

**TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP**  
**KHOA ĐIỆN TỬ - BỘ MÔN CÔNG NGHỆ THÔNG TIN**

---



**BÀI TẬP KẾT THÚC MÔN HỌC**  
**MÔN: LẬP TRÌNH PYTHON**

**NGÀNH : KỸ THUẬT MÁY TÍNH**

**HỆ : ĐẠI HỌC CHÍNH QUY**

**THÁI NGUYỄN – 2025**

**TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP**  
**KHOA ĐIỆN TỬ - BỘ MÔN CÔNG NGHỆ THÔNG TIN**

---



**BÀI TẬP KẾT THÚC MÔN HỌC**  
**MÔN: LẬP TRÌNH PYTHON**

<b>GIÁO VIÊN HƯỚNG DẪN</b>	<b>:</b>	<b>TS.Nguyễn Văn Huy</b>
<b>HỌ TÊN SINH VIÊN</b>	<b>:</b>	<b>Nguyễn Tuấn Anh</b>
<b>MSSV</b>	<b>:</b>	<b>K225480106095</b>
<b>LỚP</b>	<b>:</b>	<b>K58KTP</b>
<b>LINK YOUTUBE</b>	<b>:</b>	

**BÀI TẬP KẾT THÚC MÔN HỌC**  
**MÔN HỌC: LẬP TRÌNH PYTHON**  
**BỘ MÔN : CÔNG NGHỆ THÔNG TIN**

Sinh viên: Nguyễn Tuấn Anh

MSSV: K225480106095

Lớp: K58KTP

Ngành: Kỹ Thuật Máy Tính

Giáo viên hướng dẫn: TS. Nguyễn Văn Huy

Ngày giao nhiệm vụ: 19/05/2025

Ngày hoàn thành nhiệm vụ: 07/06/2025

Tên đề tài: *Máy tính đơn giản (Simple Calculator GUI)*

Yêu cầu: Viết chương trình máy tính có giao diện GUI cho phép người dùng nhập hai số và chọn phép toán +, -, ×, ÷ để tính toán.

**GIÁO VIÊN HƯỚNG DẪN**

*(Ký và ghi rõ họ tên)*

## PHIẾU GHI ĐIỂM

**Sinh viên:** Nguyễn Tuấn Anh

**Lớp:** K58KTP

**GVHD:** TS. Nguyễn Văn Huy

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

Xếp loại : ..... Điểm : .....

Thái Nguyên, ngày    tháng    năm 2025.

**GIÁO VIÊN HƯỚNG DẪN**

*(Ký và ghi rõ họ tên)*

## **LỜI CAM ĐOAN**

Em xin cam đoan rằng bài tập kết thúc môn Lập trình Python do em tự thực hiện và là kết quả của riêng em dưới sự hướng dẫn của thầy Nguyễn Văn Huy. Các nội dung, số liệu và kết quả trình bày trong bài là trung thực, không sao chép từ bất kỳ nguồn nào một cách trái phép và không được trích dẫn rõ ràng.

Em cam kết mọi thông tin, số liệu và kết quả trong báo cáo là chính xác và trung thực. Nếu có bất kỳ sai phạm nào trong quá trình thực hiện, em xin hoàn toàn chịu trách nhiệm trước nhà trường và giảng viên phụ trách.

Em xin chân thành cảm ơn!

Họ và tên

Nguyễn Tuấn Anh

## MỤC LỤC

<b>LỜI CAM ĐOAN .....</b>	<b>3</b>
<b>DANH MỤC BẢNG VẼ VÀ ĐỒ THỊ .....</b>	<b>5</b>
<b>LỜI MỞ ĐẦU .....</b>	<b>6</b>
<b>CHƯƠNG I: GIỚI THIỆU ĐẦU BÀI .....</b>	<b>7</b>
1.1. Tên đề tài .....	7
1.2. Mô tả bài toán .....	7
1.3. Mục tiêu của chương trình .....	8
1.4. Thách thức và giải pháp .....	8
<b>CHƯƠNG II: CƠ SỞ LÝ THUYẾT .....</b>	<b>10</b>
2.1. Lập trình Python .....	<b>Error! Bookmark not defined.</b>
2.2. Thư viện Tkinter .....	<b>Error! Bookmark not defined.</b>
2.3. Biến liên kết và xử lý sự kiện .....	<b>Error! Bookmark not defined.</b>
2.4. Bắt lỗi bằng try – except .....	10
<b>CHƯƠNG III: THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH .....</b>	<b>12</b>
3.1. Sơ đồ khối hệ thống .....	12
3.1.1. Các module chính của chương trình .....	12
3.2. Sơ đồ khối các thuật toán chính .....	13
3.2.1. Thuật toán hàm calculate() .....	<b>Error! Bookmark not defined.</b>
3.2.2. Thuật toán hàm reset .....	<b>Error! Bookmark not defined.</b>
3.3. Cấu trúc dữ liệu .....	<b>Error! Bookmark not defined.</b>
3.4. Chương trình .....	14
<b>CHƯƠNG 4. THỰC NGHIỆM VÀ KẾT LUẬN .....</b>	<b>15</b>
4.1. Thực nghiệm .....	15
4.1.1. Ghi lại kết quả các tính năng .....	15
4.1.2. Mô tả các tính năng .....	20
4.2. Kết luận .....	23
4.2.1. Những điều học được trong quá trình thực hiện .....	23
4.2.2. Hướng cải tiến trong tương lai .....	24
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>25</b>

## DANH MỤC BẢNG VẼ VÀ ĐỒ THỊ

*Hình 3.2.1: Sơ đồ khối thuật toán hàm calculate()*

*Hình 3.1.2: Biểu đồ phân cấp chức năng*

*Hình 3.2.2: Sơ đồ khối thuật toán hàm reset()*

*Hình 4.1.1: Kết quả nhập đúng*

*Hình 4.1.2: Kết quả nhập sai định dạng số*

*Hình 4.1.3: Kết quả nhập chia cho 0*

*Hình 4.1.4: code giao diện đồ họa GUI*

*Hình 4.1.5: code nhập dữ liệu cho các số*

*Hình 4.1.6: code chọn phép tính và tính toán*

## LỜI MỞ ĐẦU

Ngày nay, việc xây dựng các ứng dụng có giao diện đồ họa (GUI) ngày càng trở nên phổ biến, giúp người dùng tương tác dễ dàng và trực quan hơn với phần mềm. Python là một trong những ngôn ngữ lập trình linh hoạt, hỗ trợ mạnh mẽ việc phát triển các ứng dụng GUI thông qua nhiều thư viện như Tkinter, PyQt, hay Kivy.

Trong khuôn khổ môn học Lập trình Python, em đã chọn đề tài “Xây dựng máy tính đơn giản bằng giao diện đồ họa (GUI)” nhằm áp dụng các kiến thức đã học vào một ứng dụng thực tiễn. Đề tài này không chỉ giúp em rèn luyện kỹ năng lập trình, tư duy logic mà còn làm quen với cách thiết kế giao diện người dùng cơ bản trong Python. Ứng dụng máy tính được xây dựng cho phép thực hiện các phép toán cơ bản như cộng, trừ, nhân, chia với giao diện thân thiện và dễ sử dụng.

Bài báo cáo này trình bày toàn bộ quá trình xây dựng ứng dụng, từ phân tích yêu cầu, thiết kế giao diện đến cài đặt và kiểm thử chương trình. Trong quá trình thực hiện, em đã học hỏi và tích lũy được nhiều kinh nghiệm quý báu.

Em xin chân thành cảm ơn thầy Nguyễn Văn Huy đã tận tình hướng dẫn, tạo điều kiện để em hoàn thành đề tài này. Em rất mong nhận được sự góp ý của thầy để có thể hoàn thiện hơn trong các dự án sau.



# CHƯƠNG I: GIỚI THIỆU ĐẦU BÀI

## 1.1. Tên đề tài

- Xây dựng chương trình minh họa trò chơi Blackjack có giao diện người dùng.

## 1.2. Mô tả bài toán

- Trò chơi Blackjack (còn gọi là "Xì Dách" trong tiếng Việt) là một trò chơi bài nổi tiếng, nơi mục tiêu của người chơi là có tổng điểm bài gần hoặc bằng 21 hơn người chơi đối thủ (thường là nhà cái). Người chơi sẽ được chia hai lá bài ban đầu, sau đó có thể chọn rút thêm bài để nâng cao tổng điểm. Tuy nhiên, nếu tổng điểm vượt quá 21, người chơi sẽ "bust" (bị cháy) và thua ván đó.

Trong đề tài này, bài toán được đặt ra là xây dựng một chương trình mô phỏng trò chơi Blackjack dưới dạng giao diện đồ họa (GUI) sử dụng Python. Trò chơi sẽ gồm hai người chơi: người dùng (USER) và máy (BOT). Người dùng có thể tương tác với các nút chức năng như "Rút bài", "Hạ bài", và "Chơi lại". Giao diện cần trực quan, đẹp mắt, hiển thị hình ảnh các lá bài, kết quả ván chơi và điểm số.

Chương trình cần đảm bảo:

- Tuân thủ luật chơi cơ bản của Blackjack.
- Có giao diện đồ họa sử dụng thư viện Tkinter.
- Hiển thị hình ảnh của từng lá bài sử dụng thư viện Pillow.
- Quản lý logic trò chơi bao gồm việc rút bài, tính điểm, xác định thắng thua.
- Cho phép khởi tạo lại ván chơi mới.

### 1.3. Mục tiêu của chương trình

- Giao diện người dùng trực quan, thân thiện, có bàn chơi, bài và các nút chức năng.
- Hỗ trợ tính năng rút bài, hạ bài, chơi lại, hiển thị kết quả thắng/thua.
- BOT (người chơi đối kháng) phải tự động chơi theo luật chuẩn.
- Hiển thị đầy đủ tình trạng của bài, điểm số, và tên người chơi.

### 1.4. Thách thức và giải pháp

- Việc quản lý điểm số trong Blackjack không đơn giản do sự linh hoạt của lá bài A (vừa có thể tính là 1, vừa có thể là 11). Cần thuật toán chính xác để xử lý tình huống này.
- Cần xây dựng BOT có khả năng tự chơi hợp lý. Việc BOT chơi không quá thông minh nhưng cần đúng luật là điều bắt buộc..
- Giao diện đồ họa cần rõ ràng, ổn định và dễ dùng, yêu cầu lập trình chính xác trong Tkinter, xử lý hình ảnh bài với Pillow.
- Việc xử lý hình ảnh, resize và ghép ảnh bài theo bộ và giá trị cũng là một bước không thể thiếu
- Kiểm tra và kiểm soát các tình huống như cháy bài, hòa điểm, hoặc xuất hiện Blackjack phải được lập trình rõ ràng, tránh lỗi logic.

### 1.5. Kiến thức áp dụng

**Python cơ bản:** Biến, vòng lặp, điều kiện, hàm, nhập xuất dữ liệu.

**Lập trình hướng đối tượng (OOP):** Thiết kế lớp (class), đóng gói, kế thừa, sử dụng đối tượng để chia nhỏ các thành phần của trò chơi như: lá bài (Card), bộ bài (Deck), tay bài (Hand), và trò chơi tổng thể (Game).

**Tkinter:** Thư viện tiêu chuẩn của Python để xây dựng giao diện người dùng đồ họa (GUI). Dùng để tạo cửa sổ, các nút điều khiển, khung chứa, hiển thị thông tin và hình ảnh.

**Pillow (PIL):** Thư viện xử lý ảnh, dùng để mở, resize và chuyển ảnh về định dạng có thể hiển thị trong Tkinter.

**Tư duy logic và thiết kế phần mềm:** Phân chia module hợp lý, thiết kế giao diện rõ ràng, đảm bảo tính ổn định và dễ mở rộng cho chương trình.

## CHƯƠNG II: CƠ SỞ LÝ THUYẾT

### 2.1. Danh sách

- Danh sách là cấu trúc dữ liệu được sử dụng chủ yếu để lưu trữ các lá bài trong tay của người chơi hoặc BOT. Mỗi tay bài là một danh sách chứa nhiều đối tượng kiểu BJ\_Card. Việc thêm một lá bài mới vào tay đơn giản chỉ là thao tác append một phần tử vào danh sách. Khi cần tính điểm, chương trình sẽ duyệt danh sách này và cộng dồn giá trị các lá bài. Đây là một thao tác quan trọng và diễn ra thường xuyên trong suốt quá trình chơi, đặc biệt khi người chơi hoặc BOT rút thêm bài.

### 2.2. Từ điển

- Từ điển là cấu trúc ánh xạ được dùng để liên kết giá trị của từng lá bài với điểm số cụ thể. Ví dụ: từ điển values = {'A':11, '2':2, ..., 'K':10} giúp chương trình dễ dàng tra cứu điểm số tương ứng với từng rank. Ngoài ra, từ điển cũng được sử dụng để chuyển đổi ký hiệu bài (J, Q, K) và bộ bài (♠, ♥, ♦, ♣) thành tên file ảnh tương ứng trong thư mục images, phục vụ cho việc hiển thị ảnh lá bài chính xác.

### 2.3. Hướng đối tượng

- Việc tổ chức chương trình theo mô hình hướng đối tượng giúp chia nhỏ các phần logic phức tạp thành các thành phần đơn giản và dễ quản lý. Mỗi thực thể trong trò chơi (lá bài, bộ bài, tay bài, trò chơi) đều được xây dựng thành class riêng với các thuộc tính và phương thức phù hợp. Điều này không chỉ giúp chương trình dễ mở rộng mà còn giúp lập trình viên dễ hình dung được luồng dữ liệu và hành vi của hệ thống trong từng bước.

### 2.4. Tkinter (GUI)

- Tkinter là thư viện chuẩn của Python dùng để xây dựng các giao diện đồ họa cơ bản. Trong đề tài này, Tkinter được sử dụng để tạo cửa sổ chính, các khung chứa bài, các nút điều khiển như "Rút bài", "Hạ bài", "Chơi lại", và các nhãn hiển thị kết quả. Tkinter hỗ trợ việc tổ chức bố cục bằng pack(), grid() hoặc place(), giúp linh hoạt trong việc hiển thị giao diện theo thiết kế mong muốn. Các sự kiện nhấn nút được xử lý qua việc gán hàm callback tương ứng cho từng nút.

## 2.5. Pillow (PIL)

- Pillow là thư viện mở rộng từ PIL, chuyên dùng cho xử lý ảnh trong Python. Trong chương trình Blackjack, Pillow đóng vai trò quan trọng trong việc đọc các ảnh PNG của lá bài từ thư mục images, thực hiện thao tác resize ảnh về kích thước chuẩn để hiển thị đồng bộ và gọn gàng, sau đó sử dụng `ImageTk.PhotoImage()` để tích hợp ảnh vào Label trong giao diện Tkinter. Nhờ Pillow, các lá bài trở nên sinh động, đẹp mắt và nâng cao tính thẩm mỹ cho toàn bộ ứng dụng.

## CHƯƠNG III: THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH

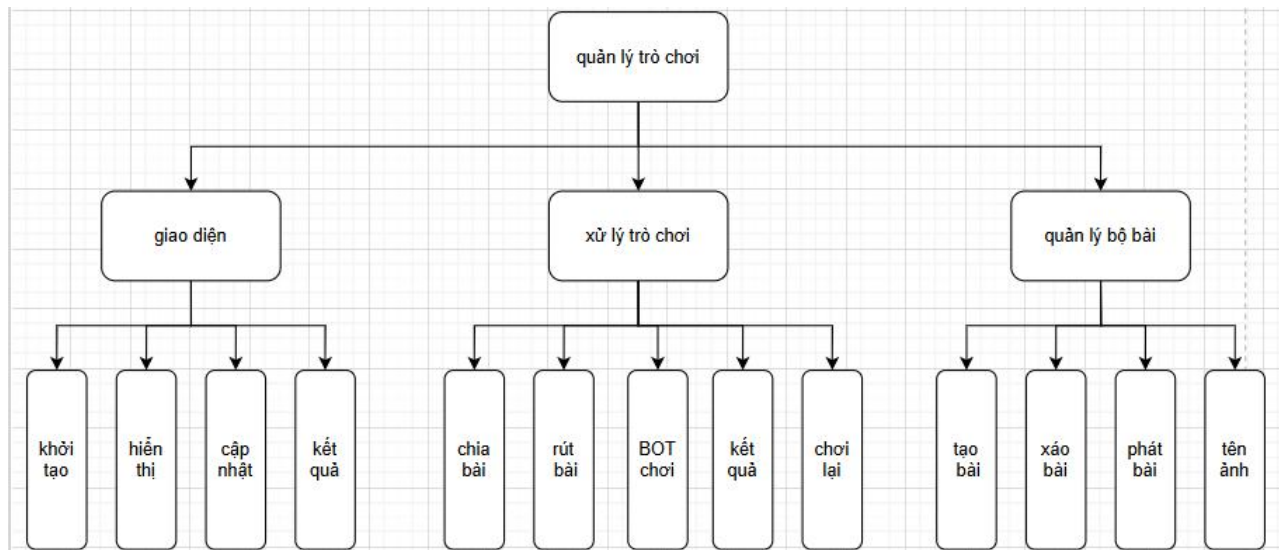
### 3.1. Sơ đồ khối hệ thống

#### 3.1.1. Các module chính của chương trình

- Chương trình gồm ba module chính: cards.py chịu trách nhiệm quản lý các lá bài và bộ bài; game.py điều khiển logic trò chơi, xử lý các hành động của người chơi và BOT; main.py là nơi tổ chức giao diện người dùng bằng thư viện Tkinter, đồng thời đóng vai trò là nơi liên kết các phần khác lại với nhau. Từ đó hình thành nên một hệ thống thống nhất, vận hành ổn định từ lúc khởi tạo cho đến khi xác định người thắng cuộc và bắt đầu ván mới.

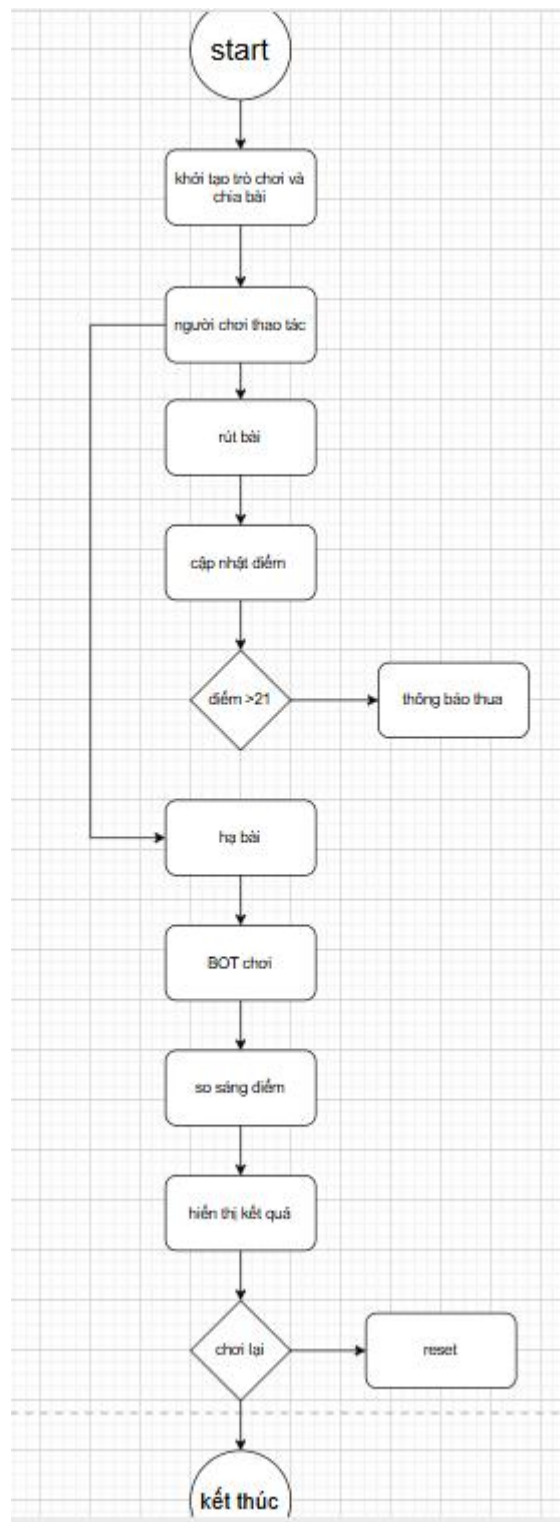
#### 3.1.2. Biểu đồ phân cấp chức năng

- Dựa trên phân tích yêu cầu của bài toán, ta có biểu đồ phân cấp chức năng sau:



Hình 3.1.2: Biểu đồ phân cấp chức năng

### 3.2. Sơ đồ khối các thuật toán chính



Hình 3.2.1: Sơ đồ khối thuật toán chính

### 3.3. Mô hình dữ liệu

- Dữ liệu trong chương trình được tổ chức theo hướng đối tượng. Lớp BJ\_Card đại diện cho một lá bài cụ thể với thuộc tính suit, rank và phương thức value(). Lớp BJ\_Deck tạo và lưu trữ toàn bộ 52 lá bài, có phương thức deal\_card() để phát bài. Lớp BJ\_Hand là nơi lưu trữ các lá bài của một người chơi (hoặc BOT), có các phương thức thêm bài, tính điểm và kiểm tra Blackjack. Lớp BJ\_Game là trung tâm điều phối, gọi các hành vi và xử lý sự kiện trong mỗi lượt chơi..

### 3.4. Chương trình

- Giao diện được thiết kế bằng Tkinter với màu nền xanh lá đậm tạo cảm giác giống bàn bài thật. Mỗi lá bài được hiển thị bằng ảnh PNG có viền đen rõ ràng trên nền trắng. Các nút lệnh được bố trí hợp lý giúp người dùng dễ thao tác. Khi rút bài hoặc hạ bài, màn hình sẽ tự động cập nhật trạng thái mới, bao gồm hình ảnh lá bài mới, tổng điểm và kết quả nếu có. Giao diện được lập trình sao cho tối giản, hiệu quả nhưng vẫn giữ được tính thẩm mỹ và trực quan



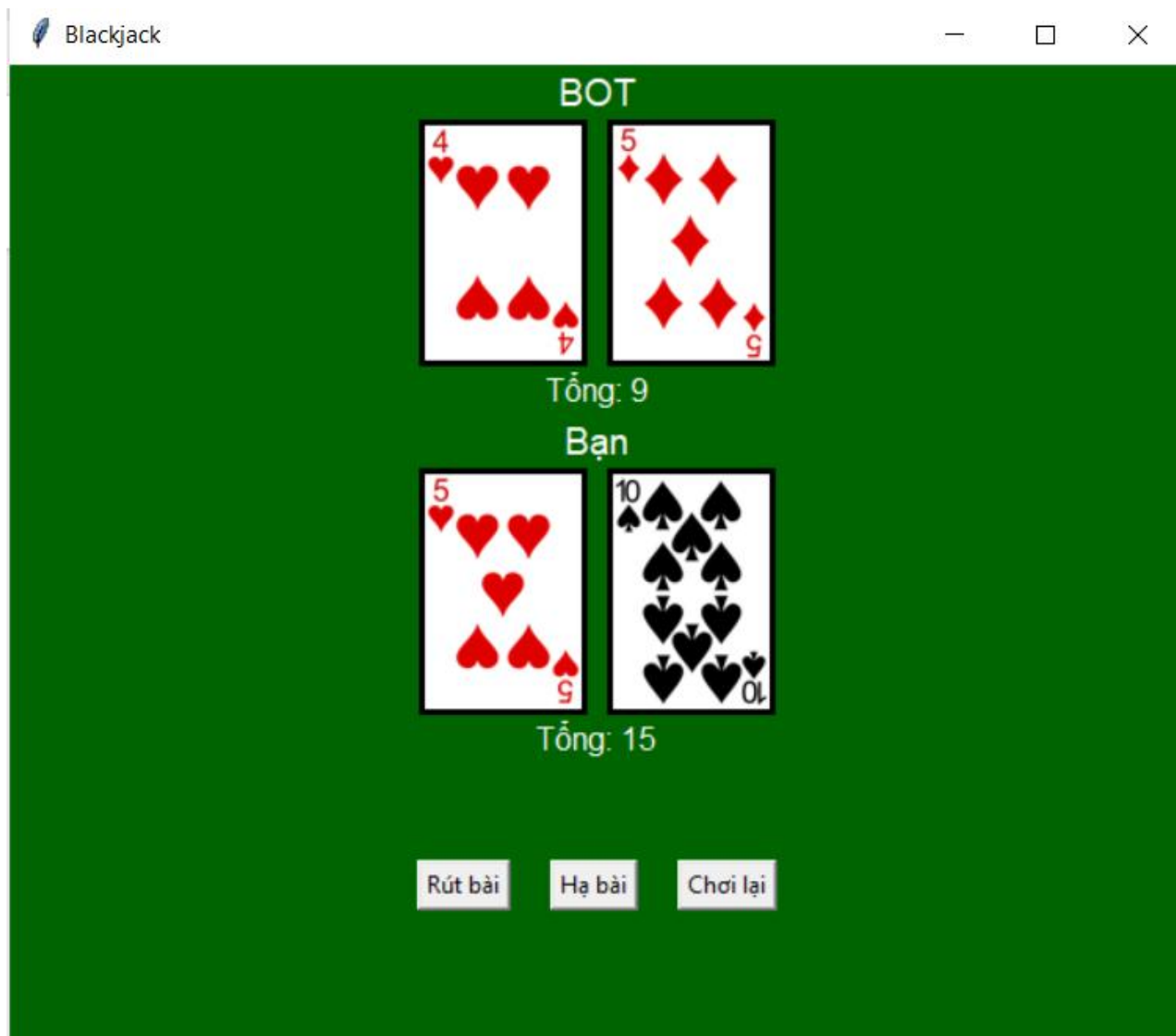
## CHƯƠNG 4. THỰC NGHIỆM VÀ KẾT LUẬN

### 4.1. Thực nghiệm

- Sau khi xây dựng hoàn chỉnh, chương trình được kiểm tra bằng nhiều lượt chơi liên tục để đảm bảo tính ổn định. Tất cả chức năng chính đều được thử nghiệm bao gồm chia bài, rút bài, hạ bài và chơi lại. Trong quá trình thử nghiệm, nhóm ghi nhận chương trình hoạt động ổn định, không xảy ra lỗi logic, điểm số tính toán đúng trong mọi trường hợp đặc biệt như có nhiều lá A hoặc BOT phải rút liên tiếp.
- Giao diện hiển thị đầy đủ thông tin, màu sắc dễ nhìn, thao tác nút nhảy và mượt. Hình ảnh lá bài được hiển thị đồng bộ với các hành động tương ứng. Kết quả thắng thua được đưa ra kịp thời, đúng với luật Blackjack. Khi người chơi rút quá 21 điểm, chương trình lập tức thông báo thua, và khi BOT rút cháy hoặc người chơi thắng bằng Blackjack, chương trình cũng đưa ra phản hồi tương ứng.

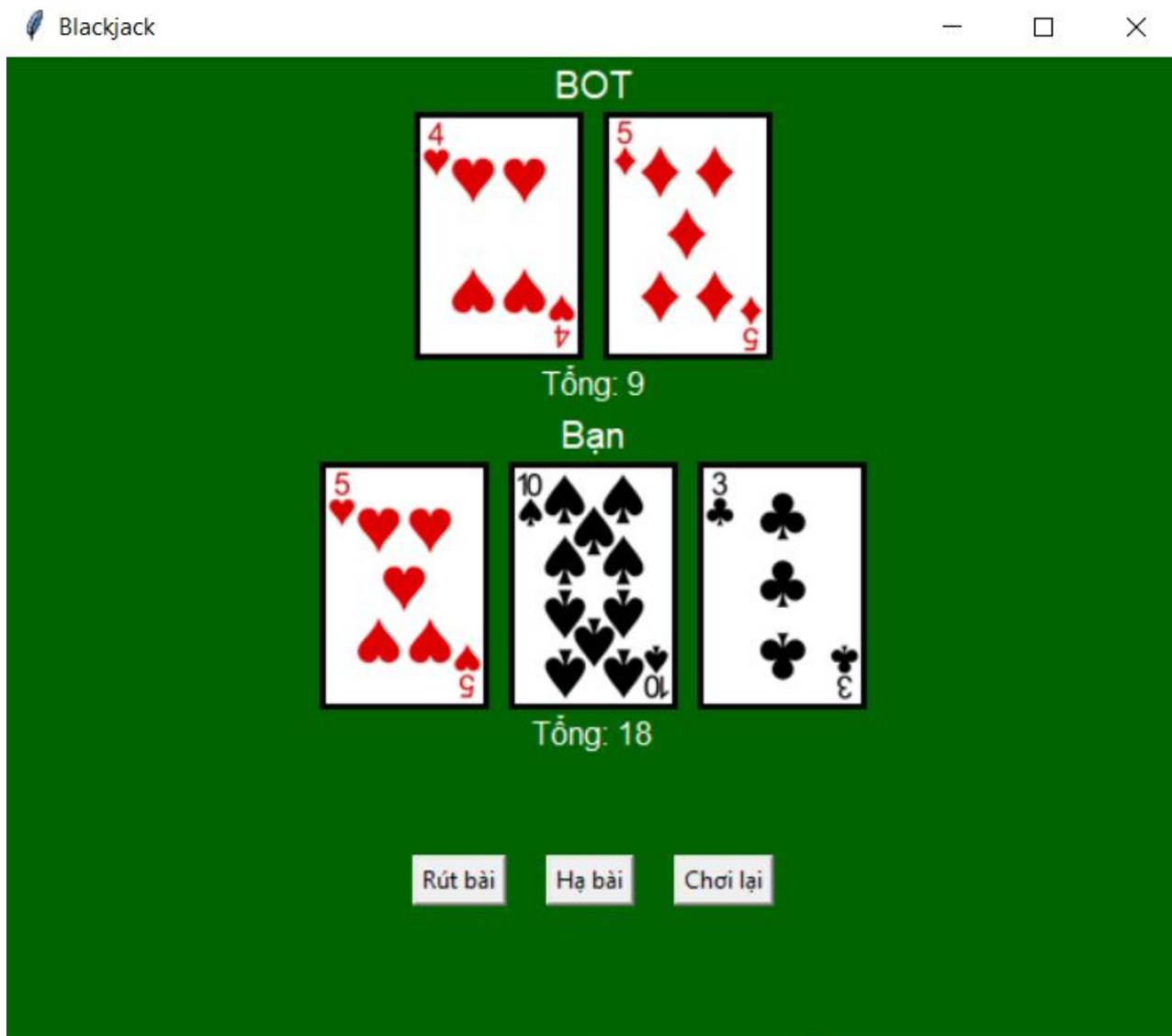
#### 4.1.1. Ghi lại kết quả các tính năng

- Tính năng chia bài tự động: Khi bắt đầu trò chơi hoặc nhấn nút "Chơi lại", chương trình sẽ chia hai lá bài đầu tiên cho người chơi và BOT, đồng thời hiển thị chúng trên giao diện cùng với điểm số.



Hình 4.1.1: Hiển thị tính năng chia bài tự động

- Tính năng rút bài: Người chơi có thể nhấn nút "Rút bài" để lấy thêm một lá bài. Mỗi lần rút bài, lá bài mới sẽ được hiển thị trên màn hình, điểm số được cập nhật tức thì và kiểm tra tình trạng cháy bài tự động.



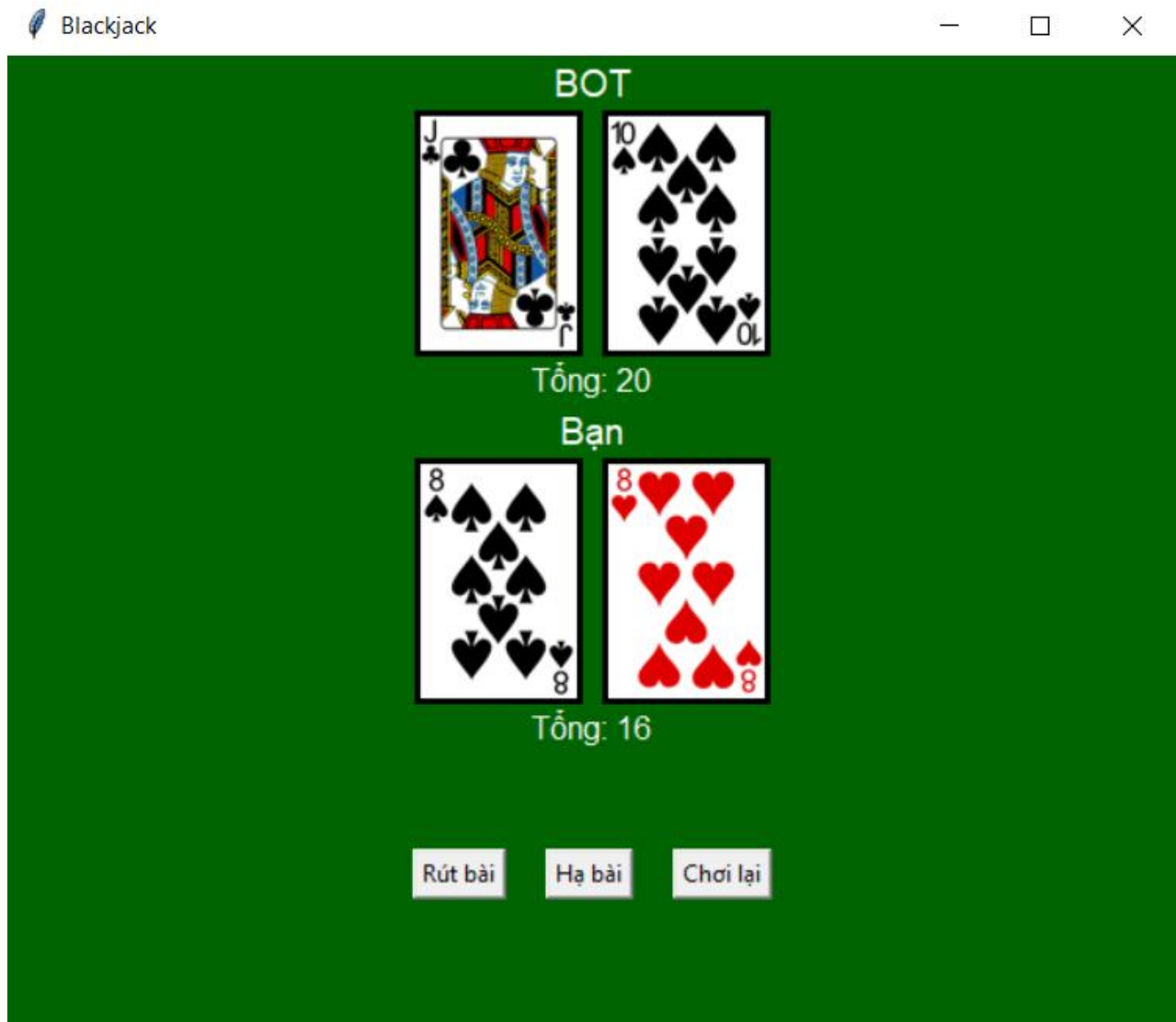
Hình 4.1.2: Kết quả tính năng rút bài

- Tính năng hạ bài: Sau khi hài lòng với bộ bài đang có, người chơi có thể nhấn nút "Hạ bài". Khi đó, BOT sẽ bắt đầu lượt chơi của mình: tự động rút bài cho đến khi đạt từ 17 điểm trở lên, sau đó chương trình sẽ tiến hành so sánh điểm và thông báo kết quả (thắng, thua hoặc hòa).



Hình 4.1.3: Kết quả tính năng hạ bài

- Tính năng chơi lại: Sau mỗi ván chơi, người dùng có thể nhấn nút "Chơi lại" để hệ thống tự động khởi tạo lại bộ bài, làm mới giao diện và bắt đầu một ván chơi mới mà không cần khởi động lại chương trình.



*Hình 4.1.4: Kết quả tính năng chơi lại*

-Tính năng thông báo kết quả: Kết quả thắng, thua hoặc hòa được thông báo rõ ràng trên màn hình, giúp người chơi biết được tình trạng của ván bài sau khi kết thúc.

## Đánh giá:

Chương trình đã hoàn thiện đầy đủ theo yêu cầu đề bài, mô phỏng chính xác trò chơi Blackjack với đầy đủ các chức năng cần thiết. Giao diện đẹp, dễ sử dụng, hoạt động ổn định.

### 4.1.2. Mô tả các tính năng

- Tính năng hiển thị hình ảnh bài:

Mỗi lá bài đều có ảnh minh họa rõ ràng, có viền đen trên nền trắng, giúp dễ nhận diện và tạo cảm giác chân thực như đang chơi bài ngoài đời..

```
def draw_cards(self):
    for widget in self.card_frame.winfo_children():
        widget.destroy()

    self.card_images = []

    def draw_hand(hand, label_text):
        tk.Label(self.card_frame, text=label_text, font=("Arial", 14), bg="#006400", fg="white").pack()
        frame = tk.Frame(self.card_frame, bg="#006400")
        frame.pack()

        for card in hand.cards:
            try:
                img = Image.open(card.image_filename()).resize((80, 120))
                photo = ImageTk.PhotoImage(img)
                self.card_images.append(photo)

                card_label = tk.Label(frame, image=photo, bd=2, relief="solid",
                                      highlightbackground="black", highlightthickness=1,
                                      bg="white") # Nền trắng
                card_label.pack(side=tk.LEFT, padx=5)
            except:
                card_label = tk.Label(frame, text=str(card), bd=2, relief="solid",
                                      highlightbackground="black", highlightthickness=1,
                                      bg="white", width=10, height=6)
                card_label.pack(side=tk.LEFT, padx=5)
```

Hình 4.1.4: code giao diện đồ họa

- Tính năng BOT tự động:

BOT hoạt động theo chiến lược đơn giản nhưng hiệu quả: luôn rút bài nếu điểm < 17 và dừng lại khi điểm  $\geq 17$  hoặc bị cháy. Điều này đảm bảo tính công bằng và bám sát luật chơi thực tế.

```
def stand(self):  
    while self.dealer.value() < 17:  
        self.dealer.add_card(self.deck.deal_card())  
  
    self.is_over = True  
    self.result = self.check_winner()  
  
def check_winner(self):  
    if self.player.is_blackjack():  
        return "Blackjack! Bạn thắng!"  
    if self.dealer.value() > 21:  
        return "BOT cháy! Bạn thắng!"  
    if self.player.value() > 21:  
        return "Bạn cháy! mày Thua!"  
    if self.player.value() > self.dealer.value():  
        return "Bạn thắng!"  
    elif self.player.value() < self.dealer.value():  
        return "BOT thắng!"  
    else:  
        return "Hòa!"
```

*Hình 4.1.5: code BOT tự động*

- Tính năng tính điểm linh hoạt: Lá bài A được xử lý thông minh, có thể tính là 11 hoặc 1 tùy theo tình huống để tránh cháy bài. Cơ chế này giúp người chơi có trải nghiệm gần với thực tế hơn.

```
class BJ_Hand:
    def __init__(self):
        self.cards = []

    def add_card(self, card):
        self.cards.append(card)

    def value(self):
        total = sum(card.value() for card in self.cards)
        aces = sum(1 for card in self.cards if card.rank == 'A')
        while total > 21 and aces:
            total -= 10
            aces -= 1
        return total
```

*Hình 4.1.6: code tính điểm linh hoạt*



## 4.2. Kết luận

- Tổng thể, các tính năng đã được tích hợp một cách hợp lý, đồng bộ và mang tính hoàn chỉnh cho một trò chơi Blackjack cơ bản. Từ việc chia bài tự động, tính điểm linh hoạt, rút bài thông minh, đến BOT có chiến lược tự động và giao diện người dùng trực quan – tất cả được phối hợp nhịp nhàng tạo nên một trải nghiệm chơi game đơn giản nhưng thú vị. Các nút điều khiển phản hồi nhanh, hình ảnh bài hiển thị rõ ràng giúp người dùng dễ nắm bắt trạng thái trò chơi. Các xử lý quan trọng như điểm số A (11/1), kiểm tra cháy bài, hoặc xác định kết quả đều hoạt động ổn định và chính xác. Điều này cho thấy chương trình không chỉ đáp ứng tốt về mặt kỹ thuật mà còn đảm bảo tiêu chí thân thiện và dễ sử dụng đối với người dùng cuối.

### 4.2.1. Những điều học được trong quá trình thực hiện

- Tự tìm hiểu và sử dụng thành thạo thư viện Tkinter để thiết kế giao diện người dùng, biết cách tạo và xử lý các thành phần như Label, Frame, Button, Canvas, đồng thời tổ chức layout sao cho hợp lý, dễ thao tác và thẩm mỹ.
- Nắm vững kỹ thuật làm việc với thư viện Pillow để đọc, resize và hiển thị ảnh các lá bài một cách trực quan, đảm bảo hiệu suất xử lý hình ảnh trong GUI.
- Làm chủ tư duy lập trình hướng đối tượng trong Python thông qua việc tự xây dựng và kết nối các lớp như Card, Deck, Hand, Game, đảm bảo tính đóng gói và mở rộng.
- Rèn luyện khả năng phân tích bài toán, thiết kế thuật toán linh hoạt – đặc biệt trong xử lý giá trị lá bài A để tránh trường hợp "bust" không hợp lệ.
- Cải thiện khả năng tổ chức code theo module, đảm bảo mỗi phần có chức năng riêng biệt, dễ bảo trì và mở rộng sau này.
- Tăng cường kỹ năng viết báo cáo, giải thích logic xử lý, trình bày luồng xử lý rõ ràng, logic, khoa học.

#### **4.2.2. Hướng cải tiến trong tương lai**

- Bổ sung các hiệu ứng âm thanh, hiệu ứng lật bài để tăng tính sinh động cho trò chơi.
- Tích hợp tính năng lưu lịch sử điểm số, kết quả các ván chơi để người chơi có thể theo dõi tiến trình chơi.
- Phát triển thêm chế độ chơi nhiều người qua mạng LAN hoặc qua máy chủ đơn giản.
- Thêm lựa chọn mức độ khó khác nhau cho BOT, từ cơ bản đến nâng cao (AI chơi chiến lược).
- Cải thiện giao diện thân thiện hơn cho thiết bị cảm ứng (mobile/tablet).
- Tích hợp tính năng hướng dẫn chơi, giúp người mới có thể nắm bắt cách chơi dễ dàng.

## TÀI LIỆU THAM KHẢO

1. Documentation Tkinter – <https://docs.python.org/3/library/tkinter.html>
2. Pillow (PIL Fork) documentation – <https://pillow.readthedocs.io/en/stable/>
3. Python OOP Tutorial – <https://realpython.com/python3-object-oriented-programming/>
4. Blackjack Game Rules – <https://bicyclecards.com/how-to-play/blackjack/>
5. StackOverflow discussions on Blackjack logic and Aces – <https://stackoverflow.com/>
6. Tài liệu môn học và slide bài giảng Python – Khoa CNTT  
việc nâng cấp BOT để trở nên thông minh hơn, hỗ trợ nhiều người chơi cùng lúc, hoặc kết nối qua mạng LAN để chơi trực tuyến. Ngoài ra, có thể bổ sung hiệu ứng âm thanh, hình ảnh động, lưu lịch sử kết quả và thêm chế độ luyện tập với nhiều cấp độ khó khác nhau để tăng tính hấp dẫn cho trò chơi.