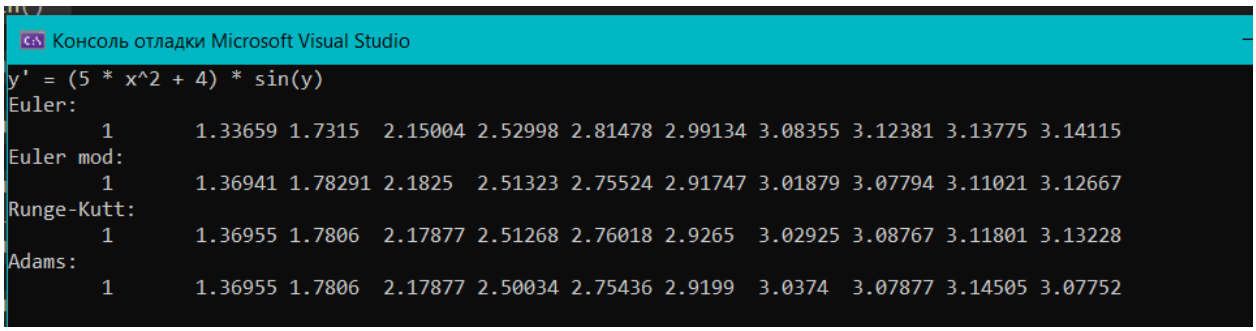
Фролова Анастасия, ВМК-20

Вычмат, 12 вариант

```
#include <iostream>
#include <locale>
#include <vector>
#include <string>
#include <iomanip>
using namespace std;

//функция
double f(double x, double y)
	{
		return (6 * pow(x, 2) +4 ) * sin(y);
	}

	//метод Эйлера
	void euler(double x0, double y0, double h, double e)
	{
		double x = x0, y = y0;

		for (int i = 0; i < 11; i++)
		{
			cout << "\t" << y;

			y = y + h * f(x, y);

			x = x + h;
		}
	}

	//модифицированный метод эйлера
	void euler_modif(double x0, double y0, double h, double e)
	{
		double x = x0, y = y0;

		for (int i = 0; i < 11; i++)
		{
			cout << "\t" << y;

			y = y + h * f(x + h / 2, y + h / 2 * f(x, y));

			x = x + h;
		}
	}

	//метод Рунге-Кутта
	void runge_kutt(double x0, double y0, double h, double e)
	{
```

```
47.           vector<double> z(4);
48.
49.           double x = x0, y = y0;
50.           for (int i = 0; i < 11; i++)
51.           {
52.                 //вычисляем коэффициенты
53.                 z[0] = h * f(x, y);
54.                 z[1] = h * f(x + h / 2, y + z[0] / 2);
55.                 z[2] = h * f(x + h / 2, y + z[1] / 2);
56.                 z[3] = h * f(x + h, y + z[2]);
57.
58.                 cout << "\t" << y;
59.
60.                 y = y + (z[0] + 2 * z[1] + 2 * z[2] + z[3]) / 6;
61.
62.                 x = x + h;
63.           }
64.     }
65.
66.     //метод Адамса
67.     void adams(double x0, double y0, double h, double e)
68.     {
69.           vector<double> z(4), yk(4), xk(4);
70.
71.           double x = x0, y = y0;
72.           double t = 0;
73.
74.           yk[0] = y0;
75.           xk[0] = x0;
76.
77.           cout << setw(9) << y0;
78.
79.           for (int i = 1; i < 11; i++)
80.           {
81.                       if (i < 4)
82.                       {
83.                             z[0] = h * f(x, y);
84.                             z[1] = h * f(x + h / 2, y + z[0] / 2);
85.                             z[2] = h * f(x + h / 2, y + z[1] / 2);
86.                             z[3] = h * f(x + h, y + z[2]);
87.
88.                             y = y + (z[0] + 2 * z[1] + 2 * z[2] +
   z[3]) / 6;
89.                             x = x + h;
90.
91.                             yk[i] = y;
92.                             xk[i] = x;
93.                       }
94.                       else
95.                       {
```

```
96.                             t = 55 * f(xk[3], yk[3]) - 59 *
   f(xk[2], yk[2]) + 37 * f(xk[1], yk[1]) - 9 * f(xk[0], yk[0]);
97.                             y = y + (h / 24) * t;
98.
99.                             xk[0] = xk[0] + h;
100.                            xk[1] = xk[1] + h;
101.                            xk[2] = xk[2] + h;
102.                            xk[3] = xk[3] + h;
103.
104.                            yk[0] = yk[1];
105.                            yk[1] = yk[2];
106.                            yk[2] = yk[3];
107.                            yk[3] = y;
108.                    }
109.                cout << "\t" << y;
110.        }
111.        return;
112.    }
113.    int main()
114.    {
115.        double x0 = 0, y0 = 1, h = 0.1, e = 0.001;
116.
117.        cout << "y' = (5 * x^2 + 4) * sin(y)" << endl;
118.
119.        cout << "Euler:" << endl;
120.        euler(x0, y0, h, e);
121.        cout << endl;
122.
123.        cout << "Euler mod:" << endl;
124.        euler_modif(x0, y0, h, e);
125.        cout << endl;
126.
127.        cout << "Runge-Kutt:" << endl;
128.        runge_kutt(x0, y0, h, e);
129.        cout << endl;
130.
131.        cout << "Adams:" << endl;
132.        adams(x0, y0, h, e);
133.        cout << endl;
134.
135.        return 0;
136.    }
```


Консоль отладки Microsoft Visual Studio
y' = (5 * x^2 + 4) * sin(y)
Euler:
1 1.33659 1.7315 2.15004 2.52998 2.81478 2.99134 3.08355 3.12381 3.13775 3.14115
Euler mod:
1 1.36941 1.78291 2.1825 2.51323 2.75524 2.91747 3.01879 3.07794 3.11021 3.12667
Runge-Kutt:
1 1.36955 1.7806 2.17877 2.51268 2.76018 2.9265 3.02925 3.08767 3.11801 3.13228
Adams:
1 1.36955 1.7806 2.17877 2.50034 2.75436 2.9199 3.0374 3.07877 3.14505 3.07752