



WYDZIAŁ ELEKTRONIKI,
TELEKOMUNIKACJI
I INFORMATYKI

Dokumentacja Projektu grupowego

Dokumentacja techniczna projektu

Wydział Elektroniki, Telekomunikacji i Informatyki

Politechnika Gdańska

Nazwa i akronim projektu: <i>Kalkulator centylowy - KC</i>	Zlecniodawca: <i>dr. inż. Barbara Stawarz-Graczyk</i>	
Numer zlecenia: <i>14@KMIO'2023</i>	Kierownik projektu: <i>Aleksandra Rykowska</i>	Opiekun projektu: <i>dr. inż. Barbara Stawarz-Graczyk</i>

Nazwa / kod dokumentu: Dokumentacja techniczna produktu – DTP	Nr wersji: <i>1.20</i>
Odpowiedzialny za dokument: <i>Piąstka Aleksandra Szarafiński Radosław</i>	Data pierwszego sporządzenia: <i>04.01.2022r.</i>
	Data ostatniej aktualizacji: <i>07.01.2022r.</i>
	Semestr realizacji Projektu grupowego: <i>1</i>

Historia dokumentu

Wersja	Opis modyfikacji	Rozdział / strona	Autor modyfikacji	Data
1.00	Wstępna wersja	całość	Piąstka Aleksandra Szarafiński Radosław	04.01.2022r.
1.10	Poprawka w pkt. 2	pkt. 2	Piąstka Aleksandra Szarafiński Radosław	06.01.2022r.
1.20	Poprawka w pkt. 2.2 i 2.3	pkt 2.2 i 2.3	Piąstka Aleksandra	07.01.2022r.

{UWAGA: w II semestrze dokumentacja może być rozszerzeniem dokumentacji z semestru I (nowa wersja dokumentu), może być też nowym plikiem}

Spis treści

1	Wprowadzenie - o dokumencie.....	3
1.1	Cel dokumentu	3
1.2	Zakres dokumentu.....	3
1.3	Odbiorcy	3
1.4	Terminologia	3
2	Dokumentacja techniczna projektu	3

1 Wprowadzenie - o dokumencie

1.1 Cel dokumentu

Celem dokumentu jest udokumentowanie informacji dotyczących produktu, jego cech funkcjonalnych, parametrów technicznych, schematów blokowych, oprogramowania, wyników działania, zdjęć produktu, pomiarów, testów oraz innych elementów wymaganych przez opiekuna i klienta.

1.2 Zakres dokumentu

W zakres dokumentu wchodzi informacje dotyczące obecnej wersji produktu, opis jego cech funkcjonalnych, parametrów technicznych, oprogramowania oraz wyniki działania (zdjęcia produktu, testy). W dokumencie nie są uwzględnione rozważania teoretyczne tzn. rozeznanie rynku, analiza ryzyka itp.

1.3 Odbiorcy

zleceniobiorca: Katedra Metrologii i Optoelektroniki

zleceniodawca: dr. inż. Barbara Stawarz-Graczyk

członkowie zespołu projektowego:

Aleksandra Rykowska

Magdalena Podlińska

Kamil Myćka

Aleksandra Piąstka

Radosław Szarafiński

1.4 Terminologia

Brak terminologii wymagającej objaśnienia.

2 Dokumentacja techniczna projektu

2.1 Cechy funkcjonalne

2.1.1 Prosty interfejs, który umożliwia wprowadzenie danych o dziecku.

2.1.2 Możliwość zapisania tych danych do pliku.

2.1.3 Wykreślenie wykresów na podstawie tych danych.

2.2 Parametry techniczne

2.2.1 Wymagany system operacyjny:

- Linux
- Windows
- MacOS

2.2.2 Wymagane oprogramowanie:

- Python3

2.2.3 Wymagania sprzętowe:

- karta graficzna (nie jest to aplikacja konsolowa, wymagane jest jakiekolwiek środowisko graficzne)

2.2.4 Wymagane miejsce na dysku:

- 1 GB pamięci masowej (obecnie program zajmuje 12,1 MB, nie przewiduje się, aby przekroczył 1 GB)

2.2.5 Licencja

- GPLv2

2.2.6 Język

- Polski
- W planach dodanie języka angielskiego

2.3 Oprogramowanie

2.3.1 Python3

2.3.2 Biblioteki:

- tkinter
- matplotlib

2.3.3 Wykorzystane funkcje:

- Funkcja wykonywana po kliknięciu przycisku. W przypadku wprowadzenia błędnych danych wyświetli komunikat proszący o wprowadzenie poprawnych danych. Jeśli jednak pole wieku pozostanie puste, nie zostanie to uznane za błąd, a program sam wpisze miesiąc o 1 większy niż wprowadzony poprzednio. Jeśli wprowadzone dane będą poprawne zostaną one dodane do pliku dane.txt, a przycisk zostanie wyłączony, żeby uniknąć błędnych, wtórnych kliknięć ze strony użytkownika. Wyświetli się również komunikat o tym, że dane zostały poprawnie zapisane.

```
def button_click():
    error_label = tk.Label(main, text="Wprowadź poprawne dane!",
fg="red")
    # if any of the inputs is empty or not a number than show error
message
    if is_empty_or_not_number():
        error_label.grid(row=12, column=0, columnspan=2)
        # hide window after 3 seconds
        main.after(3000, error_label.destroy)
    else:
        # if age is empty then increase previous age from data.txt file
by 1
        if age_input.get() == "":
            with open("data.txt", "r") as file:
                data = file.read()
                data = data.split("\n")
                data_list = []
                for i in data:
                    data_list.append(i.split(","))
                age_input.insert(0, str(int(data_list[-1][1]) + 1))
        # disable button after click
        input_button['state'] = 'disabled'
        # hide window after 3 seconds
        main.after(3000, error_label.destroy)
        success_label = tk.Label(main, text="Dane zostały zapisane!",
fg="green")
        success_label.grid(row=12, column=0, columnspan=2)
        # hide window after 3 seconds
        main.after(3000, success_label.destroy)
        # append data to file
        append_data()
        create_charts(main)
```

- Funkcja dodająca dane do pliku tekstowego w formacie: płeć, wiek, wzrost, waga, obwód głowy. Każdy wiersz to oddzielny rekord.

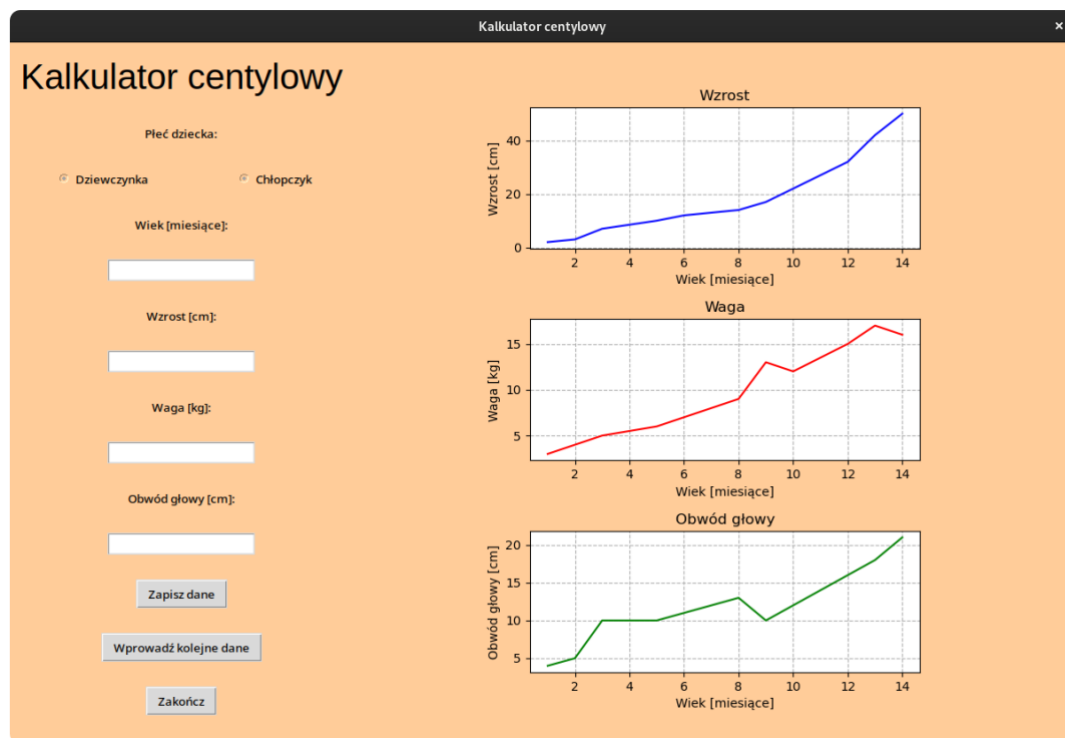
```
def append_data():
    with open("data.txt", "a") as file:
        file.write(
            f"\n{selected_gender.get()}, {age_input.get()},
{height_input.get()}, {weight_input.get()}, "
            f" {head_input.get()}")
```

- Funkcja odpowiadająca za rysowanie wykresów. Zaczyna ona od wczytania zawartości pliku "dane.txt" i zapisania jego zawartości do dwuwymiarowej listy, oddzielając parametry znakiem ",". Każdy wiersz odpowiada innemu rekordowi. Następnie na podstawie tych danych generowane są wykresy, na których oś x oznacza wiek dziecka, a oś y wzrostowi, wadze albo obwodzie głowy, w zależności od wykresu. Każdy wykres posiada swoją własną legendę oraz kolor, w celu łatwiejszego rozróżnienia.

```
def draw_plot():  
    # read data from file  
    with open("data.txt", "r") as file:  
        data = file.read()  
  
    # split data by new line  
    data = data.split("\n")  
  
    # create list from provided data  
    data_list = []  
    for i in data:  
        data_list.append(i.split(","))  
  
    # create lists for each  
    age = []  
    height = []  
    weight = []  
    head = []  
  
    # fill lists with data  
    for i in range(len(data_list)):  
        age.append(int(data_list[i][1]))  
        height.append(int(data_list[i][2]))  
        weight.append(int(data_list[i][3]))  
        head.append(int(data_list[i][4]))  
  
    # create 3 plots  
    # (3, 1, 1) means 3 rows, 1 column, 1st plot  
    fig = plt.figure()  
    ax1 = fig.add_subplot(311)  
    ax2 = fig.add_subplot(312)  
    ax3 = fig.add_subplot(313)  
  
    # plot age and height  
    ax1.plot(age, height, color="blue")  
    ax1.grid(linestyle="--")  
    ax1.set_title("Wzrost")  
    ax1.set_xlabel("Wiek [miesiące]")  
    ax1.set_ylabel("Wzrost [cm]")  
  
    # plot age and weight  
    ax2.plot(age, weight, color="red")  
    ax2.grid(linestyle="--")  
    ax2.set_title("Waga")  
    ax2.set_xlabel("Wiek [miesiące]")  
    ax2.set_ylabel("Waga [kg]")  
  
    # plot age and head circumference  
    ax3.plot(age, head, color="green")  
    ax3.grid(linestyle="--")  
    ax3.set_title("Obwód głowy")  
    ax3.set_xlabel("Wiek [miesiące]")  
    ax3.set_ylabel("Obwód głowy [cm]")  
  
    # improve layout  
    plt.tight_layout()  
    # increase plot height  
    fig.set_figheight(8)  
    # decrease plot width  
    fig.set_figwidth(5)  
    # decrease space between plots  
    fig.subplots_adjust(hspace=0.5)  
  
    # change background color to the same as main window  
    fig.patch.set_facecolor(global_variables.background_color)  
  
    # draw widget  
    canvas = FigureCanvasTkAgg(fig, main)  
    canvas.draw()  
  
    # move plot to the right  
    canvas.get_tk_widget().grid(row=0, column=6, rowspan=20, padx=150,  
                                pady=15) # possible improvement
```

2.4 Wyniki działania aplikacji

2.4.1 Zdjęcie produktu



2.4.2 Wyniki pomiarów, testów

```
growth-chart-calculator - data.txt
File Edit View Navigate Code Refactor Run Tools Git Window Help
growth-chart-calculator data.txt
Project
growth-chart-calculator ~/HDD/Projekty/grc
venv
chart.py
data.txt
global_variables.py
LICENSE
main.py
README.md
test.py
todo.md
ui.py
External Libraries
Scratches and Consoles
8 boy, 1, 2, 3, 4
9 boy, 2, 3, 4, 5
7 boy, 3, 7, 5, 10
6 boy, 5, 10, 6, 10
5 boy, 6, 12, 7, 11
4 boy, 8, 14, 9, 13
3 boy, 8, 14, 9, 13
2 boy, 9, 17, 13, 10
1 boy, 10, 22, 12, 12
10 boy, 12, 32, 15, 16
1 boy, 13, 42, 17, 18
2 boy, 14, 50, 16, 21
3 boy, 15, 51, 20, 30
Run: main x
/usr/bin/python3 /home/kamil/HDD/Projekty/growth-chart-calculator/main.py
10:19 LF UTF-8 4 spaces Python 3.11 P main
```

