

Фролова А. ВМК-20.

## Лабораторная №1, вариант 12.

Язык C++

Вариант №12. (а)  $x + 2^x + 5 = 0$ , (б)  $-2x^3 - x^2 + 5x + 1 = 0$ .

Код программы. Вариант а)

```
1. #include <iostream>
2. #include <cmath>
3. using namespace std;
4.
5. double f(double x) //функция
6. {
7.     return x + pow(2, x) + 5;
8. }
9.
10. double df(double x) //первая производная
11. {
12.     return log(2)*pow(2,x) + 1;
13. }
14.
15. double ddf(double x) //вторая производная
16. {
17.     return log(2)*pow(2,x)*log(2);
18. }
19.
20. double halfy(double a, double b, double e) //метод
    половинных отрезков
21. {
22.     double c;
23.     int k = 0;
24.
25.     while (abs(b - a) > 2 * e)
26.     {
27.         c = (a + b) / 2;
28.         if (f(a) * f(c) < 0)
29.             b = c;
30.         else
31.         {
32.             if (f(b) * f(c) < 0)
33.                 a = c;
34.         }
35.         k++;
36.     }
37.     cout << "Итераций: " << k << endl;
38.     return (a + b) / 2;
39. }
```

```

40.
41.     double comba(double a, double b, double e) //комбинированный
        метод хорд и касательных
42.     {
43.         double c, d;
44.         int k = 0;
45.         while ((b - a) > 2 * e)
46.         {
47.             k++;
48.             c = (a * f(b) - b * f(a)) / (f(b) - f(a));
49.             if (f(a) * ddf(a) > 0)
50.             {
51.                 d = a - f(a) / df(a);
52.                 a = d;
53.                 b = c;
54.             }
55.             else
56.             {
57.                 if (f(b) * ddf(b) > 0)
58.                 {
59.                     d = b - f(b) / df(b);
60.                     b = d;
61.                     a = c;
62.                 }
63.             }
64.         }
65.         cout << "Итераций: " << k << endl;
66.         return (a + b) / 2;
67.     }
68.
69.
70.     int main()
71.     {
72.         setlocale(LC_ALL, "Russian");
73.         double e = 0.00001;
74.
75.         double a = -6.2, b = -4.7;
76.
77.         cout << "a) f(x) = x + pow(2,x) + 5" << endl << endl;
78.
79.         cout << "Метод половинного деления " << endl;
80.         cout << "f(0) = " << halfy(a, b, e) << endl << endl;
81.
82.         cout << "Комбинированный метод хорд и касательных " <<
endl;
83.         cout << "f(0) = " << comba(a, b, e) << endl;
84.     }
85.

```

*Вывод в консоль:*

```
Консоль отладки Microsoft Visual Studio
a)  $f(x) = x + \text{pow}(2, x) + 5$ 

Метод половинного деления
Итераций: 17
 $f(0) = -5.03059$ 

Комбинированный метод хорд и касательных
Итераций: 2
 $f(0) = -5.03059$ 
```

**Код программы. Вариант б)**

```
1. #include <iostream>
2. #include <cmath>
3. using namespace std;
4.
5. double f(double x) //функция
6. {
7.     return -2 * pow(x, 3) - pow(x, 2) + 5 * x + 1;
8. }
9.
10. double df(double x) //первая производная
11. {
12.     return -6*pow(x,2) - 2*x + 5;
13. }
14.
15. double ddf(double x) //вторая производная
16. {
17.     return -12*x - 2;
18. }
19.
20. double halfy(double a, double b, double e) //метод
    половинных отрезков
21. {
22.     double c;
23.     int k = 0;
24.
25.     while (abs(b - a) > 2 * e)
26.     {
27.         c = (a + b) / 2;
28.         if (f(a) * f(c) < 0)
29.             b = c;
30.         else
31.         {
32.             if (f(b) * f(c) < 0)
33.                 a = c;
```

```

34.         }
35.         k++;
36.     }
37.     cout << "Итераций: " << k << "\t";
38.     return (a + b) / 2;
39. }
40.
41.     double сомба(double a, double b, double e) //комбинированный
        метод хорд и касательных
42.     {
43.         double c, d;
44.
45.         int k = 0;
46.         while (abs(b - a) > 2 * e)
47.         {
48.             k++;
49.             c = (a * f(b) - b * f(a)) / (f(b) - f(a));
50.             if (f(a) * ddf(a) > 0)
51.             {
52.                 d = a - f(a) / df(a);
53.                 a = d;
54.                 b = c;
55.             }
56.             else
57.             {
58.                 if (f(b) * ddf(b) > 0)
59.                 {
60.                     d = b - f(b) / df(b);
61.                     b = d;
62.                     a = c;
63.                 }
64.             }
65.         }
66.         cout << "Итераций: " << k << "\t";
67.         return (a + b) / 2;
68.     }
69.
70.     int main()
71.     {
72.         setlocale(LC_ALL, "Russian");
73.         double e = 0.001;
74.
75.         float a, b;
76.         // всего три корня
77.         // (-2, -1)
78.         // (-1, 0)
79.         // ( 1, 2)
80.
81.         cout << "b) -2*pow(x,3) - pow(x,2) + 5*x + 1" << endl <<
endl;

```

```

82.
83.     for (int i = 1; i <= 3; i++)
84.     {
85.         //выбираем отрезки
86.         if (i == 1)
87.         { a = -2; b = -1;}
88.         else
89.             if (i == 2)
90.                 { a = 1; b = 2; }
91.             else { a = -1; b = 0;}
92.
93.         //работаем с выбранным отрезком
94.         cout << "- - - - -" << endl;
95.         cout << i << " корень на отрезке (" << a << "; " <<
            b << "): " << endl << endl;
96.
97.         cout << "Метод половинного деления " << endl;
98.         cout << "f(0) = " << halfy(a, b, e) << endl << endl;
99.
100.        if (i == 3) break; //для третьего корня не вычисляем
            по второму методу, тк это не представляется возможным
101.
102.        cout << "Комбинированный метод хорд и касательных "
            << endl;
103.        cout << "f(0) = " << comba(a, b, e) << endl << endl;
104.
105.    }
106. }

```

```

Консоль отладки Microsoft Visual Studio
- - - - -
1 корень на отрезке (-2; -1):

Метод половинного деления
Итераций: 9      f(0) = -1.75879

Комбинированный метод хорд и касательных
Итераций: 3      f(0) = -1.7594

- - - - -
2 корень на отрезке (1; 2):

Метод половинного деления
Итераций: 9      f(0) = 1.4541

Комбинированный метод хорд и касательных
Итераций: 3      f(0) = 1.45465

- - - - -
3 корень на отрезке (-1; 0):

Метод половинного деления
Итераций: 9      f(0) = -0.196289

```