

**TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP**  
**KHOA ĐIỆN TỬ**

**Bộ môn: Công nghệ Thông tin**



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

**Sinh viên: Nguyễn Thị Thu Hiền**

**Lớp: K58KTP**

**Giáo viên hướng dẫn : TS. Nguyễn Văn Huy**

**Thái Nguyên – 2025**

## BÀI TẬP KẾT THÚC MÔN HỌC

### MÔN HỌC: LẬP TRÌNH PYTHON

#### BỘ MÔN : CÔNG NGHỆ THÔNG TIN

Sinh viên: Nguyễn Thị Thu Hiền

Lớp: K58KPT.01

Ngành: Kỹ thuật phần mềm

Giáo viên hướng dẫn: TS. Nguyễn Văn Huy

Ngày giao đề: 20/5/2025

Ngày hoàn thành: 10/6/2025

Tên đề tài : *Xây game Blackjack với GUI: chia bài, hit/stand, tính điểm.*

Yêu cầu :

- **Đầu vào – đầu ra:**
  - Đầu vào: Nút “Hit”, “Stand”.
  - Đầu ra: Hiển thị lá bài (Label/Icon), điểm, kết quả.
- **Tính năng yêu cầu:**
  - Sử dụng class BJ\_Card, BJ\_Deck, BJ\_Hand, BJ\_Game.
  - GUI cập nhật khi rút bài.
  - Xử lý bust (>21) và so sánh với dealer.

**GIÁO VIÊN HƯỚNG DẪN**

*(Ký và ghi rõ họ tên)*

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

*Thái Nguyên, ngày....tháng.....năm 20....*

**GIÁO VIÊN HƯỚNG DẪN**

*(Ký ghi rõ họ tên)*

## LỜI NÓI ĐẦU

Trong thời đại công nghệ thông tin phát triển mạnh mẽ, việc học và ứng dụng các ngôn ngữ lập trình hiện đại vào thực tiễn đang ngày càng trở nên quan trọng. Python là một trong những ngôn ngữ lập trình phổ biến nhất hiện nay nhờ cú pháp đơn giản, dễ tiếp cận và khả năng mở rộng linh hoạt trong nhiều lĩnh vực như trí tuệ nhân tạo, phân tích dữ liệu, phát triển ứng dụng và game.

Thông qua quá trình học tập môn Lập trình Python, em đã có cơ hội tìm hiểu và vận dụng các kiến thức đã học để xây dựng một ứng dụng hoàn chỉnh. Đề tài “*Xây dựng trò chơi Blackjack với GUI*” được lựa chọn nhằm kết hợp giữa lập trình hướng đối tượng, quản lý module và thiết kế giao diện người dùng bằng Tkinter, giúp củng cố kỹ năng lập trình, tư duy phân tích và khả năng tổ chức chương trình theo hướng cấu trúc rõ ràng.

Báo cáo này trình bày quá trình nghiên cứu, thiết kế và xây dựng trò chơi Blackjack – một game bài đơn giản nhưng thú vị, phù hợp với người học lập trình ở cấp độ cơ bản đến trung cấp. Đề tài giúp người học vận dụng hiệu quả kiến thức về class, module, sự kiện GUI, cũng như hiểu rõ cách phối hợp giữa xử lý logic và cập nhật giao diện thời gian thực.

Em xin chân thành cảm ơn thầy giáo **TS. Nguyễn Văn Huy** đã tận tình giảng dạy và hướng dẫn trong suốt quá trình thực hiện đề tài. Mặc dù đã cố gắng hoàn thành tốt nhất trong khả năng, song báo cáo không thể tránh khỏi những thiếu sót. Kính mong nhận được sự đóng góp ý kiến từ thầy giáo, bạn bè để em có thể học hỏi và hoàn thiện hơn trong tương lai.

Sinh viên thực hiện  
(ký và ghi rõ họ tên)

## MỤC LỤC

<b>LỜI NÓI ĐẦU .....</b>	<b>4</b>
<b>DANH MỤC BẢNG BIỂU, HÌNH VẼ.....</b>	<b>6</b>
<b>CHƯƠNG 1. GIỚI THIỆU ĐẦU BÀI .....</b>	<b>7</b>
1.1. Tên đề tài.....	7
1.2. Lý do chọn đề tài.....	7
1.3. Mục tiêu đề tài.....	7
1.4. Phương pháp thực hiện.....	7
<b>CHƯƠNG 2. CƠ SỞ LÝ THUYẾT.....</b>	<b>9</b>
2.1. Ngôn ngữ lập trình Python.....	9
2.2. Lập trình hướng đối tượng OOP .....	9
2.3. Giao diện đồ họa với Tkinter .....	10
2.4. Các công cụ và thư viện sử dụng .....	11
2.5. Tìm hiểu về BlackJack.....	11
<b>CHƯƠNG 3. 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 hệ thống .....	12
3.1.2. Biểu đồ chức năng.....	13
3.1.3. Biểu đồ Use-Case hệ thống game BlackJack.....	14
3.2. Sơ đồ khối các thuật toán chính .....	15
3.3. Cấu trúc dữ liệu .....	17
3.4. Chương trình .....	19
<b>CHƯƠNG 4. THỰC NGHIỆM VÀ KẾT LUẬN.....</b>	<b>22</b>
4.1. Thực nghiệm .....	22
4.2. Kết luận .....	25
<b>PHỤ LỤC .....</b>	<b>27</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>28</b>

## **DANH MỤC BẢNG BIỂU, HÌNH VẼ**

Hình 2.1. Biểu tượng ngôn ngữ Python

Bảng 2.1. Các thành phần chính của Tkinter

Hình 3.1. Biểu đồ phân cấp chức năng

Hình 3.2. Biểu đồ Use - Case

Hình 3.3. Sơ đồ khối thuật toán trò chơi

Bảng 3.1. Tóm tắt thành phần cấu trúc lưu trữ

Hình 3.4. Biểu đồ lớp game BlackJack

Bảng 3.2. Mô tả các hàm trong chương trình chính

Hình 3.5. Danh sách các file trong folder BLACKJACK\_GUI

Hình 3.6. Đường dẫn đến file blackjack\_gui\_multi.py trên Anaconda Prompt

Hình 3.7. Chạy file blackjack\_gui\_multi.py

Hình 3.8. Giao diện game Blackjack

Hình 3.9. Giao diện khi vào trò chơi

Hình 3.10. Điểm số của tất cả người chơi với dealer

Hình 3.11. Player 1 > điểm dealer và =21 nên thắng

Hình 3.12. Player 2 >21 điểm nên thua

Hình 3.13. Player 3 > điểm dealer nên thắng

Hình 3.14. Số tiền cược sau mỗi ván

# CHƯƠNG 1. GIỚI THIỆU ĐẦU BÀI

## 1.1. Tên đề tài.

*“Xây game Blackjack với GUI: chia bài, hit/stand, tính điểm.”*

## 1.2. Lý do chọn đề tài.

Trong quá trình học môn Lập trình Python, việc áp dụng kiến thức về lập trình hướng đối tượng (OOP) và giao diện đồ họa (GUI) là rất quan trọng để rèn luyện tư duy lập trình và kỹ năng xây dựng ứng dụng thực tế. Game Blackjack là một trò chơi phổ biến, có luật chơi đơn giản nhưng đủ để triển khai nhiều kỹ thuật lập trình như: quản lý class, xử lý sự kiện, cập nhật giao diện theo thời gian thực và tính toán logic.

Đề tài “Blackjack GUI đơn giản” không chỉ giúp em củng cố kiến thức về class, kế thừa, và module trong Python, mà còn giúp em làm quen với thư viện Tkinter để xây dựng giao diện người dùng. Bên cạnh đó, việc triển khai game còn tạo động lực học tập vì mang tính giải trí, giúp hiểu sâu hơn về cách một ứng dụng tương tác hoạt động trong thực tế.

## 1.3. Mục tiêu đề tài.

Mục tiêu của đề tài là xây dựng một trò chơi Blackjack có giao diện đồ họa đơn giản sử dụng ngôn ngữ lập trình Python. Thông qua đề tài này, em mong muốn áp dụng kiến thức về lập trình hướng đối tượng để xây dựng các lớp cơ bản như **BJ\_Card**, **BJ\_Deck**, **BJ\_Hand**, và **BJ\_Game**, đồng thời sử dụng thư viện Tkinter để thiết kế giao diện người dùng thân thiện với các nút chức năng như “Hit” và “Stand”. Đề tài hướng đến việc xử lý các thao tác trong game như chia bài, rút bài, tính điểm, xử lý tình huống thua cuộc khi vượt quá 21 điểm (bust), và xác định kết quả thắng thua giữa người chơi và dealer. Qua đó, em rèn luyện thêm kỹ năng lập trình giao diện, xử lý sự kiện, và cập nhật trạng thái trò chơi theo thời gian thực, từ đó nâng cao khả năng phát triển ứng dụng Python có tính tương tác cao.

## 1.4. Phương pháp thực hiện.

Để triển khai đề tài “Xây game Blackjack với GUI: chia bài, hit/stand, tính điểm”, em thực hiện theo từng bước cụ thể như sau:

Trước tiên, em xây dựng logic trò chơi bằng cách sử dụng các lớp đã được định nghĩa sẵn hoặc tự xây dựng theo yêu cầu: **BJ\_Card** để đại diện cho lá bài, **BJ\_Deck** để quản lý bộ bài, **BJ\_Hand** để quản lý các lá bài người chơi hoặc

dealer đang có, và BJ\_Game để xử lý toàn bộ luật chơi. Các class này giúp tổ chức mã nguồn rõ ràng, dễ bảo trì và mở rộng.

Sau khi hoàn thiện phần xử lý logic, em sử dụng thư viện Tkinter để thiết kế giao diện đồ họa (GUI). Giao diện sẽ bao gồm hai khung (Frame) chính: một dành cho người chơi và một dành cho dealer. Mỗi khung sẽ hiển thị các lá bài dưới dạng Label hoặc hình ảnh (icon), cùng với điểm số tương ứng. Giao diện cũng bao gồm hai nút điều khiển: “Hit” để người chơi rút thêm bài, và “Stand” để kết thúc lượt và chuyển quyền chơi sang dealer.

Khi người chơi nhấn “Hit”, chương trình sẽ gọi phương thức từ BJ\_Game để rút thêm bài và cập nhật điểm số. GUI sẽ phản hồi ngay bằng cách hiển thị lá bài mới và kiểm tra nếu tổng điểm vượt quá 21 (bust) hoặc đạt Blackjack (ví dụ như rút được A và 10). Nếu người chơi chọn “Stand”, dealer sẽ tự động rút bài theo luật, sau đó chương trình sẽ so sánh điểm giữa hai bên và hiển thị kết quả qua Label hoặc MessageBox.

- **Các thách thức có thể gặp phải**

- Quản lý GUI động: Việc cập nhật giao diện khi có lá bài mới xuất hiện yêu cầu quản lý tốt các widget như Label hoặc ảnh, tránh việc đè lên hoặc rò rỉ tài nguyên.
- Tính điểm đặc biệt của lá bài A: Lá A có thể được tính là 1 hoặc 11, đòi hỏi xử lý linh hoạt để không làm người chơi bị bust không cần thiết.
- Đồng bộ giữa logic và giao diện: Phải đảm bảo rằng trạng thái logic game luôn đồng bộ với giao diện hiển thị, tránh gây nhầm lẫn cho người chơi.
- Kiểm soát lượt chơi: Cần đảm bảo rằng người chơi và dealer không thực hiện hành động sai lượt (ví dụ, người chơi chưa “Stand” thì dealer không được rút).
- Xử lý sự kiện và kết thúc game: Sau khi game kết thúc, các nút “Hit” và “Stand” cần bị vô hiệu hóa hoặc khởi động lại ván mới đúng cách.

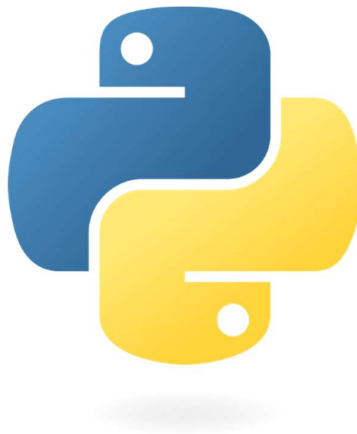


## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

### 2.1. Ngôn ngữ lập trình Python

Python là một ngôn ngữ lập trình bậc cao, đa mục đích, có cú pháp đơn giản, dễ học, dễ đọc, rất phù hợp cho cả người mới bắt đầu và các ứng dụng chuyên sâu. Python hỗ trợ lập trình thủ tục, lập trình hướng đối tượng (OOP) và cả lập trình hàm.

Trong đề tài này, Python được sử dụng để xây dựng toàn bộ logic trò chơi, giao diện đồ họa cũng như xử lý các thuật toán liên quan đến trò chơi bài Blackjack.



Hình 2.1. Biểu tượng ngôn ngữ Python

### 2.2. Lập trình hướng đối tượng OOP

Lập trình hướng đối tượng là kỹ thuật lập trình tập trung vào việc xây dựng các đối tượng và lớp có chứa dữ liệu và các hành vi. Các tính chất cơ bản bao gồm:

- Đóng gói: gom nhóm dữ liệu và phương thức liên quan vào một lớp.
- Kế thừa: cho phép lớp con sử dụng lại thuộc tính và hành vi của lớp cha.
- Đa hình: cùng một hành động có thể được thực hiện theo nhiều cách khác nhau.

Áp dụng trong đề tài (dựa trên cấu trúc chương 9) trò chơi sử dụng các lớp chính sau:

- BJ\_Card: kế thừa từ Card, đại diện cho lá bài, có khả năng kiểm tra xem có phải là bài úp không.
- BJ\_Deck: kế thừa từ Deck, cho phép tạo bộ bài Blackjack và chia bài

- BJ\_Hand: kế thừa từ Hand, xử lý việc tính điểm Blackjack, xử lý bài A linh hoạt (tính 1 hoặc 11).
- BJ\_Game: điều khiển luồng chơi giữa người dùng và Dealer, xử lý thắng/thua.

### 2.3. Giao diện đồ họa với Tkinter

Tkinter là thư viện tiêu chuẩn của Python dùng để xây dựng giao diện người dùng đồ họa (GUI – Graphical User Interface). Nó là một giao diện (wrapper) cho thư viện GUI Tcl/Tk, được tích hợp sẵn trong Python, nên không cần cài đặt thêm.

Tkinter cho phép tạo ra các ứng dụng với cửa sổ, nút bấm, ô nhập, hình ảnh, menu... một cách đơn giản và trực quan. Đây là lựa chọn lý tưởng cho các bài tập nhỏ, trò chơi đơn giản hoặc ứng dụng desktop cơ bản.

- Các thành phần chính trong thư viện Tkinter:

Thành phần	Mô tả
Tk()	Tạo cửa sổ chính của ứng dụng
Frame	Khung dùng để nhóm các widget khác
Label	Dùng để hiển thị văn bản hoặc hình ảnh
Button	Nút bấm tương tác với người dùng
PhotoImage	Dùng để hiển thị hình ảnh (ảnh .png)
messagebox	Hộp thoại hiện thông báo hoặc cảnh báo
pack(), grid(), place()	Các phương thức bố trí giao diện (layout)

Bảng 2.1. Các thành phần chính của Tkinter

Ví dụ : Tạo cửa sổ chào mừng đồ họa Tkinter

```
import tkinter as tk
root = tk.Tk()
root.title("Xin chào")
root.geometry("300x100")
```

```
label = tk.Label(root, text="Chào mừng đến với Tkinter!")
label.pack()
root.mainloop()
```

## 2.4. Các công cụ và thư viện sử dụng

- Ngôn ngữ lập trình: Python
- Thư viện: Tkinter , random , os, sys
- Trình soạn thảo code: Visual Studio code

## 2.5. Tìm hiểu về BlackJack

**Blackjack**, hay còn gọi là 21, là một trong những trò chơi bài phổ biến nhất trên thế giới, đặc biệt là trong các sòng bạc. Người chơi không đấu với nhau mà đấu với Dealer (nhà cái), mục tiêu là có tổng điểm gần 21 nhất nhưng không vượt quá.

- Luật chơi Blackjack

Người chơi được chia các lá bài có giá trị điểm. Mỗi người chơi cố gắng đạt tổng điểm là 21 mà không vượt quá. Các lá bài số có giá trị bằng chính số đó. Lá Át được tính là 1 hoặc 11 (tùy theo điều kiện có lợi cho người chơi). Các lá J, Q, K đều tính là 10 điểm.

Máy tính đóng vai người chia bài (dealer) và thi đấu với từ một đến bảy người chơi. Khi vòng chơi bắt đầu, máy tính chia cho tất cả người chơi (bao gồm cả máy) mỗi người hai lá bài. Người chơi có thể nhìn thấy tất cả lá bài của mình và máy tính cũng hiển thị tổng điểm cho họ. Tuy nhiên, một lá bài của người chia bài sẽ được giữ ẩn trong lúc này.

Tiếp theo, mỗi người chơi sẽ lần lượt được rút thêm bài. Mỗi người có thể rút từng lá bài một theo ý muốn. Nhưng nếu tổng điểm vượt quá 21 (gọi là “bust”), người đó thua. Nếu tất cả người chơi đều “bust”, máy tính sẽ lật lá bài đầu tiên của mình và vòng chơi kết thúc. Nếu không, trò chơi tiếp tục. Người chia bài phải rút thêm bài cho đến khi tổng điểm của nó ít nhất là 17. Nếu người chia bài “bust”, tất cả người chơi còn lại không “bust” sẽ thắng. Nếu không, điểm số của mỗi người chơi còn lại sẽ được so sánh với người chia bài. Nếu điểm của người chơi lớn hơn, người đó thắng; nhỏ hơn thì thua; bằng nhau thì hòa (gọi là “push”).

## CHƯƠNG 3. THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH

### 3.1. Sơ đồ khối hệ thống

#### 3.1.1. Các module chính của hệ thống

- **BJ\_Card**: Module lá bài
  - Chức năng: Đại diện cho một lá bài duy nhất trong bộ bài.
  - Thuộc tính chính:
    - + *rank*: Giá trị lá bài (A, 2, ..., K).
    - + *suit*: Chất bài (♥, ♦, ♣, ♠).
  - Phương thức chính:
    - + *\_\_str\_\_*: Trả về chuỗi mô tả lá bài, ví dụ "A♥".
  - Ý nghĩa: Là đơn vị nhỏ nhất trong game, dùng để biểu diễn từng lá bài riêng biệt, phục vụ cho việc hiển thị và tính điểm.
- **BJ\_Deck** : Module bộ bài
  - Chức năng: Tạo và quản lý 52 lá bài chuẩn.
  - Thuộc tính chính:
    - + *cards*: Danh sách 52 đối tượng **BJ\_Card**.
  - Phương thức chính:
    - + *shuffle()*: Xáo trộn bài ngẫu nhiên.
    - + *deal()*: Rút 1 lá từ bộ bài (pop khỏi danh sách).
  - Ý nghĩa: Cung cấp lá bài cho cả người chơi và dealer, đảm bảo tính ngẫu nhiên và công bằng.
- **BJ\_Hand**: Module tay bài
  - Chức năng: Quản lý tập hợp các lá bài trên tay của người chơi hoặc dealer.
  - Thuộc tính chính:
    - + *cards*: Danh sách các **BJ\_Card** hiện có trên tay.
  - Phương thức chính:
    - + *add\_card(card)*: Thêm một lá vào tay bài.
    - + *get\_value()*: Tính tổng điểm, xử lý đặc biệt cho quân A (1 hoặc 11).
    - + *is\_bust()*: Kiểm tra nếu điểm > 21.
  - Ý nghĩa: Giúp quản lý điểm số và tình trạng của người chơi/dealer trong suốt ván chơi.
- **BJ\_Game**: Module điều khiển trò chơi

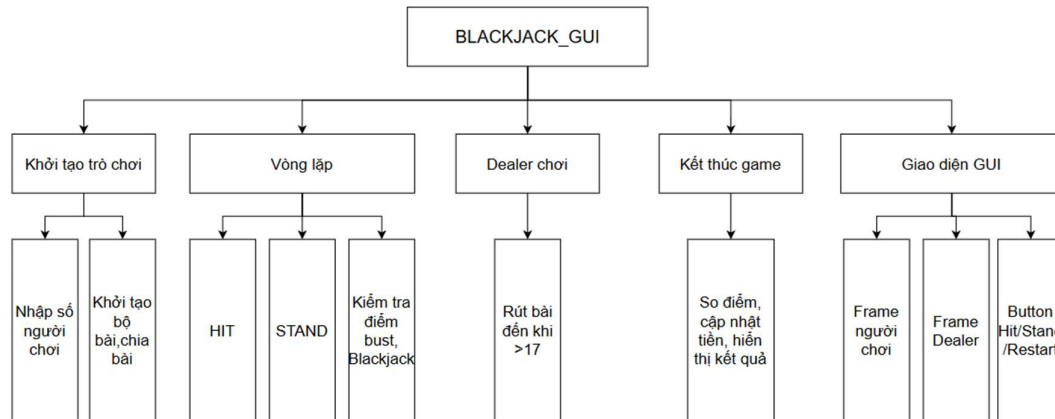
- Chức năng: Điều phối toàn bộ logic ván chơi Blackjack.
- Thuộc tính chính:
  - + *deck*: Bộ bài hiện tại (**BJ\_Deck**).
  - + *players*: Danh sách các tay bài của người chơi (**BJ\_Hand**).
  - + *dealer*: Tay bài của dealer (**BJ\_Hand**).
- Phương thức chính:
  - + *start\_game()*: Bắt đầu một ván chơi mới, chia bài.
  - + *player\_hit(player\_index)*: Người chơi rút thêm bài.
  - + *player\_stand(player\_index)*: Người chơi dừng lượt.
  - + *dealer\_play()*: Dealer tự động rút bài đến khi đủ điểm.
  - + *compare\_scores()*: So sánh điểm để xác định thắng thua.
- Ý nghĩa: Là module trung tâm của game, kết nối giữa dữ liệu và giao diện.
- **BlackjackGUI**: Giao diện người dùng với Tkinter
- Chức năng: Hiển thị toàn bộ trò chơi trên cửa sổ GUI.
- Chứa các thành phần:
  - + *Frame người chơi*: Hiển thị bài và điểm của từng người chơi.
  - + *Frame dealer*: Hiển thị bài và điểm của nhà cái.
- Các nút điều khiển:
  - + *Hit*: Rút thêm bài.
  - + *Stand*: Kết thúc lượt.
  - + *Restart*: Bắt đầu ván mới.
  - + *Label và MessageBox*: Hiển thị kết quả, thông báo.
- Ý nghĩa: Giúp người chơi tương tác trực quan với game, xử lý sự kiện người dùng và cập nhật màn hình.

### 3.1.2. Biểu đồ chức năng

Hệ thống game Blackjack có giao diện đồ họa cho phép người chơi đối đầu với máy (dealer) theo luật chơi Blackjack. Các chức năng chính gồm:

- Quản lý ván chơi: khởi tạo bộ bài, chia bài, xử lý lượt chơi, tính điểm và xác định thắng thua.
- Giao diện người dùng: hiển thị lá bài, điểm số, và thông báo kết quả qua các nút “Hit” và “Stand”.
- Xử lý luật chơi: tính điểm, nhận biết “Blackjack”, xử lý “bust”, và để dealer rút bài tự động.

- Biểu đồ phân cấp chức năng:



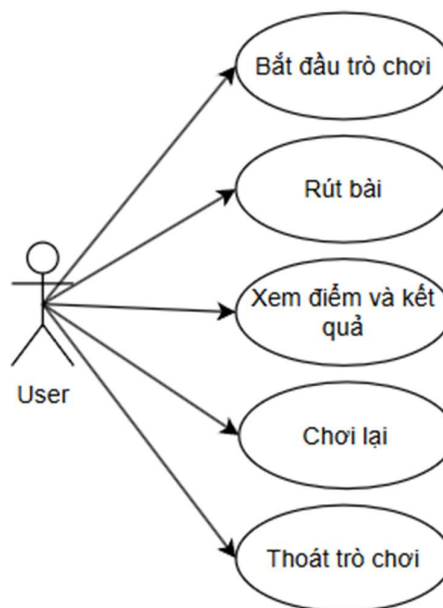
Hình 3.1. Biểu đồ phân cấp chức năng

### 3.1.3. Biểu đồ Use-Case hệ thống game *BlackJack*

- Tác nhân: User (Tương tác trực tiếp tới hệ thống).



- Biểu đồ Use – Case với tác nhân User



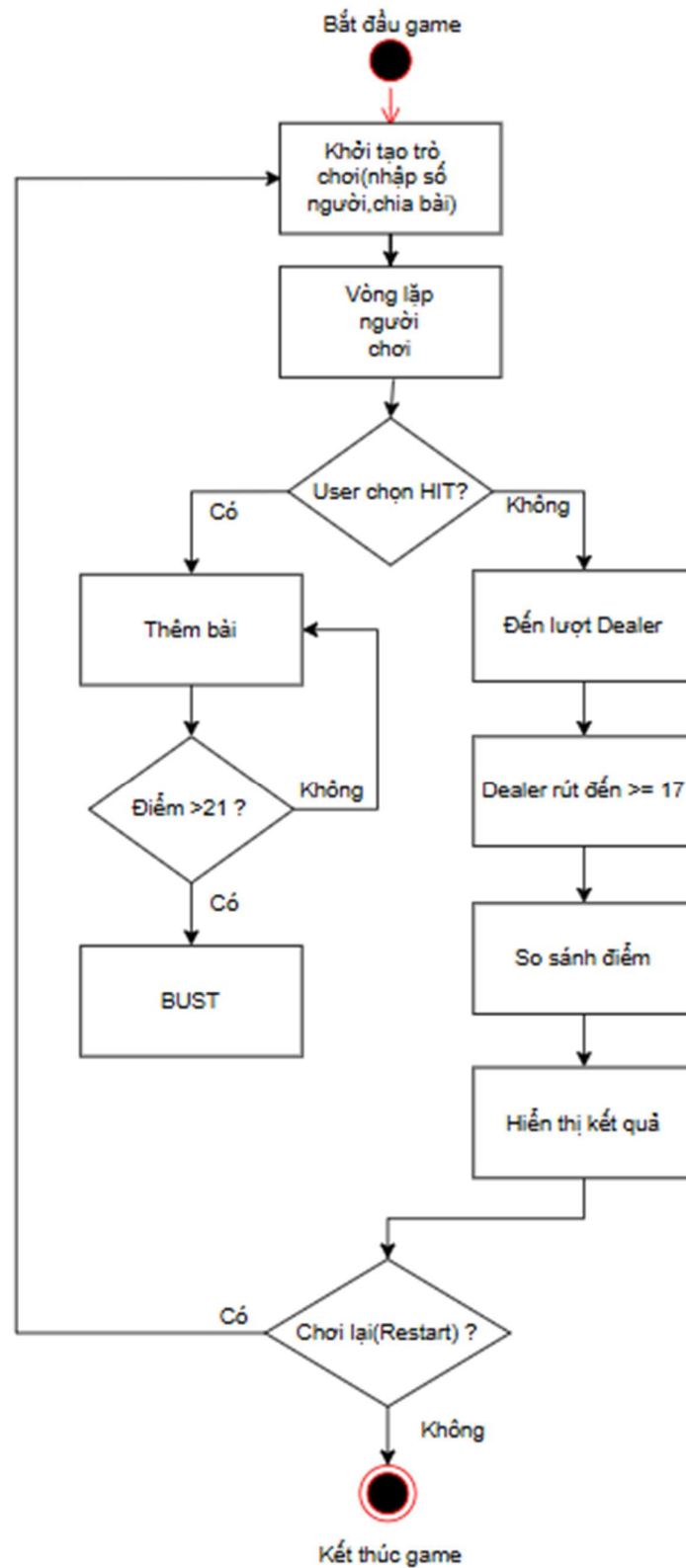
Hình 3.2. Biểu đồ Use - Case

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

Hệ thống game Blackjack hoạt động theo quy trình gồm nhiều bước xử lý. Các khối thuật toán chính được mô tả như sau:

- **Khối 1:** Khởi tạo trò chơi
  - Chức năng: Nhận thông tin số lượng người chơi, tạo bộ bài, chia bài cho các người chơi và dealer.
  - Đầu vào: Số lượng người chơi (người dùng nhập).
  - Đầu ra: Các đối tượng Hand (tay bài) chứa các lá bài ban đầu.
- **Khối 2:** Vòng lặp người chơi (Hit/Stand)
  - Chức năng: Người chơi chọn “Hit” để rút thêm bài, hoặc “Stand” để kết thúc lượt. Nếu vượt quá 21 điểm (bust) thì dừng.
  - Đầu vào: Sự kiện nút “Hit” hoặc “Stand”.
  - Đầu ra: Tay bài người chơi được cập nhật; kiểm tra thắng thua.
- **Khối 3:** Lượt chơi của Dealer
  - Chức năng: Dealer rút bài đến khi điểm  $\geq 17$ .
  - Đầu vào: Tay bài ban đầu của Dealer.
  - Đầu ra: Tay bài dealer hoàn chỉnh.
- **Khối 4:** So sánh kết quả
  - Chức năng: So điểm người chơi và dealer để xác định thắng/thua/hòa.
  - Đầu vào: Điểm người chơi và dealer.
  - Đầu ra: Kết quả hiển thị trên GUI.

- Sơ đồ khối thuật toán trò chơi



Hình 3.3. Sơ đồ khối thuật toán trò chơi



### 3.3. Cấu trúc dữ liệu

Chương trình game Blackjack sử dụng lập trình hướng đối tượng để quản lý dữ liệu. Thay vì sử dụng cơ sở dữ liệu kiểu bảng như SQL, game sử dụng các lớp (class) để lưu trữ thông tin về lá bài, bộ bài, tay người chơi, và ván chơi. Cụ thể:

- Class **BJ\_Card**
  - Chức năng: Đại diện cho 1 lá bài trên tay.
  - Thuộc tính:
    - + *rank*: Hạng bài (A, 2, 3, ..., K).
    - + *suit*: Chất bài (♠, ♥, ♣, ♦).
    - + *is\_face\_up*: Trạng thái lật bài (True/False).
  - Phương thức chính
    - + *value()*: Trả về giá trị điểm số của lá bài theo quy tắc Blackjack.
- Class **BJ\_Deck**
  - Chức năng: Đại diện cho bộ bài gồm 52 lá.
  - Thuộc tính:
    - + *cards*: Danh sách các đối tượng **BJ\_Card**.
  - Phương thức chính:
    - + *populate()*: Tạo đầy đủ bộ bài.
    - + *shuffle()*: Trộn bài.
    - + *deal(hands, per\_hand)*: Chia bài cho tay người chơi.
- Class **BJ\_Hand**
  - Chức năng: Đại diện cho tay bài của 1 người chơi hoặc dealer.
  - Thuộc tính:
    - + *cards*: Danh sách các lá bài trong tay.
    - + *name*: Tên người chơi (Player hoặc Dealer).
  - Phương thức chính:
    - + *add(card)*: Thêm lá bài vào tay.
    - + *total()*: Tính tổng điểm tay bài (xử lý cả A).
    - + *is\_busted()*: Kiểm tra bust (>21 điểm).
- Class **BJ\_Game**
  - Chức năng: Quản lý toàn bộ tiến trình ván chơi.
  - Thuộc tính:

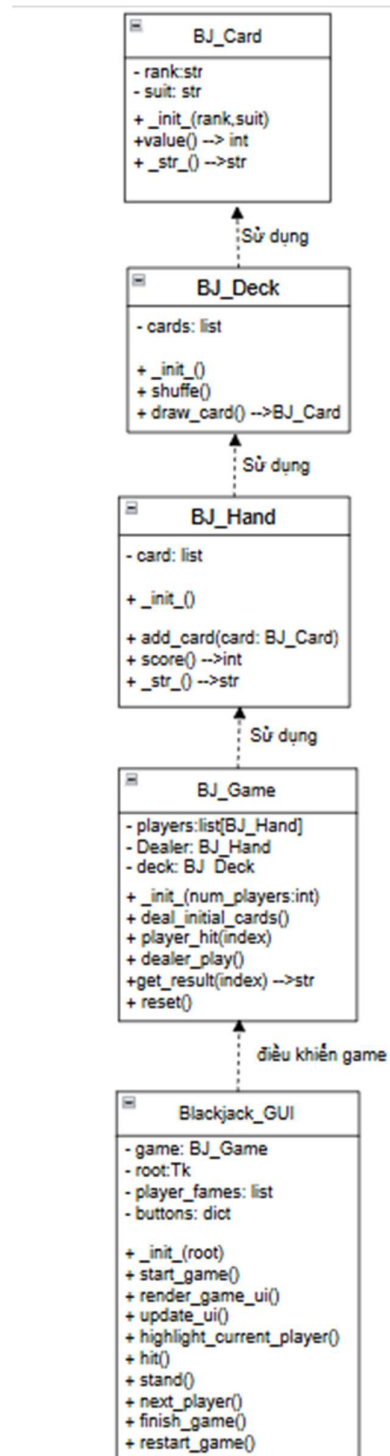
- + *deck*: Bộ bài đang dùng.
- + *player*: Tay người chơi.
- + *dealer*: Tay nhà cái.
- Phương thức chính:
  - + *hit()*: Xử lý khi người chơi chọn rút bài.
  - + *stand()*: Xử lý khi người chơi dừng lại.
  - + *check\_winner()*: So sánh điểm để xác định thắng thua.
  - + *restart()*: Khởi động lại ván chơi.
- Các biến hỗ trợ trong GUI
  - *player\_frame, dealer\_frame*: Frame chứa các lá bài.
  - *label\_score, messagebox*: Hiển thị điểm số và kết quả.

Class / Thành phần	Mục đích
BJ_Card	Đại diện lá bài
BJ_Deck	Bộ bài
BJ_Hand	Tay bài
BJ_Game	Quản lý trò chơi
GUI Variables	Hiển thị giao diện

Bảng 3.1. Tóm tắt thành phần cấu trúc lưu trữ

### 3.4. Chương trình

Để biểu thị rõ hơn mô tả các hàm được sử dụng trong chương trình chính, ta thông qua biểu đồ lớp để thấy rõ các mối quan hệ:



Hình 3.4. Biểu đồ lớp game BlackJack

- Mô tả các hàm trong các hàm trong chương trình chính

Tên lớp	Tên hàm	Chức năng và ý nghĩa
BJ_Card	<code>__init__(rank, suit)</code>	Khởi tạo một lá bài với hạng (rank) và chất (suit).
	<code>__str__()</code>	Trả về chuỗi biểu diễn lá bài, ví dụ: "A♠".
	<code>value()</code>	Tính và trả về giá trị điểm số của lá bài.
BJ_Deck	<code>init ()</code>	Tạo bộ bài gồm 52 lá.
	<code>shuffle()</code>	Xáo trộn bộ bài ngẫu nhiên.
	<code>deal_card()</code>	Rút một lá bài từ bộ bài.
BJ_Hand	<code>init ()</code>	Khởi tạo một tay bài trống.
	<code>add_card(card)</code>	Thêm lá bài vào tay bài.
	<code>score()</code>	Tính điểm tổng của tay bài, xử lý đặc biệt cho quân Át.
	<code>is_bust()</code>	Trả về True nếu điểm lớn hơn 21.
	<code>__str__()</code>	Trả về chuỗi biểu diễn các lá bài trong tay.
BJ_Game	<code>__init__(num_players)</code>	Khởi tạo trò chơi với số người chơi xác định.
	<code>deal_initial_cards()</code>	Chia 2 lá bài cho mỗi người chơi và dealer.
	<code>player_hit(index)</code>	Người chơi rút thêm bài.
	<code>player_stand()</code>	Người chơi kết thúc lượt.
	<code>dealer_play()</code>	Dealer tự động rút bài cho đến khi đủ điểm.
	<code>get_winner(index)</code>	So sánh điểm để xác định kết quả thắng – thua – hòa.
BlackjackGUI	<code>__init__(root)</code>	Khởi tạo giao diện người dùng chính.
	<code>start_game()</code>	Đọc số người chơi, tạo trò chơi, và chia bài ban đầu.
	<code>render_game_ui()</code>	Vẽ giao diện các khung thông tin người chơi và dealer.
	<code>update_ui()</code>	Cập nhật thông tin điểm và bài trên giao diện.

<b>Tên lớp</b>	<b>Tên hàm</b>	<b>Chức năng và ý nghĩa</b>
	highlight_current_player()	Tô sáng người chơi đang được chơi lượt.
	hit()	Sự kiện người chơi bấm "Hit" → gọi player_hit.
	stand()	Sự kiện người chơi bấm "Stand" → chuyển lượt.
	next_player()	Di chuyển đến người chơi tiếp theo hoặc cho dealer chơi.
	finish_game()	Tính toán kết quả trận đấu và cập nhật tiền cược.
	restart_game()	Trở về màn hình khởi đầu để chơi lại.

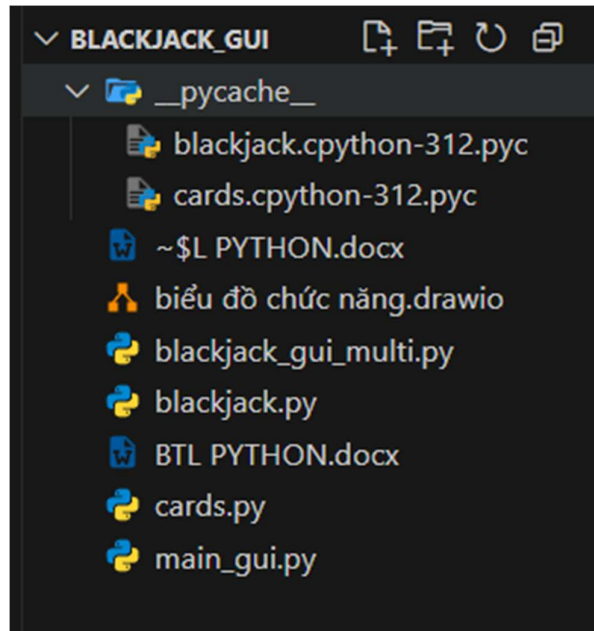
Bảng 3.2. Mô tả các hàm trong chương trình chính

## CHƯƠNG 4. THỰC NGHIỆM VÀ KẾT LUẬN

### 4.1. Thực nghiệm

Sau khi đã tạo chương trình xây dựng game BlackJack\_GUI bằng python với các hàm đã có, ta có một giao diện game GUI BlackJack hoàn thiện.

**Bước 1:** Tạo Folder để lưu file code : Folder **BLACKJACK\_GUI** với danh sách các file như sau:



Hình 3.5. Danh sách các file trong folder BLACKJACK\_GUI

**Bước 2:** File blackjack\_gui\_multi.py là file dùng để chạy chương trình game blackjack. Ở đây, ta sử dụng môi trường Anaconda để chạy các chương trình code python.

Vào Start trên màn hình chính → Gõ: Anaconda Promt(python) → Đưa đường dẫn đến file **blackjack\_gui\_multi.py** trong thư mục **BLACK\_GUI** tại Desktop bằng cú pháp dẫn “cd”.

```
(python) C:\Users\Dell>cd Desktop  
  
(python) C:\Users\Dell\Desktop>cd BLACKJACK_GUI
```

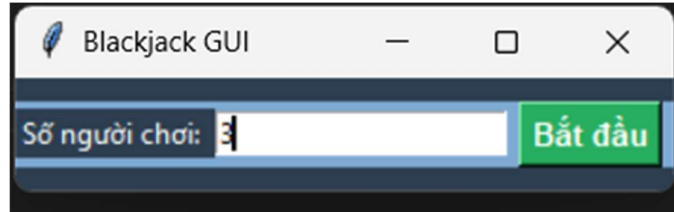
Hình 3.6. Đường dẫn đến file blackjack\_gui\_multi.py trên Anaconda Promt

Sau đó ta sử dụng cú pháp “python\_ “tên file”” để chạy code file , hiện giao diện game :

Ví dụ: python blackjack\_gui\_multi.py

```
(python) C:\Users\Dell\Desktop\BLACKJACK_GUI>python blackjack_gui_multi.py
```

Hình 3.7. Chạy file blackjack\_gui\_multi.py



Hình 3.8. Giao diện game Blackjack

Tại giao diện( hình 3.8 ) bao gồm số người chơi và ô bắt đầu ở bên cạnh . Người chơi chính sẽ nhập số lượng người tham gia trò chơi và chọn ô “bắt đầu” để vào giao diện game. Ví dụ: 3 người.



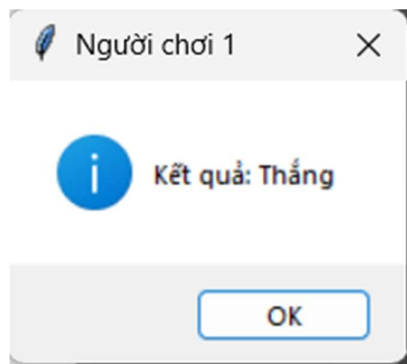
Hình 3.9. Giao diện khi vào trò chơi

Tại giao diện này, gồm có 3 người chơi và một Dealer. Mỗi người chơi sẽ có 2 lá bài mặc định trước đó hiện rõ ràng các lá và điểm số. Tiền cược sẽ giới hạn ở 100\$ hoặc có thể thay đổi nếu muốn. Sau đó, những người chơi sẽ lựa chọn xem có rút thêm lá bài hay không? Chọn “Hit” nếu muốn rút , “Stand” nếu không. Mỗi lần rút bài, hệ thống sẽ tự cộng điểm vào số điểm hiện tại của mỗi lượt chơi, nếu người chơi rút điểm cộng >21 thì sẽ không được rút thêm mà phải stand. Nếu trước đó người chơi chọn Stand thì tiếp tục lượt chơi cho người tiếp theo .

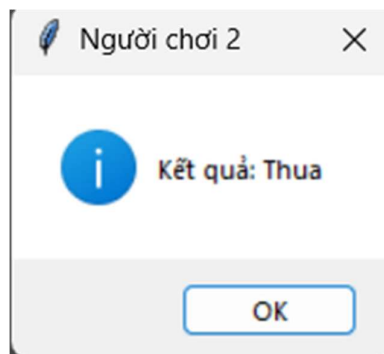
Sau khi tất cả người chơi đều được có lượt của mình thì hệ thống sẽ xem xét điểm của tất cả người chơi, so sánh với điểm của Dealer . Nếu thắng dealer thì hiện Popup báo win và ngược lại. Và nếu muốn chơi lại , chọn Restart.



Hình 3.10. Điểm số của tất cả người chơi với dealer

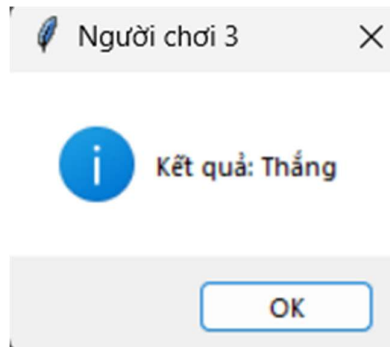


Hình 3.11. Player 1 > điểm dealer và =21 nên thắng

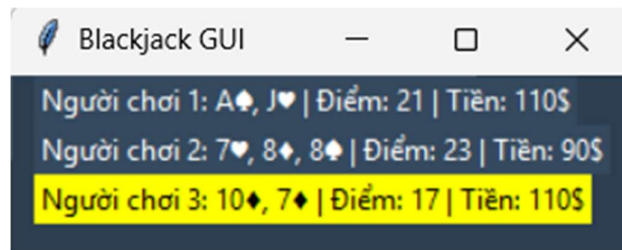


Hình 3.12. Player 2 >21 điểm nên thua





Hình 3.13. Player 3 > điểm dealer nên thắng



Hình 3.14. Số tiền cược sau mỗi ván

Nếu sau ván game chọn Restart, thì lập tức giao diện trở về khi nhập số người chơi và chọn ô “Bắt đầu”.

## 4.2. Kết luận

Qua quá trình xây dựng trò chơi Blackjack bằng ngôn ngữ Python và thư viện tkinter, nhóm đã hoàn thành sản phẩm có giao diện đồ họa thân thiện, hỗ trợ từ 1 đến 6 người chơi cùng lúc. Chương trình đã triển khai đầy đủ các chức năng cơ bản như chia bài, rút bài (Hit), dừng lượt (Stand), tính điểm, so sánh với Dealer và xác định kết quả thắng, thua hoặc hòa. Ngoài ra, sản phẩm còn có hệ thống tính tiền cược, cập nhật số dư của từng người chơi và cho phép khởi động lại ván chơi dễ dàng qua nút Restart. Các nút chức năng được cải tiến màu sắc rõ ràng, giúp người dùng dễ quan sát và thao tác.

Thông qua đề tài này, nhóm đã củng cố kiến thức về lập trình hướng đối tượng, nắm vững cách tổ chức chương trình theo mô hình lớp như BJ\_Card, BJ\_Deck, BJ\_Hand, BJ\_Game và biết cách phối hợp giao diện với xử lý logic bên trong. Việc sử dụng tkinter cũng giúp nhóm hiểu rõ hơn về cách thiết kế giao diện đồ họa cơ bản trong Python. Ngoài ra, nhóm còn học được cách xử lý điểm số đặc biệt trong trò chơi Blackjack, như cách tính quân Át (A) một cách linh hoạt để tránh quắc (bust).

Trong thời gian tới, em mong muốn sẽ cải tiến trò chơi bằng cách hiển thị hình ảnh lá bài sinh động hơn, cho phép người chơi tùy chọn mức cược thay

vì cố định, bổ sung hiệu ứng âm thanh và hiển thị lịch sử chơi. Đồng thời, nhóm cũng hướng đến việc cải thiện trí tuệ nhân tạo cho Dealer để tăng tính cạnh tranh và hấp dẫn cho trò chơi.

Một lần nữa, em xin được chân thành cảm ơn Thầy giáo TS.Nguyễn Văn Huy vì những giờ học vui nhộn của thầy, những bài học trên lớp, vô số những bài tập về nhà và tiết học tâm huyết của thầy giáo đã giúp em có nền tảng và yêu thích môn lập trình Python. Trong quá trình làm bài, em có những sai sót, mong thầy sẽ cho em những lời nhận xét đáng giá để em cố gắng hơn vào những lần sau ạ.

Em xin chân thành cảm ơn thầy!

## PHỤ LỤC

Link GitHub báo cáo bài tập lớn + code + slide:

[https://github.com/Thuhien2004/PYTHON\\_2025.git](https://github.com/Thuhien2004/PYTHON_2025.git)

Mã QR github :



Link Youtube báo cáo: <https://youtu.be/c8mV7RQh4bM>



## TÀI LIỆU THAM KHẢO

1. Dawson, M. (2010). Python programming for the absolute beginner (3rd ed.). Course Technology PTR, CHAPTER 9.
2. Tkinter GUI Programming Grayson, J. (2017). Python and Tkinter Programming. Manning Publications.
3. Nguyễn Văn Hiệp. (2020). Giáo trình Python cơ bản và nâng cao. NXB Đại học Quốc gia Hà Nội.
4. <https://github.com>: Một số repo xây dựng game Blackjack bằng Python và tkinter.
5. RealPython. (2024). Build a Simple GUI Blackjack Game in Python. Truy cập từ: <https://realpython.com/>
6. <https://chatgpt.com/c/>