

**TRƯỜNG ĐẠI HỌC VĂN LANG
KHOA CÔNG NGHỆ THÔNG TIN**



ĐỒ ÁN 1
MÔN HỌC LẬP TRÌNH PYTHON NÂNG CAO

Chủ đề:

QUẢN LÝ THỜI GIAN HẲNG NGÀY

SVTH: NGUYỄN THÀNH NHÂN

MSSV: 2274802010600

LỚP: 241_71ITSE31003_0201

GVHD: HUỖNH THÁI HỌC

TP. Hồ Chí Minh – 10/2024

Mục lục

| | |
|------------------------------|----|
| 1. Giao diện: | 3 |
| 2. Chức năng: | 7 |
| 3. Mã chương trình: | 8 |
| 3.1 Lớp chính | 10 |
| 3.2 Thuộc tính chính | 11 |
| 3.3 Các hàm của class | 11 |
| 4. GitHub: | 11 |

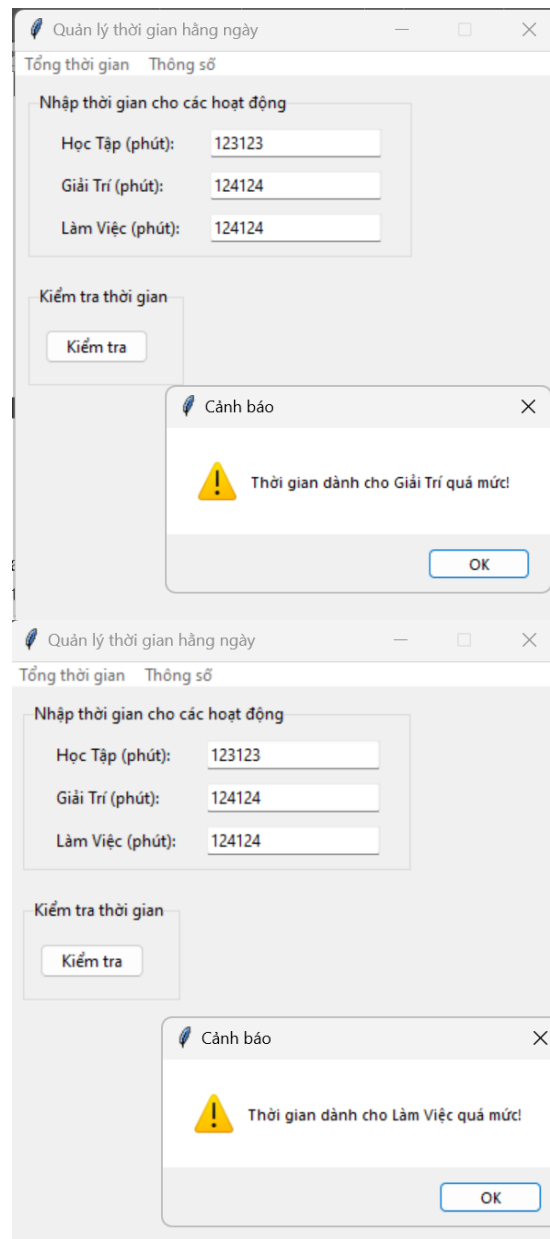
1. Giao diện:

- Ứng dụng sử dụng thư viện **Tkinter** để xây dựng giao diện người dùng.
- Giao diện bao gồm các ô nhập liệu cho thời gian dành cho các hoạt động hàng ngày như **Học Tập**, **Giải Trí**, và **Làm Việc** cùng với nút kiểm tra.

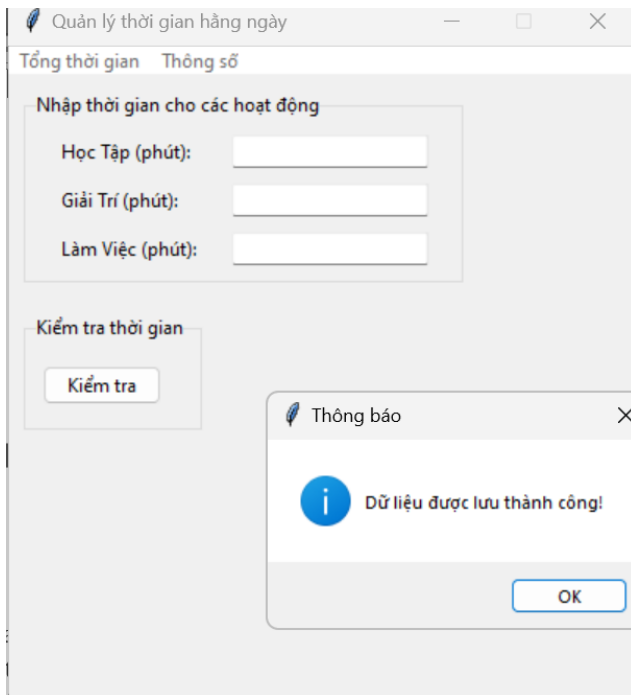
- Cung cấp cảnh báo khi các mức tiêu thụ thời gian vượt ngưỡng:
 - Cảnh báo sẽ xuất hiện khi thời gian dành cho bất kỳ hoạt động nào vượt quá giới hạn đã được xác định trước cho từng hoạt động.

The screenshot displays a software window titled "Quản lý thời gian hằng ngày" (Daily Time Management). It features two tabs: "Tổng thời gian" (Total Time) and "Thông số" (Statistics). The "Tổng thời gian" tab is active, showing a section "Nhập thời gian cho các hoạt động" (Enter time for activities) with three input fields: "Học Tập (phút):" (Study (minutes)) containing "123123", "Giải Trí (phút):" (Entertainment (minutes)) containing "124124", and "Làm Việc (phút):" (Work (minutes)) containing "124124". Below this is a "Kiểm tra thời gian" (Check time) section with a "Kiểm tra" (Check) button. A warning dialog box titled "Cảnh báo" (Warning) is overlaid on the main window, displaying a yellow warning icon and the text "Thời gian dành cho Học Tập quá mức!" (Study time is excessive!). An "OK" button is at the bottom of the dialog.

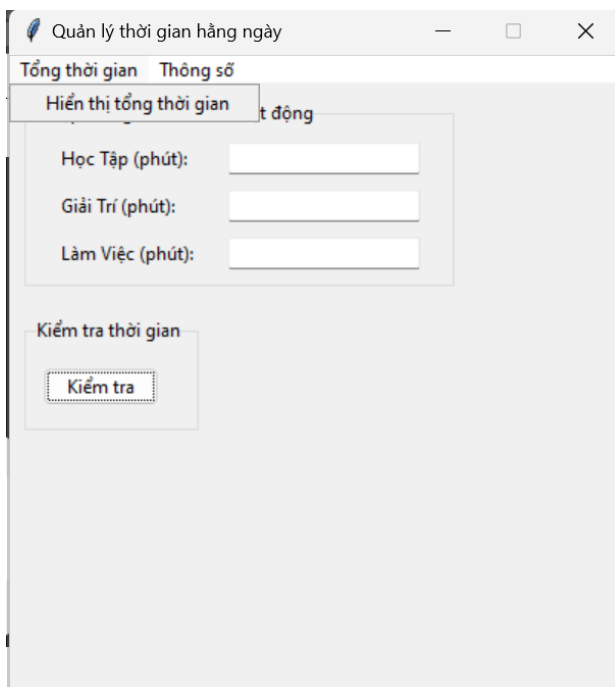
| Hoạt động | Thời gian (phút) |
|-----------|------------------|
| Học Tập | 123123 |
| Giải Trí | 124124 |
| Làm Việc | 124124 |

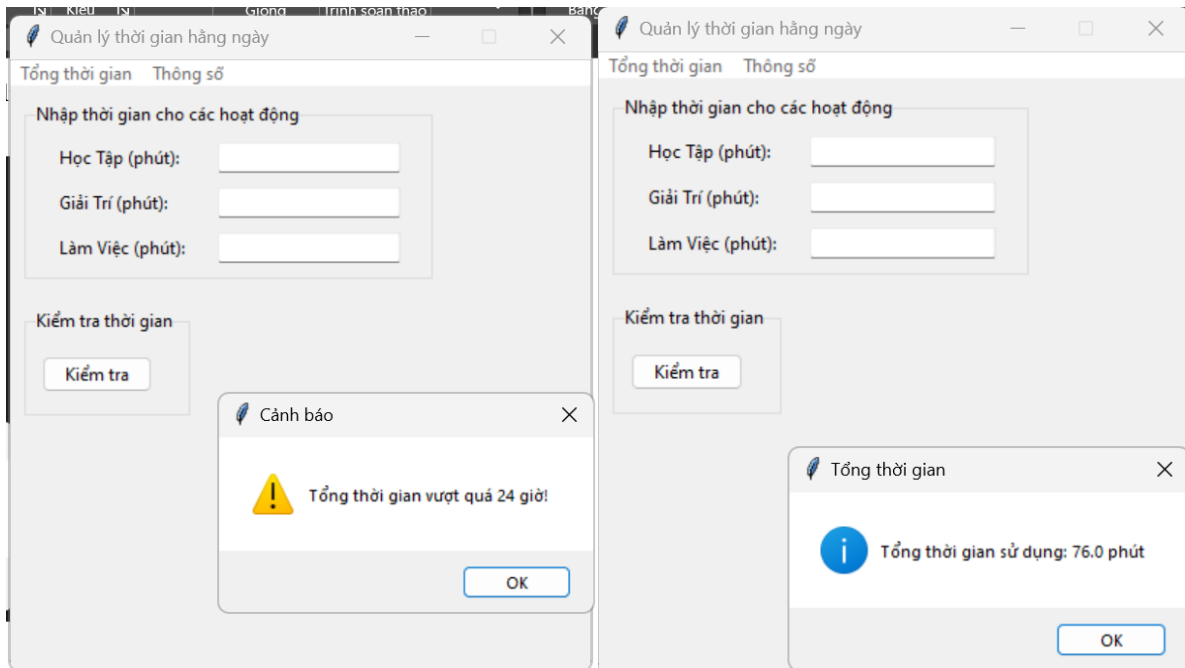


- Sau khi đưa ra các cảnh báo, thông báo sẽ xuất hiện cho biết dữ liệu đã được lưu thành công và các số liệu sẽ biến mất để chuẩn bị cho lần nhập liệu tiếp theo.

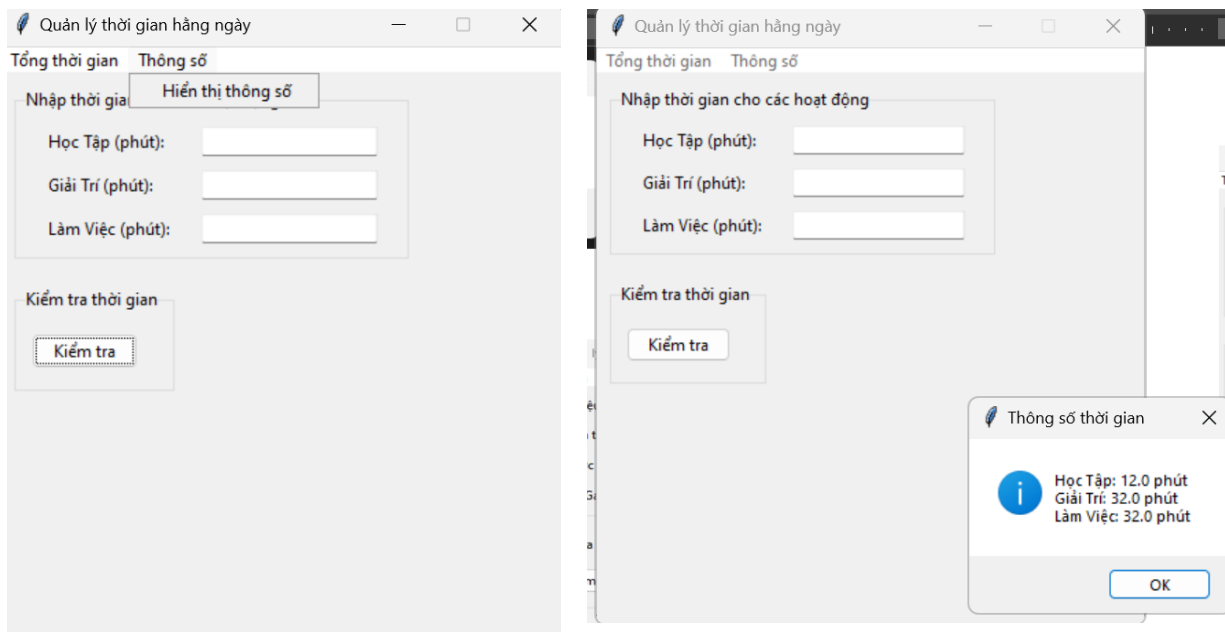


- Giao diện cũng cho phép người dùng xem tổng thời gian đã sử dụng cho các hoạt động và thông số vừa nhập gần đây và đưa ra cảnh báo nếu thời gian vượt quá 24 giờ.





- Giao diện xem thông số vừa nhập vào:



2. Chức năng:

- **Nhập dữ liệu:** Người dùng có thể nhập thời gian dành cho từng hoạt động (Học Tập, Giải Trí, Làm Việc) và nhấn nút kiểm tra để lưu dữ liệu.

- **Cảnh báo:** Hiển thị cảnh báo nếu thời gian dành cho hoạt động nào vượt quá ngưỡng đã định. Nếu tổng thời gian vượt quá 24 giờ, người dùng cũng sẽ nhận được cảnh báo.
- **Hiển thị tổng thời gian:** Tính toán và hiển thị tổng thời gian sử dụng cho các hoạt động dựa trên dữ liệu nhập vào.
- **Lưu trữ và hiển thị thông số:** Ứng dụng lưu trữ và cho phép người dùng xem các thông số thời gian dành cho từng hoạt động đã nhập gần đây.

3. Mã chương trình:

```
import tkinter as tk
from tkinter import ttk
from tkinter import messagebox

class QuanLyThoiGian:
    def __init__(self, root):
        self.root = root
        self.root.title("Quản lý thời gian hằng ngày")
        self.root.geometry("400x400")
        self.root.config(bg="#f0f0f0") # Màu nền dễ nhìn
        self.root.resizable(False, False) # Khóa phóng to cửa sổ

        # Danh sách cảnh báo
        self.canhbao_list = []
        # Ngưỡng cảnh báo (phút)
        self.nguonccanhbao = {"Học Tập": 300, "Giải Trí": 240, "Làm Việc": 180}
        # Tổng thời gian
        self.total_time = 1440 # 1440 phút = 24 giờ
        # Nơi chứa giá trị vừa nhập
        self.recent_study = 0
        self.recent_entertainment = 0
        self.recent_work = 0

        # Giao diện
        self.giao_dien()
        self.thanh_menu()

    def nhap_data(self):
        try:
            study = float(self.entry_study.get())
            entertainment = float(self.entry_entertainment.get())
            work = float(self.entry_work.get())

            # Lưu các giá trị vừa nhập gần nhất
            self.recent_study = study
            self.recent_entertainment = entertainment
            self.recent_work = work
```



```

# Kiểm tra nếu vượt quá thời gian cho từng hoạt động
if study > self.nguoncanhbao["Học Tập"]:
    canhbao_message = "Thời gian dành cho Học Tập quá mức!"
    self.canhbao_list.append(canhbao_message)
    messagebox.showwarning("Cảnh báo", canhbao_message)
if entertainment > self.nguoncanhbao["Giải Trí"]:
    canhbao_message = "Thời gian dành cho Giải Trí quá mức!"
    self.canhbao_list.append(canhbao_message)
    messagebox.showwarning("Cảnh báo", canhbao_message)
if work > self.nguoncanhbao["Làm Việc"]:
    canhbao_message = "Thời gian dành cho Làm Việc quá mức!"
    self.canhbao_list.append(canhbao_message)
    messagebox.showwarning("Cảnh báo", canhbao_message)

# Xóa các ô sau khi nhấn nút kiểm tra
self.entry_study.delete(0, tk.END)
self.entry_entertainment.delete(0, tk.END)
self.entry_work.delete(0, tk.END)

messagebox.showinfo("Thông báo", "Dữ liệu được lưu thành công!")

except ValueError:
    messagebox.showerror("Lỗi", "Vui lòng nhập giá trị số hợp lệ và không để trống!")

def show_thongso(self):
    hienthi2 = (f"Học Tập: {self.recent_study} phút\n"
               f"Giải Trí: {self.recent_entertainment} phút\n"
               f"Làm Việc: {self.recent_work} phút")
    messagebox.showinfo("Thông số thời gian", hienthi2)

def show_tongthoigian(self):
    tong_thoigian = self.recent_study + self.recent_entertainment + self.recent_work
    if tong_thoigian > self.total_time:
        messagebox.showwarning("Cảnh báo", f"Tổng thời gian vượt quá 24 giờ!")
    else:
        messagebox.showinfo("Tổng thời gian", f"Tổng thời gian sử dụng: {tong_thoigian} phút")

def giao_dien(self):
    frame_nhap = ttk.LabelFrame(self.root, text="Nhập thời gian cho các hoạt động", padding=(10, 5))
    frame_nhap.grid(row=0, column=0, padx=10, pady=10, columnspan=2)

    # Tạo Label trong frame nhập
    label_study = ttk.Label(frame_nhap, text="Học Tập (phút):")
    label_study.grid(row=0, column=0, padx=10, pady=5, sticky="w")

```

```

self.entry_study = tk.Entry(frame_nhap)
self.entry_study.grid(row=0, column=1, padx=10, pady=5)
self.entry_study.focus()

label_entertainment = tk.Label(frame_nhap, text="Giải Trí (phút):")
label_entertainment.grid(row=1, column=0, padx=10, pady=5, sticky="w")
self.entry_entertainment = tk.Entry(frame_nhap)
self.entry_entertainment.grid(row=1, column=1, padx=10, pady=5)

label_work = tk.Label(frame_nhap, text="Làm Việc (phút):")
label_work.grid(row=2, column=0, padx=10, pady=5, sticky="w")
self.entry_work = tk.Entry(frame_nhap)
self.entry_work.grid(row=2, column=1, padx=10, pady=5)

# Button kiểm tra thời gian
frame_ketqua = tk.LabelFrame(self.root, text="Kiểm tra thời gian", padding=(10, 5))
frame_ketqua.grid(row=1, column=0, padx=10, pady=10, columnspan=2, sticky='w')

submit_button = tk.Button(frame_ketqua, text="Kiểm tra", command=self.nhap_data)
submit_button.grid(row=1, column=0, columnspan=2, pady=10)

def thanh_menu(self):
    menu_bar = tk.Menu(self.root)
    self.root.config(menu=menu_bar)

    # Thêm menubar
    tongthoigian_menu = tk.Menu(menu_bar, tearoff=0)
    menu_bar.add_cascade(label="Tổng thời gian", menu=tongthoigian_menu)
    tongthoigian_menu.add_command(label="Hiển thị tổng thời gian", command=self.show_tongthoigian)

    thongso_menu = tk.Menu(menu_bar, tearoff=0)
    menu_bar.add_cascade(label="Thông số", menu=thongso_menu)
    thongso_menu.add_command(label="Hiển thị thông số", command=self.show_thongso)

if __name__ == "__main__":
    root = tk.Tk()
    app = QuanLyThoiGian(root)
    root.mainloop()

```

3.1 Lớp chính

- **Class QuanLyThoiGian:** Đây là lớp chính của ứng dụng.

3.2 Thuộc tính chính

- **root**: Cửa sổ chính của ứng dụng.
- **canhbao_list**: Danh sách các cảnh báo về thời gian vượt mức.
- **nguongcanhbao**: Từ điển chứa ngưỡng cảnh báo cho từng hoạt động (Học Tập, Giải Trí, Làm Việc).
- **total_time**: Tổng thời gian tối đa cho phép trong ngày (1440 phút).
- **recent_study, recent_entertainment, recent_work**: Biến để lưu trữ giá trị thời gian dành cho từng hoạt động gần nhất.

3.3 Các hàm của class

- **init(self, root)**: Hàm khởi tạo, thiết lập giao diện và các thuộc tính ban đầu cho ứng dụng.
- **nhap_data(self)**: Hàm xử lý khi người dùng nhập thời gian cho các hoạt động và nhấn nút kiểm tra.
- **show_thongso(self)**: Hàm hiển thị các thông số thời gian dành cho từng hoạt động gần nhất.
- **show_tongthoigian(self)**: Hàm hiển thị tổng thời gian đã sử dụng cho các hoạt động.
- **giao_dien(self)**: Hàm thiết lập giao diện người dùng.
- **thanh_menu(self)**: Hàm tạo thanh menu cho ứng dụng.

4. GitHub:

<https://github.com/NhanDet/NGUYENTHANHNHAN/tree/main/Do%20An>