

Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра Автоматизированных систем управления

Индивидуальное домашнее задание

По дисциплине «Математическое программирование»

Методы оптимизации

Студент

Агеев А.А.

Группа ПИ-18

Руководитель

Профессор

Качановский Ю.П.

Липецк 2021г

Задание кафедры

Разработать алгоритм поиска минимума функции многих переменных методом штрафов.

Описание метода

Метод штрафных функций является одним из наиболее простых и широко применяемых методов решения нелинейных задач оптимизации. Идея метода основана на преобразовании исходной задачи с ограничениями в одну или последовательность задач безусловной оптимизации. Суть такого преобразования заключается в следующем. С помощью функций, задающих ограничения, строится так называемый штраф, который добавляется к целевой функции исходной задачи так, чтобы нарушение какого-нибудь ограничений становилось невыгодным с точки зрения полученной задачи безусловной оптимизации.

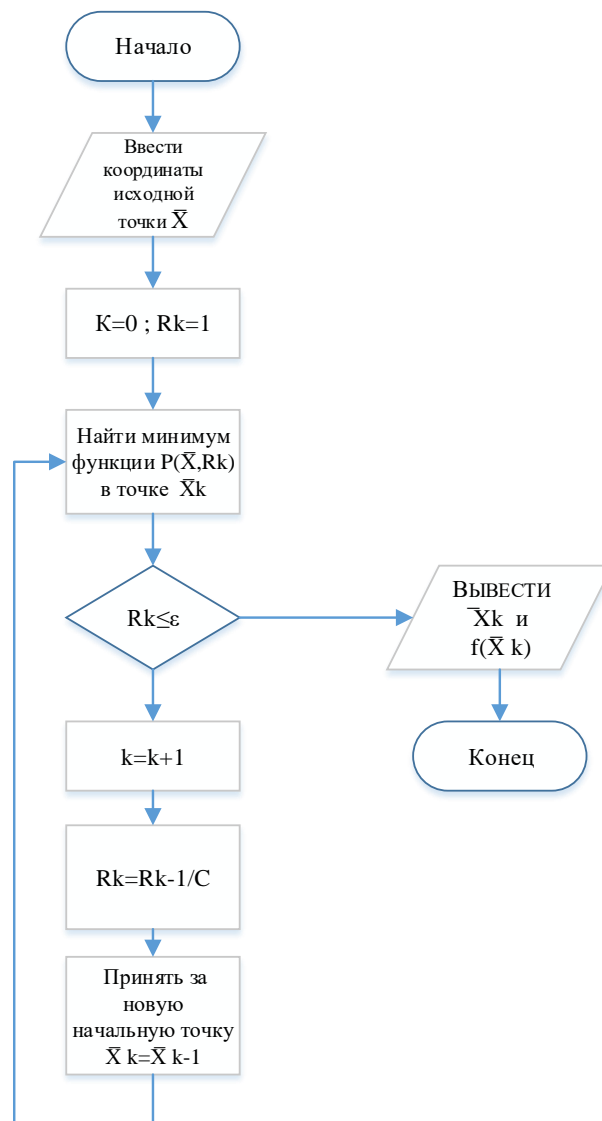


Рисунок 1 – Алгоритм метода штрафов

Описание применения

Программа «PenaltyMethod» предназначена для поиска минимума функции многих переменных методом штрафов.

Содержимое программы:

PenaltyMethod.exe – исполняемый файл. Является стартом программы.

Пример задачи

Функция: $f(x) = 1 + (x_1 - 6)^2 + (x_2 - 8)^2$

Ограничение: $g(x) = 4 - x_1 - x_2 \geq 0$

Квадратичный штраф

Преобразуем ограничение в равенство:

$$h(x) = 4 - x_1 - x_2 = 0$$

Используем квадратичный штраф. Введем штрафную функцию (ШФ) вида

$$P(x, R) = 1 + (x_1 - 6)^2 + (x_2 - 8)^2 + R(4 - x_1 - x_2)^2$$

Запишем уравнения, определяющие стационарную точку функции $P(x, R)$:

$$\frac{\partial P}{\partial x_1} = 2(x_1 - 6) - 2R(4 - x_1 - x_2) = 0$$

$$\frac{\partial P}{\partial x_2} = 2(x_2 - 8) - 2R(4 - x_1 - x_2) = 0$$

Выразим из первого уравнения

$$2R(4 - x_1 - x_2) = 2(x_1 - 6)$$

И подставим во второе уравнение:

$$2(x_2 - 8) - 2(x_1 - 6) = 2x_2 - 2x_1 - 4 = 0 \rightarrow x_2 = x_1 + 2$$

Тогда

$$2R(4 - x_1 - (x_1 + 2)) = 2(x_1 - 6) \rightarrow R(2 - 2x_1) = x_1 - 6 \rightarrow$$

$$2Rx_1 + x_1 - 2R - 6 = 0 \rightarrow x_1 = \frac{2R + 6}{2R + 1}$$

Переходя к пределу, получим

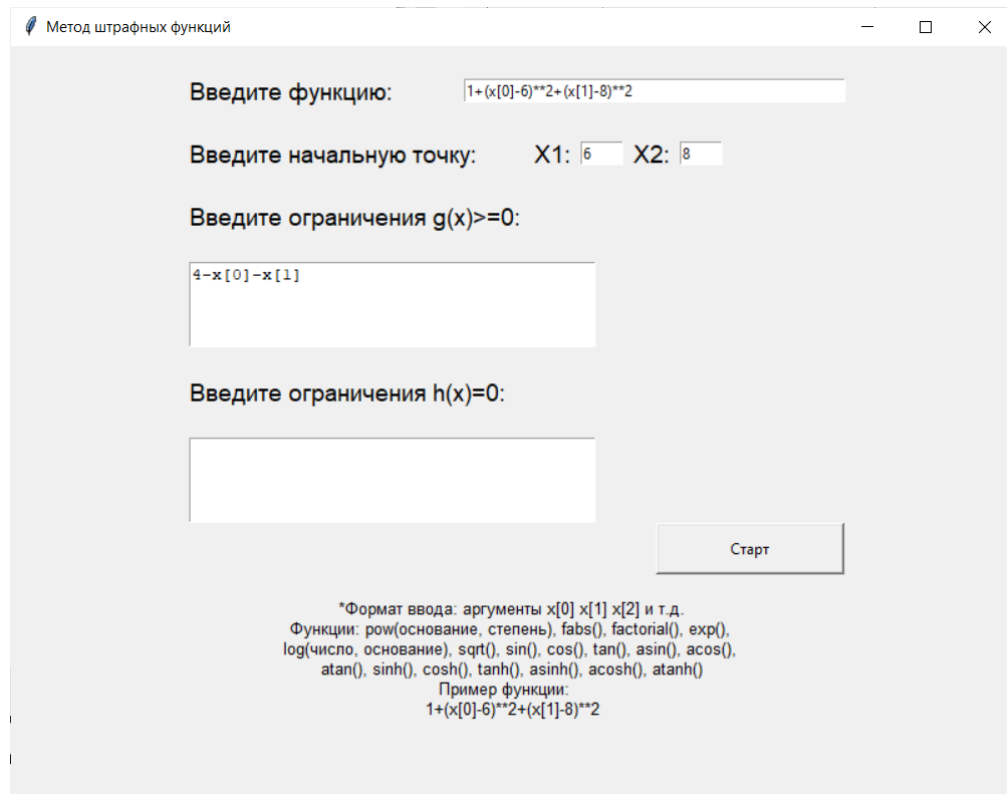
$$x_1 = \lim_{R \rightarrow \infty} \frac{2R + 6}{2R + 1} = 1$$

Соответственно, $x_2 = x_1 + 2 = 3$. Таким образом, метод сходится в точке $x^* = (1; 3)^T$ и $f^* = 51$.

Пример работы

Работа запускается через исполняемый файл PenaltyMethod.exe

После чего вводится уравнение



The screenshot shows a Windows application window titled "Метод штрафных функций". It contains several input fields and a "Старт" button. The first input field, labeled "Введите функцию:", contains the expression $1+(x[0]-6)^2+(x[1]-8)^2$. The second input field, labeled "Введите начальную точку:", has two sub-fields: "X1:" with the value 6 and "X2:" with the value 8. The third input field, labeled "Введите ограничения $g(x) \geq 0$:", contains the expression $4-x[0]-x[1]$. The fourth input field, labeled "Введите ограничения $h(x)=0$:", is empty. Below the input fields is a "Старт" button. At the bottom of the window, there is a block of text providing input format instructions, a list of supported functions, and an example function.

Метод штрафных функций

Введите функцию: $1+(x[0]-6)^2+(x[1]-8)^2$

Введите начальную точку: X1: 6 X2: 8

Введите ограничения $g(x) \geq 0$:

$4-x[0]-x[1]$

Введите ограничения $h(x)=0$:

Старт

*Формат ввода: аргументы $x[0]$ $x[1]$ $x[2]$ и т.д.
Функции: pow(основание, степень), fabs(), factorial(), exp(),
log(число, основание), sqrt(), sin(), cos(), tan(), asin(), acos(),
atan(), sinh(), cosh(), tanh(), asinh(), acosh(), atanh()
Пример функции:
 $1+(x[0]-6)^2+(x[1]-8)^2$

Рисунок 2 – Внешний вид программы

После нажатия на кнопку «Старт», появляется сообщение об ошибке или успешном выполнении операции. После чего создается файл result.txt в который будет помещено решение задачи.

```
result.txt – Блокнот
Файл Правка Формат Вид Справка

=====
Метод штрафов
=====

Функция: 1+(x[0]-6)**2+(x[1]-8)**2
Начальная точка: [6;8]
=====

Метод штрафа, итерация № 0
-----
X= [6, 8]

Метод Коши
-----
x = [6, 8]
Градиент = [1.1050000000000004, 1.1050000000000004]
Норма градиента = 1.5627059864222705

Метод Коши
-----
x = [5.545454545454545, 7.545454545454545]
Градиент = [0.10500000000000398, 0.10500000000000398]
Норма градиента = 0.14849242404918062

Метод Коши
-----
x = [5.545454545454546, 7.545454545454546]
Градиент = [1.0190909090908917, 1.0190909090908917]
Норма градиента = 1.441212184927466

Метод Коши
-----
x = [5.166666666666667, 7.166666666666667]
Градиент = [0.11000000000001009, 0.11000000000001009]
Норма градиента = 0.15556349186105473
r = 1.0

Метод штрафа, итерация № 1
-----
X= [5.166666666666667, 7.166666666666667]
```

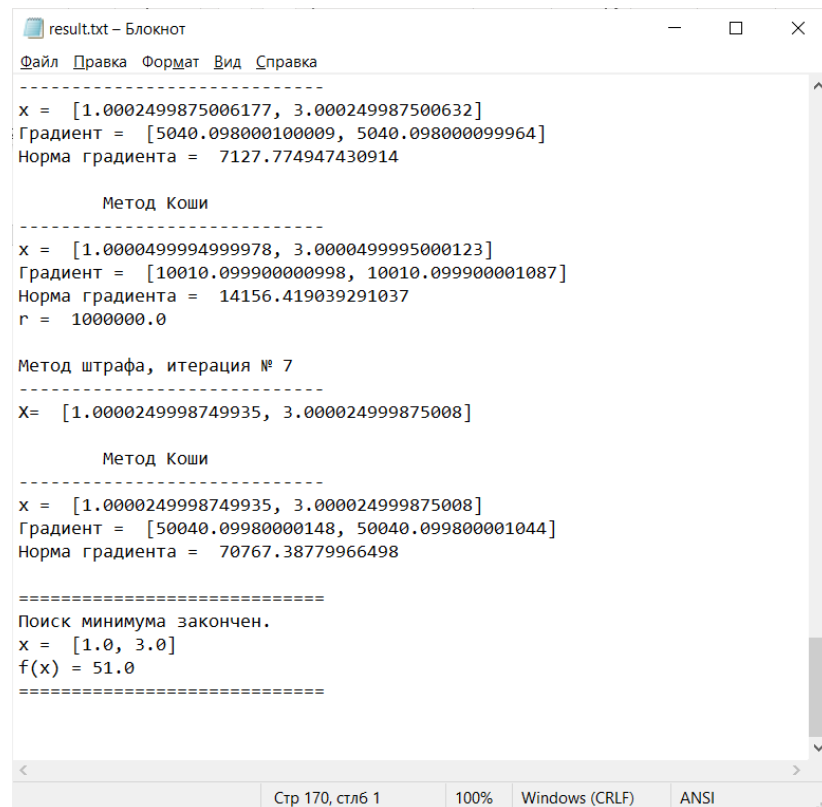
Рисунок 3 – Открытие файла с результатами.

Анализ результата.

Сравним результаты созданной программы с результатами лабораторной работы №4

Для сравнения используется ранее разобранный уравнение:

$$f(x) = 1 + (x_1 - 6)^2 + (x_2 - 8)^2$$



```
result.txt - Блокнот
Файл  Правка  Формат  Вид  Справка

-----
x = [1.0002499875006177, 3.000249987500632]
Градиент = [5040.098000100009, 5040.098000099964]
Норма градиента = 7127.774947430914

Метод Коши
-----
x = [1.0000499994999978, 3.0000499995000123]
Градиент = [10010.0999000000998, 10010.099900001087]
Норма градиента = 14156.419039291037
r = 1000000.0

Метод штрафа, итерация № 7
-----
x= [1.0000249998749935, 3.000024999875008]

Метод Коши
-----
x = [1.0000249998749935, 3.000024999875008]
Градиент = [50040.09980000148, 50040.099800001044]
Норма градиента = 70767.38779966498

=====
Поиск минимума закончен.
x = [1.0, 3.0]
f(x) = 51.0
=====

Стр 170, столб 1    100%    Windows (CRLF)    ANSI
```

Рисунок 4 – Результат вычисления программой «PenaltyMethod»

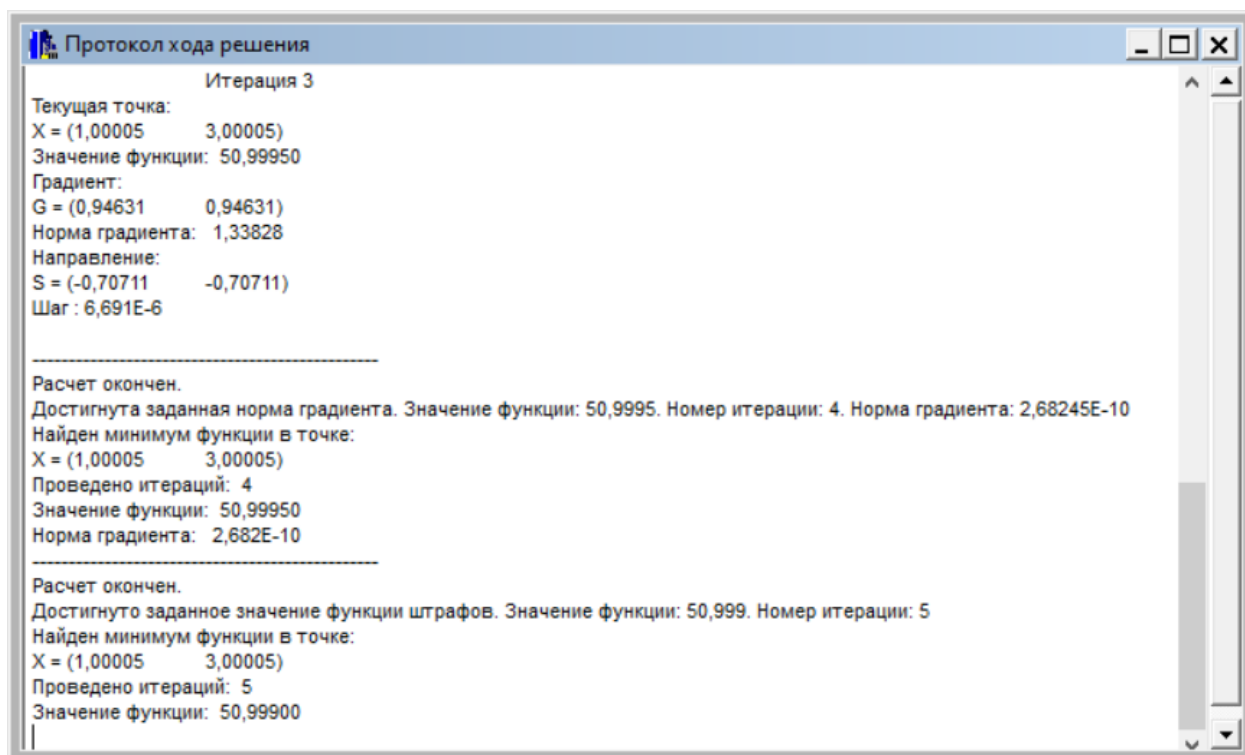


Рисунок 5 – Результат вычисления программой «ConditionalOptimization»

Возьмем результаты вычисления из лабораторной работа №4 и произведем анализ.

Метод штрафов				
	Кол-во итераций	Градиент	Минимум в точке	Значение функции
Результат ЛР4	5	2.682E-10	x1= 1.00005 x2= 3.00005	50.99900
Результат ИДЗ	8	[50040.09980000148, 50040.099800001044]	x1= 1.0 x2= 3.0	51.0
При верном ответе: x1= 1, x2= 3, f(x) = 51				

Как можно увидеть, программа нахождения точки минимума функции:

$$f(x) = 1 + (x_1 - 6)^2 + (x_2 - 8)^2,$$

реализованная в ИДЗ, выдает более точный, при этом выполняет на 3 итерации больше.

Вывод

При выполнении индивидуального домашнего задания (ИДЗ), были изучены различные методы оптимизации и реализован метод штрафов.

Приложение А

PenaltyMethod.py

```
from tkinter import *
from sympy import *
import sys
from tkinter import messagebox

def mydiff(f, x):
    i = 0
    result = []
    while i < len(x):
        x[i] = x[i] + 0.1
        temp = eval(f)
        x[i] = x[i] - 0.1
        temp2 = eval(f)
        i = i + 1
        result.append((temp - temp2) / 0.1)
    return result

def Cauchy(f, x):
    k = 0
    M = 100
    eps = 0.1
    a=[0] * (len(x) + 1)
    alfa = 0
    xk = []

    while k < M:
        print("\n\tМетод Коши")
        print("-----")
        print("x = ", x)
        xk = []
        grad = mydiff(f, x)
        print("Градиент = ", grad)
        summa = 0

        for element in grad:
            summa = summa + element**2
        normagr = summa**(1/2)
        print("Норма градиента = ", normagr)
        if normagr < eps:
            return x
        else:
            for i in range(1, len(x)):
                delta = 0.1
                a[i] = a[i-1] + delta

            fnew = []
            for i in range(len(x)):
```

```

xa = []
for j in range(len(x)):
    xa.append(x[j] - a[i] * grad[j])
temp = f
temp = temp.replace("x", "xa")
fnew.append(eval(temp))

maxi = 0
max = fnew[0]
for i in range(len(fnew)):
    if fnew[i] > max:
        max = fnew[i]
        maxi = i

a[-1] = a[maxi] + len(fnew) * delta
xa = []
for j in range(len(x)):
    xa.append(x[j] - a[-1] * grad[j])

temp = f
temp = temp.replace("x", "xa")
fnew.append(eval(temp))
temp = f
for i in range(len(x)):
    temp = temp.replace('x[' + str(i) + ']', '(' + str(eval(str(x[i]))) + ' - ' + str(grad[i]) + ')')

var = Symbol('a')
temp = simplify(temp)
temp = diff(temp, var)
step = eval(str(solve(temp, var)))
alfa = step[0]
for j in range(len(x)):
    xk.append(x[j] - alfa * grad[j])
arrayx = []

for i in range(len(x)):
    arrayx.append(xk[i] - x[i])
summa = 0

for element in arrayx:
    summa = summa + element**2
newnorm = summa**(1/2)

if newnorm < eps and abs (eval(f) - fnew[-1]) < eps:
    return xk
else:
    for i in range (len(xk)):
        if abs(xk[i] - x[i]) > eps:
            x = xk
            break

```

```

        else:
            return xk
    k = k + 1
return(x)

def penalty():
    f = txt.get()
    x = []
    x.append(int(txtx1.get()))
    x.append(int(txtx2.get()))
    hList = textBoxh.get(1.0, "end-1c")
    h = []

    for line in hList.split("\n"):
        if not line.strip():
            continue
        h.append(line.lstrip())

    gList = textBoxg.get(1.0, "end-1c")
    g = []

    for line in gList.split("\n"):
        if not line.strip():
            continue
        g.append(line.lstrip())

    C = 10
    r = 0.1
    M = 100
    eps = 0.0001
    k = 0
    while k < M:
        func = f
        print("Метод штрафа, итерация №", k)
        print("-----")
        print('X= ', x)

        str_h='('
        for element in h:
            str_h=str_h + '+' + '(' + element + ')**2'
        str_h=str_h + ')'
        if len(str_h) == 2:
            str_h = '0'

        str_g = '('
        for element in g:
            if eval(element) < 0:
                str_g = str_g + '+' + '(' + element + ')**2'
            str_g=str_g + ')'
        if len(str_g) == 2:
            str_g = '0'

```

```

P = str(eval(str(r/2))) + '*' + '(' + str_h + '+' + str_g + ')'

func = func + "+" + str(P)

x = Cauchy(func, x)

str_h='('
for element in h:
    str_h = str_h + '+' + '(' + element + ')**2'
str_h=str_h + ')'
if len(str_h) == 2:
    str_h = '0'

str_g = '('
for element in g:
    if eval(element) < 0:
        str_g = str_g + '+' + '(' + element + ')**2'
str_g=str_g + ')'
if len(str_g) == 2:
    str_g = '0'

P = str(eval(str(r/2))) + '*' + '(' + str_h + '+' + str_g + ')'

if eval(P) < eps:
    return [x, f]
else:
    func = func + "+" + P
    x = Cauchy(func, x)
    r = C * r
    print("r = ", r)
    print("")
    k = k + 1
return [x, f]

def clicked():
    try:
        orig_stdout = sys.stdout
        fl = open('result.txt', 'w')
        sys.stdout = fl
        print("\n=====")
        print("          Метод штрафов")
        print("=====\\n")
        print("Функция: ", txt.get())
        print("Начальная точка: [" + txtx1.get() + ";" + txtx2.get() + "]")
        print("=====\\n")
        temp = penalty()
        x = temp[0]
        f = temp[1]
        print("\n=====")
        print("Поиск минимума закончен.")
        for i in range (len(x)):

```

```
x[i]= abs(round(x[i], 4))
print("x = " , x )
print("f(x) = " + str(eval(f)))
print("=====\n")
sys.stdout = orig_stdout
fl.close()
messagebox.showinfo('Успех', 'Результат сохранился в файл result.txt')
except:
    messagebox.showerror('Ошибка', 'Неверные значения')

window = Tk()
window.geometry('800x600')
window.title("Метод штрафных функций")
frame1 = Frame(window)
frame1.pack()
frame2 = Frame(window)
frame2.pack()
frame3 = Frame(window)
frame3.pack()
frame4 = Frame(window)
frame4.pack()
frame5 = Frame(window)
frame5.pack()
frame6 = Frame(window)
frame6.pack()
frame7 = Frame(window)
frame7.pack()
frame8 = Frame(window)
frame8.pack()
lbl1 = Label(frame1, text="\n Введите функцию:\t\t\t\t\t\n", font=("Arial", 14))
lbl1.pack(side=LEFT)
txt = Entry(frame1,width=50)
txt.pack(side=LEFT)
lbl2 = Label(frame2, text="Введите начальную точку:\t", font=("Arial", 14))
lbl2.pack(side=LEFT)
lbl3 = Label(frame2, text=" X1: ", font=("Arial", 14))
lbl3.pack(side=LEFT)
txtx1 = Entry(frame2,width=5)
txtx1.pack(side=LEFT)
lbl4 = Label(frame2, text=" X2: ", font=("Arial", 14))
lbl4.pack(side=LEFT)
txtx2 = Entry(frame2,width=5)
txtx2.pack(side=LEFT)
lbl5 = Label(frame2, text="\t", font=("Arial", 14))
lbl5.pack(side=LEFT)
lbl6 = Label(frame3, text="\nВведите ограничения g(x)>=0:\t\t\t\t\t\n",
font=("Arial", 14))
lbl6.pack(side=LEFT)
textBoxg=Text(frame4, height=4, width=40)
textBoxg.pack(side=LEFT)
```

```
lbl7 = Label(frame4, text="\t\t ", font=("Arial", 14))
lbl7.pack(side=LEFT)
lbl8 = Label(frame5, text="\nВведите ограничения h(x)=0:\t\t\t\t\t\n",
font=("Arial", 14))
lbl8.pack(side=LEFT)
textBoxh=Text(frame6, height=4, width=40)
textBoxh.pack(side=LEFT)
lbl9 = Label(frame6, text="\t\t ", font=("Arial", 14))
lbl9.pack(side=LEFT)
lbl10 = Label(frame7, text="\t\t\t\t\t ", font=("Arial", 14))
lbl10.pack(side=LEFT)
btnRead=Button(frame7, height=2, width=20, text="Старт", command=clicked)
btnRead.pack(side=LEFT)
lbl11 = Label(frame8, text="\n*Формат ввода: аргументы x[0] x[1] x[2] и т.д.\nФунк-
ции: pow(основание, степень), fabs(), factorial(), exp(), \nlog(число, основание
), sqrt(), sin(), cos(), tan(), asin(), acos(), \natatan(), sinh(), cosh(), tanh(),
asinh(), acosh(), atanh()\nПример функции: \n1+(x[0]-6)**2+(x[1]-
8)**2", font=("Arial", 10))
lbl11.pack(side=LEFT)
window.mainloop()
```