**Студент: Моисеенков И.П.**

**Группа: М8О-208Б-19**

**Номер по списку: 21**

**Тема: Знакомство с языком МИКРОЛИСП.**

**Отображение программ из МИКРОЛИСПа в С++.**

**Лабораторная работа N2**

**Распечатка файла golden21.cpp.**

**#include "mlisp.h"**

**extern double a;**

**extern double b;**

**double fun(double x);**

**double golden\_\_section\_\_search(double a, double b);**

**double golden\_\_start(double a, double b);**

**extern double mphi;**

**double \_\_mip\_\_try(double a, double b, double xa, double ya, double xb, double yb);**

**bool close\_\_enough\_Q(double x, double y);**

**extern double tolerance;**

**extern double total\_\_iterations;**

**extern double xmin;**

**double a = 2., b = 6.;**

**//(define (fun x)**

**// (set! x (- x (/ 21 22)))**

**// (-(expt(- x 3) 4) (expt(atan x) 3) 2)**

**//)**

**double fun(double x) {**

**x = x - 21. / 22.;**

**return expt(x - 3., 4.) - expt(atan(x), 3.) - 2.;**

**}**

**//(define (golden-section-search a b)**

**// (let(**

**// (xmin(if(< a b)(golden-start a b)(golden-start b a )))**

**// )**

**// (newline)**

**// xmin**

**// )**

**//)**

**double golden\_\_section\_\_search(double a, double b) {**

**{**

**double**

**xmin((a < b ? golden\_\_start(a, b) : golden\_\_start(b, a)));**

**newline();**

**return xmin;**

**}**

**}**

**//(define (golden-start a b)**

**// (set! total-iterations 0)**

**// (let(**

**// (xa (+ a (\* mphi(- b a))))**

**// (xb (+ b (-(\* mphi(- b a)))))**

**// )**

**// (try a b xa (fun xa) xb (fun xb))**

**// )**

**//)**

**double golden\_\_start(double a, double b) {**

**total\_\_iterations = 0.;**

**{**

**double**

**xa(a + (mphi \* (b - a))),**

**xb(b + (-(mphi \* (b - a))));**

**return \_\_mip\_\_try(a, b, xa, fun(xa), xb, fun(xb));**

**}**

**}**

**//(define mphi (\* (- 3(sqrt 5))(/ 2.0)))**

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

**//(define (try a b xa ya xb yb)**

**// (if(close-enough? a b)**

**// (\* (+ a b)0.5)**

**// (let() (display "+")**

**// (set! total-iterations (+ total-iterations 1))**

**// (cond((< ya yb)(set! b xb)**

**// (set! xb xa)**

**// (set! yb ya)**

**// (set! xa (+ a (\* mphi(- b a))))**

**// (try a b xa (fun xa) xb yb)**

**// )**

**// (else (set! a xa)**

**// (set! xa xb)**

**// (set! ya yb)**

**// (set! xb (- b (\* mphi(- b a))))**

**// (try a b xa ya xb (fun xb))**

**// )**

**// );cond...**

**// );let...**

**// );if...**

**//)**

**double \_\_mip\_\_try(double a, double b, double xa, double ya, double xb, double yb) {**

**return (close\_\_enough\_Q(a, b) ?**

**((a + b) \* 0.5) :**

**( display("+"),**

**total\_\_iterations = total\_\_iterations + 1,**

**(ya < yb ? (b = xb,**

**xb = xa,**

**yb = ya,**

**xa = a + mphi \* (b - a),**

**\_\_mip\_\_try(a, b, xa, fun(xa), xb, yb)**

**) : (a = xa,**

**xa = xb,**

**ya = yb,**

**xb = b - mphi \* (b - a),**

**\_\_mip\_\_try(a, b, xa, ya, xb, fun(xb))**

**)**

**)**

**)**

**);**

**}**

**//(define (close-enough? x y)**

**// (<(abs (- x y))tolerance))**

**bool close\_\_enough\_Q(double x, double y) {**

**return (abs(x - y) < tolerance);**

**}**

**double tolerance = 0.001;**

**double total\_\_iterations = 0.;**

**double xmin = 0.;**

**//(set! xmin(golden-section-search a b))**

**// (display"Interval=\t[")**

**// (display a)**

**// (display" , ")**

**// (display b)**

**// (display"]\n")**

**// (display"Total number of iteranions=")**

**//total-iterations**

**// (display"xmin=\t\t")**

**//xmin**

**// (display"f(xmin)=\t")**

**//(fun xmin)**

**int main() {**

**xmin = golden\_\_section\_\_search(a, b);**

**display("Interval=\t[");**

**display(a);**

**display(" , ");**

**display(b);**

**display("]\n");**

**display("Total number of iterations=");**

**display(total\_\_iterations); newline();**

**display("xmin=\t\t");**

**display(xmin); newline();**

**display("f(xmin)=\t");**

**display(fun(xmin)); newline();**

**std::cin.get();**

**return 0;**

**}**

**Распечатка файла golden21.ss.**

**;golden21**

**(define a 2)(define b 6)**

**(define (fun x)**

**(set! x (- x (/ 21 22)))**

**(-(expt(- x 3) 4) (expt(atan x) 3) 2)**

**)**

**(define (golden-section-search a b)**

**(let(**

**(xmin(if(< a b)(golden-start a b)(golden-start b a )))**

**)**

**(newline)**

**xmin**

**)**

**)**

**(define (golden-start a b)**

**(set! total-iterations 0)**

**(let(**

**(xa (+ a (\* mphi(- b a))))**

**(xb (+ b (-(\* mphi(- b a)))))**

**)**

**(try a b xa (fun xa) xb (fun xb))**

**)**

**)**

**(define mphi (\* (- 3(sqrt 5))(/ 2.0)))**

**(define (try a b xa ya xb yb)**

**(if(close-enough? a b)**

**(\* (+ a b)0.5)**

**(let() (display "+")**

**(set! total-iterations (+ total-iterations 1))**

**(cond((< ya yb)(set! b xb)**

**(set! xb xa)**

**(set! yb ya)**

**(set! xa (+ a (\* mphi(- b a))))**

**(try a b xa (fun xa) xb yb)**

**)**

**(else (set! a xa)**

**(set! xa xb)**

**(set! ya yb)**

**(set! xb (- b (\* mphi(- b a))))**

**(try a b xa ya xb (fun xb))**

**)**

**);cond...**

**);let...**

**);if...**

**)**

**(define (close-enough? x y)**

**(<(abs (- x y))tolerance))**

**(define tolerance 0.001)**

**(define total-iterations 0)**

**(define xmin 0)**

**(set! xmin(golden-section-search a b))**

**(display"Interval=\t[")**

**(display a)**

**(display" , ")**

**(display b)**

**(display"]\n")**

**(display"Total number of iteranions=")**

**total-iterations**

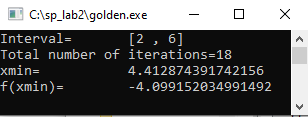
**(display"xmin=\t\t")**

**xmin**

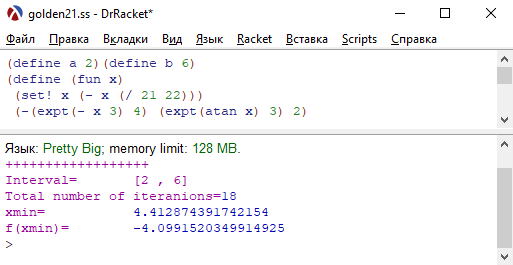
**(display"f(xmin)=\t")**

**(fun xmin)**

**Скриншот запуска в C++**

****

**Скриншот запуска в DrRacket.**

****