

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM

KHOA CÔNG NGHỆ THÔNG TIN

BỘ MÔN LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG



TRẦN NGUYỄN DUY LINH – 20110516

NGUYỄN TRẦN THẢO QUYÊN – 20110042

Đề Tài:

TÌM HIỂU 4 TRỤ CỘT TRONG OOP

**VIẾT CHƯƠNG TRÌNH MINH HỌA CHO CÁC ĐẶC ĐIỂM CỦA OOP TRONG
PYTHON**

GIÁO VIÊN HƯỚNG DẪN:

HUỲNH XUÂN PHỤNG

KHÓA 2021 – 2022

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Họ và tên Sinh viên 1: Trần Nguyễn Duy Linh. MSSV 1: 20110516

Họ và tên Sinh viên 2: Nguyễn Trần Thảo Quyên. MSSV 2: 20110042

Ngành: Công nghệ Thông tin

Tên đề tài: **Tìm hiểu 4 trụ cột trong OOP. Viết chương trình minh họa cho các đặc điểm của OOP trong ngôn ngữ Python**

Họ và tên Giáo viên hướng dẫn: Huỳnh Xuân Phụng

NHẬN XÉT

1. Về nội dung đề tài & khối lượng thực hiện:

.....

.....

.....

.....

.....

2. Ưu điểm:

.....

.....

.....

.....

.....

3. Khuyết điểm

.....
.....
.....
.....
.....

4. Đề nghị cho bảo vệ hay không ?

5. Đánh giá loại :

6. Điểm :

Tp. Hồ Chí Minh, ngày tháng năm 2021

Giáo viên hướng dẫn

(Ký & ghi rõ họ tên)

LỜI CẢM ƠN

Đầu tiên, chúng em xin gửi lời cảm ơn chân thành nhất đến Ban giám hiệu trường Đại học Sư Phạm Kỹ Thuật TP.HCM đã tạo điều kiện về cơ sở vật chất với hệ thống thư viện hiện đại, đa dạng các loại sách, tài liệu giúp thuận lợi cho việc tìm kiếm, nghiên cứu và học tập một cách hiệu quả dù trong hoàn cảnh dịch bệnh như hiện nay.

Xin cảm ơn thầy ThS. Huỳnh Xuân Phụng - giảng viên bộ môn Lập trình hướng đối tượng đã quan tâm giúp đỡ, hướng dẫn tận tình, tâm huyết cho chúng em trong suốt quá trình học tập và tìm hiểu bộ môn Lập trình hướng đối tượng . Từ đó, giúp chúng em tích lũy thêm nhiều kiến thức và có cái nhìn sâu sắc hơn để hoàn thành đề tài cuối kỳ này.

Có lẽ kiến thức là vô hạn mà sự tiếp nhận kiến thức của bản thân mỗi người luôn tồn tại những hạn chế nhất định. Do đó trong quá trình hoàn thành tiểu luận, chắc chắn không tránh khỏi những thiếu sót. Chúng em rất mong nhận được những góp ý đến từ thầy để bài cuối kỳ nhóm em được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

ĐỀ CƯƠNG CHI TIẾT

1. Tên học phần: Lập Trình Hướng Đối Tượng Mã học phần: OOPR230279

2. Tên Tiếng Anh: Object Oriented Programming

3. Số tín chỉ: 3 tín chỉ (2/1/6) (2 tín chỉ lý thuyết, 1 tín chỉ thực hành/thí nghiệm)

Phân bố thời gian: 15 tuần (2 tiết lý thuyết + 1*2 tiết thực hành + 6 tiết tự học/ tuần)

4. Các giảng viên phụ trách học phần:

1/ GV phụ trách chính: GV.TS. Lê Văn Vinh

2/ Danh sách giảng viên cùng GD:

2.1/ GVC. ThS. Nguyễn Minh Đạo

2.2/ GV. ThS. Nguyễn Trần Thị Văn

5. Điều kiện tham gia học tập học phần

Môn học tiên quyết: Không

Môn học trước: Kỹ Thuật Lập Trình (PRTE230385)

6. Mô tả học phần (Course Description)

Học phần nhằm trang bị cho người học kiến thức nền tảng về phương pháp lập trình hướng đối tượng như: lớp, đối tượng, tính đóng gói, tính thừa kế, tính đa hình, giao tiếp lập trình (interface), chồng toán tử, phương thức và lớp tĩnh, phương thức và lớp trừu tượng. Bên cạnh đó, người học có cơ hội triển khai các ứng dụng hướng đối tượng trên các ngôn ngữ lập trình khác nhau.

7. Mục tiêu học phần (Course Goals)

| Mục tiêu (Goals) | Mô tả (Goal description) <i>(Học phần này trang bị cho sinh viên)</i> | Chuẩn đầu ra CTĐT | Trình độ năng lực |
|---------------------|-----------------------------------------------------------------------------|----------------------|-------------------|
|---------------------|-----------------------------------------------------------------------------|----------------------|-------------------|

| | | | |
|-----------|--------------------------------------------------------------------------------------------|----------------------------------------|-------------|
| G1 | So sánh được các phương pháp luận trong lập trình hướng đối tượng. | ELO2 (1.2) | 3 |
| G2 | Có khả năng xây dựng các chương trình thể hiện tính chất căn bản nhất của hướng đối tượng. | ELO4 (2.1) ELO5 (2.2) ELO7 (2.4) | 4 3 3 |
| G3 | Có kỹ năng thuyết trình các vấn đề về phương pháp lập trình hướng đối tượng | ELO10 (3.2) | 3 |
| G4 | Khả năng vận dụng kỹ thuật lập trình hướng đối tượng để giải quyết vấn đề trong thực tế. | ELO15 (4.4) ELO16 (4.5) | 3 4 |

8. Chuẩn đầu ra của học phần

| Chuẩn đầu ra học phần | | Mô tả <i>Sau khi học xong môn học này, người học có thể:</i> | Chuẩn đầu ra | Trình độ năng lực |
|------------------------------|-------------|------------------------------------------------------------------------------------------------------------------|----------------------------------------|--------------------------|
| G1 | G1.1 | So sánh được những nguyên lý cơ bản của phương pháp lập trình hướng đối tượng để xây dựng chương trình máy tính. | ELO2 (1.2) | 3 |
| G2 | G2.1 | Xây dựng được các chương trình thể hiện tính chất căn bản nhất của hướng đối tượng. | ELO4 (2.1) ELO5 (2.2) ELO7 (2.4) | 4 3 3 |
| G3 | G3.1 | Trình bày trước đám đông sử dụng phương tiện trình chiếu | ELO10 (3.2) | 3 |

| | | | | |
|-----------|-------------|-------------------------------------------------------------------------------------|----------------|---|
| G4 | G4.1 | Vận dụng kỹ thuật lập trình hướng đối tượng để giải quyết một vấn đề trong thực tế. | ELO15 (4.4) | 3 |
| | | | ELO16 (4.5) | 4 |

9. Đạo đức khoa học:

Các bài tập ở nhà, bài kiểm tra phải được thực hiện từ chính bản thân sinh viên. Nếu bị phát hiện có sao chép thì xử lý các sinh viên có liên quan bằng hình thức đánh giá **0** (không) điểm quá trình và cuối kỳ.

10. Nội dung chi tiết môn học:

| Tuần | Nội dung | Chuẩn đầu ra môn học | Trình độ năng lực | Phương pháp dạy học | Phương pháp đánh giá |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|--------------------------|---------------------------------|-----------------------------|
| 1-2 | Chương 1: Giới thiệu về lập trình hướng đối tượng | | | | |
| | A./Các nội dung GD chính trên lớp (4) Nội dung GD lý thuyết: + Giới thiệu về phương pháp lập trình hướng đối tượng. + Lịch sử phát triển + Các tính chất và khái niệm trong lập trình hướng đối tượng + Nền tảng về ngôn ngữ lập trình hướng đối tượng (C#.NET/C++/Java/Python) | G1.1 | 3 | - Thuyết trình - Hỏi đáp | Trả lời câu hỏi ngắn |
| | B/ Các nội dung cần tự học ở nhà: (8) Làm các bài tập về nhà | | | | |
| 3-6 | Chương 2: Lớp và đối tượng | | | | |
| | A./Các nội dung GD chính trên lớp (4) | G1.1 G2.1 | 3 3 | - Thuyết trình | Đánh giá qua |

| | | | | | |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---|--------------------------------------------------|-------------------------------------------|
| | Nội dung GD lý thuyết: + Định nghĩa lớp, đối tượng + Các thành phần trong một lớp (Thuộc tính, Hàm thuộc tính, Hàm tạo, Hàm hủy, Phương thức) + Lớp tĩnh + Thành viên tĩnh trong lớp + Xây dựng ứng dụng hướng đối tượng cơ bản | | | - Dạy học theo nhóm - Thực hành trên máy tính | thực hiện Dự án học tập |
| | B/ Các nội dung cần tự học ở nhà: (8) | | | | |
| 7-11 | Chương 3: Tính kế thừa và tính đa hình | | | | |
| | A./Các nội dung GD chính trên lớp (4) | G2.1 | 3 | - Thuyết trình - Dạy học theo nhóm | Đánh giá qua thực hiện Dự án học tập nhóm |
| | Nội dung GD lý thuyết: + Tính kế thừa + Lớp trừu tượng và thành viên trừu tượng trong lớp + Tính đa hình + Xây dựng ứng dụng với tính kế thừa | G2.1 | 3 | | |
| | | G3.1 | 3 | | |
| | | G4.1 | 4 | | |
| | B/ Các nội dung cần tự học ở nhà: (6) | | | | |

| | | | | | |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|------------------|--------------------------------------------------------------------|-------------------------------------------|
| | | | | | |
| 12-13 | Chương 4: Hàm chồng toán tử | | | | |
| | A/ Các nội dung và PPGD chính trên lớp: (4) Nội dung GD lý thuyết: + Ý nghĩa của hàm chồng toán tử + Các loại hàm chồng toán tử + Xây dựng ứng dụng với hàm chồng toán tử | G2.1 G2.1 G3.1 G4.1 | 3 3 3 4 | - Thuyết trình - Dạy học theo nhóm - Thực hành trên máy tính | Đánh giá qua thực hiện Dự án học tập |
| | B/ Các nội dung cần tự học ở nhà: (6) | | | | |
| 14-15 | Chương 5: Giao tiếp lập trình (interface) và kiểu dữ liệu generic | | | | |
| | A/ Các nội dung và PPGD chính trên lớp: (4) Nội dung GD lý thuyết: + Giao tiếp lập trình (interface) + Kiểu dữ liệu generic + Xây dựng ứng dụng với interface và kiểu generic | G2.1 G2.1 G3.1 G4.1 | 3 3 3 4 | - Thuyết trình - Dạy học theo nhóm - Thực hành trên | Đánh giá qua thực hiện Dự án học tập nhóm |

| | | | | | |
|--|----------------------------------------------|--|--|-------------|--|
| | | | | máy tính | |
| | B/ Các nội dung cần tự học ở nhà: (6) | | | | |

11. Đánh giá kết quả học tập

- Thang điểm: **10**
- Kế hoạch kiểm tra như sau:

| Hình thức KT | Nội dung | Thời điểm | Chuẩn đầu ra đánh giá | Trình độ năng lực | Phương pháp đánh giá | Công cụ KT | Tỉ lệ (%) |
|----------------|--------------------------------------------------------|-----------|-----------------------|-------------------|-----------------------------------------------------|-----------------|-----------|
| Bài tập | | | | | | | 50 |
| BT#1 | Viết ứng dụng có khai báo các lớp (mang tính đóng gói) | Tuần 3-4 | G1.1 G2.1 | 3 3 | Đánh giá qua thực hiện Dự án học tập cá nhân / nhóm | Hồ sơ học tập | 10 |
| BT#2 | Viết ứng dụng hướng đối tượng có tính chất kế thừa | Tuần 3-5 | G1.1 G2.1 | 3 3 | Đánh giá qua thực | Câu hỏi + Hồ sơ | 15 |

| | | | | | | | |
|------------------------------------|------------------------------------------------------------------------------------|-------------------|--------------|--------|--------------------------------------------------------------------------|-------------------------------------|-----------|
| | | | | | hiện Dự án học tập cá nhân / nhóm | học tập | |
| BT#3 | Viết ứng dụng hướng đối tượng có tính chất đa hình | Tuần 6-9 | G1.1 G2.1 | 3 3 | Đánh giá qua thực hiện Dự án học tập cá nhân / nhóm | Câu hỏi + Hồ sơ học tập | 15 |
| BT#4 | Viết ứng dụng hướng đối tượng có áp dụng chồng toán tử, kiểu dữ liệu generic | Tuần 10- 11 | G1.1 G2.1 | 3 3 | Đánh giá qua thực hiện Dự án học tập cá nhân / nhóm | Câu hỏi + Hồ sơ học tập | 10 |
| Tiểu luận - Báo cáo cuối kỳ | | | | | | | 50 |
| | Xây dựng ứng dụng theo khảo sát thực tế theo phương | Tuần 14- 15 | G3.1 G4.1 | 3 4 | Đánh giá qua thực | Câu hỏi + Hồ sơ | |

| | | | | | | | |
|--|--------------------------------|--|--|--|-----------------------------------|---------|--|
| | pháp lập trình hướng đối tượng | | | | hiện Dự án học tập cá nhân / nhóm | học tập | |
| | | | | | | | |

| CĐR môn học | Hình thức kiểm tra | | | | |
|----------------|--------------------|-------|-------|-------|-----------|
| | BT #1 | BT #2 | BT #3 | BT #4 | Tiểu luận |
| G1.1 | x | x | x | x | |
| G2.1 | x | x | x | x | |
| G3.1 | | | | | x |
| G4.1 | | | | | x |

12. Tài liệu học tập

- Sách, giáo trình chính:

[1]. Dan Clark, **Beginning C# Object-Oriented Programming**, Apress, 2nd Edition, 2011.

- Sách (TLTK) tham khảo:

[1]. Jesse Liberty, **Programming C#**, O'Reilly, 2nd Edition, 2002.

[2]. Anders Hejlsberg, **The C# Programming Language**, Addison-Wesley, 4th Edition, 2010.

[3]. E Balagurusamy, **Object-Oriented programming with C++**, McGraw-Hill, 4th Edition, 2001.

[4]. Simon Kendal, **Object-Oriented Programming with Java**, Vetus Publishing ApS, 2009.

13. Ngày phê duyệt lần đầu: 11/08/2021

14. Cấp phê duyệt:

Trưởng khoa

Trưởng BM

Nhóm biên soạn

TS. Lê Văn Vinh

TS. Huỳnh Xuân Phụng

TS. Lê Văn Vinh

15. Tiến trình cập nhật ĐCCT

| | |
|----------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| Lần 1: Nội Dung Cập nhật ĐCCT lần 1: ngày 23 tháng 8 năm 2021 | <người cập nhật ký và ghi rõ họ tên) (Đã ký) Lê Văn Vinh Tổ trưởng Bộ môn: Huỳnh Xuân Phụng |
|----------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|

GVHD

SV THỰC HIỆN

HUỲNH XUÂN PHỤNG

Trần Nguyễn Duy Linh

Nguyễn Trần Thảo Quyên

MỤC LỤC

| | |
|----------------------------------------------------------------|-----------|
| BẢNG TÓM CÁC HÌNH VẼ, KÝ HIỆU, CHỮ VIẾT TẮT | 17 |
| TÓM TẮT KHÓA LUẬN | 18 |
| PHẦN I: MỞ ĐẦU | 19 |
| 1. Tính cấp thiết của đề tài: | 19 |
| 2. Mục đích của đề tài: | 20 |
| 3. Đối tượng nghiên cứu:..... | 20 |
| 4. Phạm vi nghiên cứu:..... | 20 |
| 5. Phân tích đánh giá:..... | 20 |
| 6. Kết quả đạt được: | 21 |
| PHẦN II: NỘI DUNG: | 22 |
| CHƯƠNG 1: GIỚI THIỆU CHUNG: | 22 |
| 1.1. Lập trình hướng đối tượng là gì? | 22 |
| 1.1.1. Khái niệm: | 22 |
| 1.1.2. Sự khác nhau giữa đối tượng và lớp: | 22 |
| 1.1.3. Các nguyên lý cơ bản của OOP: | 22 |
| 1.1.4. Các ưu điểm của lập trình hướng đối tượng: | 23 |
| 1.2. Ngôn ngữ lập trình Python? | 23 |
| 1.2.1. Khái niệm: | 23 |
| 1.2.2. Các ứng dụng của Python:..... | 23 |
| 1.2.3. Đặc điểm:..... | 24 |
| 1.2.4. Ưu điểm và nhược điểm của ngôn ngữ Python: | 24 |
| 1.3. Lập trình hướng đối tượng với Python? | 25 |
| 1.3.1. Giới thiệu về OOP trong Python..... | 25 |
| 1.3.2. Các nguyên lý:..... | 25 |
| 1.3.3. Lớp (Class) và đối tượng (Object) | 26 |

| | |
|---------------------------------------------------------------------------------------------|----|
| 1.3.4. Phương thức (Method) | 26 |
| CHƯƠNG 2: TÌM HIỂU VỀ 4 TRỤ CỘT CỦA LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG TRONG NGÔN NGỮ PYTHON: | 27 |
| 2.1. Các khái niệm cơ bản: | 27 |
| 2.1. Tính kế thừa (Inheritance): | 27 |
| 2.2. Tính đóng gói (Encapsulation): | 29 |
| 2.3. Tính đa hình (Polymorphism): | 30 |
| 2.4. Tính trừu tượng (Abstraction): | 32 |
| 2.5. So sánh các trụ cột của OOP giữa ngôn ngữ Python với C#: | 34 |
| CHƯƠNG 3: CHƯƠNG TRÌNH PYTHON ỨNG DỤNG BỐN TRỤ CỘT CỦA LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG: | 36 |
| 3.1. Giới thiệu sơ lược về chương trình: | 36 |
| 3.2. Sơ đồ của chương trình: | 37 |
| 3.3. Ý tưởng: | 37 |
| 3.4. Cách cài đặt để chạy được chương trình: | 38 |
| 3.5. Hướng dẫn sử dụng chương trình: | 38 |
| PHẦN 3: KẾT LUẬN | 39 |
| 1. Những kết quả đạt được: | 39 |
| 2. Đóng góp mới và đề xuất mới: | 40 |
| 3. Ưu nhược điểm và hướng phát triển của đề tài: | 40 |
| 4. Kết luận: | 40 |
| TÀI LIỆU THAM KHẢO | 42 |
| PHỤ LỤC | 42 |

BẢNG TÓM CÁC HÌNH VẼ, KÝ HIỆU, CHỮ VIẾT TẮT

| HÌNH VẼ / KÝ HIỆU/ CHỮ VIẾT TẮT | GIẢI THÍCH |
|---------------------------------|---------------------------------------------------------|
| OOP | Object-oriented programming - Lập trình hướng đối tượng |
| IoT | Internet of Things - Internet vạn vật |
| Obj | Object – Đối tượng |
| | |

TÓM TẮT KHÓA LUẬN

Sự bùng nổ về cuộc cách mạng 4.0 đã khiến cụm từ Internet Of Things hay vạn vật kết nối internet trở nên không còn quá xa lạ với nhiều người. Rất nhiều người nói rằng nó có thể sẽ trở thành kẻ thay đổi toàn bộ cục diện của thị trường, bằng cách khiến cho mọi vật đều được kết nối với nhau. Ngày nay chúng ta thấy có rất nhiều thành quả từ chính IoT mang lại. Rất nhiều công ty hiện nay đã và đang dốc hết sức lực để phát triển thật mạnh IoT. Chính vì tầm quan trọng đó nên nhóm chúng em quyết định chọn đề tài liên quan tới IoT, để từ đề án này mà nhóm có thêm kinh nghiệm cho công việc trong tương lai. Cụ thể hơn nhóm nhận thấy được tầm quan trọng của việc **“Quản lý thông tin vé máy bay”**. Trong 15 tuần học vừa qua, được sự giảng dạy, giúp đỡ và hỗ trợ của GV, TS. Huỳnh Xuân Phụng - giảng viên bộ môn Lập trình hướng đối tượng. Thầy để chỉ dạy nhóm em về 4 trụ cột trong OOP và viết chương trình minh họa cho các đặc điểm của OOP trong ngôn ngữ C#. Vào 5 tuần cuối, nhóm em đã chọn đề tài **“Tìm hiểu 4 trụ cột trong OOP. Viết chương trình minh họa cho các đặc điểm của OOP trong ngôn ngữ Python”** ngoài tự học lý thuyết trên ở những website uy tín, thì hằng tuần nhóm báo cáo về tiến độ thực hiện, được thầy góp ý, đánh giá và đặt những câu hỏi thực tế nhằm cho sinh viên có thể trau dồi thêm kiến thức. Từ đó, nhóm em đã rút kết ra nhiều kinh nghiệm và thực hiện chương trình: **“Quản lý thông tin vé máy bay”**. Trong suốt quá trình làm dự án, nhóm em có gặp một số vấn đề: thời gian giữa các bạn trong nhóm, những lỗi trong chương trình... Tuy nhiên, thì nhóm đã cố gắng tìm hiểu để có thể vượt qua những vấn đề trên bằng cách xem lại những video bài giảng của GV trên fhqx lần đọc thêm tài liệu trên mạng. Cuối cùng, thì nhóm em đã hoàn thành chương trình nhằm phục vụ cho việc hoàn thành môn học và nếu thêm nhiều điều kiện để phát triển một cách hoàn thiện nhằm có thể hỗ trợ cho các dịch vụ khác của xã hội.

PHẦN I: MỞ ĐẦU

1. Tính cấp thiết của đề tài:

Có vô số ngôn ngữ lập trình được sử dụng để phát triển ứng dụng trên khắp thế giới, mỗi loại đều có các tính năng cụ thể cũng như ưu và nhược điểm. Một trong những ngôn ngữ lập trình được ưa chuộng nhất hiện nay là Python. Python đã trở thành một trong số ít ngôn ngữ có mức độ phù hợp trong hiện tại cũng như có tiềm năng cao cho tương lai. Kể từ khi nó được thành lập vào năm 1991 bởi một nhà phát triển tên là Guido Van Rossum, phạm vi tiếp cận của Python đã phát triển theo từng ngày, với một số công ty đa quốc gia sử dụng nó cho các hoạt động phát triển ứng dụng của họ. Nhóm em đã áp dụng 4 trụ cột chính trong OOP và viết chương trình cho các đặc điểm đó bằng ngôn ngữ Python. Lập trình hướng đối tượng (tiếng Anh: Object-oriented programming, viết tắt: OOP) là một kỹ thuật hỗ trợ, cho phép lập trình viên trực tiếp làm việc với các đối tượng mà họ định nghĩa lên. Hiệu quả của kỹ thuật này giúp tăng năng suất, đơn giản hoá độ phức tạp khi bảo trì cũng như mở rộng phần mềm. Hiện nay có khá nhiều ngôn ngữ lập trình theo hướng đối tượng như C++, Java, PHP,... và còn cả Python. Khái niệm về OOP trong Python tập trung vào việc tạo code sử dụng lại. Khái niệm này còn được gọi là DRY (Don't Repeat Yourself).

Ngày nay khi chất lượng cuộc sống của con người được nâng cao thì con người ngày càng đòi hỏi nhiều hơn về các nhu cầu khác cho bản thân một trong số đó là nhu cầu về công nghệ thiết bị hiện đại là phục vụ mọi mặt trong cuộc sống của con người. Với sự phát triển của công nghệ thông tin luôn cố gắng đáp ứng các yêu cầu của con người. Việc đưa công nghệ thông tin áp dụng vào quản lý đã và đang được đưa vào thực hiện để thể hiện tốc độ và độ chính xác. Với việc tạo ra chương trình: **“Quản lý thông tin vé máy bay”** nhóm em muốn tạo ra một hệ thống giúp các sân bay có thể dễ dàng quản lý thông tin vé máy bay lẫn thông tin khách hàng của mình một cách đơn giản, tối ưu và hiệu quả nhất. Hiện nay máy bay đã trở thành phương tiện di chuyển gần gũi và cũng được nhiều người sử dụng vì vậy cần trang bị kiến thức này để hành khách có thể nhanh chóng và linh hoạt, chủ động hơn với việc làm thủ tục tại sân bay, giúp hệ thống sân bay dễ dàng kiểm soát

được thông tin hành khách, giúp hành khách nhanh chóng hoàn tất thủ tục tại sân bay, đồng thời đảm bảo được an ninh sân bay.

2. Mục đích của đề tài:

Nhằm bổ sung kiến thức nền tảng về lập trình hướng đối tượng, tìm hiểu về bốn trụ cột trong OOP

Làm quen thêm được một ngôn ngữ lập trình mới “ Python”, có từng nghe nhắc tới, tuy nhiên đến giờ mới có thể tìm hiểu sâu hơn và ứng dụng vào lập trình hướng đối tượng

Viết chương trình minh họa cho các đặc điểm của OOP trong ngôn ngữ Python

3. Đối tượng nghiên cứu:

Tập trung nghiên cứu, bổ sung kiến thức và tìm hiểu sâu về các trụ cột trong OOP, phân tích, so sánh các trụ cột đó trong ngôn ngữ Python với các ngôn ngữ khác, đưa ra kết luận và ứng dụng vào chương trình mà chương trình đó có đầy đủ những đặc điểm mà trong OOP.

4. Phạm vi nghiên cứu:

Các kiến thức về liên quan đến ngành công nghệ thông tin nói chung, lập trình hướng đối tượng nói riêng và tìm hiểu, nghiên cứu về ngôn ngữ lập trình Python

5. Phân tích đánh giá:

❖ Những thuận lợi:

- Nhóm được Thầy Huỳnh Xuân Phụng hỗ trợ tận tình. Tạo điều kiện cho nhóm được báo cáo hằng tuần, để tăng khả năng làm việc nhóm trong tình hình dịch bệnh này. Thầy đặt ra những câu hỏi hay không chỉ để nhóm em trau dồi thêm kiến thức mà còn rút kết được những kinh nghiệm những thiếu sót, để hằng tuần tiến bộ và hoàn thành đề tài cuối kì hoàn thiện nhất.
- Các kiến thức về OOP được thầy trang bị đầy đủ khi học bằng ngôn ngữ C#. Khi tự tìm hiểu ở ngôn ngữ Python thì cũng không gặp quá nhiều khó khăn, nguồn tài liệu phong phú về Python.
- Tuy lịch học có nhiều khi trái múi giờ với nhau. Nhưng nhóm em luôn có sự phối hợp làm việc, hỗ trợ lẫn nhau trong suốt quá trình thực hiện đề tài

❖ **Những khó khăn:**

- Vì tình hình dịch bệnh, nên việc tra đổi với bạn chung nhóm hết sức khó khăn. Hầu như thời gian của nhóm em thì không được khớp lắm. Nên thường tra đổi trên Zalo, Facebook,...
- Chưa có kiến thức sâu với ngôn ngữ mới – Python. Vì thời gian tìm hiểu còn hẹp
- Thời gian để nhóm báo cáo và làm tiểu luận thì trùng với thời gian thi cuối kỳ của một số môn khác. Nên là có một vài tuần nội dung báo cáo chưa được cụ thể lắm

6. Kết quả đạt được:

Suốt 5 tuần vừa tìm hiểu nội dung để làm báo cáo vừa làm đồ án cuối kỳ, nhóm em đã đạt được:

- Những kiến thức về nền lập trình hướng đối tượng, các trụ cột, những nội dung liên quan
- Học tập và làm việc với ngôn ngữ mới – Python, các cú pháp, câu lệnh cơ bản, ứng dụng Python kết hợp với OOP
- Kỹ năng thuyết trình, trả lời câu hỏi
- Kỹ năng làm việc nhóm
- Viết được chương trình minh họa

PHẦN II: NỘI DUNG:

CHƯƠNG 1: GIỚI THIỆU CHUNG:

1.1. Lập trình hướng đối tượng là gì?

1.1.1. Khái niệm:

Lập trình hướng đối tượng (OOP) là một kỹ thuật lập trình cho phép lập trình viên tạo ra các đối tượng trong code trừu tượng hóa các đối tượng.

Một đối tượng bao gồm 2 thông tin: thuộc tính và phương thức.

- Thuộc tính chính là những thông tin, đặc điểm của đối tượng. Ví dụ: con người có các đặc tính như mắt, mũi, tay, chân...
- Phương thức là những thao tác, hành động mà đối tượng đó có thể thực hiện. Ví dụ: một người sẽ có thể thực hiện hành động nói, đi, ăn, uống, . . .

Một lớp là một kiểu dữ liệu bao gồm các thuộc tính và các phương thức được định nghĩa từ trước. Đây là sự trừu tượng hóa của đối tượng. Khác với kiểu dữ liệu thông thường, một lớp là một đơn vị (trừu tượng) bao gồm sự kết hợp giữa các phương thức và các thuộc tính. Hiểu nôm na hơn là các đối tượng có các đặc tính tương tự nhau được gom lại thành một lớp đối tượng.

1.1.2. Sự khác nhau giữa đối tượng và lớp:

- Lớp có thể hiểu nó như là khuôn mẫu, đối tượng là một thực thể thể hiện dựa trên khuôn mẫu đó. Ví dụ: Ta nói về loài chó, có thể hiểu nó là class (lớp) chó có:
 - + Các thông tin, đặc điểm: 4 chân, 2 mắt, có đuôi, có chiều cao, có cân nặng, màu lông...
 - + Các hành động như: sủa, đi, ăn, ngủ...
- Đối tượng thì chính là con chó Phú Quốc ta đang nuôi trong nhà cũng mang đặc tính của lớp chó.

1.1.3. Các nguyên lý cơ bản của OOP:

- Tính đóng gói (Encapsulation)

- Tính kế thừa (Inheritance)
- Tính đa hình (Polymorphism)
- Tính trừu tượng (Abstraction)

1.1.4. Các ưu điểm của lập trình hướng đối tượng:

- Dựa trên nguyên lý kế thừa, trong quá trình mô tả các lớp có thể loại bỏ những chương trình bị lặp, dư. Và có thể mở rộng khả năng sử dụng các lớp mà không cần thực hiện lại. Tối ưu và tái sử dụng code hiệu quả.
- Đảm bảo rút ngắn thời gian xây dựng hệ thống và tăng năng suất thực hiện.
- Sự xuất hiện của 2 khái niệm mới là lớp và đối tượng chính là đặc trưng của phương pháp lập trình hướng đối tượng. Nó đã giải quyết được các khuyết điểm của phương pháp lập trình hướng cấu trúc để lại. Ngoài ra 2 khái niệm này đã giúp biểu diễn tốt hơn thế giới thực trên máy tính.

1.2. Ngôn ngữ lập trình Python?

1.2.1. Khái niệm:

Python là ngôn ngữ lập trình đa năng ra đời năm 1991 do Guido van Rossum sáng tạo ra. Đây là ngôn ngữ lập trình với những điểm mạnh như dễ đọc, dễ nhớ, dễ học. Với cấu trúc rõ ràng, thuận tiện nên thu hút khá nhiều người mong muốn học ngôn ngữ này.

So với các ngôn ngữ khác, Python có cấu trúc cú pháp ít hơn. Cấu trúc của Python cho phép người sử dụng viết mã lệnh với số lần gõ phím ít nhất.

Lúc đầu Python được phát triển trên cơ sở Unix. Nhưng theo sự phát triển của thời đại, ngôn ngữ lập trình này được mở rộng sang mọi hệ điều hành từ MS Dos đến Mac Os, Linux,... Dù sự phát triển này của Python được đóng góp bởi nhiều cá nhân, nhưng Guido van Rossum vẫn là người nắm vai trò chủ chốt trong việc quyết định sự phát triển của Python.

1.2.2. Các ứng dụng của Python:

- Làm trang Web với Framework của Python
- Python dùng để tạo nguyên mẫu phần mềm
- Ứng dụng trong khoa học và tính toán

- Viết tool để tự động hóa công việc
- Khoa học máy tính
- Lĩnh vực IoT – Internet Vạn Vật
- Làm dự án Blockchain với Python
- Làm game với PyGame
- Machine Learning
- Là ngôn ngữ được sử dụng cho mục đích giảng dạy

1.2.3. Đặc điểm:

- Ngôn ngữ lập trình đơn giản, dễ học – dễ học
- Miễn phí, mã nguồn mở
- Khả năng di động linh hoạt
- Khả năng mở rộng và có thể nhúng
- Ngôn ngữ thông dịch cấp cao
- Thư viện tiêu chuẩn lớn để giải quyết những tác vụ phổ biến
- Hướng đối tượng

Mọi thứ trong Python đều là hướng đối tượng. Lập trình hướng đối tượng (OOP) giúp giải quyết những vấn đề phức tạp một cách trực quan. Với OOP, có thể phân chia những vấn đề phức tạp thành những tập nhỏ hơn bằng cách tạo ra các đối tượng. Python hỗ trợ cả lập trình hướng đối tượng, một trong những tính năng chính của nó. Nó cũng hỗ trợ nhiều kế thừa, không giống như Java.

1.2.4. Ưu điểm và nhược điểm của ngôn ngữ Python:

❖ Ưu điểm chính của việc sử dụng Python

Python có một số tính năng độc đáo giúp việc lập trình trở nên đơn giản hơn nhiều. Một số tính năng giúp làm việc với lợi thế của Python:

- Dễ đọc và dễ học
- Giảm chi phí bảo trì
- Tránh tác hại từ lỗi phần mềm

- Khả năng ứng dụng rộng rãi
- Quản lý bộ nhớ
- Đơn giản và nhanh chóng
- Mã hóa không đồng bộ
- Tích hợp với các ngôn ngữ khác
- Tích hợp ứng dụng doanh nghiệp

❖ **Nhược điểm chính của việc sử dụng Python:**

Cùng với một số ưu điểm, Python có một số hạn chế trong các lĩnh vực hiệu suất và bảo mật. Sau đây là một số nhược điểm đáng kể của việc sử dụng Python.

- Tốc độ thực thi chậm
- Tiêu thụ bộ nhớ lớn
- Không thích hợp cho phát triển trò chơi và thiết bị di động
- Phát hiện lỗi trong mã
- Quyền truy cập cơ sở dữ liệu
- Hạn chế thiết kế
- Khó kiểm tra

1.3. Lập trình hướng đối tượng với Python?

1.3.1. Giới thiệu về OOP trong Python

Lập trình hướng đối tượng (tiếng Anh: Object-oriented programming, viết tắt: OOP) là một kỹ thuật hỗ trợ, cho phép lập trình viên trực tiếp làm việc với các đối tượng mà họ định nghĩa lên. Hiệu quả của kỹ thuật này giúp tăng năng suất, đơn giản hoá độ phức tạp khi bảo trì cũng như mở rộng phần mềm. Hiện nay có khá nhiều ngôn ngữ lập trình theo hướng đối tượng như C++, Java, PHP,... và còn cả Python.

1.3.2. Các nguyên lý:

Trong Python, khái niệm về OOP tuân theo một số nguyên lý cơ bản là tính đóng gói, tính kế thừa và tính đa hình, tính trừu tượng.

- Tính kế thừa (Inheritance): Nguyên tắc này cho phép xây dựng một lớp mới dựa trên

1 lớp đã khai báo từ trước. Lớp con có thể sử dụng lại các thuộc tính và phương thức của lớp cha mà không cần khai báo lại. Tùy thuộc vào từng ngôn ngữ cho phép việc kế thừa 1 hoặc nhiều class cha.

- Tính đóng gói (Encapsulation): các dữ liệu và phương thức có liên quan với nhau được đóng gói thành 1 lớp. Tức là mỗi lớp được xây dựng để thực hiện một nhóm chức năng đặc trưng của riêng lớp đó. Che giấu các thông tin của lớp đó đối với bên ngoài thể hiện ở public, protected, private đối với từng thuộc tính và phương thức.
- Tính đa hình (Polymorphism): là khái niệm mà hai hoặc nhiều lớp có những phương thức giống nhau nhưng có thể thực thi theo những cách thức khác nhau.
- Tính trừu tượng (Abstraction): tổng quát hóa phương thức của đối tượng không quan tâm phương thức thực hiện như thế nào, được thể hiện bởi interface (có các tên phương thức nhưng ko có body của phương thức, khi class nào impliment interface thì thực hiện nó).

1.3.3. Lớp (Class) và đối tượng (Object)

Class và Object là hai khái niệm cơ bản trong lập trình hướng đối tượng.

Đối tượng (Object) là những thực thể tồn tại có hành vi.

- Ví dụ đối tượng là một xe ô tô có tên hãng, màu sắc, loại nguyên liệu, hành vi đi, dừng, đỗ, nổ máy...

Lớp (Class) là một kiểu dữ liệu đặc biệt do người dùng định nghĩa, tập hợp nhiều thuộc tính đặc trưng cho mọi đối tượng được tạo ra từ lớp đó.

Thuộc tính là các giá trị của lớp. Sau này khi các đối tượng được tạo ra từ lớp, thì thuộc tính của lớp lúc này sẽ trở thành các đặc điểm của đối tượng đó.

Phân biệt giữa Đối tượng (Object) và Lớp (Class):

- Đối tượng (Object): có trạng thái và hành vi.
- Lớp (Class): có thể được định nghĩa như là một template mô tả trạng thái và hành vi mà loại đối tượng của lớp hỗ trợ. Một đối tượng là một thực thể (instance) của một lớp

1.3.4. Phương thức (Method)

Phương thức (Method) là các hàm được định nghĩa bên trong phần thân của một lớp. Chúng được sử dụng để xác định các hành vi của một đối tượng.

CHƯƠNG 2: TÌM HIỂU VỀ 4 TRỤ CỘT CỦA LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG TRONG NGÔN NGỮ PYTHON:

2.1. Các khái niệm cơ bản:

Python là một ngôn ngữ hướng đối tượng. Vì vậy hầu hết mọi thứ trong Python đều là những đối tượng với những thuộc tính (properties) và phương thức (methods).

OOP là một kỹ thuật lập trình cho phép tạo ra các đối tượng để trừu tượng hóa 1 đối tượng thực tế (đưa các đối tượng trong thực tế vào trong code). Cho phép lập trình viên tương tác với các đối tượng này.

Một đối tượng bao gồm: thuộc tính (attributes) và phương thức (methods).

- Thuộc tính (attributes) chính là những thông tin, đặc điểm của đối tượng. VD: Con mèo (màu sắc, có đuôi, 2 tai, bốn chân, ...)
- Phương thức (methods) là những thao tác, hành động mà đối tượng đó có thể thực hiện. VD: Con mèo (ăn, chạy nhảy, cắn, gào, rên,)

Lớp có thể hiểu là một bản thiết kế để tạo ra một thực thể nào đó, là tập hợp nhiều thuộc tính đặc trưng cho đối tượng được tạo ra từ lớp này. VD: Con mèo (tên, màu sắc, có đuôi, tai, chân, ...)

Đối tượng là một thực thể của 1 lớp nào đó, được tạo ra từ lớp đó. VD: Mèo mun (tên là mèo mun, màu đen, có đuôi dài, 2 tai, 4 chân,).

2.1. Tính kế thừa (Inheritance):

- Cho phép một lớp (class) có thể kế thừa các thuộc tính và phương thức từ các lớp khác đã được định nghĩa. Lớp đã có gọi là lớp cha (base class hoặc parent class), lớp mới phát sinh gọi là lớp con (child class hoặc derived class). Lớp con kế thừa tất cả thành phần của lớp cha, có thể mở rộng các thành phần kế thừa và bổ sung thêm các thành phần mới.
- Overriding (Ghi đè) trong Python

+ Python cho phép ghi đè lên các phương thức của lớp cha. Có thể thực hiện việc ghi đè phương thức của lớp cha nếu muốn có tính năng khác biệt hoặc đặc biệt trong lớp con.

- Ngoài ra Python có hai hàm `isinstance()` và `issubclass()` được dùng để kiểm tra mối quan hệ của hai lớp và instance.

- + Hàm `issubclass(classA, classB)` trả về `True` nếu class A là lớp con của class B.

- + Hàm `isinstance(obj, Class)` trả về `True` nếu obj là một instance của lớp Class hoặc là một instance của lớp con của Class.

- **Cú pháp:**

```
class BaseClass:
```

```
    Body of base class
```

```
class DerivedClass(BaseClass):
```

```
    Body of derived class
```

- **Ví dụ:**

```

class info:
    graduationyear = 2024

    def __init__(self, fname, lname):
        self.firstname = fname
        self.lastname = lname

    def name(self):
        print("Name : ", self.firstname,
              self.lastname)

    def age(self, age):
        print("Age : ", age)
        address = "22 NDC"

class Student(info):

```

⇒ Class Student sẽ được kế thừa toàn bộ các thuộc tính và phương thức của Class info và các thuộc tính, phương thức của chính nó (graduationyear / year)

2.2. Tính đóng gói (Encapsulation): Dữ liệu và phương thức có liên quan với nhau được đóng gói thành 1 lớp. Tức là mỗi lớp được xây dựng để thực hiện một nhóm chức năng đặc trưng của riêng lớp đó. Che giấu các thông tin của lớp đó đối với bên ngoài thể hiện ở public, protected, private đối với từng thuộc tính và phương thức.

- Cú pháp:

Sử dụng OOP trong Python, để có thể hạn chế quyền truy cập vào trạng thái bên trong của đối tượng. Điều này ngăn chặn dữ liệu bị sửa đổi trực tiếp, được gọi là đóng gói. Trong Python, chúng ta biểu thị thuộc tính private này bằng cách sử dụng dấu gạch dưới làm tiền tố: “_” hoặc “__”.

Public: công khai.

Protected: có thể truy cập bên trong class hoặc class kế thừa, bắt đầu với “_”

Private: chỉ có thể sử dụng nội bộ bên trong class, bắt đầu với “__”

- **Ví dụ:**

```
class info:
    public = "pub"
    __protected = "pro"
    __private = "priv"
    print(__private) #thực hiện được
    def __init__(self, fname, lname):
        self.firstname = fname
        self.lastname = lname
    def __name(self):
        print("Name : ", self.firstname, self.lastname)
    def age(self, age):
        print("Age : ", age)

class Encap(info):
    pass
    #thanCT
    print(info.public)
    info.__name()
    s1 = Encap("A", "B")
    print(s1.__protected)
    print(s1.__private) #lỗi
```

⇒ Các biến public, __protected có thể truy xuất từ bên ngoài (__protected truy xuất thông qua class kế thừa), riêng __private không thể truy xuất từ bên ngoài nên khi thực hiện sẽ xảy ra lỗi, chỉ có thể sử dụng ngay bên trong class chứa nó

2.3. Tính đa hình (Polymorphism): Khái niệm mà hai hoặc nhiều lớp có những phương thức giống nhau nhưng có thể thực thi theo những cách thức khác nhau.

- **Ví dụ:**

➤ Tính đa hình đối với hàm: có một số hàm trong Python có thể tương thích để được thực thi với nhiều kiểu dữ liệu. Một trong những hàm như vậy là hàm len(). Nó có thể thực thi với nhiều kiểu dữ liệu trong Python.

```
D: > Python > ham.py
1 print(len(" Nguyen Tran Thao Quyen "))
2 print(len(["Tran", "Nguyen", "Duy", "Linh"]))
3 print(len({"Name": "Linh", "Address": "Tien Giang"}))

PS C:\Users\thaoq> python -u "d:\Python\tempCodeRunnerFile.py"
24
4
2
```

Có thể thấy rằng nhiều kiểu dữ liệu khác nhau như chuỗi ký tự, list, tuple, set, và từ điển có thể hoạt động với hàm len(). Tuy nhiên, Có thể thấy rằng nó hoạt động và trả về dữ liệu tùy thuộc vào kiểu dữ liệu mà nó làm việc với.

➤ Tính đa hình trong kế thừa:

- Trong Python, tính đa hình cho phép chúng ta định nghĩa các phương thức trong lớp con có cùng tên với các phương thức trong lớp cha. Trong kế thừa, lớp con kế thừa các phương thức từ lớp cha. Tuy nhiên, chúng cũng có thể sửa đổi một phương thức trong một lớp con mà nó đã kế thừa từ lớp cha. Điều này đặc biệt hữu ích trong trường hợp phương thức được kế thừa từ lớp cha không hoàn toàn phù hợp với lớp con.

- Khi một phương thức trong lớp con có cùng tên, cùng tham số hoặc ký hiệu và cùng kiểu trả về giống như một phương thức trong lớp cha của nó, thì phương thức trong lớp con được cho là ghi đè phương thức trong lớp cha.

- Tính đa hình cho phép chúng ta truy cập vào các phương thức và thuộc tính bị ghi đè mà có cùng tên với lớp cha.

```
D: > Python > vd3.py > ...
1 class Hoa:
2     def giới_thieu (self):
3         print("Đây là bông hoa")
4     def mau_sac (self):
5         print("Đa dạng màu sắc")
6 class Hoa_Hong(Hoa):
7     def mau_sac (self):
8         print("Màu đỏ")
9 class Hoa_Lan(Hoa):
10    def mau_sac (self):
11        print("Màu trắng")
12 flo = Hoa ()
13 rose = Hoa_Hong()
14 orc = Hoa_Lan()
15 flo.giới_thieu()
16 flo.mau_sac()
17
18 rose.giới_thieu()
19 rose.mau_sac()
20
21 orc.giới_thieu()
22 orc.mau_sac()
```

```
PS C:\Users\thaoq> python -u "d:\Python\tempCodeRunnerFile.py"
Đây là bông hoa
Đa dạng màu sắc
Đây là bông hoa
Màu đỏ
Đây là bông hoa
Màu trắng
```

- **Nạp chồng toán tử:** đối với kiểu dữ liệu ký tự, danh sách,... Nạp chồng toán tử trong Python được hiểu như phép nối các phần tử lại với nhau, hoặc là phép cộng thông thường (+) đối với các kiểu dữ liệu số nguyên, số thực

```
class example:
    def __init__(self, a, b):
        self.a = a
        self.b = b
    def __add__(self, other):
        a = self.a + other.a
        b = self.b + other.b
        return a, b

t1 = example(5, 2)
t2 = example(-1, 3)
print(t1 + t2)           #(4, 5)
```

- Nếu chỉ gọi phương thức `__init__` thì Python sẽ không thể cộng vì không biết cách cộng 2 đối tượng này với nhau. Ta cần định nghĩa một phương thức đặc biệt để thực hiện toán tử này, việc này gọi là nạp chồng toán tử.
- Trong ví dụ này, phương thức `__add__` sẽ định nghĩa để thực hiện toán tử (+). Khi dùng toán tử + để cộng 2 đối tượng thì `__add__` sẽ tự động được gọi.

2.4. Tính trừu tượng (Abstraction):

Tổng quát hóa phương thức của đối tượng không quan tâm phương thức thực hiện như thế nào, được thể hiện bởi interface (có các tên phương thức nhưng ko có body của phương thức, khi class nào impliment interface thì thực hiện nó).

Mục đích: ẩn tất cả dữ liệu hoặc quy trình không liên quan của ứng dụng để giảm độ phức tạp và tăng hiệu quả sử dụng của phần mềm.

VD: khi sử dụng điện thoại, ta chỉ tương tác với các tính năng, còn những hoạt động nền sau tính năng đó đều không được thể hiện ra ngoài.

Trong Python, tính trừu tượng được thể hiện thông qua class trừu tượng. Class chứa các phương thức trừu tượng (abstract method) gọi là lớp trừu tượng (abstract class)

- Các phương thức trừu tượng không chứa các lệnh thực thi.
- Class trừu tượng không thể tự khởi tạo, chỉ có thể khởi tạo thông qua các class con kế thừa lại nó.
- Một class con kế thừa lại abstract class phải khai báo lại toàn bộ các pt trừu tượng có trong abstract class
- Abstract class có thể xem là một bản thiết kế cho các class khác.

- **Cú pháp:**

Để có thể khai báo được một abstract class trong Python, thì class này bắt buộc phải được kế thừa từ một ABC (Abstract Base Classes) của Python

Và để gọi được class này trong chương trình phải import nó. Cú pháp import như sau:

```
from abc import ABC
```

Cú pháp khai báo abstract class:

```
class ClassName(ABC):
```

```
# code
```

Trong đó: ClassName là tên của abstract class mà muốn khai báo.

- **Ví dụ:**

```
# Abstraction - Trừu tượng hóa
from abc import ABC, abstractmethod
class DaGiac(ABC):
    # khai báo phương thức trừu tượng
    @abstractmethod
    def canh(self):
        pass
    def chuvi(self):
        pass

class HinhChuNhat(DaGiac):
    def __init__(self, dai, rong):
        self.dai = dai
        self.rong = rong
    def canh(self):
        print("Hình chu nhật có 4 cạnh")
    def chuvi(self):
        print("Chu vi hình chu nhật = ", (self.dai + self.rong) * 2)
        return (self.dai + self.rong) * 2

class HinhVuong(DaGiac):
    def __init__(self, canhv):
        self.canhv = canhv
    def canh(self):
        print("Hình vuông có 4 cạnh bằng nhau")
    def chuvi(self):
        print("Chu vi hình vuông = ", self.canhv * 4)
        return self.canhv * 4

hv = HinhVuong(8)
hv.chuvi()
hv.canh()

hcn = HinhChuNhat(10,5)
hcn.canh()
hcn.chuvi()
```

Chu vi hình vuông = 32
Hình vuông có 4 cạnh bằng nhau
Hình chu nhật có 4 cạnh
Chu vi hình chu nhật = 30

⇒ Khai báo lớp trừu tượng ABC, các lớp kế thừa lại nó sẽ sử dụng phương thức trừu tượng theo các cách khác nhau và cho ra kết quả khác nhau

2.5. So sánh các trụ cột của OOP giữa ngôn ngữ Python với C#:

C# cũng là một trong những ngôn ngữ lập trình được yêu thích nhất. Dù cả Python và C# đều dựa trên concept OOP, dễ đọc, dễ viết, phát triển nhanh và performance tốt song giữa chúng vẫn tồn tại những điểm khác biệt, cũng như có những ưu và nhược điểm riêng.

| | C# | PYTHON |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ĐIỂM MẠNH | <ul style="list-style-type: none"> Được sự hỗ trợ của Common Language Infrastructure nên tốc độ vượt trội, trải nghiệm ứng dụng cũng tốt hơn. Nếu đã hiểu được Java, C/C++ thì làm việc với C# sẽ dễ dàng hơn và ngược lại. Được tích hợp NET. Framework. | <ul style="list-style-type: none"> Ngôn ngữ lập trình có tính động, có mã nguồn mở, dễ đọc dễ hiểu Có cộng đồng hỗ trợ lớn, ngày càng phát triển Có thư viện package rất đa dạng Thủ tục, cú pháp đơn giản, dễ học |
| ĐIỂM YẾU | <ul style="list-style-type: none"> Vì là ngôn ngữ lập trình tĩnh nên tốc độ build sẽ chậm, cần nhiều bước khi thực hiện. Cú pháp, thủ tục phức tạp Chỉ được hỗ trợ trên NET. framework | <ul style="list-style-type: none"> Hiệu suất thấp hơn C# Cần nhiều thời gian tìm hiểu vì cú pháp khác so với C#, Java... |
| Tính đóng gói | Các phạm vi truy cập dữ liệu <ul style="list-style-type: none"> Public Protected Private Internal Protected Internal | Các phạm vi tương tự ở C# <ul style="list-style-type: none"> Public (không có ký hiệu) Protected (bắt đầu bằng “_”) Private (bắt đầu bằng “__”) |
| Tính trừu tượng | <phạm vi> abstract class <tên> Chứa các phương thức ảo virtual , phương thức trừu tượng abstract | From abc import ABC , abstractmethod Cần gọi @abstractmethod khi viết code Không có phương thức virtual |

| Tính kế thừa | <p><phạm vi> class <con>:cha</p> <p>Không hỗ trợ đa kế thừa, nhưng có thể dùng định nghĩa interface để triển khai</p> <p>Không kế thừa được các thuộc tính private</p> | <p>class <con>(class cha):</p> <p>Hỗ trợ đa kế thừa, tuy nhiên interface trong Python lại khá mơ hồ.</p> <p>Tương tự C#, Python cũng không kế thừa được các thuộc tính private</p> |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tính đa hình | <ul style="list-style-type: none"> • Đa hình tĩnh Overloading • Đa hình động Overriding • Nạp chồng toán tử • Nạp chồng phương thức | <ul style="list-style-type: none"> • Có thể dùng đa hình overriding nhưng không cần khai báo từ khóa • Chỉ có thể nạp chồng toán tử |

CHƯƠNG 3: CHƯƠNG TRÌNH PYTHON ỨNG DỤNG BÓN TRỤ CỘT CỦA LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG:

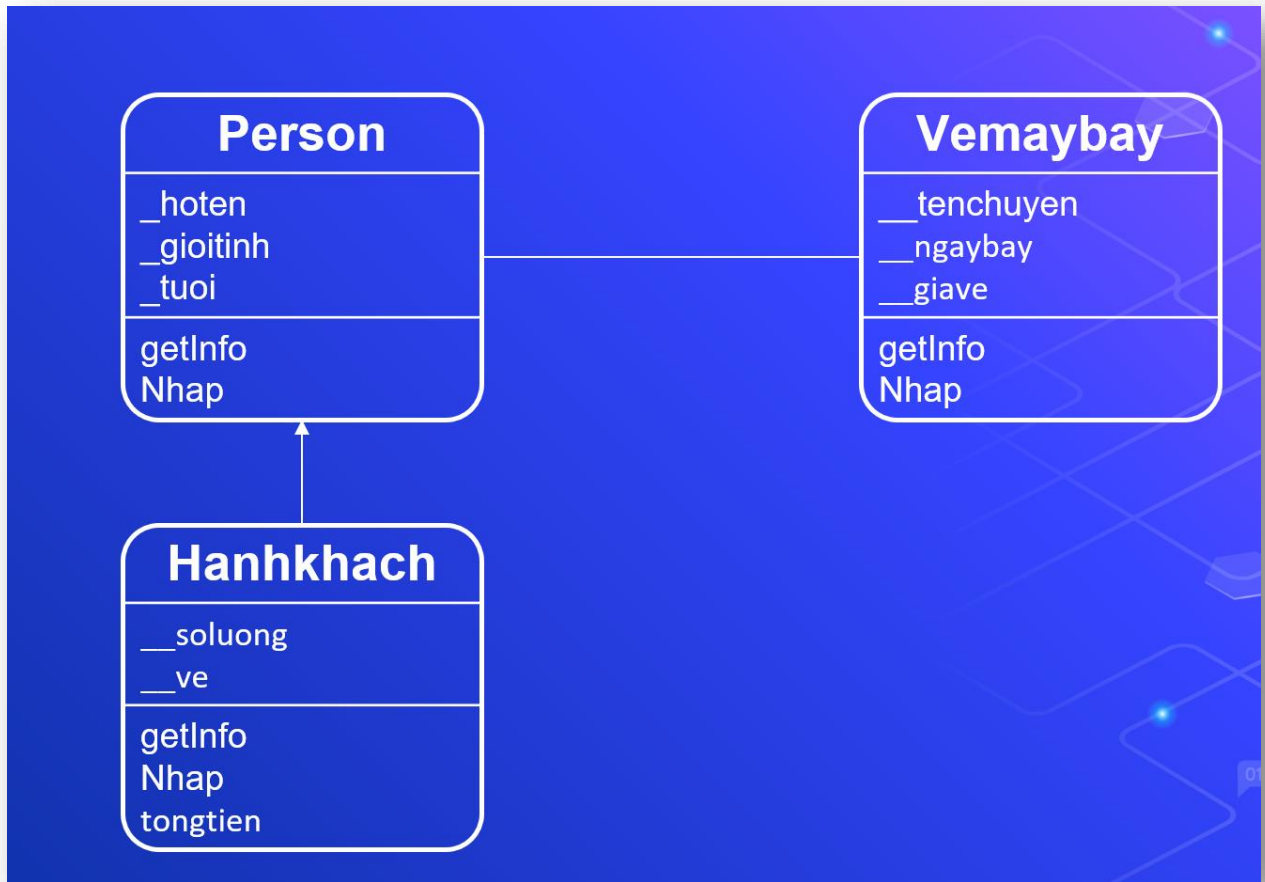
3.1. Giới thiệu sơ lược về chương trình:

Chương trình: “**Quản lý thông tin vé máy bay**” được ứng dụng bón trụ cột trong lập trình hướng đối tượng bằng ngôn ngữ Python. Chương trình này để quản lý thông tin (họ tên, giới tính, tuổi,...) của khách hàng khi mua vé. Để quản lý thông tin và thao tác một cách nhanh chóng. Ngoài ra chương trình còn thực hiện được lần lượt các chức năng:

0. Thoát chương trình
1. Khởi tạo danh sách khách hàng
2. Xuất thông tin khách hàng
3. Xuất tổng tiền vé phải trả cho từng khách hàng
4. Sắp xếp thông tin khách hàng theo thứ tự tổng tiền tăng dần
5. Sắp xếp thông tin khách hàng theo thứ tự tổng tiền giảm dần
6. Sửa thông tin khách hàng

7. Xóa thông tin khách hàng

3.2. Sơ đồ của chương trình:



3.3. Ý tưởng:

- Xác định vấn đề cần giải quyết: Xác định mục tiêu yêu cầu của đề tài (Tìm hiểu 4 trụ cột trong OOP. Viết chương trình minh họa cho các đặc điểm của OOP trong ngôn ngữ Python)
- Thiết kế giải pháp
 - + Ngôn ngữ: ngôn ngữ lập trình Python
 - + Xác định thuật toán cho chương trình: mỗi người sẽ có các thuộc tính họ tên giới tính tuổi...Mỗi hành khách sẽ kế thừa từ lớp người ngoài các thuộc tính như họ tên giới tính tuổi thì còn có các thuộc tính số lượng vé mua và mảng lưu trữ thông tin vé đã mua. Vé máy bay sẽ có các thuộc tính tên chuyến bay, ngày bay và giá vé.

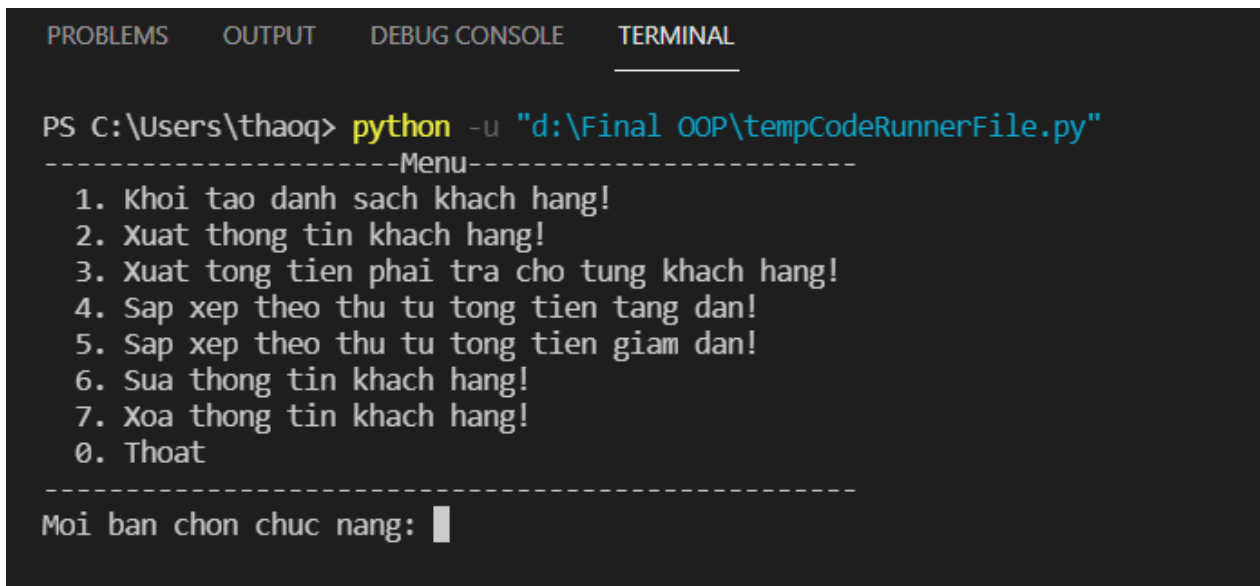
- Viết chương trình thực hiện theo giải pháp đã đề ra.
- Biên dịch chương trình (chuyển đổi mã lập trình về mã máy).
- Chạy chương trình, kiểm tra và sửa lỗi cho chương trình.

3.4. Cách cài đặt để chạy được chương trình:

- Python 3 hoặc Python 2
- Ứng dụng để chạy code Python như Visual Studio Code/ Sublime Text...

3.5. Hướng dẫn sử dụng chương trình:

1. Bật phần mềm Visual Studio Code/ Sublime Text...
2. Mở File “Final.py” trong thư mục “ Final OOP”
3. Chương trình chạy và người dùng có thể sử dụng thông qua lựa chọn trên Terminal
4. Chọn chức năng trên hộp Menu:



```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\thaoq> python -u "d:\Final OOP\tempCodeRunnerFile.py"
-----Menu-----
 1. Khởi tạo danh sách khách hàng!
 2. Xuất thông tin khách hàng!
 3. Xuất tổng tiền phải trả cho từng khách hàng!
 4. Sắp xếp theo thu tu tổng tiền tăng dần!
 5. Sắp xếp theo thu tu tổng tiền giảm dần!
 6. Sửa thông tin khách hàng!
 7. Xóa thông tin khách hàng!
 0. Thoát
-----
Mời bạn chọn chức năng: █
  
```

✓ **Nhập “1” để tạo danh sách khách hàng: Nhập lần lượt các thông tin sau**

- Số lượng khách hàng
- Họ và tên khách hàng
- Giới Tính
- Tuổi
- Nhập số lượng vé (một người có thể mua nhiều vé, nếu mua trên 3 vé thì sẽ được discount 10% tổng số tiền)
- Tên chuyến bay

- Ngày bay
- Giá vé
- ✓ **Nhập “2” để in các thông tin vừa nhập**
- ✓ **Nhập “3” để in tổng tiền của mỗi khách hàng. Nếu khách hàng mua trên 3 vé máy bay thì sẽ được discount 10%**
- ✓ **Nhập “4” để sắp xếp thứ tự khách hàng theo tổng tiền tăng dần** thì chương trình sẽ thông báo “ Sắp xếp thành công! Hãy quay lại chức năng 2 để kiểm tra”. Tiếp tục nhập “2” để xuất danh sách tho thứ tự
- ✓ **Nhập “5” để sắp xếp thứ tự khách hàng theo tổng tiền giảm dần** thì chương trình sẽ thông báo “ Sắp xếp thành công! Hãy quay lại chức năng 2 để kiểm tra”. Tiếp tục nhập “2” để xuất danh sách tho thứ tự
- ✓ **Khi muốn sửa thông tin khách hàng nào đó thì nhập “6”**. Thực hiện nhập lại các thông tin như ở chức năng 1. Và chọn chức năng “2” để xuất ra lần nữa
- ✓ **Để xóa thông tin khách hàng thì nhập “7”**. Và quay về chức năng “2” để in thông tin của khách hàng còn lại.

PHẦN 3: KẾT LUẬN

1. Những kết quả đạt được:

Qua 5 tuần tìm hiểu và báo cáo cuối kì, nhóm em không chỉ trau dồi về kiến thức, còn được học hỏi, được những lời góp ý của thầy, những câu hỏi khi báo cáo tiến độ thực hiện qua đó nhằm giúp nhóm em hoàn thiện hơn về bài cuối kỳ và có thể tạo nên một chương trình **“Quản lý thông tin vé máy bay”** sử dụng 4 trụ cột chính của OOP trên ngôn ngữ lập trình Python. Chương trình có thể thực hiện các chức năng sau:

1. Nhập thông tin khách hàng:

Nhập lần lượt: Họ tên, tuổi, số lượng vé mua, tên chuyến bay, ngày bay, giá trị của vé.

2. In thông tin vừa nhập ra màn hình

Xuất ra màn hình những thông tin vừa nhập. Nếu một người mua nhiều hơn hoặc bằng 2 vé máy bay thì chương trình in ra thông tin vé thứ nhất và thông tin vé thứ hai.

3. Tính tổng số tiền vé của mỗi khách hàng. Nếu mua trên 3 vé sẽ được giảm 10%.
4. Sắp xếp khách hàng theo tổng tiền thứ tự tăng dần.
5. Sắp xếp khách hàng theo tổng tiền thứ tự tăng dần.
6. Chỉnh sửa thông tin đã nhập
7. Xóa thông tin
8. Thoát khỏi chương trình (Bước này trong menu chương trình là 0)

Hiểu biết thêm được một ngôn ngữ lập trình mới – Python và ứng dụng vào lập trình hướng đối tượng

Kỹ năng thuyết trình, vấn đáp và kỹ năng làm việc nhóm

2. Đóng góp mới và đề xuất mới:

Ngoài những tính năng ở trên ra thì muốn thêm một số tính năng nữa như đọc thông tin khách hàng mua vé từ file có sẵn và có thể thao tác trên file thì sẽ tiện với người dùng chương trình. Tiết kiệm được thời gian nhập thủ công.

Tạo ra một giao diện đẹp bắt mắt người dùng và dễ sử dụng

3. Ưu nhược điểm và hướng phát triển của đề tài:

❖ Về ưu điểm:

- ✓ Menu dễ dàng sử dụng, không quá phức tạp.
- ✓ Gần như đầy đủ các chức năng.
- ✓ Các chức năng thực thi nhanh chóng.
- ✓ Độ chính xác tương đối cao.

❖ Về khuyết điểm và hướng phát triển:

- ✓ Chưa hoàn thành ứng dụng hiện hữu.
- ✓ Chưa có giao diện đẹp.
- ✓ Chưa hoàn thiện đầy đủ các chức năng như: đọc File, ghi File và có thể thao tác trên File để người sử dụng có thể tiết kiệm được thời gian nhập dữ liệu.

4. Kết luận:

Lời đầu tiên, nhóm em xin chân thành cảm ơn thầy. Trong suốt 15 tuần vừa qua đã dạy chúng em, biết nhiều kiến thức và trau dồi cho chúng em, không chỉ dừng lại ở đó thầy còn ôn tập kiến thức qua những bài tập vào một số tuần.

Sau khi kết thúc môn học và hoàn thành dự án cuối kỳ với chương trình “ **Quản lý thông tin vé máy bay**” được tạo nên bởi 4 trụ cột chính trong OOP bằng ngôn ngữ Python. Và thực hiện những chức năng như trên. Điều quan trọng nhất là nhóm em có thể tự học thêm một ngôn ngữ lập trình mới, mặc dù thời gian không quá dài để có thể hiểu quá sâu. Tuy nhiên nhờ những lời góp ý và những câu hỏi của thầy đặt ra nhóm em kịp sửa chữa lại những sai lầm khi không tìm hiểu trên những trang web sai nội dung và những lần nhóm em hiểu sai vấn đề. Nếu trong chương trình có gì thiếu sót. Xin thầy bỏ qua.

TÀI LIỆU THAM KHẢO

1. Video clip những bài giảng của GV trên fhqx.
 - Những định nghĩa tham khảo trên:
2. <https://viblo.asia/p/4-nguyen-ly-co-ban-trong-oop-va-vi-du-de-hieu-bang-python-jvElaaoolkw>
3. <https://topdev.vn/blog/oop-la-gi/#cac-nguyen-ly-co-ban-cua-oop>
4. <https://www.youtube.com/watch?v=D23kOfSmfwA>

PHỤ LỤC

1. Kỹ thuật lập trình: là môn cơ sở ngành không chỉ của các ngành công nghệ thông tin mà một số ngành khác như tự động hóa, toán - tin,... cũng phải học môn này. Về cơ bản kỹ thuật lập trình nhắc lại những cấu trúc dữ liệu cơ bản mà trong học phần cấu trúc dữ liệu đã học, học về các thuật toán cơ bản, các tối ưu mã nguồn, kỹ thuật viết mã nguồn....
2. Internet Of Things hay vạn vật kết nối internet: đề cập đến hàng tỷ thiết bị vật lý trên khắp thế giới hiện được kết nối với internet, thu thập và chia sẻ dữ liệu. Nhờ bộ xử lý giá rẻ và mạng không dây, có thể biến mọi thứ, từ viên thuốc sang máy bay, thành một phần của IoT. Điều này bổ sung sự “thông minh kỹ thuật số” cho các thiết bị, cho phép chúng giao tiếp mà không cần có con người tham gia và hợp nhất thế giới kỹ thuật số và vật lý.
3. Trừu tượng hóa: là quá trình đơn giản hoá một thứ cho đến khi không thể đơn giản hoá được nữa, mà không làm mất đi yếu tính của nó.

=====**HẾT**=====