

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP

KHOA ĐIỆN TỬ

Bộ Môn: Công Nghệ Thông Tin



BÀI TẬP LỚN
LẬP TRÌNH PYTHON

Sinh viên : Nguyễn Trung Kiên

Mssv : K225480106032

GVGD : TS : Nguyễn Văn Huy

Linkgithub : https://github.com/kien2511/bai_tap_cuoi_ky_python

Link youtube : <https://youtu.be/pOAO8b0bAuM>

THÁI NGUYÊN - 2025

TRƯỜNG DHKTCN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

KHOA ĐIỆN TỬ

Độc lập - Tự do - Hạnh phúc

BÀI TẬP KẾT THÚC MÔN HỌC

MÔN HỌC : LẬP TRÌNH PYTHON

BỘ MÔN : CÔNG NGHỆ THÔNG TIN

Sinh viên : Nguyễn Trung Kiên

Lớp : K58

Nghành : Kỹ Thuật Máy Tính

Giáo viên hướng dẫn : TS Nguyễn Văn Huy

Ngày giao đề : 20/05/2025

Ngày hoàn thành : 08/06/2025

Tên đề tài : **bài tập số 3: Trivia Challenge có GUI & file I/O**

Yêu cầu :

Đọc file, bắt lỗi file không tồn tại hoặc format sai.

GUI: Label câu hỏi, Entry đáp án, nút “Nộp”, Label điểm.

Tính điểm, chuyển câu hỏi kế tiếp.

Nút “Kết thúc” hiển thị tổng điểm.

GIẢNG VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

Lời Cam Đoan

Em xin cam đoan bài tập lớn môn *Python* với đề tài "**Bài 3. Trivia Challenge có GUI & file I/O**" là sản phẩm do chính em thực hiện. Trong quá trình làm bài, em đã tuân thủ đúng quy định về đạo đức học thuật, không sao chép, sử dụng hoặc lấy lại sản phẩm của người khác dưới bất kỳ hình thức nào.

Mọi tài liệu, nguồn tham khảo (nếu có) đều đã được em trích dẫn rõ ràng trong phần tài liệu tham khảo của bài làm. Nếu phát hiện có sự gian lận hoặc vi phạm quy định, em xin hoàn toàn chịu trách nhiệm trước nhà trường và giảng viên bộ môn.

Sinh viên thực hiện

Họ và tên: Nguyễn Trung Kiên

Mã sinh viên: K225480106032

Lớp: K58KTP.01

Ngày cam đoan: 08/06/2025

(Ký và ghi rõ họ tên)

LỜI CẢM ƠN

Em xin chân thành cảm ơn Thầy Nguyễn Văn Huy đã tận tình giảng dạy và hướng dẫn em trong suốt quá trình thực hiện bài tập lớn môn Python. Sự hỗ trợ và chỉ dẫn của Thầy là nguồn động lực quan trọng giúp em hoàn thành bài làm này.

Sinh viên thực hiện

Họ và tên: Nguyễn Trung Kiên

Mã sinh viên: K22548106032

NHẬN XÉT CỦA GIÁO VIÊN CHĂM

.....

.....

.....

.....

.....

Thái nguyên , ngày..., tháng ...,năm 2025

GIÁO VIÊN HƯỚNG DẪN

(ký và ghi rõ họ tên)

MỤC LỤC

<i>Mục lục :</i>	1
<i>Lời nói đầu :</i>	3
<i>Chương 1: Giới thiệu đầu bài</i>	3
1.1 đặt vấn đề :	3
1.2 mô tả bài toán :	3
1.3 tính năng chính :	3
1.4 thách thức đặt ra :	3
1.5 kiến thức vận dụng :	4
<i>Chương 2: Cơ sở lý thuyết</i>	5
2.1 xử lý file trong python(I/O) :	5
2.2 giao diện đồ họa với Tkinter :	5
2.3 lập trình hướng đối tượng OOP :	5
2.4 kiểu dữ liệu chuỗi và xử lý nhập liệu :	6
2.5 xử lý điều kiện và vòng lặp :	6
<i>Chương 3 : Thiết kế và xây dựng chương trình</i>	7
3.1 sơ đồ khối hệ thống :	7
3.2 sơ đồ các thuật toán chính :	10
3.3 cấu trúc dữ liệu :	12
3.4 chương trình :	13
<i>Chương 4 : Thực nghiệm và kết luận</i>	17
4.1 thực nghiệm :	17
4.2 kết luận :	21
4.3 tài liệu tham khảo :	22

LỜI NÓI ĐẦU

Trong thời đại công nghệ phát triển mạnh mẽ, việc xây dựng các ứng dụng tương tác với người dùng thông qua giao diện đồ họa (GUI) ngày càng trở nên phổ biến và cần thiết. Đặc biệt, các trò chơi giáo dục đơn giản như "Trivia Challenge" (Thử thách kiến thức) không chỉ giúp người học ôn tập kiến thức một cách hiệu quả mà còn là bài tập lập trình thực tế để rèn luyện kỹ năng về xử lý file, giao diện người dùng, và tư duy lập trình.

Chương trình **Trivia Challenge** được xây dựng bằng ngôn ngữ lập trình Python, sử dụng thư viện **Tkinter** để thiết kế giao diện người dùng thân thiện, dễ sử dụng. Bên cạnh đó, chương trình còn kết hợp với kỹ thuật **xử lý tệp tin (File I/O)** để lưu trữ và truy xuất dữ liệu các câu hỏi từ file văn bản. Điều này giúp nội dung câu hỏi có thể dễ dàng thay đổi hoặc mở rộng mà không cần sửa mã nguồn chương trình.

Thông qua việc phát triển ứng dụng này, người thực hiện không chỉ củng cố được các kiến thức nền tảng về Python mà còn tiếp cận với cách tổ chức chương trình theo hướng đối tượng (OOP), làm việc với sự kiện trong GUI, và đảm bảo tính linh hoạt, dễ mở rộng cho phần mềm. Đây là một bước chuẩn bị quan trọng cho các dự án lập trình lớn hơn trong tương lai.

CHƯƠNG 1 : GIỚI THIỆU ĐẦU BÀI

1.1 Đặt vấn đề

Trong quá trình học tập, việc ôn luyện và kiểm tra kiến thức qua các trò chơi dạng câu hỏi – đáp (quiz) là một phương pháp hiệu quả giúp tăng sự hứng thú và khả năng ghi nhớ cho người học. Trên cơ sở đó, bài tập xây dựng chương trình **Trivia Challenge** đã được giao nhằm rèn luyện khả năng lập trình, thiết kế giao diện người dùng và thao tác với dữ liệu bên ngoài.

1.2 Mô tả bài toán

Chương trình "Trivia Challenge" là một ứng dụng trò chơi hỏi – đáp dạng trắc nghiệm đơn giản. Ứng dụng sẽ lần lượt đọc từng cặp câu hỏi và đáp án từ một tệp văn bản có định dạng sẵn. Sau đó, người dùng sẽ nhập câu trả lời vào giao diện, chương trình sẽ kiểm tra và phản hồi đúng/sai tương ứng, đồng thời cập nhật điểm số.

1.3 Tính năng chính

Chương trình cần đảm bảo chức năng :

- Cho phép người dùng nhập tên tệp câu hỏi khi bắt đầu chơi
- Đọc dữ liệu câu hỏi từ file văn bản với định dạng mỗi câu hỏi gồm : dòng câu hỏi , dòng đáp án , dòng trống ngăn cách.
- Hiện thị từng câu hỏi trên giao diện đồ họa (GUI)
- Nhận câu trả lời từ người dùng và so sánh đáp án đúng
- Cập nhật và hiện thị điểm số theo từng câu trả lời.
- Kết thúc trò chơi khi hết câu hỏi và hiện thị tổng điểm

1.4 Thách thức đặt ra

Một số thách thức trong quá trình thực hiện chương trình:

- Thiết kế giao diện đơn giản nhưng trực quan và dễ sử dụng bằng thư viện Tkinter
- Đảm bảo việc xử lý file ổn định, có kiểm tra lỗi khi file không tồn tại hoặc không đúng định dạng
- So sánh câu trả lời không phân biệt hoa - thường , và cần xử lý nhập liệu đầu vào hợp lý
- Quản lý luồng câu hỏi liên tiếp một cách logic , đảm bảo không lặp hoặc bỏ sót

1.5 Kiến thức vận dụng

- Ngôn ngữ lập trình python : sử dụng cú pháp , hàm, biến , cấu trúc điều kiện - lặp và lập trình hướng đối tượng .
- Tkinter : xây dựng giao diện đồ họa , xử lý sự kiện (event) , tương tác với người dùng.
- Xử lý file (file I/O) : mở file , đọc từng dòng , kiểm tra lỗi và đóng file đúng cách.
- Thông báo hộp thoại (messagebox , simpledialog) : hỗ trợ phản hồi và nhập thông tin từ người dùng.

CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT

Trong quá trình xây dựng chương trình “Trivia Challenge”, người lập trình cần vận dụng nhiều kiến thức chuyên môn từ ngôn ngữ python và các thư viện tiêu chuẩn.

2.1 Xử lý file trong python (file I/O)

Xử lý tệp là một trong những kỹ năng cơ bản và quan trọng trong lập trình. Trong chương trình, các câu hỏi và đáp án được lưu trữ trong tệp văn bản định dạng .txt, và được đọc ra từng dòng.

- *Open(filename , mode)* : dùng để mở tệp . ví dụ *open(“trivia.txt” , “r” , encoding = “utf-8”)*.
- *Readline ()* : đọc từng dòng một từ file .
- *Strip ()* : xóa ký tự xuống dòng và khoảng trắng thừa ở đầu / cuối dòng.
- *Try - except* : xử lý ngoại lệ , ví dụ như không tìm thấy tệp (*FileNotFoundException*)

2.2 Giao diện đồ họa với Tkinter

Tkinter là thư viện tiêu chuẩn trong Python dùng để xây dựng giao diện người dùng (GUI). Một số thành phần (widget) cơ bản đã được sử dụng:

- *Tk()* : tạo cửa sổ chính.
- *Label* : hiển thị văn bản tĩnh như câu hỏi và điểm số.
- *Entry* : cho phép người dùng nhập dữ liệu .
- *Button* : tạo các nút như “nộp” , “kết thúc”.
- *Messagebox* : hiển thị thông báo (đúng, sai , điểm số, ..).
- *Simpdialog.askstring()* : yêu cầu người dùng nhập tên file đầu vào.

2.3 Lập trình hướng đối tượng (OOP)

Chương trình được thiết kế theo hướng đối tượng với lớp TriviaGUI chứa toàn bộ logic giao diện và xử lý sự kiện. Một số khái niệm liên quan:

- *Lớp (class)*: Mô hình hóa đối tượng gồm thuộc tính và phương thức.
- *Phương thức khởi tạo (__init__)*: Hàm đặc biệt chạy khi khởi tạo đối tượng.
- *Phương thức (method)*: Các hành động như *submit_answer*, *next_question*.

2.4 Kiểu dữ liệu chuỗi và xử lý nhập liệu

- *So sánh chuỗi*: để tránh phân biệt chữ hoa – thường, sử dụng *.lower()* để so sánh.
- Kiểm tra nhập liệu rỗng, chuẩn hóa khoảng trắng bằng *.strip()*.

2.5 Xử lý điều kiện và vòng lặp

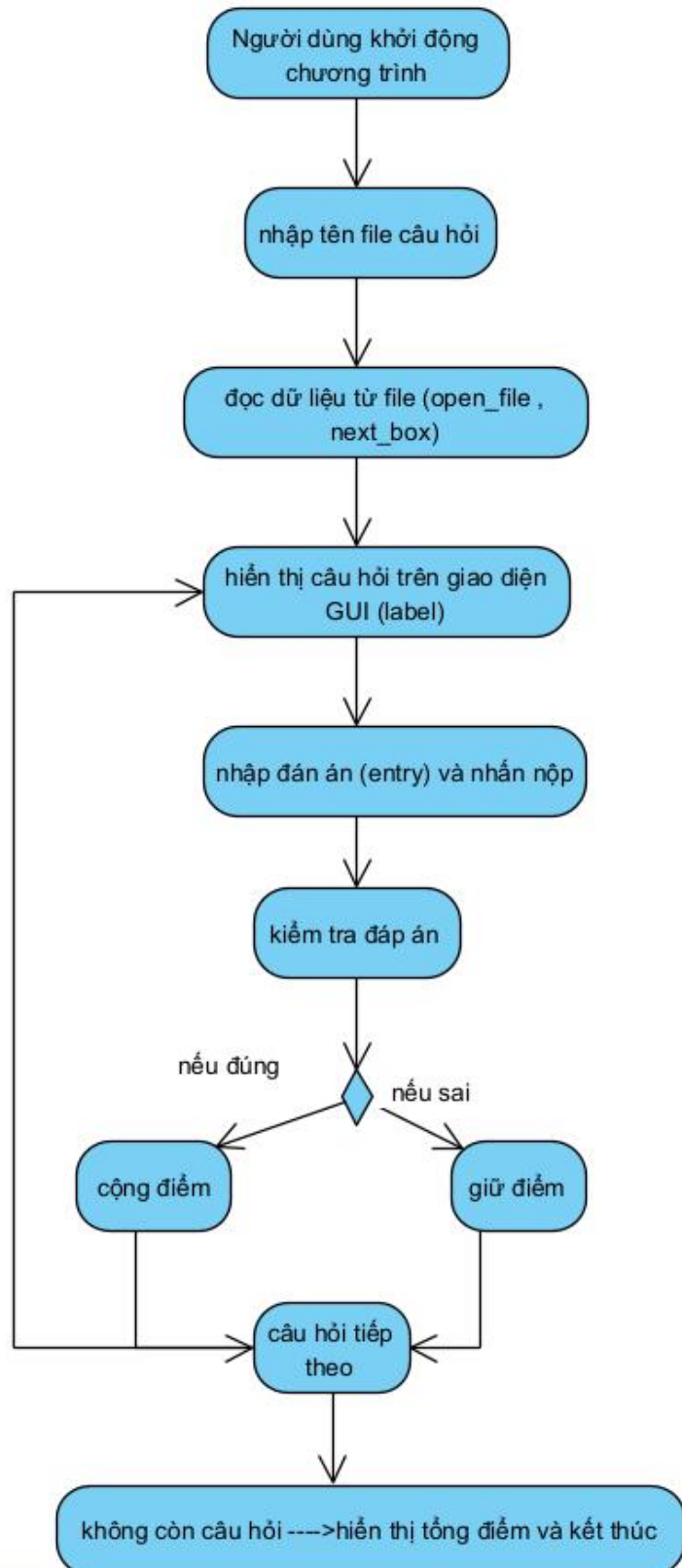
- Câu lệnh điều kiện *if-else*: Kiểm tra đáp án đúng/sai.
- Vòng lặp qua từng câu hỏi: Được thực hiện bằng việc gọi liên tục hàm *next_block()* cho đến khi hết câu hỏi.

CHƯƠNG 3 : THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH

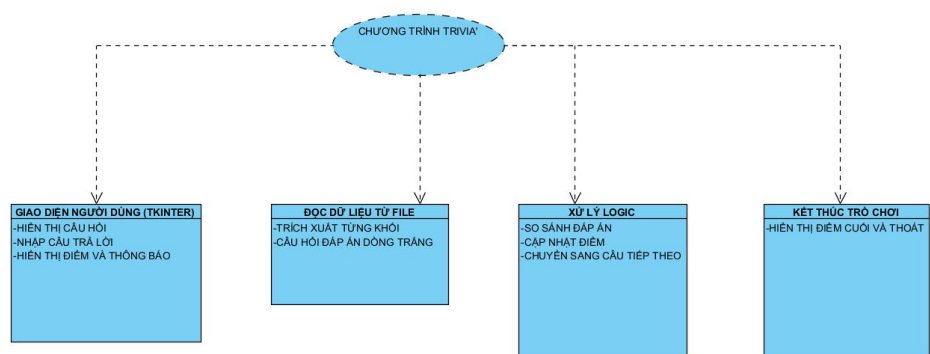
3.1 Sơ đồ khối hệ thống

Hệ thống chương trình Trivia Challenge được chia thành các module chính như sau:

- *Giao diện người dùng (GUI)*: Xây dựng bằng thư viện tkinter, gồm các nút, ô nhập và nhãn để tương tác với người dùng.
- *Xử lý file đầu vào*: Đọc dữ liệu câu hỏi từ file văn bản.
- *Xử lý câu trả lời*: So sánh câu trả lời người dùng nhập với đáp án đúng.
- *Quản lý điểm*: Tăng điểm nếu người dùng trả lời đúng và hiển thị điểm tổng khi kết thúc.

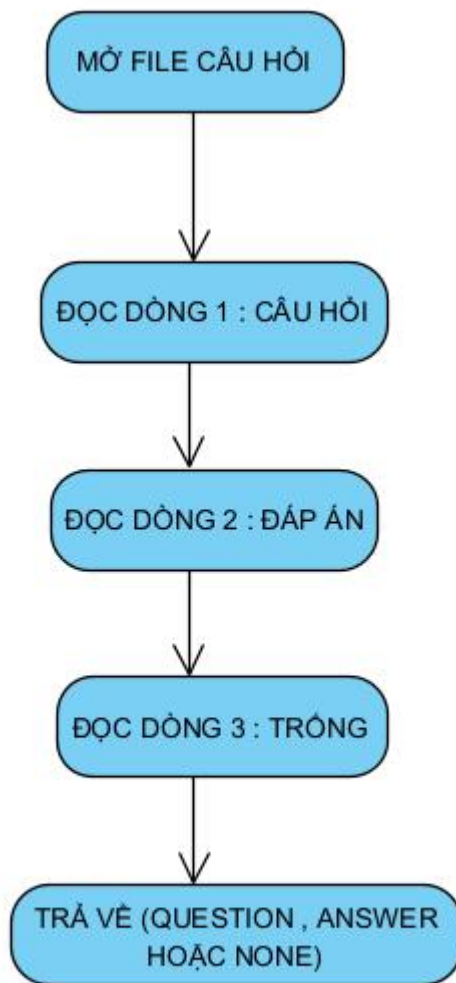


3.1.1 sơ đồ khối hệ thống

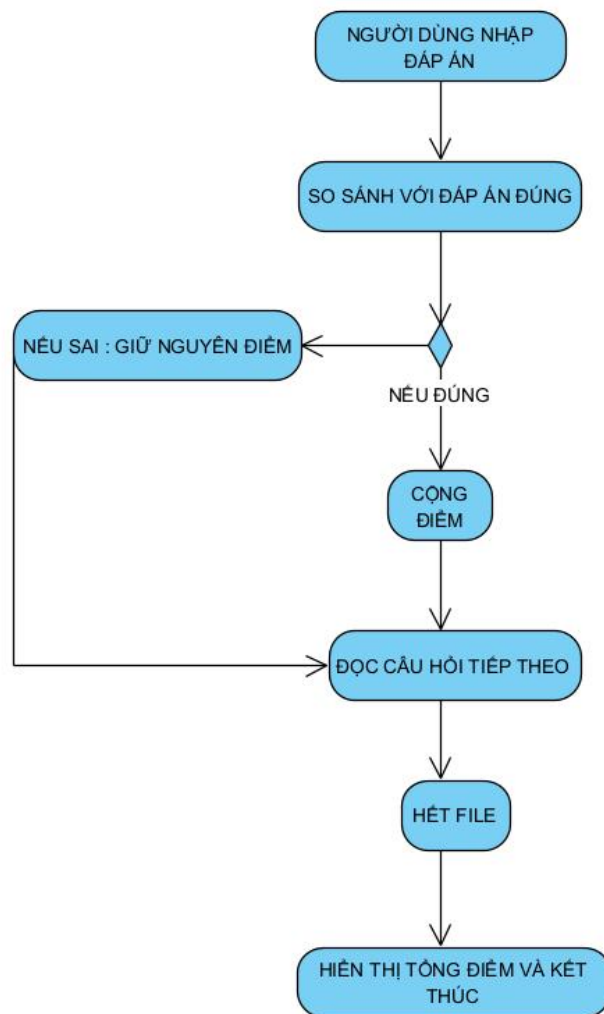


3.1.2 Biểu đồ phân cấp chức năng

3.2 Sơ đồ các thuật toán chính



3.2.1 Khởi xử lý



3.2.2 Sơ đồ thuật toán xử lý câu hỏi

3.3 Cấu trúc dữ liệu

3.3.1 dữ liệu câu hỏi lưu trữ trong tệp văn bản *trivia.txt*

Trong chương trình , toàn bộ dữ liệu câu hỏi được lưu trữ dưới dạng văn bản với ,mỗi câu hỏi gồm 3 dòng liên tiếp như sau :

- Dòng 1 : câu hỏi
- Dòng 2 : câu trả lời đúng
- Dòng 3 : Dòng trống để phân tách

Trong nội dung *Trivia.txt* :

What is 2+2?

4

1

What is the capital of France?

Paris

1

3.3.2 Cấu trúc dữ liệu trong chương trình

Sau khi đọc file, chương trình chuyển đổi dữ liệu từ dạng văn bản sang danh sách các bộ 3 (tuple), mỗi phần tử có dạng:

(*question* : *str* , *answer*: *str*) lưu vào thuộc tính *seft.question* và *seft.answer* để hiển thị và xử lý

Danh sách kết quả sau khi xử lý :

[(“ What is 2+2?” , “ 4 “ , “1”) ,
 (“What is the capital of France” , “paris” , “1”)]

3.4 chương trình

```
import tkinter as tk
from tkinter import messagebox, simpledialog
import os
import random
import time

# --- File I/O functions ---
def open_file(filename):
    try:
        return open(filename, "r", encoding="utf-8")
    except FileNotFoundError:
        messagebox.showerror("Lỗi", f"Không tìm thấy file: {filename}")
        return None

def next_block(file):
    while True:
        question = file.readline()
        if question == "":
            return None, None
        question = question.strip()
        if question != "":
            break
    while True:
        answer = file.readline()
        if answer == "":
            return None, None
        answer = answer.strip()
        if answer != "":
            break
    file.readline() # dòng trống phân cách
    return question, answer

# --- Logic class (Model / Controller) ---
class TriviaGame:
    def __init__(self, filename):
        self.filename = filename
        self.file = open_file(self.filename)
        if not self.file:
            raise FileNotFoundError("Không mở được file câu hỏi.")
        self.score = 0
        self.question, self.answer = next_block(self.file)

    def check_answer(self, user_input):
        correct = user_input.strip().lower() == self.answer.lower()
        if correct:
            self.score += 1
        return correct, self.answer
```

```

def next_question(self):
    self.question, self.answer = next_block(self.file)
    return self.question is not None

def close(self):
    if self.file:
        self.file.close()

# --- GUI class (View) ---
class TriviaGUI:
    def __init__(self, master, game: TriviaGame):
        self.master = master
        self.master.title(" Trivia Challenge ")
        self.master.geometry("700x550")
        self.master.configure(bg="#fffaf0")
        self.game = game

        self.create_widgets()
        self.show_question()

    def create_widgets(self):
        self.lbl_title = tk.Label(self.master, text=" Thử thách
kiến thức ", font=("Comic Sans MS", 22, "bold"),
                                bg="#fffaf0", fg="#ff4500")
        self.lbl_title.pack(pady=15)

        self.frame = tk.Frame(self.master, bg="#fffaf0")
        self.frame.pack(pady=10)

        self.lbl_score = tk.Label(self.frame, text=f"★ Điểm:
{self.game.score}", font=("Verdana", 13, "bold"),
                                bg="#fffaf0", fg="#228b22")
        self.lbl_score.pack(pady=5)

        self.lbl_question = tk.Label(self.frame, text="",
wraplength=600, font=("Helvetica", 15),
                                bg="#fffaf0", fg="#000080",
justify="center")
        self.lbl_question.pack(pady=15)

        self.entry_answer = tk.Entry(self.frame, font=("Helvetica",
13), width=45, justify="center", bd=4, relief="sunken")
        self.entry_answer.pack(pady=10)
        self.entry_answer.bind("<Return>", lambda e:
self.submit_answer())
        self.entry_answer.focus()

        self.button_frame = tk.Frame(self.master, bg="#fffaf0")

```

```

self.button_frame.pack(pady=15)

self.btn_submit = tk.Button(self.button_frame, text="✓  
Nộp", font=("Helvetica", 13, "bold"),  
                             command=self.submit_answer,  
                             bg="#4CAF50", fg="white", width=14, height=2)  
self.btn_submit.grid(row=0, column=0, padx=15)

self.btn_end = tk.Button(self.button_frame, text="⊖ Kết  
thúc", font=("Helvetica", 13, "bold"),  
                         command=self.end_game,  
                         bg="#f44336", fg="white", width=14, height=2)  
self.btn_end.grid(row=0, column=1, padx=15)

self.canvas = tk.Canvas(self.master, width=700, height=180,  
                         bg="#ffffaf", highlightthickness=0)  
self.canvas.pack()

def show_question(self):  
    self.lbl_question.config(text=self.game.question)  
    self.entry_answer.delete(0, tk.END)  
    self.entry_answer.focus()

def submit_answer(self):  
    user_input = self.entry_answer.get()  
    correct, correct_answer =  
self.game.check_answer(user_input)  
    self.lbl_score.config(text=f"★ Điểm: {self.game.score}")  
    if correct:  
        self.animated_correct_feedback()  
    else:  
        messagebox.showinfo("✗ Sai rồi!", f"Đáp án đúng là:  
{correct_answer}")  
        if self.game.next_question():  
            self.show_question()  
        else:  
            self.end_game()

def animated_correct_feedback(self):  
    for i in range(5):  
        self.master.update()  
        self.lbl_question.config(fg=random.choice(["green",  
"blue", "orange", "purple"]))  
        time.sleep(0.1)  
        self.lbl_question.config(fg="#000080")  
        messagebox.showinfo(" Đúng rồi!", "Chính xác! Bạn thật  
tuyệt!")

def end_game(self):

```

```

        self.canvas.delete("all")
        self.canvas.create_text(350, 90, text=f" Trò chơi kết
thúc! Điểm số: {self.game.score}",
                                font=("Helvetica", 18, "bold"),
fill="#8b0000")
        self.master.after(2000, self.master.destroy)
        self.game.close()

# --- Main ---
if __name__ == "__main__":
    root = tk.Tk()
    filename = simpdialog.askstring(" File câu hỏi", "Nhập tên
file câu hỏi:", initialvalue="trivia.txt")
    if filename:
        try:
            game = TriviaGame(filename)
            app = TriviaGUI(root, game)
            root.mainloop()
        except FileNotFoundError:
            pass

```

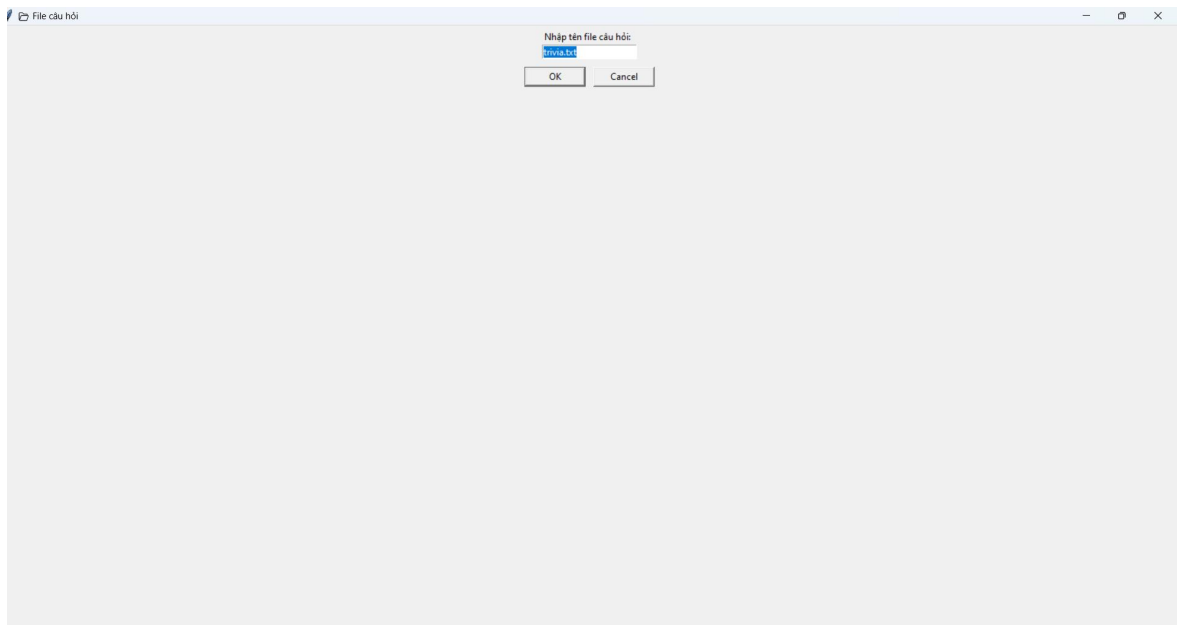
CHƯƠNG 4 : THỰC NGHIỆM VÀ KẾT LUẬN

4.1 Thực nghiệm

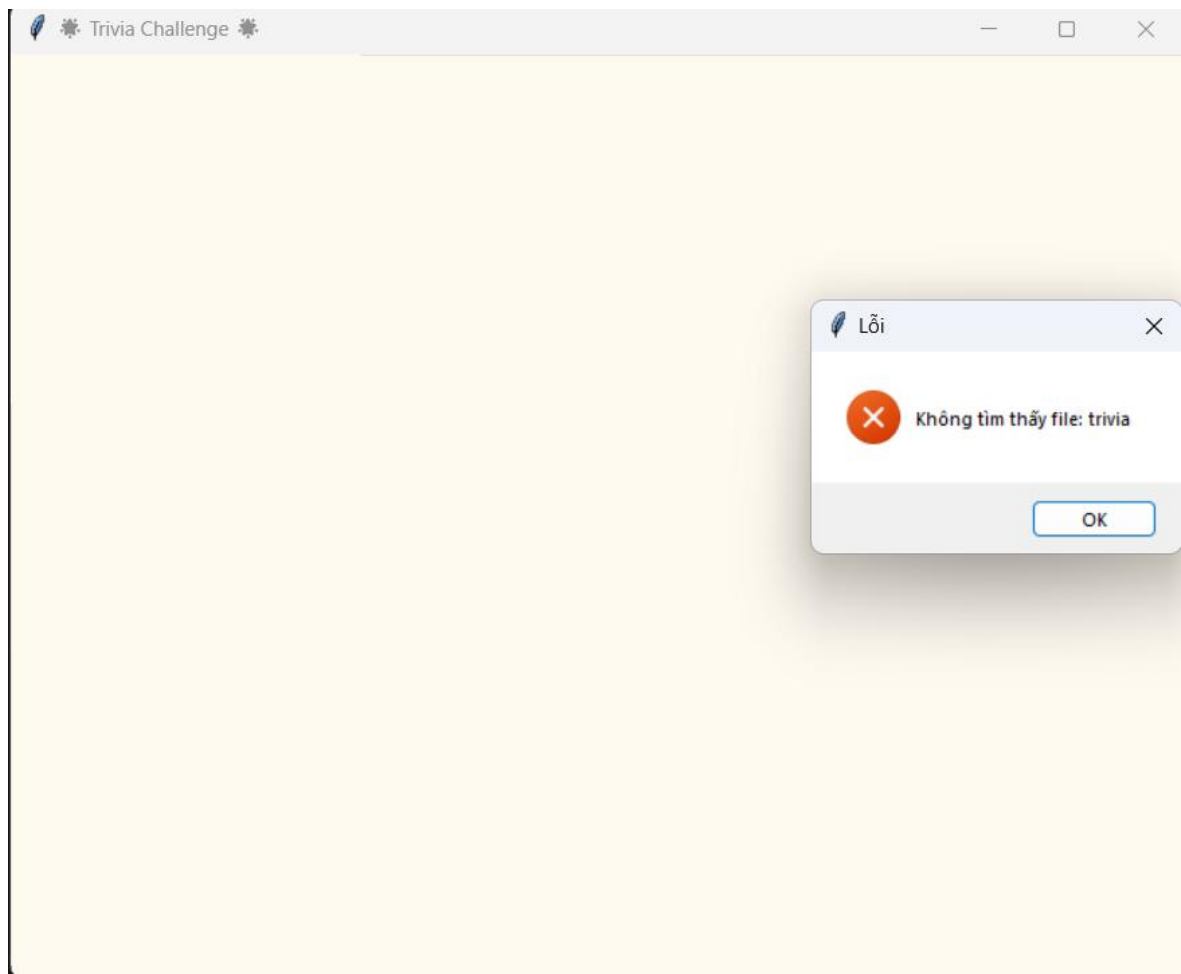
Mục tiêu :

Thực thi chương trình, kiểm tra tính năng, đảm bảo hoạt động đúng theo yêu cầu.

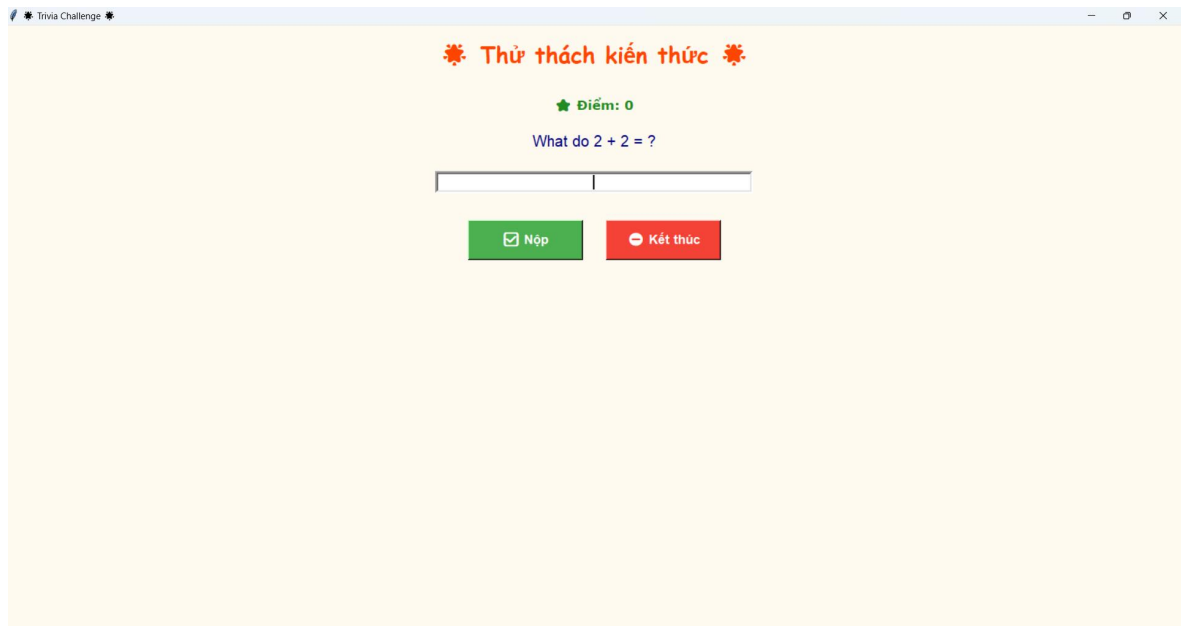
Ảnh minh họa



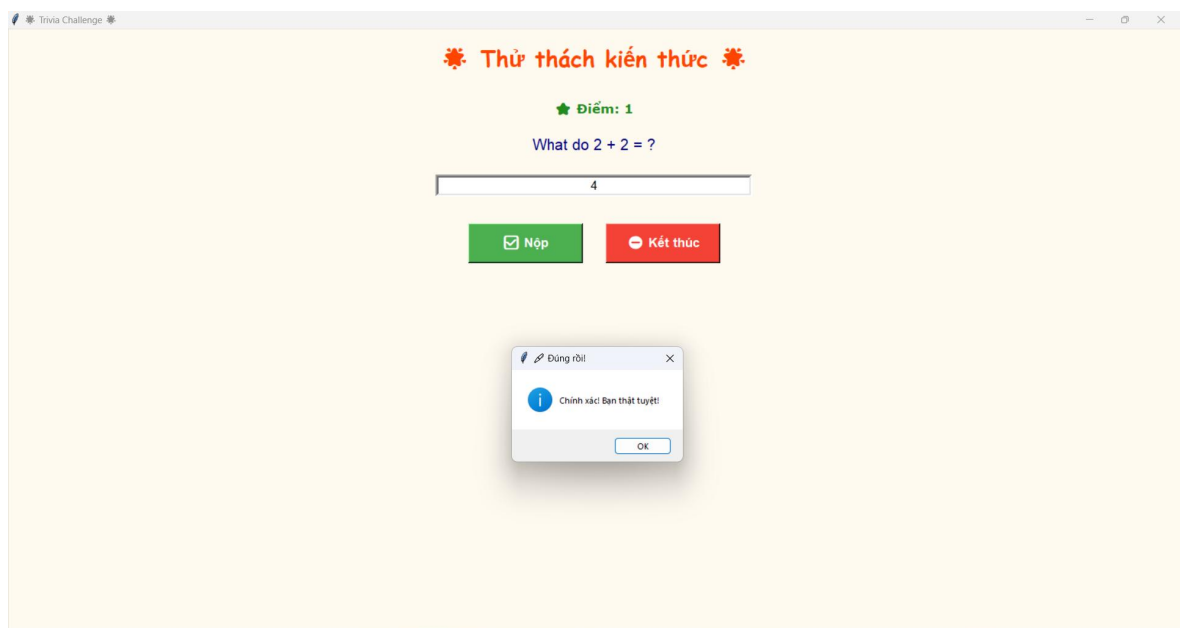
Nhập tên file đúng , chương trình mở và đọc nội dung



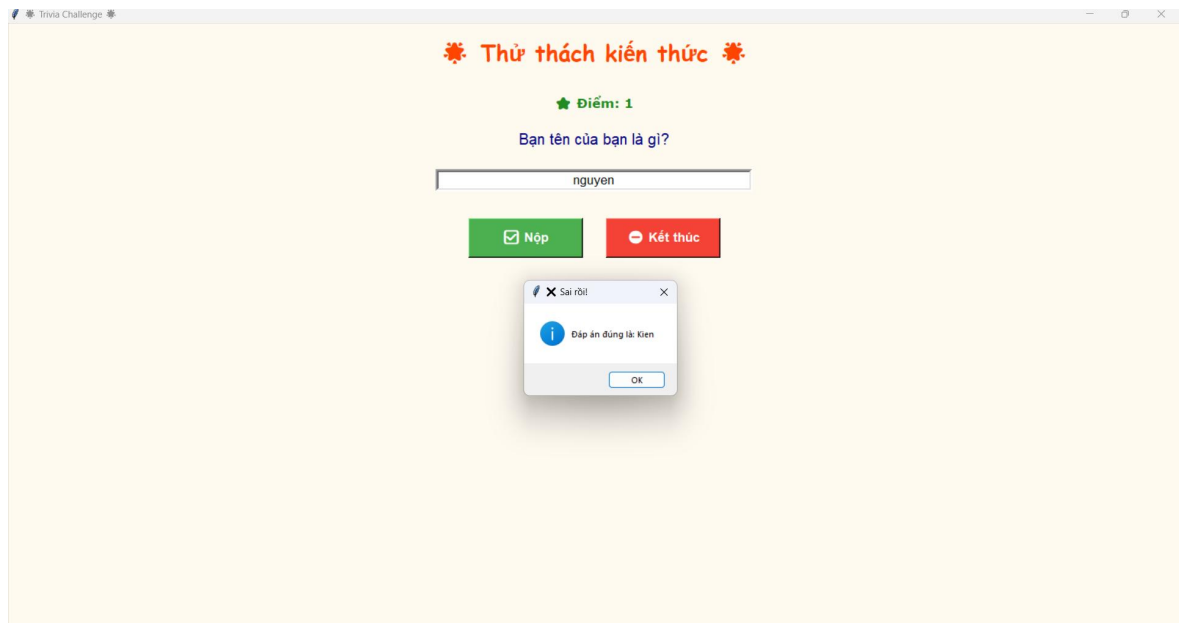
Nhập tên file sai



Nhập tên file thành công chương trình sẽ hiện ra câu hỏi , nhập đáp án , nộp , kết thúc



Khi nhập đáp án đúng chương trình sẽ thông báo và cộng điểm



Khi nhập đáp án sai chương trình sẽ thông báo đáp án đúng là:.. và không được cộng điểm



Khi ấn nút kết thúc hoặc hết câu hỏi chương trình sẽ thông báo tổng điểm và kết thúc chương trình

4.2 Kết luận

4.2.1 Sản phẩm đã thực hiện được:

- Giao diện đồ họa thân thiện sử dụng *tkinter*.
- Đọc câu hỏi từ file và xử lý nhập dữ liệu từ người dùng.
- Kiểm tra đúng/sai và cập nhật điểm số theo thời gian thực.
- Thông báo khi kết thúc và đóng file an toàn.

4.2.2 bài học rút ra

- Nắm vững cách sử dụng *Tkinter* để tạo giao diện đơn giản.
- Thành thạo thao tác với file trong Python (*open()*, *readline()*).
- Biết tổ chức chương trình theo hướng module và lập trình hướng đối tượng (*OOP*).
- Biết xử lý ngoại lệ (*try/except*) để tạo chương trình ổn định hơn.

4.2.3 hướng phát triển trong tương lai

- Cho phép hiển thị nhiều lựa chọn trắc nghiệm (ABC) thay vì chỉ nhập text.
- Cho phép nhập và chỉnh sửa câu hỏi thông qua giao diện quản lý.
- Lưu điểm của người chơi và tạo bảng xếp hạng.
- Cải thiện giao diện đẹp hơn, có hiệu ứng và âm thanh phản hồi.

4.3 tài liệu tham khảo

*W3Schools. **Python Tkinter Tutorial**. Hướng dẫn lập trình giao diện với tkinter cơ bản.*

Nguyễn Văn Hiệp. Giáo trình Python cơ bản. Nhà xuất bản Giao thông Vận tải, 2020. Chương trình đọc/ghi file, xử lý chuỗi, và giao diện tkinter.

<https://chat.openai.com/>