

```

1 using System;
2 using static System.Math;
3 //многомерная минимизация
4 //методы нахождения минимума функции
5 //Павлов Дмитрий 206 группа
6 // A0 = Д + П*i = 5 + 17i
7 // A1 = И + А*i = 10 + i
8 // A2 = М + В*i = 14 + 3i
9 // A3 = Л*i = 13i
10 // f(x, y) = |A0+A1*z+A2*z^2+A3*z^3|^2
11 // функция имеет три локальных минимума
12 //в x=-0.969085 y=0.245384
13 //x=0.435595 y=-0.923594
14 //x=0.436993 y=1.03343
15 namespace M.O._2
16 {
17     class Program
18     {
19         const double Eps = 1e-6;
20         const double Left = 0;
21         const double Right = 1.5;
22
23         const double X1= 0,X2=0;//точка, с которой начинается минимизация
24         static double Funct(double x,double y)//возвращает значение функции
25             в точке
26         {
27             return Pow(-39*x*x*y+14*x*x-6*x*y+10*x+13*y*y*y-14*y*y-y+5,2)
28                 +Pow(13*x*x*x+3*x*x-39*x*y*y+28*x*y+x-3*y*y-10*y+17,2);
29         }
30         static double Derivative_x(double x,double y)//первая производная по
31             x
32         {
33             return 2*(-39*x*x*y+14*x*x-6*x*y+10*x+13*y*y*y-14*y*y-y+5)*(-78*x*y
34                 +28*x-6*y+10)+2*(13*x*x*x+3*x*x-39*x*y*y+28*x*y+x-3*y*y-10*y+17)*
35                 (26*x*x+6*x-39*y*y+28*y+1);
36         }
37         static double Derivative_y(double x, double y)//первая производная
38             по y
39         {
40             return 2 * (-39 * x * x * y + 14 * x * x - 6 * x * y + 10 * x +
41                 13 * y * y * y - 14 * y * y - y + 5) * (-39 * x * x - 6 * x +
42                 39 * y * y - 28 * y - 1) + 2 * (13 * x * x * x + 3 * x * x -
43                 39 * x * y * y + 28 * x * y + x - 3 * y * y - 10 * y + 17) *
44                 (-78 * x * y + 28 * x - 6 * y - 10);
45         }
46         static double F(double x, double y, double o)//слегка видоизмененная
47             функция для МНГС // o - альфа
48         {
49             double q1 = Derivative_x(x, y);
50             double q2 = Derivative_y(x, y);
51             return Pow(-39 *(x-o*q1) * (x - o * q1) * (y - o * q2) + 14 * (x
52                 - o * q1) * (x - o * q1) - 6 * (x - o * q1) * (y - o * q2) +

```

```

10 * (x - o * q1) + 13 * (y - o * q2) * (y - o * q2) * (y - o *
* q2) - 14 * (y - o * q2) * (y - o * q2) - (y - o * q2) + 5,
2) + Pow(13 * (x - o * q1) * (x - o * q1) * (x - o * q1) + 3 *
(x - o * q1) * (x - o * q1) - 39 * (x - o * q1) * (y - o *
q2) * (y - o * q2) + 28 * (x - o * q1) * (y - o * q2) + (x - o
* q1) - 3 * (y - o * q2) * (y - o * q2) - 10 * (y-o*q2) + 17,
2);
42     }
43     static double GR(double x,double y)//золотое сечение для МНГС
44     {
45         double a = Left;
46         double b = Right;
47         double s5 = Sqrt(5);
48         double c = (3 - s5) / 2 * (b - a) + a;
49         double d = (s5 - 1) / 2 * (b - a) + a;
50         while ((b - a) / 2 > Eps)
51         {
52             if (F(x,y,c)<=F(x,y,d))
53             {
54                 b = d;
55                 d = c;
56                 c = (3 - s5) / 2 * (b - a) + a;
57             }
58             else
59             {
60                 a = c;
61                 c = d;
62                 d = (s5 - 1) / 2 * (b - a) + a;
63             }
64         }
65         return (a + b) / 2;
66     }
67     static double GoldenRatio(int p, double k)//золотое сечение p=0 - x
        фиксирован, p=1 - y фиксирован, k - фиксированная переменная
68     {
69         double a = Left;
70         double b = Right;
71         double s5 = Sqrt(5);
72         double c = (3 - s5) / 2 * (b - a) + a;
73         double d = (s5 - 1) / 2 * (b - a) + a;
74         if (p==1)
75         {
76             while ((b - a) / 2 > Eps)
77             {
78                 if (Funct(c,k) <= Funct(d,k))
79                 {
80                     b = d;
81                     d = c;
82                     c = (3 - s5) / 2 * (b - a) + a;
83                 }
84                 else
85                 {
86                     a = c;

```

```
87         c = d;
88         d = (s5 - 1) / 2 * (b - a) + a;
89     }
90 }
91 }
92 else
93 {
94     while ((b - a) / 2 > Eps)
95     {
96         if (Funct(k,c) <= Funct(k,d))
97         {
98             b = d;
99             d = c;
100             c = (3 - s5) / 2 * (b - a) + a;
101         }
102         else
103         {
104             a = c;
105             c = d;
106             d = (s5 - 1) / 2 * (b - a) + a;
107         }
108     }
109 }
110 return (a + b) / 2;
111 }
112
113 static void Main()
114 {
115     Console.WriteLine("Функция |A0+A1*z+A2*z^2+A3*z^3|^2");
116     Console.WriteLine("A0 = Д + П*i = 5 + 17i");
117     Console.WriteLine("A1 = И + А*i = 10 + i");
118     Console.WriteLine("A2 = М + В*i = 14 + 3i");
119     Console.WriteLine("A3 = Л*i = 13i");
120     Console.WriteLine("Нажмите цифру для одномерной минимизации:");
121     Console.WriteLine("1.Метод покоординатного спуска");
122     Console.WriteLine("2.Метод градиентного спуска с дроблением шага");
123     Console.WriteLine("3.Метод градиентного спуска с постоянным шагом");
124     Console.WriteLine("4.Метод наискорейшего градиентного спуска");
125     Console.WriteLine("5.Метод градиентного спуска с заранее заданным шагом");
126     int key = Convert.ToInt16(Console.ReadLine());
127     switch (key)
128     {
129         case 1:
130             CoordinateDescent();
131             break;
132         case 2:
133             CrushingGradient();
134             break;
135         case 3:
136             ConstantGradient();
```

```
137         break;
138     case 4:
139         FastestGradient();
140         break;
141     case 5:
142         DivergentSeries();
143         break;
144     }
145 }
146
147 static void CoordinateDescent()//метод покоординатного спуска
148 {
149     Console.WriteLine("метод покоординатного спуска");
150     double a = X1, b = X2;
151     double previous = Funct(a, b);
152     Console.WriteLine("x={0},y={1},F(x,y)={2}", a, b, previous);
153     a = GoldenRatio(1, b);
154     b = GoldenRatio(0, a);
155     double current = Funct(a, b);
156     while(Abs(current-previous)>Eps)
157     {
158         previous = current;
159         Console.WriteLine("x={0},y={1},F(x,y)={2}", a, b, previous);
160         a = GoldenRatio(1, b);
161         b = GoldenRatio(0, a);
162         current = Funct(a, b);
163     }
164     Console.WriteLine("Минимум достигается в точке:");
165     Console.WriteLine("x={0},y={1},F(x,y)={2}", a, b, current);
166     Console.ReadKey();
167 }
168
169 static void CrushingGradient()//метод градиентного спуска с
170     дроблением шага
171 {
172     Console.WriteLine("метод градиентного спуска с дроблением
173     шага");
174     double x = X1;
175     double y = X2;
176     double a=1;
177     double delta = 0.95;
178     Console.WriteLine("x={0},y={1},F(x,y)={2}", x, y, Funct(x,y));
179     //градиент (q1,q2) в 1 точке
180     double q1 = Derivative_x(x, y);
181     double q2 = Derivative_y(x, y);
182     //
183     double norm = Sqrt(q1 * q1 + q2 * q2);
184     while (norm >= Eps)
185     {
186         while (Funct(x-a*q1,y-a*q2)-Funct(x,y)>-a*Eps* (q1 * q1 + q2 * q2))
187         {
188             a = a * delta;
```

```
187     }
188     x = x - a * q1;
189     y = y - a * q2;
190     Console.WriteLine("x={0},y={1},F(x,y)={2}", x, y, Funct(x, y));
191     q1 = Derivative_x(x, y);
192     q2 = Derivative_y(x, y);
193     norm = Sqrt(q1 * q1 + q2 * q2);
194 }
195 Console.WriteLine("Минимум достигается в точке:");
196 Console.WriteLine("x={0},y={1},F(x,y)={2}", x, y, Funct(x, y));
197 Console.ReadKey();
198 }
199
200 static void ConstantGradient()//метод градиентного спуска с
    постоянным шагом
201 {
202     Console.WriteLine("метод градиентного спуска с постоянным
    шагом");
203     double a = 0.0001;//shag
204     double x = X1;
205     double y = X2;
206     Console.WriteLine("x={0},y={1},F(x,y)={2}", x, y, Funct(x, y));
207     //градиент (q1,q2)
208     double q1 = Derivative_x(x, y);
209     double q2 = Derivative_y(x, y);
210     //
211     double norm = Sqrt(q1 * q1 + q2 * q2);
212     while (norm>=Eps)
213     {
214         x = x - a * q1;
215         y = y - a * q2;
216         Console.WriteLine("x={0},y={1},F(x,y)={2}", x, y, Funct(x, y));
217         q1 = Derivative_x(x, y);
218         q2 = Derivative_y(x, y);
219         norm = Sqrt(q1 * q1 + q2 * q2);
220     }
221
222     Console.WriteLine("Минимум достигается в точке:");
223     Console.WriteLine("x={0},y={1},F(x,y)={2}", x, y, Funct(x, y));
224     Console.ReadKey();
225 }
226
227 static void DivergentSeries()//метод градиентного спуска с заранее
    заданным шагом
228 {
229     Console.WriteLine("метод градиентного спуска с заранее заданным
    шагом");
230     double x = X1;
231     double y = X2;
232     Console.WriteLine("x={0},y={1},F(x,y)={2}", x, y, Funct(x, y));
233     double k = 1000;
```

```
234         //градиент (q1,q2)
235         double q1 = Derivative_x(x, y);
236         double q2 = Derivative_y(x, y);
237         //
238         double norm = Sqrt(q1 * q1 + q2 * q2);
239         while (norm >= Eps)
240         {
241             k++;
242             x = x - q1/k;
243             y = y - q2/k;
244             Console.WriteLine("x={0},y={1},F(x,y)={2}", x, y, Funct(x, y));
245             q1 = Derivative_x(x, y);
246             q2 = Derivative_y(x, y);
247             norm = Sqrt(q1 * q1 + q2 * q2);
248         }
249         Console.WriteLine("Минимум достигается в точке:");
250         Console.WriteLine("x={0},y={1},F(x,y)={2}", x, y, Funct(x, y));
251         Console.ReadKey();
252     }
253
254     static void FastestGradient()//метод наискорейшего градиентного
        спуска
255     {
256         Console.WriteLine("метод наискорейшего градиентного спуска");
257         double x = X1;
258         double y = X2;
259         double a;
260         Console.WriteLine("x={0},y={1},F(x,y)={2}", x, y, Funct(x, y));
261         //градиент (q1,q2)
262         double q1 = Derivative_x(x, y);
263         double q2 = Derivative_y(x, y);
264         //
265         double norm = Sqrt(q1 * q1 + q2 * q2);
266         while (norm >= Eps)
267         {
268             a=GR(x, y);
269             x = x - a*q1;
270             y = y - a*q2;
271             Console.WriteLine("x={0},y={1},F(x,y)={2}", x, y, Funct(x, y));
272             q1 = Derivative_x(x, y);
273             q2 = Derivative_y(x, y);
274             norm = Sqrt(q1 * q1 + q2 * q2);
275         }
276         Console.WriteLine("Минимум достигается в точке:");
277         Console.WriteLine("x={0},y={1},F(x,y)={2}", x, y, Funct(x, y));
278         Console.ReadKey();
279     }
280 }
281 }
```