**Чепоков E, Karavaev C**

(bloodiesmail@gmail.com)

**DOCUMENTATION SUR LE PROGRAMME QUI EXTRAIT LA RACINE**

**(POUR L'UTILISATEUR)**

Perm

2017

# CONTENU

1.Introduction;

2. Nom, champ d'application, but du développement;

3. Base de développement;

4. Spécifications du produit logiciel;

5. Étapes et étapes du développement, tests et erreurs;

6. Personnel et rémunération;

7. En contact avec nous.

# 1. INTRODUCTION

Nous voulons vous préparer moralement à ce produit logiciel, j'espère qu'il répondra en quelque sorte à vos attentes.

Le programme est développé sur la base du code qui nous a été donné lors d'une conférence.

Le produit logiciel en question a été développé sur la base d'une mission technique du Client, qui est également présentée ci-dessous avec des modifications mineures, et se compose d'un programme d'extraction de la racine.

# 2. NOM, CHAMP D'APPLICATION, BUT DU DÉVELOPPEMENT

Notre programme "Extractor" est utilisé par les écoliers et les étudiants pour extraire la racine dans le domaine de l'éducation, peut également être applicable aux utilisateurs paresseux qui ont déjà terminé l'apprentissage, mais la situation de la vie les a forcés à extraire la racine.

# 3. BASE DE DÉVELOPPEMENT

La raison du développement était le travail à domicile, délivré à la première année de NIU wshe.

# 4. SPÉCIFICATIONS DU PRODUIT LOGICIEL

1. Fonctionnalité: les racines ne sont pas seulement arithmétiques, à partir de zéro, complexes, nombres longs, précision prédéfinie, analytique (Une partie des exigences à notre regret ne sont pas remplies en raison de l'ignorance).

2. Erreur de stabilité

3. GUI (en tenant compte de la gestion de la précision et de la poly-langue)

4. Multilangue

5. Multiplateforme

# 5. ÉTAPES DU DÉVELOPPEMENT, TESTS ET ERREURS

Nos algorithmes seront utilisés pour un programme implémenté dans un environnement de développement pycharm dans un langage Python de niveau intermédiaire.

Code du programme:

1. from tkinter import \*
2. import tkinter
3. import tkinter.ttk
4. import math
5. def create\_widgets\_in\_first\_frame():
6. # Create the label for the frame
7. first\_window\_label = tkinter.ttk.Label(first\_frame,
   * + - 1. text='Choose Language')
8. first\_window\_label.grid(column=2,
   * 1. row=0,
     2. pady=10,
     3. padx=10,
     4. sticky=(tkinter.N))
9. # Create the button for the frame
10. create\_widgets\_in\_first\_frame.add\_img = tkinter.PhotoImage(file="1.png")
11. create\_widgets\_in\_second\_frame.add\_img = tkinter.PhotoImage(file="2.png")
12. create\_widgets\_in\_third\_frame.add\_img = tkinter.PhotoImage(file="3.png")
13. create\_widgets\_in\_fourth\_frame.add\_img = tkinter.PhotoImage(file="4.png")
14. first\_window\_next\_button = tkinter.Button(first\_frame,
    * + - 1. text="Русский",
          2. image=create\_widgets\_in\_first\_frame.add\_img,
          3. command=call\_second\_frame\_on\_top)
15. first\_window\_next\_button.grid(column=1,
    * + 1. row=1,
        2. pady=10,
        3. padx=10)
16. first\_window\_next\_button = tkinter.Button(first\_frame,
    * + - 1. text="Английский",
          2. image=create\_widgets\_in\_second\_frame.add\_img,
          3. command=call\_third\_frame\_on\_top)
17. first\_window\_next\_button.grid(column=1,
    * + 1. row=2,
        2. pady=10,
        3. padx=10)
18. first\_window\_next\_button = tkinter.Button(first\_frame,
    * + - 1. text="Китайский",
          2. image=create\_widgets\_in\_third\_frame.add\_img,
          3. command=call\_fourth\_frame\_on\_top)
19. first\_window\_next\_button.grid(column=3,
    * + 1. row=1,
        2. pady=10,
        3. padx=10)
20. first\_window\_next\_button = tkinter.Button(first\_frame,
    * + - 1. text="Французский",
          2. image=create\_widgets\_in\_fourth\_frame.add\_img,
          3. command=call\_fifth\_frame\_on\_top)
21. first\_window\_next\_button.grid(column=3,
    * + 1. row=2,
        2. pady=10,
        3. padx=10)
22. first\_window\_quit\_button = tkinter.Button(first\_frame,
    * + - 1. text="Exit",
          2. command=quit\_program)
23. first\_window\_quit\_button.grid(column=4,
    * + 1. row=3,
        2. pady=10,
        3. padx=10)
24. def create\_widgets\_in\_second\_frame():
25. # Create the label for the frame
26. second\_window\_label = tkinter.ttk.Label(second\_frame,
    * + - 1. text='Введите число:')
27. second\_window\_label.grid(column=1,
    * 1. row=0,
      2. pady=10,
      3. padx=10,
      4. sticky=(tkinter.N))
28. message = StringVar()
29. entry1 = tkinter.Entry(second\_frame, text='', textvariable=message, width=50,)
30. entry1.grid(column=2,
    * 1. row=0,
      2. pady=10,
      3. padx=10,
      4. sticky=(tkinter.N))
31. second\_window\_label = tkinter.ttk.Label(second\_frame,
    * + - 1. text='Точность:')
32. second\_window\_label.grid(column=1,
    * 1. row=1,
      2. pady=10,
      3. padx=10,
      4. sticky=(tkinter.N))
33. many = StringVar()
34. message\_entry = Entry(second\_frame, text='', textvariable=many, width=20, )
35. message\_entry.grid(column=2,
    * 1. row=1,
      2. pady=10,
      3. padx=10,
      4. sticky=(tkinter.N))
36. def func1():
37. try:
    1. x = float(entry1.get())
    2. a = int(message\_entry.get())
    3. second\_window\_label1.config(
       1. text=("{0:." + str(a) + "f}").format(
       2. math.sqrt(-x) if x < 0 else math.sqrt(x))
       3. + ("i" if x < 0 else ""))
38. except ValueError:
    1. second\_window\_label1.config(text="Ошибка введите цифры")
39. second\_window\_label1 = tkinter.Label(second\_frame,
    * + - 1. text="Ответ:")
40. second\_window\_label1.grid(column=2,
    * 1. row=3,
      2. pady=10,
      3. padx=10,
      4. sticky=(tkinter.N))
41. second\_window\_label = tkinter.ttk.Label(second\_frame,
    * + - 1. text="Ответ:")
42. second\_window\_label.grid(column=1,
    * 1. row=3,
      2. pady=10,
      3. padx=10,
      4. sticky=(tkinter.N))
43. # Create the button for the frame
44. second\_window\_enter\_button = tkinter.Button(second\_frame, text='Решить', command=func1)
45. second\_window\_enter\_button.grid(column=2,
    * + 1. row=2,
        2. pady=10,
        3. padx=10)
46. second\_window\_back\_button = tkinter.Button(second\_frame,

text="Назад",

command=call\_first\_frame\_on\_top)

1. second\_window\_back\_button.grid(column=0,
   * + 1. row=5,
       2. pady=10,
       3. padx=10)
2. second\_window\_next\_button = tkinter.Button(second\_frame,

text="Выход",

command=quit\_program)

1. second\_window\_next\_button.grid(column=5,
   * + 1. row=5,
       2. pady=10,
       3. padx=10)
2. def create\_widgets\_in\_third\_frame():
3. # Create the label for the frame
4. third\_window\_label = tkinter.ttk.Label(third\_frame,
   * + - 1. text='输入一个数字:')
5. third\_window\_label.grid(column=1,
   * + 1. row=0,
       2. pady=10,
       3. padx=10,
       4. sticky=(tkinter.N))
6. message = StringVar()
7. entry1 = tkinter.Entry(third\_frame, text='', textvariable=message, width=50, )
8. entry1.grid(column=2,
   * 1. row=0,
     2. pady=10,
     3. padx=10,
     4. sticky=(tkinter.N))
9. third\_window\_label = tkinter.ttk.Label(third\_frame,
   * + - 1. text='准确度:')
10. third\_window\_label.grid(column=1,
    * + 1. row=1,
        2. pady=10,
        3. padx=10,
        4. sticky=(tkinter.N))
11. many = StringVar()
12. message\_entry = Entry(third\_frame, text='', textvariable=many, width=20, )
13. message\_entry.grid(column=2,
    * 1. row=1,
      2. pady=10,
      3. padx=10,
      4. sticky=(tkinter.N))
14. def func1():
15. try:
    1. x = float(entry1.get())
    2. a = int(message\_entry.get())
    3. third\_window\_label1.config(
       1. text=("{0:." + str(a) + "f}").format(
       2. math.sqrt(-x) if x < 0 else math.sqrt(x))
       3. + ("i" if x < 0 else ""))
16. except ValueError:
    1. third\_window\_label1.config(text="错误输入数字")
17. third\_window\_label1 = tkinter.Label(third\_frame,
    * + - 1. text="回答:")
18. third\_window\_label1.grid(column=2,
    * + 1. row=3,
        2. pady=10,
        3. padx=10,
        4. sticky=(tkinter.N))
19. third\_window\_label = tkinter.ttk.Label(third\_frame,
    * + - 1. text="回答:")
20. third\_window\_label.grid(column=1,
    * + 1. row=3,
        2. pady=10,
        3. padx=10,
        4. sticky=(tkinter.N))
21. # Create the button for the frame
22. third\_window\_enter\_button = tkinter.Button(third\_frame, text='解决', command=func1)
23. third\_window\_enter\_button.grid(column=2,
    * + 1. row=2,
        2. pady=10,
        3. padx=10)
24. # Create the button for the frame
25. third\_window\_back\_button = tkinter.Button(third\_frame,
    * + - 1. text="向后",
          2. command=call\_first\_frame\_on\_top)
26. third\_window\_back\_button.grid(column=0,
    * + 1. row=5,
        2. pady=10,
        3. padx=10,
        4. sticky=(tkinter.N))
27. third\_window\_quit\_button = tkinter.Button(third\_frame,
    * + - 1. text="输出",
          2. command = quit\_program)
28. third\_window\_quit\_button.grid(column=5,
    * + 1. row=5,
        2. pady=10,
        3. padx=10,
        4. sticky=(tkinter.N))
29. def create\_widgets\_in\_fourth\_frame():
30. # Create the label for the frame
31. fourth\_window\_label = tkinter.ttk.Label(fourth\_frame,
    * + - 1. text='Enter a number:')
32. fourth\_window\_label.grid(column=1,
    * + 1. row=0,
        2. pady=10,
        3. padx=10,
        4. sticky=(tkinter.N))
33. message = StringVar()
34. entry1 = tkinter.Entry(fourth\_frame, text='', textvariable=message, width=50, )
35. entry1.grid(column=2,
    * 1. row=0,
      2. pady=10,
      3. padx=10,
      4. sticky=(tkinter.N))
36. fourth\_window\_label = tkinter.ttk.Label(fourth\_frame,
    * + - 1. text='Accuracy:')
37. fourth\_window\_label.grid(column=1,
    * + 1. row=1,
        2. pady=10,
        3. padx=10,
        4. sticky=(tkinter.N))
38. many = StringVar()
39. message\_entry = Entry(fourth\_frame, text='', textvariable=many, width=20, )
40. message\_entry.grid(column=2,
    * 1. row=1,
      2. pady=10,
      3. padx=10,
      4. sticky=(tkinter.N))
41. def func1():
42. try:
    1. x = float(entry1.get())
    2. a = int(message\_entry.get())
    3. fourth\_window\_label1.config(
       1. text=("{0:." + str(a) + "f}").format(
       2. math.sqrt(-x) if x < 0 else math.sqrt(x))
       3. + ("i" if x < 0 else ""))
43. except ValueError:
    1. fourth\_window\_label1.config(text="Error enter numbers")
44. fourth\_window\_label1 = tkinter.Label(fourth\_frame,
    * + - 1. text="Answer:")
45. fourth\_window\_label1.grid(column=2,
    * + 1. row=3,
        2. pady=10,
        3. padx=10,
        4. sticky=(tkinter.N))
46. fourth\_window\_label = tkinter.ttk.Label(fourth\_frame,
    * + - 1. text="Answer:")
47. fourth\_window\_label.grid(column=1,
    * + 1. row=3,
        2. pady=10,
        3. padx=10,
        4. sticky=(tkinter.N))
48. # Create the button for the frame
49. fourth\_window\_enter\_button = tkinter.Button(fourth\_frame, text='Solve', command=func1)
50. fourth\_window\_enter\_button.grid(column=2,
    * + 1. row=2,
        2. pady=10,
        3. padx=10)
51. # Create the button for the frame
52. fourth\_window\_back\_button = tkinter.Button(fourth\_frame,

text="Back",

command=call\_first\_frame\_on\_top)

1. fourth\_window\_back\_button.grid(column=0,
   * + 1. row=5,
       2. pady=10,
       3. padx=10,
       4. sticky=(tkinter.N))
2. fourth\_window\_quit\_button = tkinter.Button(fourth\_frame,

text="Exit",

command=quit\_program)

1. fourth\_window\_quit\_button.grid(column=5,
   * + 1. row=5,
       2. pady=10,
       3. padx=10,
       4. sticky=(tkinter.N))
2. def create\_widgets\_in\_fifth\_frame():
3. # Create the label for the frame
4. fifth\_window\_label = tkinter.ttk.Label(fifth\_frame,
   * + - 1. text='Entrez un nombre:')
5. fifth\_window\_label.grid(column=1,
   * + 1. row=0,
       2. pady=10,
       3. padx=10,
       4. sticky=(tkinter.N))
6. message = StringVar()
7. entry1 = tkinter.Entry(fifth\_frame, text='', textvariable=message, width=50, )
8. entry1.grid(column=2,
   * 1. row=0,
     2. pady=10,
     3. padx=10,
     4. sticky=(tkinter.N))
9. fifth\_window\_label = tkinter.ttk.Label(fifth\_frame,
   * + - 1. text='Précision:')
10. fifth\_window\_label.grid(column=1,
    * + 1. row=1,
        2. pady=10,
        3. padx=10,
        4. sticky=(tkinter.N))
11. many = StringVar()
12. message\_entry = Entry(fifth\_frame, text='', textvariable=many, width=20, )
13. message\_entry.grid(column=2,
    * 1. row=1,
      2. pady=10,
      3. padx=10,
      4. sticky=(tkinter.N))
14. def func1():
15. try:
    1. x = float(entry1.get())
    2. a = int(message\_entry.get())
    3. fifth\_window\_label1.config(
       1. text=("{0:." + str(a) + "f}").format(
       2. math.sqrt(-x) if x < 0 else math.sqrt(x))
       3. + ("i" if x < 0 else ""))
16. except ValueError:
    1. fifth\_window\_label1.config(text="Erreur entrez les chiffres")
17. fifth\_window\_label1 = tkinter.Label(fifth\_frame,
    * + - 1. text="Réponse:")
18. fifth\_window\_label1.grid(column=2,
    * + 1. row=3,
        2. pady=10,
        3. padx=10,
        4. sticky=(tkinter.N))
19. fifth\_window\_label = tkinter.ttk.Label(fifth\_frame,
    * + - 1. text="Réponse:")
20. fifth\_window\_label.grid(column=1,
    * + 1. row=3,
        2. pady=10,
        3. padx=10,
        4. sticky=(tkinter.N))
21. # Create the button for the frame
22. fifth\_window\_enter\_button = tkinter.Button(fifth\_frame, text='Résoudre', command=func1)
23. fifth\_window\_enter\_button.grid(column=2,
    * + 1. row=2,
        2. pady=10,
        3. padx=10)
24. # Create the button for the frame
25. fifth\_window\_back\_button = tkinter.Button(fifth\_frame,
    * + - 1. text="Retourner",
          2. command=call\_first\_frame\_on\_top)
26. fifth\_window\_back\_button.grid(column=0,
    * + 1. row=5,
        2. pady=10,
        3. padx=10,
        4. sticky=(tkinter.N))
27. fifth\_window\_quit\_button = tkinter.Button(fifth\_frame,
    * + - 1. text="Sortir",
          2. command=quit\_program)
28. fifth\_window\_quit\_button.grid(column=5,
    * + 1. row=5,
        2. pady=10,
        3. padx=10,
        4. sticky=(tkinter.N))
29. def call\_first\_frame\_on\_top():
30. # This function can be called only from the second window.
31. # Hide the second window and show the first window.
32. second\_frame.place\_forget()
33. third\_frame.place\_forget()
34. fourth\_frame.place\_forget()
35. fifth\_frame.place\_forget()
36. first\_frame.place(relx=0.1, rely=0.1)
37. def call\_second\_frame\_on\_top():
38. # This function can be called from the first and third windows.
39. # Hide the first and third windows and show the second window.
40. first\_frame.place\_forget()
41. second\_frame.place(relx=0.1, rely=0.1)
42. def call\_third\_frame\_on\_top():
43. # This function can only be called from the second window.
44. # Hide the second window and show the third window.
45. first\_frame.place\_forget()
46. third\_frame.place(relx=0.1, rely=0.1)
47. def call\_fourth\_frame\_on\_top():
48. # This function can only be called from the second window.
49. # Hide the second window and show the third window.
50. first\_frame.place\_forget()
51. fourth\_frame.place(relx=0.1, rely=0.1)
52. def call\_fifth\_frame\_on\_top():
53. # This function can only be called from the second window.
54. # Hide the second window and show the third window.
55. first\_frame.place\_forget()
56. fifth\_frame.place(relx=0.1, rely=0.1)
57. def quit\_program():
58. root\_window.destroy()
59. ###############################
60. # Main program starts here :) #
61. ###############################
62. # Create the root GUI window.
63. root\_window = tkinter.Tk()
64. root\_window.title("Калькулятор квадратов")
65. root\_window.geometry("700x400")
66. root\_window.resizable(False, False)
67. # Create frames inside the root window to hold other GUI elements. All frames must be created in the main program, otherwise they are not accessible in functions.
68. first\_frame = tkinter.ttk.Frame(root\_window, width=750, height=450+300+200)
69. first\_frame.place(relx=0.1, rely=0.1)
70. second\_frame = tkinter.ttk.Frame(root\_window, width=750, height=450+300+200)
71. second\_frame.place(relx=0.1, rely=0.1)
72. third\_frame = tkinter.ttk.Frame(root\_window, width=750, height=450+300+200)
73. third\_frame.place(relx=0.1, rely=0.1)
74. fourth\_frame = tkinter.ttk.Frame(root\_window, width=750, height=450+300+200)
75. fourth\_frame.place(relx=0.1, rely=0.1)
76. fifth\_frame = tkinter.ttk.Frame(root\_window, width=750, height=450+300+200)
77. fifth\_frame.place(relx=0.1, rely=0.1)
78. # Create all widgets to all frames
79. create\_widgets\_in\_first\_frame()
80. create\_widgets\_in\_second\_frame()
81. create\_widgets\_in\_third\_frame()
82. create\_widgets\_in\_fourth\_frame()
83. create\_widgets\_in\_fifth\_frame()
84. # Hide all frames in reverse order, but leave first frame visible (unhidden).
85. second\_frame.place\_forget()
86. third\_frame.place\_forget()
87. fourth\_frame.place\_forget()
88. fifth\_frame.place\_forget()
89. # Start tkinter event - loop
90. root\_window.mainloop()
91. Application.EnableVisualStyles()
92. Application.SetCompatibleTextRenderingDefault(False)
93. form = MyForm()
94. Application.Run(form)

# 6. PERSONNEL ET RÉMUNÉRATION

1. Chepokov Elizar programmeur-90h sur le développement du programme, 6h sur le développement du site, 50% du paiement

2. Karavaev Alexander-5h sur le développement du site, 3 heures sur le développement de la documentation, 50% du paiement

# 7. CONTACT AVEC NOUS

Notre site: http:// bloodiesproject.gq

Thelephone: **+79638739767**