

Konfigurator sprzętu PC

Skład: Robert Baca, Arkadiusz Bodziony, Wiktor Ciskał, Mikołaj Śnieżko

Projekt wykonany w semestrze letnim roku akademickiego 2023/2024

1 Wprowadzenie

Celem tego projektu jest automatyczne pobieranie i analiza danych dotyczących cen, wydajności oraz kompatybilności ze sobą komponentów komputerowych takich jak CPU, GPU, HDD, RAM oraz płyty główne. Program pobiera dane z różnych stron internetowych, przetwarza je i prezentuje w formie tekstowej, dzięki czemu pozwala użytkownikowi wybrać konkretne komponenty oraz poznać ogólną ocenę zestawienia.

2 Przygotowanie danych

Dane są pobierane z następujących stron:

- https://www.cpubenchmark.net/cpu_value_available.html
- https://www.videocardbenchmark.net/gpu_value.html
- https://www.harddrivebenchmark.net/hdd_value.html
- <https://www.memorybenchmark.net/popular.html>
- <https://versus.com/en/motherboard>
- <https://www.money.pl/pieniadze/nbp/srednie/>

3 Importowanie niezbędnych modułów

```
1 import requests
2 from bs4 import BeautifulSoup
3 import tkinter as tk
4 from tkinter import ttk
5 from PIL import Image, ImageTk
6 import io
```

Ta sekcja importuje wszystkie niezbędne biblioteki:

- 'requests' do wysyłania zapytań HTTP
- 'BeautifulSoup' do parsowania HTML
- 'tkinter' do tworzenia GUI
- 'PIL' (Pillow) do manipulacji obrazkami

4 Funkcja 'scrapedata'

```
1 def scrape_data(url):
2     response = requests.get(url)
3     soup = BeautifulSoup(response.text, 'html.parser')
4     return soup
```

Funkcja scrapedata jest odpowiedzialna za pobieranie i parsowanie danych HTML z podanego URL. Używa requests do pobrania treści strony i BeautifulSoup do parsowania HTML.

5 Pobieranie danych o CPU

```
1 # Zapytanie dla CPU
2 soup = scrape_data('https://www.cpubenchmark.net/cpu_value_available.html')
3 cpu = [i.text for i in soup.find_all('span', class_="prdname")]
4 cpu_prices = [float(i.text.replace('$', '').replace(',','').replace(' ','')) for i in soup.find_all(
5     'span', class_="price-neww")]
6 cpu_scores = [round(float(i.text.replace(',','').replace(' ','')) / max([float(i.text.replace(',',''.
7     '.')) for i in soup.find_all('span', class_="mark-neww")])), 2) for i in soup.
8     find_all('span', class_="mark-neww")]
9 cpu_prices = dict(zip(cpu, cpu_prices))
10 cpu_scores = dict(zip(cpu, cpu_scores))
```

Ta sekcja pobiera i przetwarza dane o procesorach (CPU) z serwisu cpubenchmark.net. Zapisuje nazwy, ceny i oceny procesorów w odpowiednich strukturach danych (listach i słownikach).

6 Pobieranie danych o GPU

```
1 # Zapytanie dla GPU
2 soup = scrape_data('https://www.videocardbenchmark.net/gpu_value.html')
3 gpu = [i.text for i in soup.find_all('span', class_="prdname")]
4 gpu_prices = [float(i.text.replace('$', '').replace(',','').replace(' ','').replace('*','')) for i
5     in soup.find_all('span', class_="price-neww")]
6 gpu_scores = [round(float(i.text.replace(',','').replace(' ','')) / max([float(i.text.replace(',',''.
7     '.')) for i in soup.find_all('span', class_="mark-neww")])), 2) for i in soup.
8     find_all('span', class_="mark-neww")]
9 gpu_prices = dict(zip(gpu, gpu_prices))
10 gpu_scores = dict(zip(gpu, gpu_scores))
```

Ta sekcja działa analogicznie do poprzedniej, ale dotyczy kart graficznych (GPU). Pobiera i przetwarza dane o nazwach, cenach i ocenach GPU.

7 Pobieranie danych o HDD/SSD

```
1 # Zapytanie dla HDD
2 soup = scrape_data('https://www.harddrivebenchmark.net/hdd_value.html')
3 hdd = [i.text for i in soup.find_all('span', class_="prdname")]
4 hdd_prices = [float(i.text.replace('$', '').replace(',','').replace(' ','').replace('*','')) for i
5     in soup.find_all('span', class_="price-neww")]
6 hdd_scores = [round(float(i.text.replace(',','').replace(' ','')) / max([float(i.text.replace(',',''.
7     '.')) for i in soup.find_all('span', class_="mark-neww")])), 2) for i in soup.
8     find_all('span', class_="mark-neww")]
9 hdd_prices = dict(zip(hdd, hdd_prices))
10 hdd_scores = dict(zip(hdd, hdd_scores))
```

Sekcja pobiera i przetwarza dane o dyskach twardych (HDD) oraz dyskach półprzewodnikowych (SSD), podobnie jak w przypadku CPU i GPU. Zawiera nazwy, ceny i oceny dysków.

8 Pobieranie danych o RAM

```
1 # Zapytanie dla RAM
2 soup = scrape_data('https://www.memorybenchmark.net/popular.html')
3 ram = [i.text for i in soup.find_all('span', class_="prcname")]
4 ram_prices = [float(i.text.replace(',', '').replace('NA', '0')) for i in soup.
5               find_all('span', class_="price-neww")]
6 ram_scores = [float(i.text.replace(' %', '')) * 10.0 for i in soup.find_all('span',
7                                     class_="count")]
8 ram_data = list(zip(ram, ram_prices, ram_scores))
9 ram_data = [(name, price, score) for name, price, score in ram_data if price > 0]
10 max_ram_score = max(ram_data, key=lambda x: x[2])[2]
11 ram = [data[0] for data in ram_data]
12 ram_prices = {data[0]: data[1] for data in ram_data}
13 ram_scores = {data[0]: data[2] / max_ram_score for data in ram_data}
```

Ta sekcja pobiera i przetwarza dane o pamięciach RAM, w tym nazwy, ceny i oceny. Zawiera dodatkowe kroki przetwarzania, aby usunąć rekordy z ceną równą zero i znormalizować oceny.

9 Pobieranie danych o płytach głównych

```
1 # Zapytanie dla płyt głównych
2 soup = scrape_data('https://versus.com/en/motherboard')
3 mother_board = [i.text for i in soup.find_all('p', class_="BarItem_name___3EC0w")]
4 mother_board_prices = [float(i.text[-5:].replace(',', '')) * 100 if len(i.text) > 0
5                          else None for i in soup.find_all('div', class_="BarItem_price___3dk0c")]
6 mother_board_scores = [round(float(i.text.replace('points', '')) / max([float(i.text.
7                                replace('points', '')) for i in soup.find_all('span', class_="pointsText")])) *
8                          0.3, 3) for i in soup.find_all('span', class_="pointsText")]
9 mother_board_data = list(zip(mother_board, mother_board_prices, mother_board_scores))
10 mother_board_data = [(name, price, score) for name, price, score in mother_board_data
11                       ]
12 mother_board = [data[0] for data in mother_board_data]
13 mother_board_prices = {data[0]: data[1] for data in mother_board_data}
14 mother_board_scores = {data[0]: data[2] for data in mother_board_data}
```

Ta sekcja pobiera i przetwarza dane o płytach głównych, w tym nazwy, ceny i oceny. Wartości cen są przeliczane na podstawie odpowiednich formatów.

10 Pobieranie danych walutowych

```
1 # Pobieranie kursów walut
2 soup = scrape_data('https://www.money.pl/pieniadze/nbp/srednie/')
3 site = soup.find_all('div', class_="rt-td")
4 currencies = []
5 values = []
6 for i in range(1, len(site), 5):
7     currencies.append(site[i].text)
8 for i in range(2, len(site), 5):
9     values.append(site[i].text)
10 values = [float(i.replace(',', '')) for i in values]
11 currencies.insert(0, "PLN")
12 values.insert(0, 1)
13 exchange_rates = dict(zip(currencies, values))
```

Sekcja ta pobiera kursy walut z serwisu money.pl. Dodaje domyślny kurs dla PLN i tworzy słownik z kursami walut.

11 Funkcja 'calculate()'

```
1  # Funkcja obliczająca wyniki po naciśnięciu odpowiedniego przycisku w GUI
2  def calculate():
3      try:
4          selected_cpu = cpu_combobox.get()
5          selected_gpu = gpu_combobox.get()
6          selected_hdd = hdd_combobox.get()
7          selected_ram = ram_combobox.get()
8          selected_currency = currency_combobox.get()
9          selected_mother = mother_board_combobox.get()
10         exchange_rate = exchange_rates[selected_currency]
11
12         total_price = round((cpu_prices[selected_cpu] + gpu_prices[selected_gpu] +
13                               hdd_prices[selected_hdd] +
14                               ram_prices[selected_ram] + mother_board_prices[
15                                   selected_mother])*exchange_rates["USD"], 2)
16         total_price = round(total_price / exchange_rate, 2)
17         total_score = round(cpu_scores[selected_cpu] + gpu_scores[selected_gpu] +
18                               hdd_scores[selected_hdd] +
19                               ram_scores[selected_ram] + mother_board_scores[
20                                   selected_mother], 2)
21
22         if total_score < 0.9:
23             total_score_label.config(text=f"Łączny wynik: {total_score} (słaby)")
24             total_score_label.config(foreground="red")
25             img_url = "https://media.makeameme.org/created/oh-dude-thats-5c689f.jpg"
26         elif 0.9 <= total_score < 1.15:
27             total_score_label.config(text=f"Łączny wynik: {total_score} (średni)")
28             total_score_label.config(foreground="yellow")
29             img_url = "https://us-tuna-sounds-images.voicemod.net/64c2bcda-a203-47cd-
30                 a81a-68bf07397033-1701636231104.jpeg"
31         else:
32             total_score_label.config(text=f"Łączny wynik: {total_score} (świetny!)")
33             total_score_label.config(foreground="green")
34             img_url = "https://melmagazine.com/wp-content/uploads/2021/01/66f-1.jpg"
35
36         total_price_label.config(text=f"Łączna cena: {total_price} {selected_currency}
37             ")
38
39         response = requests.get(img_url)
40         img_data = response.content
41         img = Image.open(io.BytesIO(img_data))
42         img = img.resize((150, 100), Image.LANCZOS)
43         img = ImageTk.PhotoImage(img)
44         image_label.config(image=img)
45         image_label.image = img
46         image_label.grid(row=7, column=1, rowspan=2)
```

Funkcja calculate oblicza łączną cenę i wynik na podstawie wybranych komponentów oraz aktualnego kursu walut. Wyświetla wynik i obrazek w zależności od uzyskanego wyniku. W przypadku błędu wyświetla odpowiedni komunikat.

12 Funkcja 'showprices()'

```
1 def show_prices():
2     selected_cpu = cpu_combobox.get()
3     selected_gpu = gpu_combobox.get()
4     selected_hdd = hdd_combobox.get()
5     selected_ram = ram_combobox.get()
6     selected_mother_board = mother_board_combobox.get()
7     selected_currency = currency_combobox.get()
8     exchange_rate = exchange_rates[selected_currency]
9
10    cpu_price = round(cpu_prices[selected_cpu] * exchange_rates["USD"] /
11                      exchange_rate, 2)
12    gpu_price = round(gpu_prices[selected_gpu] * exchange_rates["USD"] /
13                      exchange_rate, 2)
14    hdd_price = round(hdd_prices[selected_hdd] * exchange_rates["USD"] /
15                      exchange_rate, 2)
16    ram_price = round(ram_prices[selected_ram] * exchange_rates["USD"] /
17                      exchange_rate, 2)
18    mother_board_price = round(mother_board_prices[selected_mother_board] *
19                               exchange_rates["USD"] / exchange_rate, 2)
20
21    prices_cpu_label.config(text=f"Cena CPU: {cpu_price} {selected_currency}")
22    prices_gpu_label.config(text=f"Cena GPU: {gpu_price} {selected_currency}")
23    prices_hdd_label.config(text=f"Cena HDD: {hdd_price} {selected_currency}")
24    prices_ram_label.config(text=f"Cena RAM: {ram_price} {selected_currency}")
25    prices_mother_label.config(text=f"Cena płyty głównej: {mother_board_price} {
26                                  selected_currency}")
27
28    root.geometry(f"{window_width + 170}x{window_height}+{position_right - 170}+{
29                    position_top}")
```

Funkcja showprices wyświetla szczegółowe ceny dla każdego wybranego komponentu, przeliczone na wybraną walutę. Aktualizuje również rozmiar okna GUI.

13 Tworzenie GUI

```
1 root = tk.Tk()
2 root.title("Podsumowanie podzespołów")
3 root.configure(bg="#2e2e2e")
4
5 screen_width = root.winfo_screenwidth()
6 screen_height = root.winfo_screenheight()
7
8 window_width = 500
9 window_height = 700
10 position_top = int(screen_height/2 - window_height/2)
11 position_right = int(screen_width/2 - window_width/2)
12 root.geometry(f"{window_width}x{window_height}+{position_right}+{position_top}")
13
14 style = ttk.Style()
15 style.configure("TLabel", background="#2e2e2e", foreground="white", font=("Arial",
16    14))
17 style.configure("TCombobox", font=("Arial", 12))
18 style.configure("TButton", font=("Arial", 12, "bold"), background="black", foreground
19    ="white")
20 style.map("TButton",
21    foreground=[('!active', 'black'), ('pressed', 'white'), ('active', 'black')],
22    background=[('!active', 'black'), ('pressed', 'black'), ('active', 'white')])
23
24 cpu.sort()
```

```

24 gpu.sort()
25 hdd.sort()
26 ram.sort()
27 mother_board.sort()
28
29 cpu_label = ttk.Label(root, text="Wybierz CPU:")
30 cpu_label.grid(row=0, column=0, padx=50, pady=10, sticky='w')
31 cpu_combobox = ttk.Combobox(root, values=cpu)
32 cpu_combobox.grid(row=0, column=1, padx=20, pady=10, sticky='e')
33
34 gpu_label = ttk.Label(root, text="Wybierz GPU:")
35 gpu_label.grid(row=1, column=0, padx=50, pady=10, sticky='w')
36 gpu_combobox = ttk.Combobox(root, values=gpu)
37 gpu_combobox.grid(row=1, column=1, padx=20, pady=10, sticky='e')
38
39 hdd_label = ttk.Label(root, text="Wybierz HDD/SSD:")
40 hdd_label.grid(row=2, column=0, padx=50, pady=10, sticky='w')
41 hdd_combobox = ttk.Combobox(root, values=hdd)
42 hdd_combobox.grid(row=2, column=1, padx=20, pady=10, sticky='e')
43
44 ram_label = ttk.Label(root, text="Wybierz RAM:")
45 ram_label.grid(row=3, column=0, padx=50, pady=10, sticky='w')
46 ram_combobox = ttk.Combobox(root, values=ram)
47 ram_combobox.grid(row=3, column=1, padx=20, pady=10, sticky='e')
48
49 mother_board_label = ttk.Label(root, text="Wybierz płytę główną:")
50 mother_board_label.grid(row=4, column=0, padx=50, pady=10, sticky='w')
51 mother_board_combobox = ttk.Combobox(root, values=mother_board)
52 mother_board_combobox.grid(row=4, column=1, padx=20, pady=10, sticky='e')
53
54 currency_label = ttk.Label(root, text="Wybierz walutę:")
55 currency_label.grid(row=5, column=0, padx=50, pady=10, sticky='w')
56 currency_combobox = ttk.Combobox(root, values=list(currencies))
57 currency_combobox.grid(row=5, column=1, padx=20, pady=10, sticky='e')
58 currency_combobox.current(0)
59
60 calculate_button = ttk.Button(root, text="Oblicz", command=calculate)
61 calculate_button.grid(row=6, column=0, columnspan=2, pady=20)
62
63 total_score_label = ttk.Label(root, text="Łączny wynik:")
64 total_score_label.grid(row=7, column=0, columnspan=1, padx=(80,0), pady=5, sticky='w',
65 )
66
67 total_price_label = ttk.Label(root, text="Łączna cena:")
68 total_price_label.grid(row=8, column=0, columnspan=1, padx=(80,0), pady=5, sticky='w',
69 )
70
71 price_button = ttk.Button(root, text="Pokaż ceny szczegółowe", command=show_prices)
72 price_button.grid(row=9, column=0, columnspan=2, pady=20)
73
74 prices_cpu_label = ttk.Label(root, text="")
75 prices_cpu_label.grid(row=10, column=0, columnspan=2, pady=5, padx=50, sticky='w')
76
77 prices_hdd_label = ttk.Label(root, text="")
78 prices_hdd_label.grid(row=10, column=1, columnspan=2, pady=5, padx=45, sticky='w')
79
80 prices_gpu_label = ttk.Label(root, text="")
81 prices_gpu_label.grid(row=11, column=0, columnspan=2, pady=5, padx=50, sticky='w')
82
83 prices_ram_label = ttk.Label(root, text="")
84 prices_ram_label.grid(row=11, column=1, columnspan=2, pady=5, padx=45, sticky='w')
85
86 prices_mother_label = ttk.Label(root, text="")
87 prices_mother_label.grid(row=12, column=1, columnspan=2, pady=5, padx=45, sticky='w')
88
89 image_label = ttk.Label(root)

```

```

88 image_label.grid(row=7, column=1, rowspan=2, padx=5, sticky='e')
89
90 wyjscie = tk.Button(root,
91                     text='Wyjście',
92                     width=10,
93                     bg='tomato',
94                     command=root.destroy,
95                     bd=4)
96 wyjscie.grid(column=1, row=13, padx=20, pady=20, sticky=tk.SE)
97
98 root.mainloop()

```

Ta sekcja tworzy graficzny interfejs użytkownika (GUI) przy użyciu tkinter. Składa się z etykiet, rozwijanych list (combobox), przycisków oraz etykiet do wyświetlania wyników i obrazków. GUI pozwala użytkownikowi wybrać komponenty, wyświetlić szczegółowe ceny oraz obliczyć łączną cenę i wynik.

14 Podsumowanie

Użyteczność programu

- **Dla entuzjastów komputerowych:** Umożliwia szybkie porównanie cen i wydajności różnych komponentów komputerowych.
- **Dla osób składających komputer:** Pomaga w wyborze najlepszych komponentów w ramach określonego budżetu i preferencji wydajnościowych.
- **Dla sprzedawców i doradców IT:** Może służyć jako narzędzie do szybkiego przedstawienia klientowi możliwych konfiguracji i ich kosztów w różnych walutach.