```
{
 "cells": [
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "Zusammengefasst:\n",
    "\n",
    "- Der Benutzer gibt seinen Namen und sein Geburtsdatum ein und wählt ein Bild aus.\n",
    "- Das Programm berechnet das Sternzeichen basierend auf dem Geburtsdatum.\n",
    "- Das Bild wird skaliert und zugeschnitten, das Sternzeichenbild hinzugefügt und der - Text mit Namen und Geburtsdatum über das Bild gelegt.\n",
    "- Das kombinierte Bild wird angezeigt und kann gespeichert werden."
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 3,
   "metadata": {},
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "Bild auswählen gestartet.\n",
      "Bildpfad: C:/Users/Admin/OneDrive - DataCraft GmbH/Bilder/Eigene Aufnahmen/20210621_104454.jpg\n",
      "Bild erfolgreich geladen und angepasst.\n",
      "Ergebnisse anzeigen gestartet.\n",
      "Name: cc, Geburtsdatum: 17.01.1968, Bildpfad: C:/Users/Admin/OneDrive - DataCraft GmbH/Bilder/Eigene Aufnahmen/20210621_104454.jpg\n",
      "Sternzeichen: Steinbock\n",
      "Bild wird auf 800x800 skaliert und zugeschnitten.\n",
      "Benutzerbild erfolgreich geladen und angepasst.\n",
      "Sternzeichenbild erfolgreich geladen und angepasst.\n",
      "Schriftart Arial erfolgreich geladen.\n",
      "Öffnen des Dialogfensters zum Speichern.\n",
      "Speicherpfad: \n",
      "Speichervorgang abgebrochen.\n"
     ]
    }
   ],
   "source": [
    "# Imports\n",
    "\n",
    "import tkinter as tk\n",
    "from tkinter import ttk, filedialog, messagebox\n",
    "from PIL import Image, ImageTk, ImageDraw, ImageFont\n",
    "import numpy as np\n",
    "from datetime import datetime\n",
    "\n",
    "\n",
```

```
    "# Funktion zur Berechnung des Sternzeichens\n",
    "\n",
    "# Diese Funktion berechnet das Sternzeichen basierend auf \n",
    "# dem Tag und Monat des Geburtsdatums.\n",
    "\n",
    "def get_zodiac_sign(day, month):\n",
    "    if (month == 12 and day >= 22) or (month == 1 and day <= 19):\n",
    "        return \"Steinbock\"\n",
    "    elif (month == 1 and day >= 20) or (month == 2 and day <= 18):\n",
    "        return \"Wassermann\"\n",
    "    elif (month == 2 and day >= 19) or (month == 3 and day <= 20):\n",
    "        return \"Fisch\"\n",
    "    elif (month == 3 and day >= 21) or (month == 4 and day <= 19):\n",
    "        return \"Widder\"\n",
    "    elif (month == 4 and day >= 20) or (month == 5 and day <= 20):\n",
    "        return \"Stier\"\n",
    "    elif (month == 5 and day >= 21) or (month == 6 and day <= 20):\n",
    "        return \"Zwilling\"\n",
    "    elif (month == 6 and day >= 21) or (month == 7 and day <= 22):\n",
    "        return \"Krebs\"\n",
    "    elif (month == 7 and day >= 23) or (month == 8 and day <= 22):\n",
    "        return \"Loewe\"\n",
    "    elif (month == 8 and day >= 23) or (month == 9 and day <= 22):\n",
    "        return \"Jungfrau\"\n",
    "    elif (month == 9 and day >= 23) or (month == 10 and day <= 22):\n",
    "        return \"Waage\"\n",
    "    elif (month == 10 and day >= 23) or (month == 11 and day <= 21):\n",
    "        return \"Skorpion\"\n",
    "    elif (month == 11 and day >= 22) or (month == 12 and day <= 21):\n",
    "        return \"Schütze\"\n",
    "\n",
    "\n",
    "# Bildauswahl-Funktion\n",
    "\n",
    "# Diese Funktion öffnet einen Dateiauswahldialog, um ein Bild auszuwählen,
\n",
    "# und zeigt es in der GUI an.\n",
    "\n",
    "def select_image():\n",
    "    print(\"Bild auswählen gestartet.\")\n",
    "    file_path = filedialog.askopenfilename()\n",
    "    if file_path:\n",
    "        try:\n",
    "            print(f\"Bildpfad: {file_path}\")\n",
    "            image = Image.open(file_path)\n",
    "            image.thumbnail((800, 800), Image.LANCZOS)\n",
    "            user_image = ImageTk.PhotoImage(image)\n",
    "            image_label.config(image=user_image)\n",
    "            image_label.image = user_image\n",
    "            image_label.file_path = file_path\n",
    "            print(\"Bild erfolgreich geladen und angepasst.\")\n",
    "        except Exception as e:\n",
    "            print(f\"Fehler beim Laden des Bildes: {e}\")\n",
    "            messagebox.showerror(\"Fehler\", f\"Bild konnte nicht geladen
```

```
werden: {e}\")\n",
    "\n",
    "\n",
    "# Bildgröße anpassen und zuschneiden\n",
    "\n",
    "# Diese Funktion skaliert und schneidet ein Bild auf eine Größe von 800x800
Pixeln zu.\n",
    "\n",
    "def resize_and_crop_to_800x800(image_array):\n",
    "    print(\"Bild wird auf 800x800 skaliert und zugeschnitten.\")\n",
    "    image = Image.fromarray(image_array)\n",
    "    width, height = image.size\n",
    "    aspect_ratio = width / height\n",
    "\n",
    "    if aspect_ratio > 1:  # Landscape orientation\n",
    "        new_height = 800\n",
    "        new_width = int(aspect_ratio * new_height)\n",
    "    else:  # Portrait orientation or square\n",
    "        new_width = 800\n",
    "        new_height = int(new_width / aspect_ratio)\n",
    "\n",
    "    # Bild proportional skalieren\n",
    "    image = image.resize((new_width, new_height), Image.LANCZOS)\n",
    "\n",
    "    # Bild zuschneiden\n",
    "    left = (new_width - 800) / 2\n",
    "    top = (new_height - 800) / 2\n",
    "    right = (new_width + 800) / 2\n",
    "    bottom = (new_height + 800) / 2\n",
    "\n",
    "    image = image.crop((left, top, right, bottom))\n",
    "    return np.array(image)\n",
    "\n",
    "\n",
    "# Ergebnisse anzeigen\n",
    "\n",
    "# Diese Funktion sammelt die Benutzereingaben, \n",
    "# berechnet das Sternzeichen, \n",
    "# lädt und verarbeitet das Bild und das Sternzeichenbild, \n",
    "# fügt die Texte hinzu und speichert das kombinierte Bild.\n",
    "\n",
    "def show_results():\n",
    "    print(\"Ergebnisse anzeigen gestartet.\")\n",
    "    name = name_entry.get()\n",
    "    birth_date = birth_date_entry.get()\n",
    "    image_path = getattr(image_label, 'file_path', None)\n",
    "    \n",
    "    print(f\"Name: {name}, Geburtsdatum: {birth_date}, Bildpfad:
{image_path}\")\n",
    "    \n",
    "    try:\n",
    "        birth_date_obj = datetime.strptime(birth_date, \"%d.%m.%Y\")\n",
    "        day, month = birth_date_obj.day, birth_date_obj.month\n",
    "        zodiac_sign = get_zodiac_sign(day, month)\n",
```

```
    "        print(f\"Sternzeichen: {zodiac_sign}\")\n",
    "    except ValueError:\n",
    "        print(\"Ungültiges Geburtsdatum.\")\n",
    "        messagebox.showerror(\"Fehler\", \"Ungültiges Geburtsdatum. Bitte
im Format TT.MM.JJJJ eingeben.\")\n",
    "        return\n",
    "\n",
    "    if not name or not birth_date or not image_path:\n",
    "        print(\"Nicht alle Felder ausgefüllt oder kein Bild
ausgewählt.\")\n",
    "        messagebox.showerror(\"Fehler\", \"Bitte füllen Sie alle Felder aus
und wählen Sie ein Bild aus.\")\n",
    "        return\n",
    "    \n",
    "    # Erstellen des Ausgabe-Bildes\n",
    "    try:\n",
    "        user_image = Image.open(image_path)\n",
    "        user_image = user_image.convert(\"RGBA\")\n",
    "        user_image_array = np.array(user_image)\n",
    "        user_image_array = resize_and_crop_to_800x800(user_image_array)\n",
    "        user_image = Image.fromarray(user_image_array)\n",
    "        user_image_width, user_image_height = user_image.size\n",
    "        print(\"Benutzerbild erfolgreich geladen und angepasst.\")\n",
    "    except Exception as e:\n",
    "        print(f\"Fehler beim Verarbeiten des Benutzerbildes: {e}\")\n",
    "        messagebox.showerror(\"Fehler\", f\"Bild konnte nicht verarbeitet
werden: {e}\")\n",
    "        return\n",
    "\n",
    "    # Laden und Einfügen des Sternzeichenbildes\n",
    "    zodiac_image_path = f\"zodiac_images/{zodiac_sign.lower()}.png\"\n",
    "    try:\n",
    "        zodiac_image = Image.open(zodiac_image_path)\n",
    "        zodiac_image.thumbnail((200, 200), Image.LANCZOS)\n",
    "        zodiac_image = zodiac_image.convert(\"RGBA\")\n",
    "        zodiac_image_width, zodiac_image_height = zodiac_image.size\n",
    "        print(\"Sternzeichenbild erfolgreich geladen und angepasst.\")\n",
    "    except FileNotFoundError:\n",
    "        print(f\"Sternzeichenbild für {zodiac_sign} nicht gefunden.\")\n",
    "        messagebox.showerror(\"Fehler\", f\"Sternzeichenbild für
{zodiac_sign} nicht gefunden.\")\n",
    "        return\n",
    "\n",
    "    combined_image = Image.new('RGBA', (user_image_width,
user_image_height))\n",
    "    combined_image.paste(user_image, (0, 0))\n",
    "    combined_image.paste(zodiac_image, (user_image_width -
zodiac_image_width, 0), zodiac_image)\n",
    "\n",
    "    # Hinzufügen von Name und Geburtsdatum\n",
    "    draw = ImageDraw.Draw(combined_image)\n",
    "    try:\n",
    "        font = ImageFont.truetype(\"arial.ttf\", 40)\n",
    "        print(\"Schriftart Arial erfolgreich geladen.\")\n",
```

```
    "    except IOError:\n",
    "        font = ImageFont.load_default()\n",
    "        print(\"Standard-Schriftart geladen.\")\n",
    "\n",
    "    text = f\"Name: {name}\\nGeburtsdatum: {birth_date}\\nDu bist:
{zodiac_sign}\"\n",
    "    text_position = (10, user_image_height - 120)\n",
    "    draw.text(text_position, text, font=font, fill=\"white\")\n",
    "\n",
    "    # Speichern des Ausgabe-Bildes\n",
    "    print(\"Öffnen des Dialogfensters zum Speichern.\")\n",
    "    save_path = filedialog.asksaveasfilename(defaultextension=\".png\",\n",
    "                                             filetypes=[(\"PNG files\",
\"*.png\"), (\"All files\", \"*.*\")])\n",
    "    print(f\"Speicherpfad: {save_path}\")\n",
    "    \n",
    "    if save_path:\n",
    "        try:\n",
    "            print(f\"Speichern des Bildes unter: {save_path}\")\n",
    "            combined_image.save(save_path)\n",
    "            print(f\"Bild erfolgreich gespeichert unter: {save_path}\")\n",
    "            messagebox.showinfo(\"Erfolg\", f\"Bild erfolgreich gespeichert
unter: {save_path}\")\n",
    "        except Exception as e:\n",
    "            print(f\"Fehler beim Speichern des Bildes: {e}\")\n",
    "            messagebox.showerror(\"Fehler\", f\"Bild konnte nicht
gespeichert werden: {e}\")\n",
    "    else:\n",
    "        print(\"Speichervorgang abgebrochen.\")\n",
    "\n",
    "    # Anzeigen des Ausgabe-Bildes\n",
    "    combined_image.show()\n",
    "\n",
    "\n",
    "# GUI-Setup\n",
    "\n",
    "# Dieser Teil des Codes erstellt die \n",
    "# GUI-Elemente (Labels, Eingabefelder, Buttons) \n",
    "# und startet die Hauptschleife der GUI.\n",
    "\n",
    "root = tk.Tk()\n",
    "root.title(\"Sternzeichen Bestimmung\")\n",
    "\n",
    "# Layout\n",
    "name_label = ttk.Label(root, text=\"Name:\")\n",
    "name_label.grid(row=0, column=0, padx=10, pady=10)\n",
    "name_entry = ttk.Entry(root)\n",
    "name_entry.grid(row=0, column=1, padx=10, pady=10)\n",
    "\n",
    "birth_date_label = ttk.Label(root, text=\"Geburtsdatum (TT.MM.JJJJ):\")\n",
    "birth_date_label.grid(row=1, column=0, padx=10, pady=10)\n",
    "birth_date_entry = ttk.Entry(root)\n",
    "birth_date_entry.grid(row=1, column=1, padx=10, pady=10)\n",
    "\n",
```

```
    "# Bildanzeige\n",
    "image_label = ttk.Label(root)\n",
    "image_label.grid(row=2, column=0, columnspan=2, padx=10, pady=10)\n",
    "\n",
    "# Bildauswahl-Button\n",
    "select_image_button = ttk.Button(root, text=\"Bild auswählen\",
command=select_image)\n",
    "select_image_button.grid(row=3, column=0, columnspan=2, padx=10,
pady=10)\n",
    "\n",
    "# Ergebnis-Button\n",
    "show_results_button = ttk.Button(root, text=\"Ergebnisse anzeigen\",
command=show_results)\n",
    "show_results_button.grid(row=4, column=0, columnspan=2, padx=10,
pady=10)\n",
    "\n",
    "# Hauptschleife starten\n",
    "root.mainloop()\n",
    "\n",
    "# Zusammengefasst:\n",
    "\n",
    "# Der Benutzer gibt seinen Namen und sein Geburtsdatum ein und wählt ein
Bild aus.\n",
    "# Das Programm berechnet das Sternzeichen basierend auf dem
Geburtsdatum.\n",
    "# Das Bild wird skaliert und zugeschnitten, \n",
    "# das Sternzeichenbild hinzugefügt und der Text mit \n",
    "# Namen und Geburtsdatum über das Bild gelegt.\n",
    "# Das kombinierte Bild wird angezeigt und kann gespeichert werden."
   ]
  }
 ],
 "metadata": {
  "kernelspec": {
   "display_name": "DataCraft",
   "language": "python",
   "name": "python3"
  },
  "language_info": {
   "codemirror_mode": {
    "name": "ipython",
    "version": 3
   },
   "file_extension": ".py",
   "mimetype": "text/x-python",
   "name": "python",
   "nbconvert_exporter": "python",
   "pygments_lexer": "ipython3",
   "version": "3.12.2"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 2
}
```