Богатов Р.Н.

Программирование на языке высокого уровня

Лекция 7.
Методы класса как подпрограммы.
Решение нелинейных уравнений

Кафедра АСОИУ ОмГТУ, 2012

Подпрограмма. Процедура. Функция. Метод

Машинный я
Подпрогра
глобальны

Pascal:
Процедура
Функция =

C:
Функция м
(процедур

```
// пример вызова процедуры на языке Assebmler
mov   ax, 0
mov   dx, 123
call  my_proc

// пример вызова процедур и функций на языке Pascal
writeln;
y := sin(x);

// пример вызова функций на языке Си
getch();
c = getch();
y = sin(x);

// пример вызова методов на языке C#
textBox1.Hide();
x = r.NextDouble();
y = Math.Sin(x);
```

C#, Java и другие:
Данные и код инкапсулированы в классы. Инкапсулированные подпрограммы называются **методами**.

```
public partial class Form1 : Form
{
    int N;
    int[] a;

    pri
    {
```

```
public partial class Form1 : Form
{
     private void button1_Click(object sender, EventArgs e)
    {
        int N = (int)numericUpDown1.Value;
        Massiv a = new Massiv(N);
        Massiv b;

        a.sluchaino(0, 100);
        a.vyvod(textBox1.Text);
        a.sort();
        a.vyvod(textBox1.Text);
        b = a;
        a.perevorot();
        ...
        label1.Text = "Максимум = " + a.maximum();
        label2.Text = "Сумма без крайних эл-в = " + a.part_sum(1, N-2);
    }
}
```

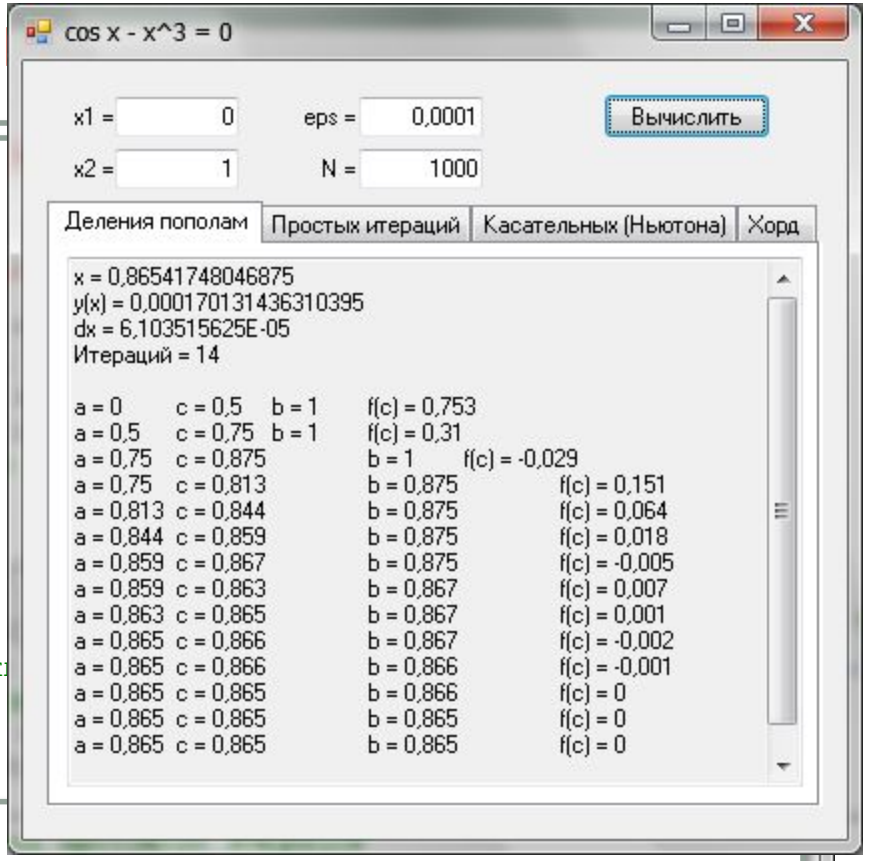
```
    {
        int sum = 0;
        for (int k = i; k <= j; k++)
            sum += a[k];
        return sum;
    }
```

Решение нелинейного ура

```
void metod_del_popolam()
{
    textBox1.Text = "";
    double a = x1, b = x2, c = 0, fc = 0;
    double fa = y(a);
    double fb = y(b);
    // проверка на наличие корней (fa*fb>=0)

    int i = 1;
    for (; i < N; i++)
    {
        c = (a + b) / 2;
        fc = y(c);
        // ...вывод протокола итерации (если нужн
        if (b - c < eps) break;
        if (fa * fc < 0)
        {
            // вывод протокола итерации
            textBox1.Text += String.Format(
                "a = {0:0.###}\t c = {1:0.###}\t b = {2:0.###}\t f(c) = {3:0.###}\r\n",
                 a, c, b, fc);


        // вывод результата
        textBox1.Text = String.Format(
            "x = {0}\r\ny(x) = {1}\r\ndx = {2}\r\nИтераций = {3}\r\n\r\n",
            c, fc, b-c, i) + textBox1.Text;
}
```



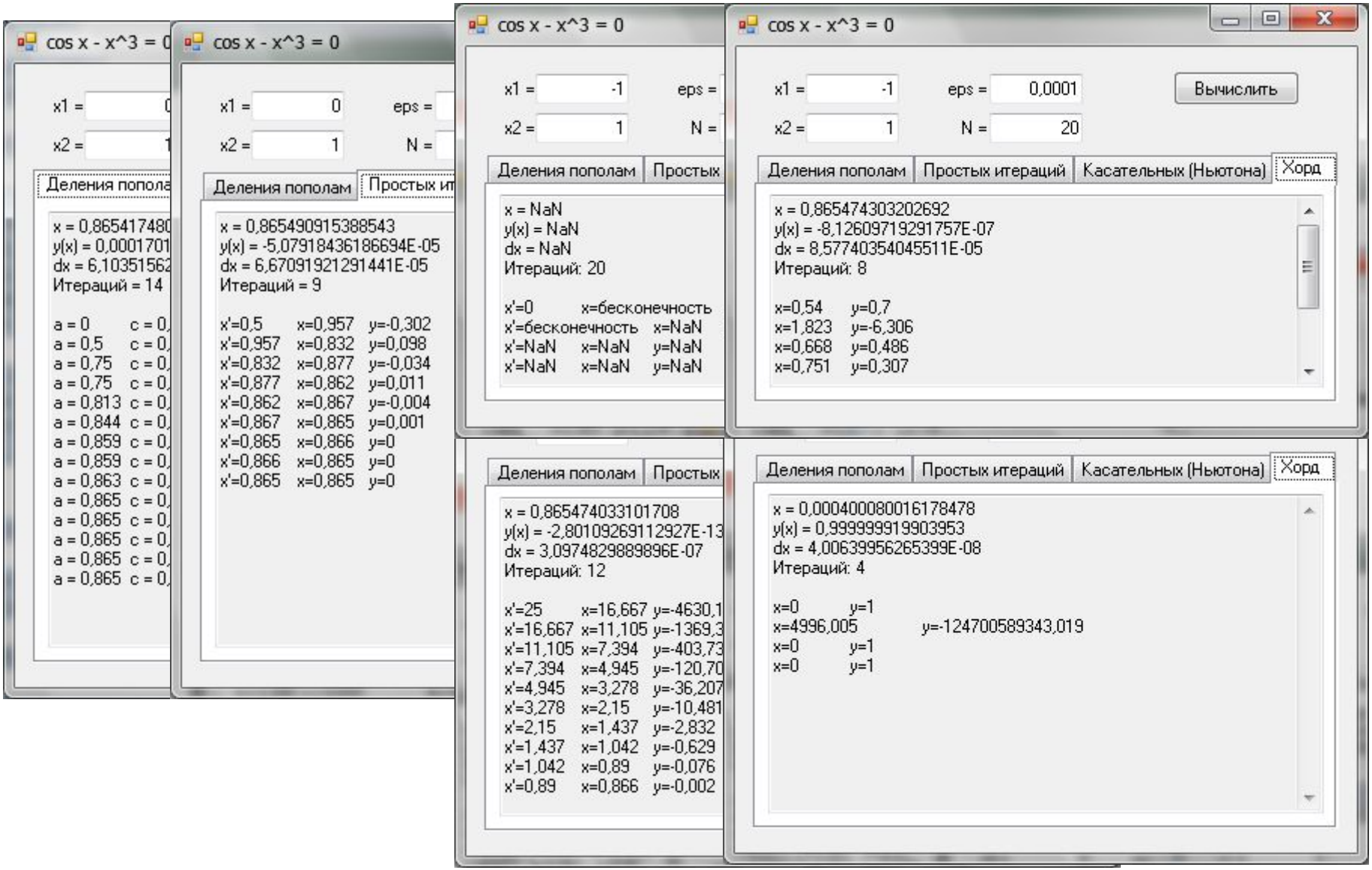
Решение нелинейного уравнения

```
double y_x(double x)
{
    return Math.Pow(Math.Cos(x), 1/3.0);
}
```

```
double dy(double x)
{
    return - Math.Sin(x) - 3 * x * x;
}
```

```
void metod_hord()
{
    textBox8.Text = "";
    double a = x1, b = x2, c = 0;
    int i = 1;
    for (; i < N; i++)
    {
        c = a - y(a) * (a - b) / (y(a) - y(b));
        // ...вывод протокола итерации(если нужно)
        if (Math.Abs(c - a) < eps)
            break;
        b = a;
        a = c;
    }
    // ...вывод или возврат результата
}
```

Решение нелинейного уравнения



Домашнее задание

В таблице ниже приведены наиболее часто используемые параметры линейных конгруэнтных генераторов, в частности, в стандартных библиотеках различных компиляторов (функция *rand()*).

Source	m	a	c	выдаваемые биты результата в *rand()* / *Random(L)*
Numerical Recipes (*англ.*)	2^{32}	1664525	1013904223	
MMIX by Donald Knuth	2^{64}	6364136223846793005	1442695040888963407	
Borland C/C++	2^{32}	22695477	1	биты 30..16 в *rand()*, 30..0 в *lrand()*
GNU Compiler Collection	2^{32}	69069	5	биты 30..16
ANSI C: Open Watcom, Digital Mars, Metrowerks, IBM VisualAge C/C++	2^{32}	1103515245	12345	биты 30..16
Borland Delphi, Virtual Pascal	2^{32}	134775813	1	биты 63..32 числа (seed * L)
Microsoft Visual/Quick C/C++	2^{32}	214013	2531011	биты 30..16
Apple CarbonLib	$2^{31}-1$	16807	0	см. Метод Лемера (*англ.*)

$$X_{k+1} = (aX_k + c) \mod m,$$

где *a* и *c* — некоторые целочисленные коэффициенты. Получаемая последовательность зависит от выбора стартового числа X_0 и при разных его значениях получаются различные последовательности случайных чисел. В то же время, многие свойства этой последовательности определяются выбором коэффициентов в формуле и не зависят от выбора стартового числа.