*/ ;
      extern double xmin/*52*/ ;
      //____________________
double a/*2*/ = 2.;

double b/*2*/ = 3.;

double fun/*3*/ (double x){
 x = (x - (11. / 12.));
      return
(x - expt((x - 2.)
       , 3.)
        - atan(x) - 1.);
       }

double golden__section__search/*7*/ (double a, double b){
 {
double xmin(((!( (!( a <= b )) || ( a == b ) ))
      ? (golden__start(a
       , b)
       )
      : (golden__start(b
       , a)
       )));
      newline();
      return
xmin;
      }
}

double golden__start/*16*/ (double a, double b){
 total__iterations = 0.;
      {
double xa((a + (mphi * (b - a)))),
      xb((b + (- (mphi * (b - a)))));
      return
__ILV__try(a
       , b
       , xa
       , fun(xa)
       , xb
       , fun(xb))
```

```
      ;
      }
}

double mphi/*25*/ = ((3. - sqrt(5.)) * (1. / 2.0e0));

double __ILV__try/*26*/ (double a, double b
       , double xa, double ya
       , double xb, double yb){
 return
(close__enough_Q(a, b)
      ? (((a + b) * 0.5e0))
      : (display("+"),
      total__iterations = (total__iterations + 1.),
      ((!( (!( ya <= yb )) || ( ya == yb ) ))
      ? (b = xb,
      xb = xa,
      yb = ya,
      xa = (a + (mphi * (b - a))),
      __ILV__try(a
       , b
       , xa
       , fun(xa)
       , xb
       , yb)
       )
      : (a = xa,
      xa = xb,
      ya = yb,
      xb = (b - (mphi * (b - a))),
      __ILV__try(a
       , b
       , xa
       , ya
       , xb
       , fun(xb))
       ))));
       }

bool close__enough_Q/*48*/ (double x, double y){
 return
 (!( (!( abs((x - y)) <= tolerance )) || ( abs((x - y)) ==
tolerance ) ));
       }
```

```
double tolerance/*50*/ = 0.001e0;

double total__iterations/*51*/ = 0.;

double xmin/*52*/ = 0.;
      int main(){
 xmin = golden__section__search(a
       , b)
       ;
      display("Interval=\t[");
      display(a);
      display(" , ");
      display(b);
      display("]\n");
      display("Total number of iteranions=");
      display(total__iterations);
       newline();
       display("xmin=\t\t");
      display(xmin);
       newline();
       display("f(xmin)=\t");
      display(fun(xmin));
       newline();
       std::cin.get();
       return 0;
       }
```

Скриншот запуска задачи на C++.
>

```
parsifal@DESKTOP-3G70RV4:~/SP/curs1$ g++ golden21.cpp -o golden21
parsifal@DESKTOP-3G70RV4:~/SP/curs1$ ./golden21
+++++++++++++++
Interval=       [2 , 3]
Total number of iteranions=15
xmin=           2.434675016371661
f(xmin)=        -0.3583063254111947
```

Контрольная задача №3 – coin21.

Полный скриншот трансляции без трассировки (крупный белый шрифт на ярком черном фоне).

>

```
parsifal@DESKTOP-3G70RV4:~/SP/curs1$ ./Mlispgen
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)
```

```
                    (display"count-change= ")
                    (cc amount KINDS-OF-COINS))
                 (else
                    (display"Improper parameter value!\ncount-change= ")-1)
            )
        )
)

(define (pier? x? y?) (not(or x? y?)))

(define (cc amount kinds-of-coins)
  (cond((= amount 0) 1)
       ((pier?(not(or (not(<= amount 0)) (= amount 0))) (= kinds-of-coins 0))
             (+ (cc amount (- kinds-of-coins 1))
             (cc (- amount (first-denomination kinds-of-coins)) kinds-of-coins)))
       (else 0)
)
)

(define (denomination-list  kinds-of-coins)
        (cond((= kinds-of-coins 0) (newline) 0)
             (else (display(first-denomination kinds-of-coins))
                   (display" ")
                   (denomination-list (- kinds-of-coins 1)))
        )
)


(define (GR-AMOUNT)
  (remainder (+ (* 100 LAST-DIGIT-OF-GROUP-NUMBER) VARIANT) 231))

(display"Variant ")
(display VARIANT)
(newline)
(newline)
(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|
```

_____

Code:

```
/*  ILV   */
#include "mlisp.h"
extern double VARIANT/*1*/ ;
        extern double LAST__DIGIT__OF__GROUP__NUMBER/*2*/ ;
        extern double KINDS__OF__COINS/*3*/ ;
        double first__denomination/*5*/ (double kinds__of__coins);
        double count__change/*16*/ (double amount);
        bool pier_Q/*40*/ (bool x_Q, bool y_Q);
         double cc/*42*/ (double amount, double kinds__of__coins);
        double denomination__list/*51*/ (double kinds__of__coins);
        double GR__AMOUNT/*60*/ ();
        //_____________
double VARIANT/*1*/ = 11.;

double LAST__DIGIT__OF__GROUP__NUMBER/*2*/ = 8.;

double KINDS__OF__COINS/*3*/ = 5.;

double first__denomination/*5*/ (double kinds__of__coins){
 return
(( kinds__of__coins == 1. )
        ? (1.)
        : ( kinds__of__coins == 2. )
        ? (2.)
        : ( kinds__of__coins == 3. )
        ? (3.)
        : ( kinds__of__coins == 4. )
        ? (10.)
        : ( kinds__of__coins == 5. )
```

```
        ? (15.)
        : (0.));
         }

double count__change/*16*/ (double amount){
 display("______\n amount: ");
        display(amount);
        newline();
        display("KINDS-OF-COINS: ");
        display(KINDS__OF__COINS);
        newline();
        {
double largest__coin(first__denomination(KINDS__OF__COINS));
        display("largest-coin: ");
        display(largest__coin);
        newline();
        return
((!( ( amount <= 0. ) || ( KINDS__OF__COINS <= 0. ) || ( largest__coin <= 0. ) ))
        ? (display("List of coin denominations: "),
        denomination__list(KINDS__OF__COINS),
        display("count-change= "),
        cc(amount
         , KINDS__OF__COINS)
         )
        : (display("Improper parameter value!\ncount-change= "),
        -1.));
        }
}

bool pier_Q/*40*/ (bool x_Q, bool y_Q){
 return
 (!( x_Q || y_Q ));
         }

double cc/*42*/ (double amount, double kinds__of__coins){
 return
```

```
(( amount == 0. )
        ? (1.)
        : pier_Q((!( (!( amount <= 0. )) || ( amount == 0. ) )), ( kinds__of__coins == 0. ))
        ? ((cc(amount
         , (kinds__of__coins - 1.))
          + cc((amount - first__denomination(kinds__of__coins))
         , kinds__of__coins)
         ))
        : (0.));
         }

double denomination__list/*51*/ (double kinds__of__coins){
 return
(( kinds__of__coins == 0. )
        ? (newline(),
        0.)
        : (display(first__denomination(kinds__of__coins)),
        display(" "),
        denomination__list((kinds__of__coins - 1.))));
         }

double GR__AMOUNT/*60*/ (){
 return
remainder(((100. * LAST__DIGIT__OF__GROUP__NUMBER) + VARIANT)
         , 231.)
         ;
         }
int main(){
 display("Variant ");
        display(VARIANT);
        newline();
        newline();
        display(count__change(100.));
        newline();
        display(count__change(GR__AMOUNT()));
        newline();
```

```
          KINDS__OF__COINS = 13.;
          display(count__change(100.));
          newline();
          display("(c) Ivenkova L.V. 2021\n");
          std::cin.get();
           return 0;
           }

  Code is saved to file coin21.cpp !
```

Распечатка файла coin21.cpp .
>
/* ILV */
#include "mlisp.h"
extern double VARIANT/*1*/ ;

```
      extern double
LAST__DIGIT__OF__GROUP__NUMBER/*2*/ ;
      extern double KINDS__OF__COINS/*3*/ ;
      double first__denomination/*5*/ (double
kinds__of__coins);
      double count__change/*16*/ (double amount);
      bool pier_Q/*40*/ (bool x_Q, bool y_Q);
       double cc/*42*/ (double amount, double
kinds__of__coins);
      double denomination__list/*51*/ (double
kinds__of__coins);
      double GR__AMOUNT/*60*/ ();
      //____________________
double VARIANT/*1*/ = 11.;

double LAST__DIGIT__OF__GROUP__NUMBER/*2*/ = 8.;

double KINDS__OF__COINS/*3*/ = 5.;

double first__denomination/*5*/ (double kinds__of__coins){
 return
(( kinds__of__coins == 1. )
      ? (1.)
      : ( kinds__of__coins == 2. )
      ? (2.)
      : ( kinds__of__coins == 3. )
      ? (3.)
      : ( kinds__of__coins == 4. )
      ? (10.)
      : ( kinds__of__coins == 5. )
      ? (15.)
      : (0.));
       }

double count__change/*16*/ (double amount){
 display("_______\n amount: ");
      display(amount);
      newline();
      display("KINDS-OF-COINS: ");
      display(KINDS__OF__COINS);
      newline();
      {
double
largest__coin(first__denomination(KINDS__OF__COINS));
```

```
      display("largest-coin: ");
      display(largest__coin);
      newline();
      return
((!( ( amount <= 0. ) || ( KINDS__OF__COINS <= 0. ) || (
largest__coin <= 0. ) ))
      ? (display("List of coin denominations: "),
      denomination__list(KINDS__OF__COINS),
      display("count-change= "),
      cc(amount
       , KINDS__OF__COINS)
       )
      : (display("Improper parameter value!\ncount-change=
"),
      -1.));
      }
}

bool pier_Q/*40*/ (bool x_Q, bool y_Q){
 return
 (!( x_Q || y_Q ));
       }

double cc/*42*/ (double amount, double kinds__of__coins){
 return
(( amount == 0. )
      ? (1.)
      : pier_Q((!( (!( amount <= 0. )) || ( amount == 0. ) )), (
kinds__of__coins == 0. ))
      ? ((cc(amount
       , (kinds__of__coins - 1.))
        + cc((amount - first__denomination(kinds__of__coins))
       , kinds__of__coins)
       ))
      : (0.));
       }

double denomination__list/*51*/ (double
kinds__of__coins){
 return
(( kinds__of__coins == 0. )
      ? (newline(),
      0.)
      : (display(first__denomination(kinds__of__coins)),
```

```
      display(" "),
      denomination__list((kinds__of__coins - 1.))));
       }

double GR__AMOUNT/*60*/ (){
 return
remainder(((100. * LAST__DIGIT__OF__GROUP__NUMBER) +
VARIANT)
       , 231.)
       ;
       }
int main(){
 display("Variant ");
      display(VARIANT);
      newline();
      newline();
      display(count__change(100.));
      newline();
      display(count__change(GR__AMOUNT()));
      newline();
      KINDS__OF__COINS = 13.;
      display(count__change(100.));
      newline();
      display("(c) Ivenkova L.V. 2021\n");
      std::cin.get();
       return 0;
       }
```

Скриншот запуска задачи на C++.

```
parsifal@DESKTOP-3G70RV4:~/SP/curs1$ g++ coin21.cpp -o coin21
parsifal@DESKTOP-3G70RV4:~/SP/curs1$ ./coin21
Variant 11


______
 amount: 100
KINDS-OF-COINS: 5
largest-coin: 15
List of coin denominations: 15 10 3 2 1
count-change= 8136
______
 amount: 118
KINDS-OF-COINS: 5
largest-coin: 15
List of coin denominations: 15 10 3 2 1
count-change= 14566
______
 amount: 100
KINDS-OF-COINS: 13
largest-coin: 0
Improper parameter value!
count-change= -1
(c) Ivenkova L.V. 2021
```

➢

Распечатка файла code-gen.cpp.
>

```
/* $j11 */
#include "code-gen.h"
using namespace std;
void tCG::init(){declarations.clear();
 Authentication = "ILV";
//              ^
// replace with your initials!!!
}

int tCG::p01(){ // S -> PROG
      string header ="/*  " + Authentication +"   */\n";
      header += "#include \"mlisp.h\"\n";
      header += declarations;
      header += "//____________________ \n";
      S1->obj = header + S1->obj;
      return 0;
}
```

```
int tCG::p02(){ //    PROG -> CALCS
      S1->obj = "int main(){\n " + S1->obj +
"std::cin.get();\n\t return 0;\n\t }\n";
      return 0;
}

int tCG::p03(){ //    PROG -> DEFS
      S1->obj += "int main(){\n "
   "display(\"No calculations!\");\n\t newline();\n\t "
   " std::cin.get();\n\t return 0;\n\t }\n";
      return 0;
}
int tCG::p04(){ //    PROG -> DEFS CALCS
      S1->obj += "int main(){\n " + S2->obj +
"std::cin.get();\n\t return 0;\n\t }\n";
      return 0;
}

int tCG::p05(){ //      E -> $id
      S1->obj = decor(S1->name);
      return 0;
}

int tCG::p06(){ //      E -> $int
      S1->obj = S1->name + ".";
      return 0;
}

int tCG::p07(){ //      E -> $dec
      S1->obj = S1->name;
      return 0;
}

int tCG::p08(){ //      E -> AREX
      return 0;
}

int tCG::p09(){ //      E -> COND
      return 0;
}

int tCG::p10(){ //      E -> CPROC
      return 0;
}
```

```
int tCG::p11(){ //    AREX -> HAREX E )
      if (S1->count == 0 && S1->name == "/")
       S1->obj = "(1. " + S1->obj + " " + S2->obj + ")";
   else S1->obj = "(" + S1->obj + " " + S2->obj + ")";
   return 0;
      return 0;
}

int tCG::p12(){ //   HAREX -> ( AROP
      S1->obj = S2->obj;
      S1->name = S2->name;
      return 0;
}

int tCG::p13(){ //   HAREX -> HAREX E
      if (S1->count == 0)
       S1->obj = S2->obj + " " + S1->name;
   else S1->obj = S1->obj + " " + S2->obj + " " + S1->name;
   ++(S1->count);
      return 0;
}

int tCG::p14(){ //    AROP -> +
      S1->obj = S1->name;
      return 0;
}

int tCG::p15(){ //    AROP -> -
      S1->obj = S1->name;
      return 0;
}

int tCG::p16(){ //    AROP -> *
      S1->obj = S1->name;
      return 0;
}

int tCG::p17(){ //    AROP -> /
      S1->obj = S1->name;
      return 0;
}

int tCG::p18(){ //   CPROC -> HCPROC )
```

```
      S1->obj += ")";
      if (S1->count >= 2) S1->obj += "\n\t ";
      return 0;
}

int tCG::p19(){ //  HCPROC -> ( $id
      S1->obj += decor(S2->name) + "(";
      return 0;
}

int tCG::p20(){ //  HCPROC -> HCPROC E
      if (S1->count) S1->obj += "\n\t , ";
   S1->obj += S2->obj;
   ++(S1->count);
      return 0;
}

int tCG::p21(){ //    COND -> ( cond BRANCHES )
      S1->obj = "(" + S3->obj + ")";
      return 0;
}

int tCG::p22(){ //BRANCHES -> ELSE
      return 0;
}

int tCG::p23(){ //BRANCHES -> CLAUS BRANCHES
      S1->obj += "\n\t: " + S2->obj;
      return 0;
}

int tCG::p24(){ //   CLAUS -> ( BOOL CLAUSB )
      S1->obj = S2->obj + "\n\t? (" + S3->obj + ")";
      return 0;
}

int tCG::p25(){ //  CLAUSB -> E
      return 0;
}

int tCG::p26(){ //  CLAUSB -> INTER CLAUSB
      S1->obj += ",\n\t" + S2->obj;
      return 0;
}
```

```
int tCG::p27(){ //    ELSE -> ( else ELSEB )
      S1->obj = "(" + S3->obj + ")";
      return 0;
}

int tCG::p28(){ //   ELSEB -> E
      return 0;
}

int tCG::p29(){ //   ELSEB -> INTER ELSEB
      S1->obj += ",\n\t" + S2->obj;
      return 0;
}

int tCG::p30(){ //    STR -> $str
      S1->obj = S1->name;
      return 0;
}

int tCG::p31(){ //    STR -> SIF
      return 0;
}

int tCG::p32(){ //    SIF -> ( if BOOL STR STR )
      S1->obj = "(" + S3->obj + "\n\t? " + S4->obj + "\n\t: "
+ S5->obj + ")";
      return 0;
}

int tCG::p33(){ //    BOOL -> $bool
      if(S1->name == "#t") S1->obj += "true";
      else S1->obj += "false";
      return 0;
}

int tCG::p34(){ //    BOOL -> $idq
      S1->obj = decor(S1->name);
      return 0;
}

int tCG::p35(){ //    BOOL -> REL
      return 0;
}
```

```
int tCG::p36(){ //    BOOL -> OR
      return 0;
}

int tCG::p37(){ //    BOOL -> ( not BOOL )
      S1->obj = "(!" + S3->obj + ")";
      return 0;
}

int tCG::p38(){ //    BOOL -> CPRED
      return 0;
}

int tCG::p39(){ //   CPRED -> HCPRED )
      S1->obj += ")";
      return 0;
}

int tCG::p40(){ //  HCPRED -> ( $idq
      S1->obj = decor(S2->name) + "(";
      return 0;
}

int tCG::p41(){ //  HCPRED -> HCPRED ARG
      if (S1->count) S1->obj += S1->count % 2 ? ", " : "\n\t ,
";
   S1->obj += S2->obj;
   ++(S1->count);
      return 0;
}

int tCG::p42(){ //    ARG -> E
      return 0;
}

int tCG::p43(){ //    ARG -> BOOL
      return 0;
}

int tCG::p44(){ //    REL -> ( = E E )
      S1->obj = "( " + S3->obj + " == " + S4->obj + " )";
      return 0;
}
```

```
int tCG::p45(){ //    REL -> ( <= E E )
      S1->obj = "( " + S3->obj + " <= " + S4->obj + " )";
      return 0;
}

int tCG::p46(){ //     OR -> HOR BOOL )
      if (S1->count == 0) S1->obj += " " + S2->obj + " )";
   else S1->obj += " || " + S2->obj + " )";
      return 0;
}

int tCG::p47(){ //    HOR -> ( or
      S1->obj = "(";
      return 0;
}

int tCG::p48(){ //    HOR -> HOR BOOL
      if (S1->count == 0) S1->obj += " " + S2->obj;
   else S1->obj += " || " + S2->obj;
   ++S1->count;
      return 0;
}

int tCG::p49(){ //    SET -> HSET E )
      S1->obj += S2->obj;
      return 0;
}

int tCG::p50(){ //   HSET -> ( set! $id
      S1->obj = decor(S3->name) + " = ";
      return 0;
}

int tCG::p51(){ // DISPSET -> ( display E )
      S1->obj = "display(" + S3->obj + ")";
      return 0;
}

int tCG::p52(){ // DISPSET -> ( display BOOL )
      S1->obj = "display(" + S3->obj + ")";
      return 0;
}
```

```
int tCG::p53(){ // DISPSET -> ( display STR )
	S1->obj = "display(" + S3->obj + ")";
	return 0;
}

int tCG::p54(){ // DISPSET -> ( newline )
	S1->obj = "newline()";
	return 0;
}

int tCG::p55(){ // DISPSET -> SET
	return 0;
}

int tCG::p56(){ //   INTER -> DISPSET
	return 0;
}

int tCG::p57(){ //   INTER -> E
	return 0;
}

int tCG::p58(){ //   CALCS -> CALC
	return 0;
}

int tCG::p59(){ //   CALCS -> CALCS CALC
	S1->obj += S2->obj;
	return 0;
}

int tCG::p60(){ //    CALC -> E
	S1->obj = "display(" + S1->obj + ");\n\t newline();\n\t
";
	return 0;
}

int tCG::p61(){ //    CALC -> BOOL
	S1->obj = "display(" + S1->obj + ");\n\t newline();\n\t
";
	return 0;
}

int tCG::p62(){ //    CALC -> STR
```

```
      S1->obj = "display(" + S1->obj + ");\n\t newline();\n\t
";
      return 0;
}

int tCG::p63(){ //    CALC -> DISPSET
      S1->obj += ";\n\t";
      return 0;
}

int tCG::p64(){ //    DEFS -> DEF
      return 0;
}

int tCG::p65(){ //    DEFS -> DEFS DEF
      S1->obj = S1->obj + "\n" + S2->obj;
      return 0;
}

int tCG::p66(){ //     DEF -> PRED
      return 0;
}

int tCG::p67(){ //     DEF -> VAR
      return 0;
}

int tCG::p68(){ //     DEF -> PROC
      return 0;
}

int tCG::p69(){ //    PRED -> HPRED BOOL )
      S1->obj += S2->obj + ";\n\t }\n";
      return 0;
}

int tCG::p70(){ //   HPRED -> PDPAR )
      S1->obj += ")";
    declarations += S1->obj + ";\n\t ";
    S1->obj += "{\n return\n ";
      return 0;
}

int tCG::p71(){ //   PDPAR -> ( define ( $idq
```

```
     S1->obj = "bool " + decor(S4->name) + "/*" + S4->line
+ "*/ (";
     S1->count = 0;
     return 0;
}

int tCG::p72(){ //   PDPAR -> PDPAR $idq
     if(S1->count) S1->obj += S1->count % 2 ? ", " : "\n\t ,
";
   S1->obj += "bool " + decor(S2->name);
   ++(S1->count);
     return 0;
}

int tCG::p73(){ //   PDPAR -> PDPAR $id
     if(S1->count) S1->obj += S1->count % 2 ? ", " : "\n\t ,
";
   S1->obj += "double " + decor(S2->name);
   ++(S1->count);
     return 0;
     }

int tCG::p74(){ //    VAR -> VARDCL E )
     declarations += "extern double " + S1->obj + "/*" + S1-
>line + "*/ ;\n\t";
   S1->obj = "double " + S1->obj + "/*" + S1->line + "*/ = "
+ S2->obj + ";\n\t";
     return 0;
}

int tCG::p75(){ //  VARDCL -> ( define $id
     S1->obj = decor(S3->name);
     return 0;
}

int tCG::p76(){ //    PROC -> HPROC BLOCK )
     S1->obj += S2->obj + "}\n";
     return 0;
}

int tCG::p77(){ //    PROC -> HPROC E )
     S1->obj += "return\n" + S2->obj + ";\n\t }\n";
     return 0;
}
```

```
int tCG::p78(){ //   HPROC -> PCPAR )
      S1->obj += ")";
   declarations += S1->obj + ";\n\t";
   S1->obj += "{\n ";
      return 0;
}

int tCG::p79(){ //   HPROC -> HPROC INTER
      S1->obj += S2->obj + ";\n\t";
      return 0;}

int tCG::p80(){ //   PCPAR -> ( define ( $id
      S1->obj = "double " + decor(S4->name) + "/*" + S4-
>line + "*/ (";
   S1->count = 0;
   S1->name = S4->name;
      return 0;
}

int tCG::p81(){ //   PCPAR -> PCPAR $id
      if (S1->count)
       S1->obj += S1->count % 2 ? ", " : "\n\t , ";
   S1->obj += "double " + decor(S2->name);
   ++(S1->count);
      return 0;
}

int tCG::p82(){ //   BLOCK -> HBLOCK E )
      S1->obj += "return\n" + S2->obj + ";\n\t}\n";
      return 0;
}

int tCG::p83(){ //  HBLOCK -> BLVAR )
      S1->obj += ";\n\t";
      return 0;
}

int tCG::p84(){ //  HBLOCK -> HBLOCK INTER
      S1->obj += S2->obj + ";\n\t";
      return 0;
}

int tCG::p85(){ //   BLVAR -> ( let ( LOCDEF
```

```
	S1->obj = "{\ndouble " + S4->obj;
	return 0;
}

int tCG::p86(){ //   BLVAR -> BLVAR LOCDEF
	S1->obj += ",\n\t" + S2->obj;
	return 0;
}

int tCG::p87(){ //  LOCDEF -> ( $id E )
	S1->obj = decor(S2->name) + "(" + S3->obj + ")";
	return 0;
}

//_____________________________
int tCG::p88(){return 0;} int tCG::p89(){return 0;}
int tCG::p90(){return 0;} int tCG::p91(){return 0;}
int tCG::p92(){return 0;} int tCG::p93(){return 0;}
int tCG::p94(){return 0;} int tCG::p95(){return 0;}
int tCG::p96(){return 0;} int tCG::p97(){return 0;}
int tCG::p98(){return 0;} int tCG::p99(){return 0;}
int tCG::p100(){return 0;} int tCG::p101(){return 0;}
int tCG::p102(){return 0;} int tCG::p103(){return 0;}
int tCG::p104(){return 0;} int tCG::p105(){return 0;}
int tCG::p106(){return 0;} int tCG::p107(){return 0;}
int tCG::p108(){return 0;} int tCG::p109(){return 0;}
int tCG::p110(){return 0;}
```