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





**BÁO CÁO BÀI 1**

*Sinh viên thực hiện :* Nguyễn Trung Kiên

*MSSV : 2274802010447*

*Giảng viên hướng dẫn :* Huỳnh Thái Học

*TP Hồ Chí Minh, ngày 4, tháng 10, năm 2024*

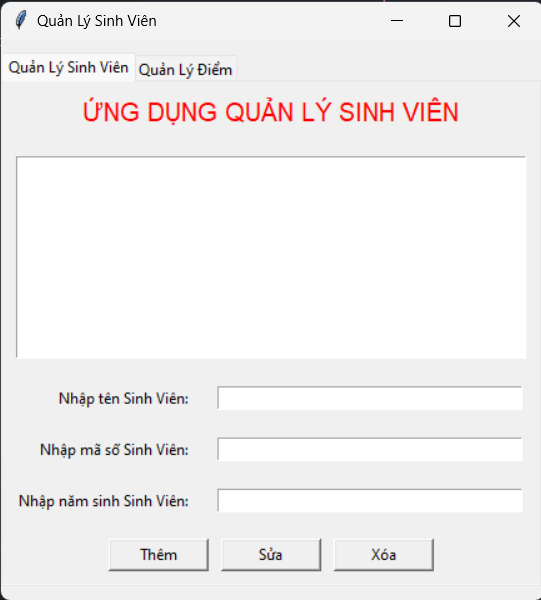
MỤC LỤC

# **Lời cảm ơn**

Em xin gửi lời cảm ơn chân thành đến Trường Đại học Văn Lang và Khoa Công nghệ Thông tin đã tạo điều kiện cho em thực hiện đồ án môn Lập trình Python nâng cao. Đặc biệt, em xin bày tỏ lòng biết ơn sâu sắc đến Thầy Huỳnh Thái Học vì đã tận tình giảng dạy, hướng dẫn và hỗ trợ em trong suốt quá trình thực hiện đồ án. Thầy đã không chỉ truyền đạt kiến thức mà còn giúp em hiểu rõ cách áp dụng lý thuyết vào thực tế, nâng cao khả năng tư duy lập trình của bản thân. Em cũng xin cảm ơn gia đình và bạn bè, những người đã luôn động viên và ủng hộ em trong suốt quá trình học tập và hoàn thành đồ án này. Qua quá trình thực hiện, em đã học được nhiều kiến thức bổ ích và rèn luyện kỹ năng quan trọng. Một lần nữa, em xin chân thành cảm ơn tất cả những ai đã giúp đỡ và đồng hành cùng em trong suốt chặng đường này.

# Phần 1 GIAO DIỆN

## **Giao diện:**



Hình 1 Giao diện của chương trình

## **Mô tả:**

Giao diện trong Python này sử dụng thư viện Tkinter để thiết kế ứng dụng quản lý sinh viên và quản lý điểm. Giao diện bao gồm hai tab chính:

### Tab "**Quản Lý Sinh Viên**":

* Tiêu đề "**ỨNG DỤNG QUẢN LÝ SINH VIÊN**" được hiển thị với màu đỏ, nổi bật ở phía trên.
* Một khung lớn bên dưới tiêu đề, là `**Listbox**` để hiển thị danh sách sinh viên.
* Bên dưới khung này, có ba ô nhập dữ liệu để nhập thông tin sinh viên:
* **"Nhập tên Sinh Viên".**
* **"Nhập mã số Sinh Viên".**
* **"Nhập năm sinh Sinh Viên".**
* Dưới cùng là ba nút lệnh: "**Thêm**", "**Sửa**", và "**Xóa**" để thao tác với thông tin sinh viên.

### Tab "**Quản Lý Điểm**":

* Tiêu đề "**QUẢN LÝ ĐIỂM SINH VIÊN**" hiển thị với màu xanh dương.
* Một khung lớn bên dưới tiêu đề, là `**Listbox**` để hiển thị danh sách sinh viên.
* Bên dưới là ba ô nhập liệu cho điểm các môn:
* **"Nhập điểm Toán".**
* **"Nhập điểm Văn".**
* **"Nhập điểm Anh".**
* Dưới cùng có các nút chức năng: "**Nhập Điểm**", "**Sắp Xếp Điểm**", "**Điểm Cao Nhất**", và "**Xuất Dữ Liệu**" để thao tác với dữ liệu điểm.

Thiết kế giao diện này giúp quản lý thông tin sinh viên và điểm số một cách trực quan và dễ dàng, với các nút chức năng chính và các trường nhập liệu rõ ràng.

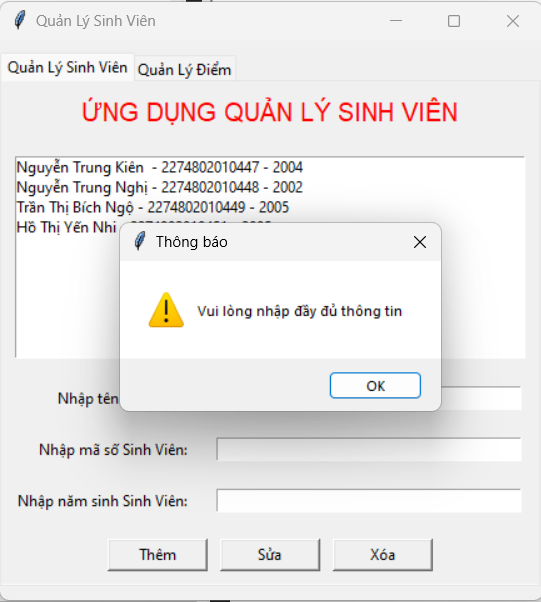
# Phần 2 Chức năng

Giao diện trong ứng dụng Python quản lý sinh viên và quản lý điểm có các chức năng chi tiết như sau:

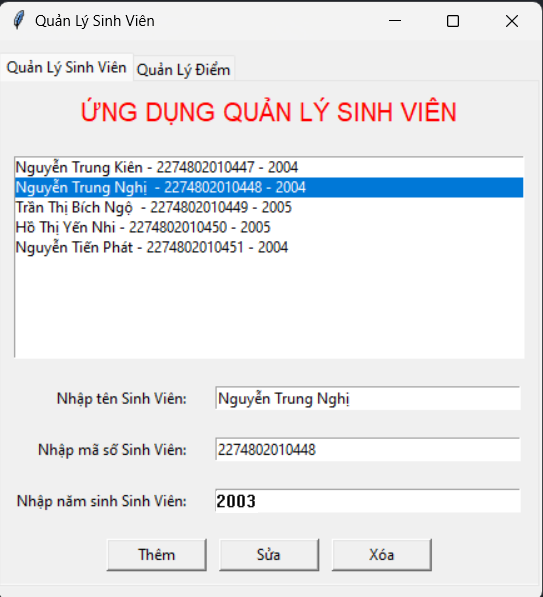
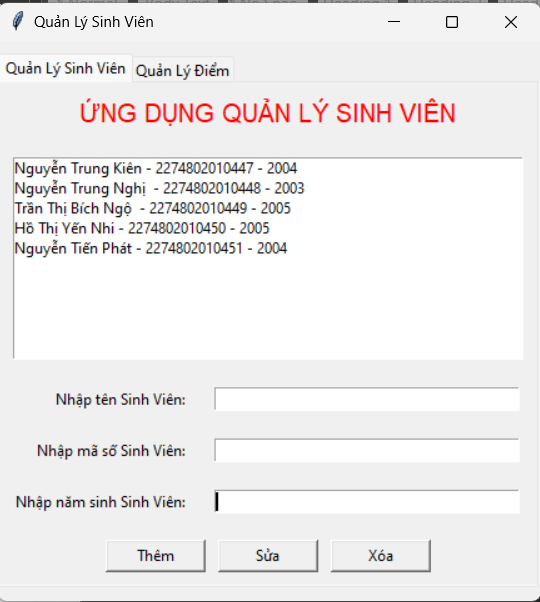
## **Tab "Quản Lý Sinh Viên"**

* Danh sách sinh viên: Hiển thị danh sách các sinh viên đã được nhập vào hệ thống. Đây là `Listbox` để hiển thị tên, mã số, và năm sinh của các sinh viên.
* Nhập tên Sinh Viên: Một ô nhập (`Entry`) cho phép người dùng nhập tên sinh viên.
* Nhập mã số Sinh Viên: Một ô nhập (`Entry`) cho phép người dùng nhập mã số sinh viên, có thể là số định danh duy nhất.
* Nhập năm sinh Sinh Viên: Một ô nhập (`Entry`) cho phép người dùng nhập năm sinh của sinh viên, dùng để xác định tuổi của sinh viên.
* Nút "Thêm":Chức năng thêm sinh viên mới vào danh sách. Khi nhấn nút này, các thông tin nhập vào từ các ô "Tên sinh viên", "Mã số sinh viên", và "Năm sinh" sẽ được lưu vào danh sách sinh viên.
* Nút "Sửa":Chức năng chỉnh sửa thông tin của sinh viên đã được chọn trong danh sách. Người dùng có thể chọn một sinh viên từ danh sách, sau đó sửa các thông tin và nhấn nút này để lưu các thay đổi.
* Nút "Xóa":Chức năng xóa sinh viên đã được chọn trong danh sách. Khi nhấn nút này, sinh viên được chọn sẽ bị loại khỏi danh sách.
* Chức năng thêm sinh viên: Đầu tiên hãy tiến hành nhập những thông tin đầy đủ vào các trường dữ liệu và click nút Thêm sau đó Listbox sẽ cập nhật vừa được thêm vào

Hình 2 Chức năng thêm sinh viên

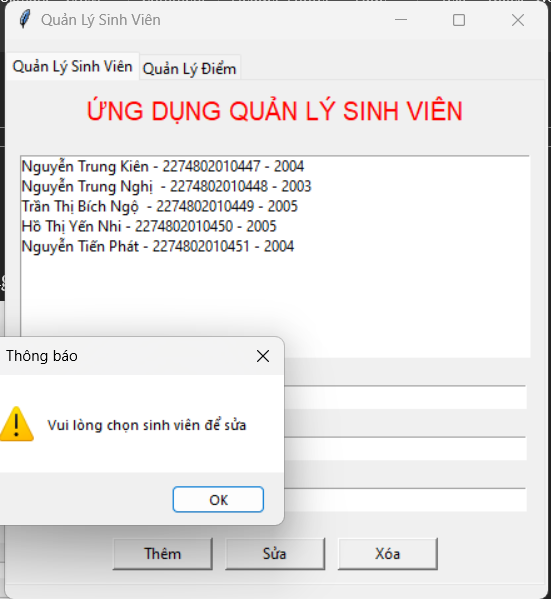
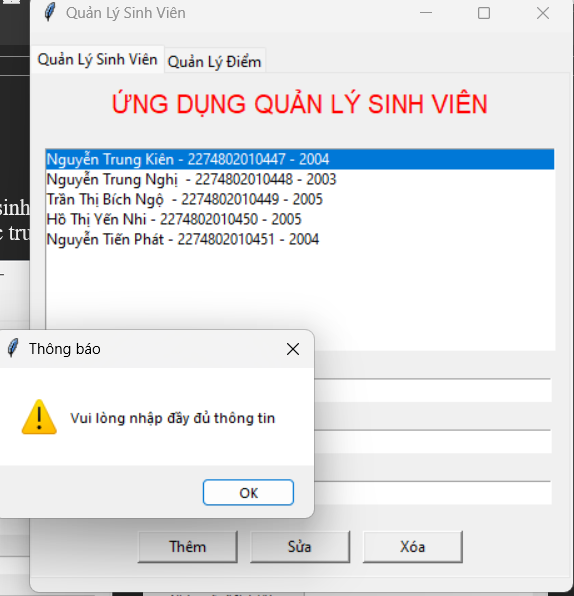
* Nhưng nếu ta nhập thiếu một trường thông tin hay không nhập bất kì trường thông tin nào thì chương trình sẽ hiện thông báo như sau:

Hình 3 Hiển thị thông báo yêu cầu nhập đầy đủ thông tin

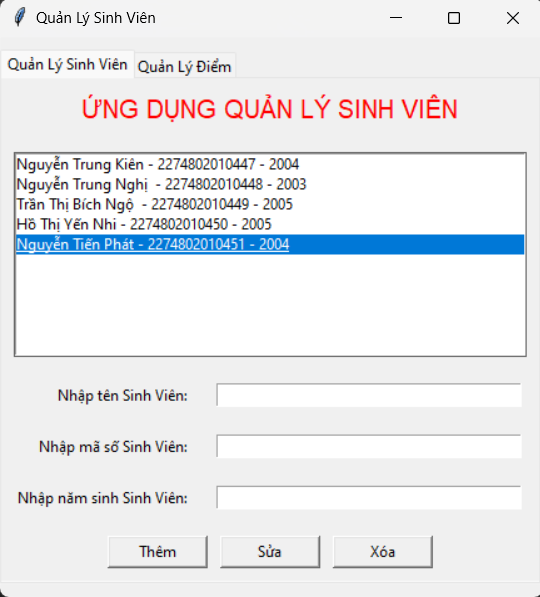
Chức năng sửa thông tin của sinh viên: Đầu tiên chọn sinh viên cần sửa thông tin và cập nhật thông tin cần sửa vào các trường bên dưới, sau đó click nút sửa.

Hình 4 Chức năng sửa thông tin sinh viên

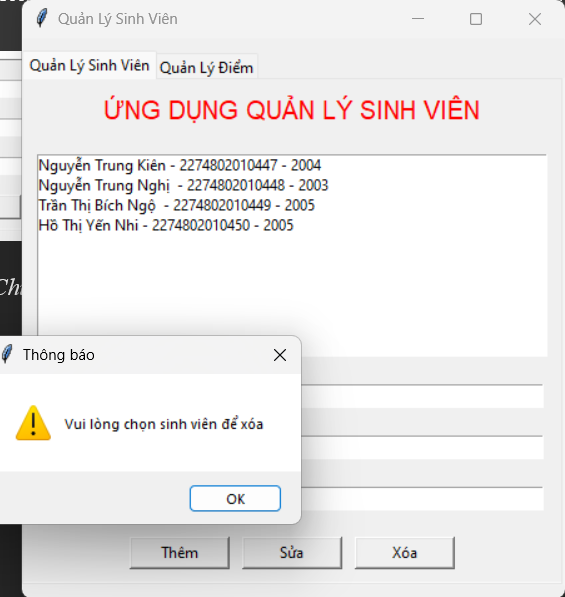
* Ở bước đầu tiên nếu không nhập đầy đủ các trường thông tin hay không chọn sinh viên để cập nhật hay thay đổi thì chương trình sẽ hiển thị thông báo sau:



Hình 5 Chức năng hiển thị thông báo

Chức năng xóa sinh viên:

Hình 6 Chức năng xóa sinh viên

* Ở chức năng xóa sinh viên nếu không chọn sinh viên để xóa thì chương trình sẽ hiện thông báo như sau:

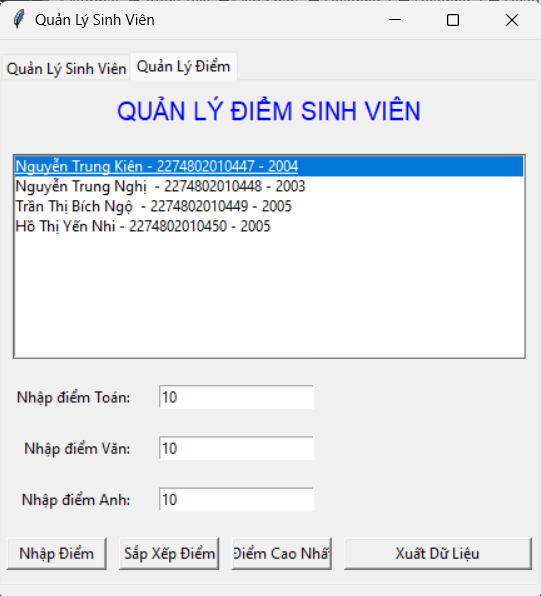
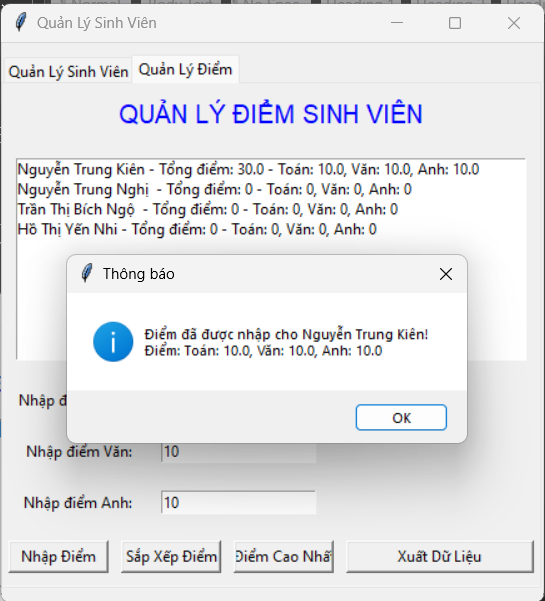
Hình 7 Chức năng xóa sinh viên

## **Tab "Quản Lý Điểm"**

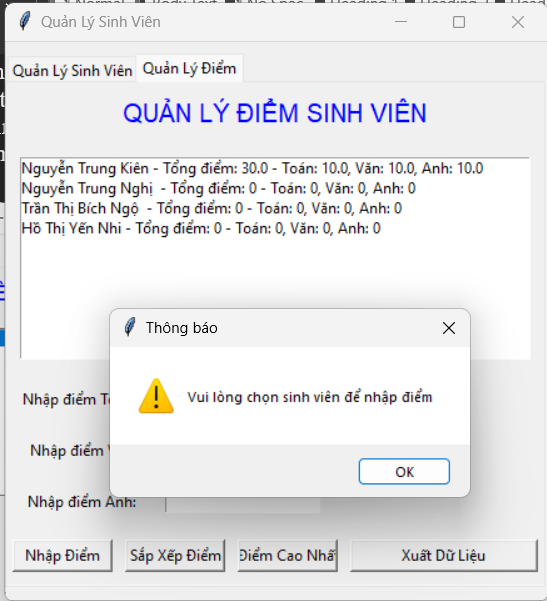
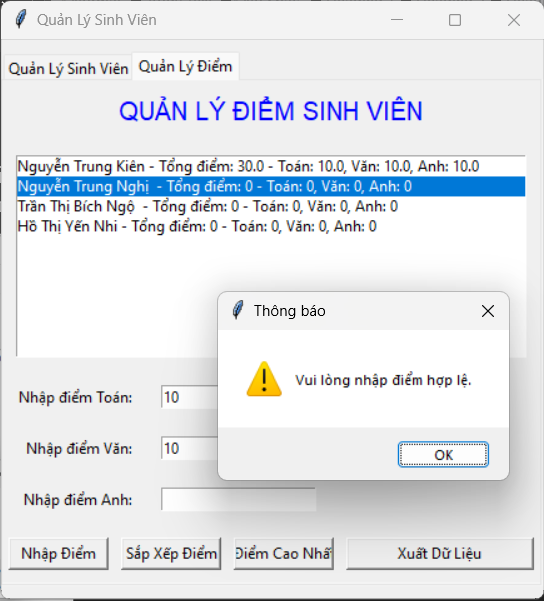
* Danh sách điểm sinh viên: Hiển thị danh sách điểm của các sinh viên đã được nhập vào ở tab Quản lý sinh viên. Đây là `Listbox` để hiển thị tên sinh viên và điểm các môn học.
* Nhập điểm Toán: Một ô nhập (`Entry`) để nhập điểm môn Toán của sinh viên.
* Nhập điểm Văn: Một ô nhập (`Entry`) để nhập điểm môn Văn của sinh viên.
* Nhập điểm Anh: Một ô nhập (`Entry`) để nhập điểm môn Anh văn của sinh viên.
* Nút "Nhập Điểm": Chức năng thêm điểm của một sinh viên vào hệ thống. Khi nhấn nút này, hệ thống sẽ lưu các điểm vừa nhập từ các ô "Điểm Toán", "Điểm Văn", và "Điểm Anh" cho sinh viên được chọn.
* Nút "Sắp Xếp Điểm": Chức năng sắp xếp danh sách sinh viên dựa trên điểm số của họ. Hệ thống có thể sắp xếp dựa trên tổng điểm hoặc điểm của một môn cụ thể (tùy thuộc vào cách triển khai).
* Nút "Điểm Cao Nhất": Chức năng tìm sinh viên có điểm số cao nhất trong danh sách. Hệ thống sẽ hiển thị sinh viên đạt điểm cao nhất trong các môn đã nhập.
* Nút "Xuất Dữ Liệu": Chức năng xuất dữ liệu điểm của sinh viên ra file hoặc dạng khác để lưu trữ hoặc in ấn. Hệ thống có thể lưu thông tin dưới dạng file CSV, Excel, hoặc văn bản đơn giản.

Chức năng nhập điểm: Đầu tiên chọn sinh viên cần nhập điểm ở Listbox, sau đó nhập điểm chính xác của từng môn vào các trường nhập điểm, sau đó click nút nhập điểm.

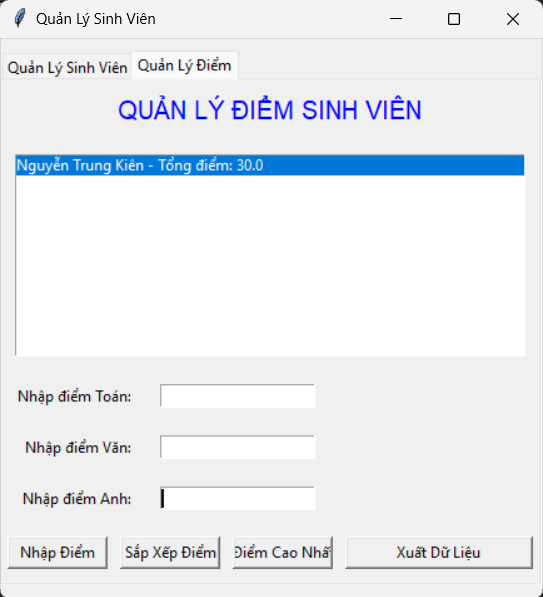
* Sau khi nhập điểm thành công chương trình sẽ hiển thị thông báo đã nhập điểm thành công cho sinh viên đã chọn và cập nhật điểm lên Listbox



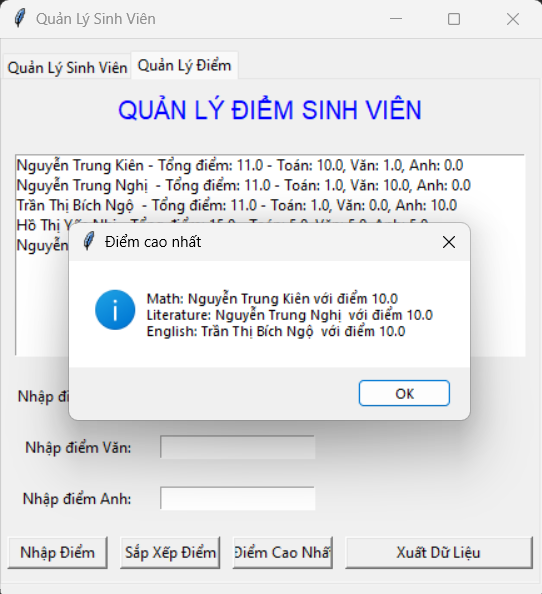
Hình 8 Chức năng nhập điểm

* Ở chức năng này nếu người dùng không chọn sinh viên để nhập điểm hay không nhập đầy đủ các trường điểm thì hệ thống sẽ hiển thị những thông báo như sau:

Hình 9 Hệ thống hiển thị thông báo yêu cầu nhập điểm hợp lệ, chọn sinh viên để nhập điểm

Chức năng sắp xếp điểm : Để thực hiện chức năng Sắp xếp điểm người dùng click vào nút Sắp Xếp Điểm, sau khi thực hiện chức năng, ở Listbox sẽ hiển thị sinh viên có số điểm trung bình cao nhất

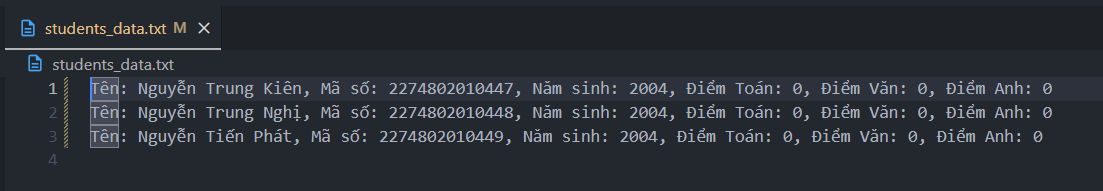
Hình 10 Chức năng sắp xếp điểm trung bình cao nhất

Chức năng xem điểm cao nhất theo môn: Ở chức năng này người dùng click vào bút Điểm Cao Nhất thì chương trình sẽ hiển thị thông báo chứa tên sinh viên với số điểm cao nhất ở từng môn.

Hình 11 Chức năng xuất dữ liệu

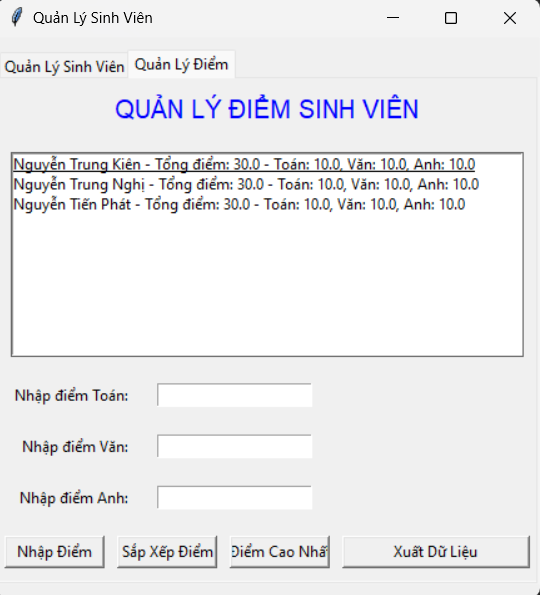
Chức năng xuất dữ liệu: ở chức năng này cho phép người dùng có thể xuất dữ liệu thành file có định dạng “.txt”, nội dung dữ liệu bao gồm (Họ tên sinh viên, mã số sinh viên, năm sinh, điểm toán, điểm văn, điểm anh).

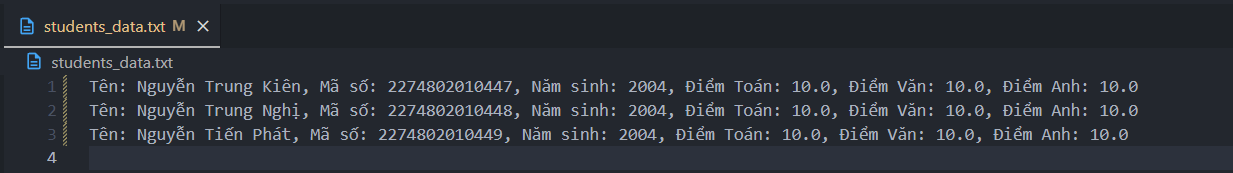
* Ở chức năng này nếu thực hiện chức năng xuất dữ liệu trước khi nhập điểm thì hệ thống sẽ xuất ra file có nội dung như sau:



Hình 12 Dữ liệu được xuất khi chưa nhập điểm

* Nếu người dùng thực hiện chức năng xuất dữ liệu sau khi thực hiện nhập điểm từng môn cho sinh viên thì hệ thống sẽ xuất ra file có nội dung như sau:





Hình 13 Dữ liệu được xuất ra khi đã nhập điểm

# **PHẦN 3 MÃ NGUỒN**

- Ứng dụng quản lý sinh viên bao gồm hai file chính:

`**main.py**`: Xây dựng giao diện bằng Tkinter với hai tab chính: "Quản Lý Sinh Viên" để thêm, sửa, xóa thông tin sinh viên và "Quản Lý Điểm" để nhập điểm, sắp xếp và xem điểm cao nhất. Giao diện cũng hỗ trợ xuất dữ liệu ra file.

`**database.py**`: Cung cấp chức năng xuất dữ liệu sinh viên và điểm số ra file `.txt`, giúp lưu trữ và quản lý thông tin dễ dàng.

Hai file này phối hợp để tạo ra một ứng dụng đơn giản và tiện dụng cho việc quản lý sinh viên.

Dưới đây là mã nguồn của hai file

## **File “main.py”:**

from tkinter import \*

from tkinter import messagebox

from tkinter import ttk

import database

from database import \*

# Khởi tạo cửa sổ

win = Tk()

win.title("Quản Lý Sinh Viên")

# Tạo Notebook để chứa các tab

notebook = ttk.Notebook(win)

notebook.pack(pady=10, expand=True)

# Tạo tab chính

main\_tab = Frame(notebook)

notebook.add(main\_tab, text="Quản Lý Sinh Viên")

# Tạo tab mới

new\_tab = Frame(notebook)

notebook.add(new\_tab, text="Quản Lý Điểm")

# biến rõng lưu dũ liệu sinh viên

students = []#bao gồm tên, mssv, năm sinh

scores = {}# bao gôm tên sinh viên và điểm của sinh viên đó

# Hàm thêm sinh viên

def add\_student():

    student\_name = name\_entry.get()

    student\_id = id\_entry.get()

    birth\_year = birth\_year\_entry.get()

    if student\_name and student\_id and birth\_year:

        student\_info = f"{student\_name} - {student\_id} - {birth\_year}"

        students.append((student\_name, student\_id, birth\_year))

        listbox.insert(END, student\_info)

        update\_student\_list()  # Cập nhật danh sách sinh viên trong tab 2

        name\_entry.delete(0, END)

        id\_entry.delete(0, END)

        birth\_year\_entry.delete(0, END)

    else:

        messagebox.showwarning("Thông báo", "Vui lòng nhập đầy đủ thông tin")

# Hàm sửa sinh viên

def edit\_student():

    selected\_index = listbox.curselection()#chọn sinh viên để sửa

    if selected\_index:

        student\_name = name\_entry.get()

        student\_id = id\_entry.get()

        birth\_year = birth\_year\_entry.get()

        if student\_name and student\_id and birth\_year:

            listbox.delete(selected\_index)

            updated\_info = f"{student\_name} - {student\_id} - {birth\_year}"

            students[selected\_index[0]] = (student\_name, student\_id, birth\_year)

            listbox.insert(selected\_index, updated\_info)

            update\_student\_list()  # Cập nhật danh sách sinh viên trong tab 2

            name\_entry.delete(0, END)

            id\_entry.delete(0, END)

            birth\_year\_entry.delete(0, END)

        else:

            messagebox.showwarning("Thông báo", "Vui lòng nhập đầy đủ thông tin")

    else:

        messagebox.showwarning("Thông báo", "Vui lòng chọn sinh viên để sửa")

# Hàm xóa sinh viên

def delete\_student():

    selected\_index = listbox.curselection()

    if selected\_index:

        listbox.delete(selected\_index)

        del students[selected\_index[0]]  # Xóa sinh viên từ danh sách

        update\_student\_list()  # Cập nhật danh sách sinh viên trong tab 2

    else:

        messagebox.showwarning("Thông báo", "Vui lòng chọn sinh viên để xóa")

# Hàm cập nhật danh sách sinh viên trong tab 2

def update\_student\_list():

    student\_listbox.delete(0, END)  # Xóa tất cả các mục trong Listbox

    for student in students:

        student\_info = f"{student[0]} - {student[1]} - {student[2]}"

        student\_listbox.insert(END, student\_info)

# Hàm nhập điểm cho sinh viên

def enter\_scores():

    selected\_index = student\_listbox.curselection()

    if selected\_index:

        student\_name = students[selected\_index[0]][0]  # Lấy tên sinh viên

        math\_score = math\_entry.get()

        literature\_score = literature\_entry.get()

        english\_score = english\_entry.get()

        # Kiểm tra xem điểm có hợp lệ không

        try:

            math\_score = float(math\_score)

            literature\_score = float(literature\_score)

            english\_score = float(english\_score)

        except ValueError:

            messagebox.showwarning("Thông báo", "Vui lòng nhập điểm hợp lệ.")

            return

        # Lưu điểm vào từ điển scores cho sinh viên được chọn

        scores[student\_name] = {

            'Math': math\_score,

            'Literature': literature\_score,

            'English': english\_score

        }

        # Cập nhật danh sách điểm trong Listbox

        update\_student\_scores\_listbox()

        messagebox.showinfo("Thông báo", f"Điểm đã được nhập cho {student\_name}!\nĐiểm: Toán: {math\_score}, Văn: {literature\_score}, Anh: {english\_score}")

        # Xóa nội dung trong các Entry sau khi nhập điểm

        math\_entry.delete(0, END)

        literature\_entry.delete(0, END)

        english\_entry.delete(0, END)

    else:

        messagebox.showwarning("Thông báo", "Vui lòng chọn sinh viên để nhập điểm")

# Hàm cập nhật danh sách sinh viên và điểm của họ trong Listbox

def update\_student\_scores\_listbox():

    student\_listbox.delete(0, END)  # Xóa tất cả các mục trong Listbox

    for student in students:

        student\_name = student[0]

        score\_info = scores.get(student\_name, {'Math': 0, 'Literature': 0, 'English': 0})

        total\_score = sum(score\_info.values())

        # Hiển thị tổng điểm và điểm từng môn

        student\_listbox.insert(END, f"{student\_name} - Tổng điểm: {total\_score} - Toán: {score\_info['Math']}, Văn: {score\_info['Literature']}, Anh: {score\_info['English']}")

# Hàm sắp xếp sinh viên theo điểm cao nhất

def sort\_by\_scores():

    sorted\_students = sorted(scores.items(), key=lambda x: sum(x[1].values()), reverse=True)

    student\_listbox.delete(0, END)  # Xóa tất cả các mục trong Listbox

    for student in sorted\_students:

        student\_listbox.insert(END, f"{student[0]} - Tổng điểm: {sum(student[1].values())}")

# Hàm lọc điểm cao nhất theo môn học

def highest\_score\_by\_subject():

    highest\_scores = {}

    for student, score in scores.items():

        for subject, value in score.items():

            if subject not in highest\_scores or value > highest\_scores[subject][1]:

                highest\_scores[subject] = (student, value)

    messagebox.showinfo("Điểm cao nhất", "\n".join([f"{subject}: {info[0]} với điểm {info[1]}" for subject, info in highest\_scores.items()]))

def export\_data():

    filename = "students\_data.txt"  # Tên file xuất ra

    database.export\_data\_to\_txt(students, scores, filename)

# Tạo Label tiêu đề cho tab quản lý sinh viên

title\_label = Label(main\_tab, text="ỨNG DỤNG QUẢN LÝ SINH VIÊN", fg="red", font="helvetica", width=30)

title\_label.grid(row=0, column=0, columnspan=3, padx=10, pady=10)

# Tạo Listbox để hiển thị danh sách sinh viên

listbox = Listbox(main\_tab, width=50, height=10)

listbox.grid(row=1, column=0, columnspan=3, padx=10, pady=10, sticky="nsew")

# Tạo Entry để nhập tên sinh viên

name\_label = Label(main\_tab, text="Nhập tên Sinh Viên:")

name\_label.grid(row=2, column=0, padx=10, pady=10, sticky="e")

name\_entry = Entry(main\_tab, width=40)

name\_entry.grid(row=2, column=1, padx=10, pady=10, sticky="w")

# Tạo Entry để nhập mã số sinh viên

id\_label = Label(main\_tab, text="Nhập mã số Sinh Viên:")

id\_label.grid(row=3, column=0, padx=10, pady=10, sticky="e")

id\_entry = Entry(main\_tab, width=40)

id\_entry.grid(row=3, column=1, padx=10, pady=10, sticky="w")

# Tạo Entry để nhập năm sinh sinh viên

birth\_year\_label = Label(main\_tab, text="Nhập năm sinh Sinh Viên:")

birth\_year\_label.grid(row=4, column=0, padx=10, pady=10, sticky="e")

birth\_year\_entry = Entry(main\_tab, width=40)

birth\_year\_entry.grid(row=4, column=1, padx=10, pady=10, sticky="w")

# Tạo các Button nằm ngang nhau trên cùng một dòng và có kích thước đồng đều

button\_frame = Frame(main\_tab)

button\_frame.grid(row=5, column=0, columnspan=3, pady=10)

add\_button = Button(button\_frame, text="Thêm", command=add\_student, width=10)

add\_button.grid(row=0, column=0, padx=5)

edit\_button = Button(button\_frame, text="Sửa", command=edit\_student, width=10)

edit\_button.grid(row=0, column=1, padx=5)

delete\_button = Button(button\_frame, text="Xóa", command=delete\_student, width=10)

delete\_button.grid(row=0, column=2, padx=5)

# Cấu hình lưới để giãn nở Listbox

main\_tab.grid\_rowconfigure(1, weight=1)

main\_tab.grid\_columnconfigure(1, weight=1)

# Tab quản lý điểm

scores\_title\_label = Label(new\_tab, text="QUẢN LÝ ĐIỂM SINH VIÊN", fg="blue", font="helvetica", width=30)

scores\_title\_label.grid(row=0, column=0, columnspan=3, padx=10, pady=10)

student\_listbox = Listbox(new\_tab, width=60, height=10)

student\_listbox.grid(row=1, column=0, columnspan=3, padx=10, pady=10, sticky="nsew")

# Tạo Label và Entry cho các môn học

math\_label = Label(new\_tab, text="Nhập điểm Toán:")

math\_label.grid(row=2, column=0, padx=10, pady=10, sticky="e")

math\_entry = Entry(new\_tab, width=20)

math\_entry.grid(row=2, column=1, padx=10, pady=10, sticky="w")

literature\_label = Label(new\_tab, text="Nhập điểm Văn:")

literature\_label.grid(row=3, column=0, padx=10, pady=10, sticky="e")

literature\_entry = Entry(new\_tab, width=20)

literature\_entry.grid(row=3, column=1, padx=10, pady=10, sticky="w")

english\_label = Label(new\_tab, text="Nhập điểm Anh:")

english\_label.grid(row=4, column=0, padx=10, pady=10, sticky="e")

english\_entry = Entry(new\_tab, width=20)

english\_entry.grid(row=4, column=1, padx=10, pady=10, sticky="w")

# Tạo các Button nằm ngang nhau trên cùng một dòng và có kích thước đồng đều cho tab điểm

scores\_button\_frame = Frame(new\_tab)

scores\_button\_frame.grid(row=5, column=0, columnspan=3, pady=10)

enter\_scores\_button = Button(scores\_button\_frame, text="Nhập Điểm", command=enter\_scores, width=10)

enter\_scores\_button.grid(row=0, column=0, padx=5)

sort\_scores\_button = Button(scores\_button\_frame, text="Sắp Xếp Điểm", command=sort\_by\_scores, width=10)

sort\_scores\_button.grid(row=0, column=1, padx=5)

highest\_score\_button = Button(scores\_button\_frame, text="Điểm Cao Nhất", command=highest\_score\_by\_subject, width=10)

highest\_score\_button.grid(row=0, column=2, padx=5)

export\_button = Button(scores\_button\_frame, text="Xuất Dữ Liệu", command=export\_data, width=20)

export\_button.grid(row=0, column=3, padx=5)

# Cấu hình lưới để giãn nở Listbox

new\_tab.grid\_rowconfigure(1, weight=1)

new\_tab.grid\_columnconfigure(1, weight=1)

# Chạy ứng dụng

win.mainloop()

## **File “database.py”:**

# database.py

def export\_data\_to\_txt(students, scores, filename):

    with open(filename, 'w', encoding='utf-8') as file:

        for student in students:

            student\_name = student[0]

            student\_id = student[1]

            birth\_year = student[2]

            score\_info = scores.get(student\_name, {'Math': 0, 'Literature': 0, 'English': 0})

            file.write(f"Tên: {student\_name}, Mã số: {student\_id}, Năm sinh: {birth\_year}, "

                       f"Điểm Toán: {score\_info['Math']}, Điểm Văn: {score\_info['Literature']}, "

                       f"Điểm Anh: {score\_info['English']}\n")

    print(f"Dữ liệu đã được xuất ra file {filename}.")

# Gọi hàm để xuất dữ liệu

if \_\_name\_\_ == "\_\_main\_\_":

    # Mô phỏng dữ liệu để xuất

    students = [("Nguyen Van A", "123456", "2000"), ("Tran Thi B", "654321", "1999")]

    scores = {

        "Nguyen Van A": {'Math': 8.5, 'Literature': 7.0, 'English': 9.0},

        "Tran Thi B": {'Math': 9.0, 'Literature': 8.5, 'English': 7.5}

    }

    export\_data\_to\_txt(students, scores, "student\_data.txt")

Phần 4 Github