

```
1 using System;
2 using System.Collections.Generic;
3 using static System.Math;
4 //одномерная минимизация
5 //методы нахождения минимума функции
6 //Павлов Дмитрий 206 группа
7 //ln(1+x^2)-sin(x)
8 //Console.Write(""); Console.WriteLine();
9 namespace M.O
10 {
11     class Program
12     {
13         const double left = 0; //левая граница
14         const double right = PI / 4; //правая граница
15         const double eps = 1e-6;
16
17
18         static void Main(string[] args)
19         {
20             Console.WriteLine("Функция ln(1+x^2)-sin(x)");
21             Console.WriteLine("Нажмите цифру для одномерной минимизации:");
22             Console.WriteLine("1.Методом пассивного поиска");
23             Console.WriteLine("2.Методом дихотомии");
24             Console.WriteLine("3.Методом золотого сечения");
25             Console.WriteLine("4.Методом Фибоначчи");
26             Console.WriteLine("5.Модифицированным методом Фибоначчи");
27             Console.WriteLine("6.Методом Ньютона-Рафсона");
28             Console.WriteLine("7.Методом секущих");
29             Console.WriteLine("8.Методом касательных(тангенсов)");
30             int key = Convert.ToInt16(Console.ReadLine());
31             switch (key)
32             {
33                 case 1:
34                     PassiveSearch();
35                     break;
36                 case 2:
37                     Dichotomy();
38                     break;
39                 case 3:
40                     GoldenRatio();
41                     break;
42                 case 4:
43                     Console.WriteLine("Метод Фибоначчи");
44                     Fibonacci(false);
45                     break;
46                 case 5:
47                     Console.WriteLine("Модифицированный метод Фибоначчи");
48                     Fibonacci(true);
49                     break;
50                 case 6:
51                     NewtonRaphson();
52                     break;
53                 case 7:
```

```
54         Secant();
55         break;
56     case 8:
57         Tangents();
58         break;
59
60     }
61 }
62
63
64 static double Funct(double x)//возвращает значение функции в точке
65 {
66     return Log(1 + Pow(x, 2)) - Sin(x);
67 }
68 static double Derivative(double x)//производная
69 {
70     return (2 * x) / (1 + Pow(x, 2)) - Cos(x);
71 }
72 static double SecondDerivative(double x)//вторая производная
73 {
74     return Sin(x) - (2 * (Pow(x, 2) - 1)) / Pow(1 + Pow(x, 2), 2);
75 }
76
77
78 static void PassiveSearch()//метод пассивного поиска
79 {
80     Console.WriteLine("Метод пассивного поиска ");
81     int i = 0;
82     int k = Convert.ToInt32(right / (eps*10000) + 1);
83     Console.Write("Число итераций: "); Console.WriteLine(k);
84     double xi = X(i, k);
85     double min = Funct(xi);
86     double xmin = xi;
87     Console.Write("Номер индекса i: "); Console.WriteLine(i);
88     Console.Write("    Значение аргумента X(i) "); ↗
89     Console.WriteLine(xi);
90     Console.Write("    Значение функции f(x(i)) "); ↗
91     Console.WriteLine(Funct(xi));
92     i++;
93     while (i <= k)
94     {
95         xi = X(i, k);
96         double p = Funct(xi);
97         if (min > p)
98         {
99             xmin = xi;
100             min = p;
101         }
102         Console.Write("Номер индекса i: "); Console.WriteLine(i);
103         Console.Write("    Значение аргумента X(i) "); ↗
104         Console.WriteLine(xi);
105         Console.Write("    Значение функции f(x(i)) "); ↗
106         Console.WriteLine(Funct(xi));
```

```
103         i++;
104     }
105     Console.Write("Минимальное значение достигается в x = ");
106     Console.WriteLine(xmin);
107     Console.Write("Значение функции в этой точке ");
108     Console.WriteLine(min);
109     Console.ReadKey();
110 }
111
112 static void Dichotomy()//метод дихотомии
113 {
114     Console.WriteLine("Метод дихотомии");
115     double delta = eps / 5;
116     int i = 0;
117     double a = left;
118     double b = right;
119     double c, d;
120     c = (a + b) / 2 - delta / 2;
121     d = (a + b) / 2 + delta / 2;
122     Console.Write("Номер индекса i: "); Console.WriteLine(i);
123     Console.Write("    Левая граница "); Console.WriteLine(a);
124     Console.Write("    Правая граница "); Console.WriteLine(b);
125     Console.Write("    Точка c "); Console.WriteLine(c);
126     Console.Write("    Значение функции в этой точке ");
127     Console.WriteLine(Funct(c));
128     Console.Write("    Точка d "); Console.WriteLine(d);
129     Console.Write("    Значение функции в этой точке ");
130     Console.WriteLine(Funct(d));
131     if (Funct(c) <= Funct(d)) b = d;
132     else a = c;
133     while ((b - a) / 2 > eps)
134     {
135         i++;
136         c = (a + b) / 2 - delta / 2;
137         d = (a + b) / 2 + delta / 2;
138         Console.Write("Номер индекса i: "); Console.WriteLine(i);
139         Console.Write("    Левая граница "); Console.WriteLine(a);
140         Console.Write("    Правая граница "); Console.WriteLine(b);
141         Console.Write("    Точка c "); Console.WriteLine(c);
142         Console.Write("    Значение функции в этой точке ");
143         Console.WriteLine(Funct(c));
144         Console.Write("    Точка d "); Console.WriteLine(d);
145         Console.Write("    Значение функции в этой точке ");
146         Console.WriteLine(Funct(d));
147         if (Funct(c) <= Funct(d)) b = d;
148         else a = c;
149     }
150     double min = Funct((a + b) / 2);
151     Console.Write("Минимальное значение достигается в x = ");
152     Console.WriteLine((a + b) / 2);
153     Console.Write("Значение функции в этой точке ");
154     Console.WriteLine(min);
```

```
148     Console.ReadKey();
149 }
150
151 static void GoldenRatio()//золотое сечение
152 {
153     Console.WriteLine("Метод золотого сечения");
154     int i = 0;
155     double a = left;
156     double b = right;
157     double c = (3 - Sqrt(5)) / 2 * (b - a) + a;
158     double d = (Sqrt(5) - 1) / 2 * (b - a) + a;
159     Console.Write("Номер индекса i:"); Console.WriteLine(i);
160     Console.Write("    Левая граница"); Console.WriteLine(a);
161     Console.Write("    Правая граница"); Console.WriteLine(b);
162     Console.Write("    Первая точка золотого сечения");
163     Console.WriteLine(c);
164     Console.Write("    Значение функции в этой точке");
165     Console.WriteLine(Funct(c));
166     Console.Write("    Вторая точка золотого сечения");
167     Console.WriteLine(d);
168     Console.Write("    Значение функции в этой точке");
169     Console.WriteLine(Funct(d));
170     while ((b - a) / 2 > eps)
171     {
172         i++;
173         if (Funct(c) <= Funct(d))
174         {
175             b = d;
176             d = c;
177             c = (3 - Sqrt(5)) / 2 * (b - a) + a;
178         }
179         else
180         {
181             a = c;
182             c = d;
183             d = (Sqrt(5) - 1) / 2 * (b - a) + a;
184         }
185         Console.Write("Номер индекса i:"); Console.WriteLine(i);
186         Console.Write("    Левая граница"); Console.WriteLine(a);
187         Console.Write("    Правая граница"); Console.WriteLine(b);
188         Console.Write("    Первая точка золотого сечения");
189         Console.WriteLine(c);
190         Console.Write("    Значение функции в этой точке");
191         Console.WriteLine(Funct(c));
192         Console.Write("    Вторая точка золотого сечения");
193         Console.WriteLine(d);
194         Console.Write("    Значение функции в этой точке");
195         Console.WriteLine(Funct(d));
196     }
197     double min = Funct((a + b) / 2);
198
199     Console.Write("Минимальное значение достигается в x = ");
200     Console.WriteLine((a + b) / 2);
```

```
192 Console.Write("Значение функции в этой точке");
193 Console.WriteLine(min);
194 Console.ReadKey();
195 }
196 static void Fibonacci(bool Modif)//метод Фибоначчи
197 {
198     double a = left;
199     double b = right;
200     double c = 0, d;
201     List<double> F = new List<double>();
202     F.Add(1);
203     if (Modif)
204     {
205         F.Add((1 + Sqrt(5)) / 2);
206     }
207     else F.Add(1);
208     int i = 1;
209     while ((b - a) / eps > F[i])
210     {
211         i++;
212         F.Add(F[i - 1] + F[i - 2]);
213     }
214     int j=1;
215     c = a + (b - a) * (Convert.ToDouble(F[i-2]) / Convert.ToDouble(F
216     [i-2 + 2]));
217     d = a + (b - a) * (Convert.ToDouble(F[i-2 + 1]) /
218     Convert.ToDouble(F[i-2 + 2]));
219     Console.Write("Число итераций: "); Console.WriteLine(i-1);
220     Console.Write("Номер индекса i:"); Console.WriteLine(j);
221     Console.Write("    Левая граница"); Console.WriteLine(a);
222     Console.Write("    Правая граница"); Console.WriteLine(b);
223     Console.Write("    Точка c "); Console.WriteLine(c);
224     Console.Write("    Значение функции в этой точке ");
225     Console.WriteLine(Func(c));
226     Console.Write("    Точка d "); Console.WriteLine(d);
227     Console.Write("    Значение функции в этой точке ");
228     Console.WriteLine(Func(d));
229     for (int n = i - 2; n > 0; n--)
230     {
231         j++;
232         if (Func(c) <= Func(d))
233         {
234             b = d;
235             d = c;
236             c = a + (b - a) * (Convert.ToDouble(F[n]) /
237             Convert.ToDouble(F[n + 2]));
238         }
239         else
240         {
241             a = c;
242             c = d;
243             d = a + (b - a) * (Convert.ToDouble(F[n + 1]) /
```

```
Convert.ToDouble(F[n + 2]));
239     }
240     Console.Write("Номер индекса i:"); Console.WriteLine(j);
241     Console.Write("    Левая граница"); Console.WriteLine(a);
242     Console.Write("    Правая граница"); Console.WriteLine(b);
243     Console.Write("    Точка c "); Console.WriteLine(c);
244     Console.Write("    Значение функции в этой точке ");
245         Console.WriteLine(Funct(c));
246     Console.Write("    Точка d "); Console.WriteLine(d);
247     Console.Write("    Значение функции в этой точке ");
248         Console.WriteLine(Funct(d));
249 }
250
251 double min = Funct(c);
252
253 Console.Write("Минимальное значение достигается в x = ");
254     Console.WriteLine(c);
255
256 Console.Write("Значение функции в этой точке");
257     Console.WriteLine(min);
258 Console.ReadKey();
259 }
260
261 static void NewtonRaphson()
262 {
263     Console.WriteLine("Метод Ньютона-Рафсона");
264     double a = left;
265     double b = right;
266     double xi = a;
267     int i = 0;
268     Console.Write("Номер индекса i: "); Console.WriteLine(i);
269     Console.Write("    Значение аргумента X(i) ");
270         Console.WriteLine(xi);
271     Console.Write("    Значение функции f(x(i)) ");
272         Console.WriteLine(Funct(xi));
273     Console.Write("    Значение производной F(X(i)) ");
274         Console.WriteLine(Derivative(xi));
275     Console.Write("    Значение второй производной F(X(i)) ");
276         Console.WriteLine(SecondDerivative(xi));
277     while (Abs(Derivative(xi)) > eps)
278     {
279         i++;
280         xi = xi - Derivative(xi) / SecondDerivative(xi);
281         Console.Write("Номер индекса i: "); Console.WriteLine(i);
282         Console.Write("    Значение аргумента X(i) ");
283             Console.WriteLine(xi);
284         Console.Write("    Значение функции f(x(i)) ");
285             Console.WriteLine(Funct(xi));
286         Console.Write("    Значение производной F(X(i)) ");
287             Console.WriteLine(Derivative(xi));
288         Console.Write("    Значение второй производной F(X(i)) ");
289             Console.WriteLine(SecondDerivative(xi));
290     }
291 }
```

```
279 Console.Write("Минимальное значение достигается в x = ");
    Console.WriteLine(xi);
280 Console.Write("Значение функции в этой точке");
    Console.WriteLine(Funct(xi));
281 Console.ReadKey();
282 }
283
284 static void Secant()
285 {
286     Console.WriteLine("Метод секущих");
287     double a = left;
288     double b = right;
289     double c = Intersection(a, b);
290     int i = 0;
291     Console.Write("Номер индекса i: "); Console.WriteLine(i);
292     Console.Write("    Левая граница "); Console.WriteLine(a);
293     Console.Write("    Правая граница "); Console.WriteLine(b);
294     Console.Write("    Точка c "); Console.WriteLine(c);
295     Console.Write("    Значение функции в этой точке ");
        Console.WriteLine(Funct(c));
296     while (Abs(Derivative(b)) > eps)
297     {
298         i++;
299         a = b;
300         b = c;
301         c = Intersection(a, b);
302         Console.Write("Номер индекса i: "); Console.WriteLine(i);
303         Console.Write("    Левая граница "); Console.WriteLine(a);
304         Console.Write("    Правая граница "); Console.WriteLine(b);
305         Console.Write("    Точка c "); Console.WriteLine(c);
306         Console.Write("    Значение функции в этой точке ");
            Console.WriteLine(Funct(c));
307     }
308     Console.ReadKey();
309 }
310
311 static void Tangents()
312 {
313     Console.WriteLine("Метод касательных(тангенсов)");
314     double a = left;
315     double b = right;
316     double c; //точка пересечения производных
317     c = (a * Derivative(a) - b * Derivative(b) - Funct(a) + Funct
        (b))/(Derivative(a)-Derivative(b));
318     int i = 0;
319     Console.Write("Номер индекса i: "); Console.WriteLine(i);
320     Console.Write("    Левая граница "); Console.WriteLine(a);
321     Console.Write("    Правая граница "); Console.WriteLine(b);
322     Console.Write("    Точка c "); Console.WriteLine(c);
323     Console.Write("    Значение функции в этой точке ");
        Console.WriteLine(Funct(c));
324     while ((Abs(b-a)>eps)&&(Abs(Derivative(c)) > eps)&&(Abs(Funct
        (a)-Funct(b))>eps))
```

```
325     {
326         i++;
327         if (Derivative(c) > 0) b = c;
328         else if (Derivative(c) < 0) a = c;
329         else break;
330         c = (a * Derivative(a) - b * Derivative(b) - Funct(a) +
331             Funct(b)) / (Derivative(a) - Derivative(b));
332         Console.Write("Номер индекса i: "); Console.WriteLine(i);
333         Console.Write("    Левая граница "); Console.WriteLine(a);
334         Console.Write("    Правая граница "); Console.WriteLine(b);
335         Console.Write("    Точка c "); Console.WriteLine(c);
336         Console.Write("    Значение функции в этой точке ");
337         Console.WriteLine(Funct(c));
338     }
339     Console.ReadKey();
340 }
341 static double X(int i, int k)//высчитывает x с индексом i для метода
342     {
343         return i * ((right - left) / k);
344     }
345 static double Intersection(double x0, double x1)//пересечение для
346     {
347         return x1 - Derivative(x1) * ((x1 - x0) / (Derivative(x1) -
348             Derivative(x0)));
349     }
350 }
```