**I) Mục tiêu** :

Quản lí chi tiêu cá nhân (Income, Outcome, Saving) theo tài khoản ngân hàng

**II) Tính chất OOP :**

**II.1 ) Các Class:**

*1) Abstract Class Transaction :*

- Thuộc tính (private) : ID, Name, Budget, Description, Status, CreatedDate, UpdatedDate

- Phương thức : Set/Get ; AddToFile; ...

*2) Class Income*

- Thừa kế Class Transaction

- Override lại các phương thức

*3) Class Outcome*

- Thừa kế Class Transaction

- Override lại các phương thức

- Thêm thuộc tính ExpectedBudget, ExpectedDate

- Thêm các phương thức xử lí ExpectedBudget, ExpectedDate

*4) Class Saving*

- Thừa kế Class Transaction

- Override lại các phương thức

**II.2) Các tính chất OOP được áp dụng:**

*1) Tính Đóng Gói* : Các thuộc tính là Private, có các phương thức (public) để xử lí giá trị thuộc tính (phương thức Set/Get)

*2) Tính Kế Thừa* : Class Income, OutCome, Saving thừa kế Class Transaction

*3) Tính Đa Hình* : Class Income, OutCome, Saving Override các phương thức Class Transaction

*4) Tính Trừu Tượng* : dùng Abstract Class Transaction

**III) Mô tả chi tiết các lớp :**

**III.1) Abstract Class Transaction :**

1) Thuộc tính (private) : ID, Name, Budget, Description, Status, CreatedDate, UpdatedDate

- (int) ID : Mã định danh (Unique)

- (string) Name : Tên/Loại giao dịch (Unique)

- (float) Budget : Tổng số tiền trong 1 loại giao dịch

- (string) Description : Mô tả, dùng để đưa lời khuyên/ lời nhắc nhở chi tiêu

- (int) Status (1: tồn tại | 0: Bị xóa) : Trạng thái Giao Dịch

- (time\_t) CreatedDate : Thời điểm tạo giao dịch

- (time\_t) UpdatedDate : Thời điểm sửa đổi giao dịch gần nhất

2) Phương thức :

a) Hàm set/get các thuộc tính

b) void AddToFile ()

- Thêm loại Income/Outcome/Saving mới (this) vào file

- Nếu có thay đổi về Budget sẽ xử lí giá trị Ví Tổng

c) void UpdateFile(int id)

- Sửa đổi thông tin của 1 loại Income/Outcome/Saving có sẵn với ID==id thành thông tin mới (this)

- Nếu có thay đổi về Budget sẽ xử lí giá trị Ví Tổng

d) void Delete(int id)

- Đổi record có ID==id thành Status = 0; Budget = 0

- xử lí giá trị Ví Tổng

e) void Show()

- Hiển thị dữ liệu trong file tương ứng

f) void Statistics()

- Show()

- Hiển thị Max, Min budget

- Hiển thị Lời nhắc/ lời khuyên tài chính từ Description

**III.2) Class Income**

1) Thuộc tính Thừa kế Class Transaction

2) Phương Thức

- Override lại các phương thức

(Không được Đổi tên/ Xóa record "WithdrawSaving")

3) Thông tin thêm:

- File Path : "Data\\Income.txt"

- Cấu trúc file :

+) Mỗi dòng là 1 record/ Loại Income

+) Mỗi cột là thuộc tính của Income được phân cách bởi dấu "|", có thứ tự :

ID | Name | Budget | Status | Description | CreatedDate | UpdatedDate

**III.3) Class Outcome**

1) Thuốc tính:

- Thừa kế Class Transaction

- (float) ExpectedBudget : Dự định ngân sách để chi cho loại Outcome này

- (time\_t) ExpectedDate : Thời gian cho xét ExpectedBudget là từ CreatedDate tới ExpectedDate. Sau ExpectedDate sẽ reset 2 thuốc tính này bằng NULL

2) Phương Thức

- Override lại các phương thức

- Set/Get ExpectedBudget, ExpectedDate (ExpectedDate > CreatedDate)

- bool IsOutOfExpectedDate (int id ) : Trả về TRUE khi Outcome (ID==id) có ExpectedDate<Now; ngược lại trả về FALSE

- bool IsOutOfExpectedBudget (int id ) : Trả về TRUE khi Outcome (ID==id) có ExpectedBudget<(tổng Budget từ CreatedDate tới ExpectedDate); ngược lại trả về FALSE

3) Thông tin thêm:

- File Path : "Data\\Outcome.txt"

- Cấu trúc file :

+) Mỗi dòng là 1 record/ Loại Outcome

+) Mỗi cột là thuộc tính của Outcome được phân cách bởi dấu "|", có thứ tự :

ID | Name | Budget | ExpectedBudget | ExpectedDate | Status | Description | CreatedDate | UpdatedDate

**III.4) Class Saving**

1) Thuộc tính

- Thừa kế Class Transaction

- (float) ExpectedBudget

2) Phương thức

- Override lại các phương thức :

(trong phương thức AddToFile/Update xét nếu :

+) Budget<ExpectedBudget : set Description "Keep saving!"

+) Budget==ExpectedBudget : Không cho add thêm tiền vào và set Description "Reached Goal!")

- Set/Get ExpectedBudget

3) Thông tin thêm:

- File Path : "Data\\Saving.txt"

- Cấu trúc file :

+) Mỗi dòng là 1 record/ Loại Saving

+) Mỗi cột là thuộc tính của Saving được phân cách bởi dấu "|", có thứ tự :

ID | Name | Budget | ExpectedBudget | Status | Description | CreatedDate | UpdatedDate

**IV) Mô tả đề tài :**

Một người muốn ghi chép lại chi tiêu của mình dựa trên 1 tài khoản ngân hàng nhất định, từ đó, thống kê và nhận lời nhắc chi tiêu hợp lí.

Có Ví tổng là tổng tiên hiên tại đang có

Thực hiện giao dịch :

- Thêm tiền vào tài khoản

- Rút tiền khỏi tài khoản

- Thiết lập mục tiêu tài chính ( cài đặt Mỗi loại Nguồn chi có MAX ngân sách sử dụng)

- Có quỹ tiết kiệm

**V) Luồng chạy :**

1) Chạy chương trình (hàm main)

- Xét File Setting có hay không :

+) Nếu KHÔNG : Đây là lần chạy đầu tiên. Tạo File Setting, cho nhập và lưu UserName, Description; Ví Tổng = 0. tạo file Income, thêm record "WithdrawSaving". Load Màn hình Home

+) Nếu CÓ : Tạo mảng DS kiểu Transaction để lưu các hoạt động giao dịch đã làm trong lần chạy app hiện tại. Load Màn hình Home

2) Màn hình Home :

- Hiển thị UserName, Ví Tổng , Description

- Menu tính năng :

+) Thêm loại giao dịch Thu / Chi : AddToFile()

+) Thêm Ngân sách Thu/Chi : Update()

+) Sửa đổi thông tin Thu/Chi : Update()

+) Xóa : Delete()

+) Thiết lập Mục tiêu tài chính : Update()

+) Xóa Thiết lập Mục tiêu tài chính : Update()

+) Tạo Quỹ Tiết kiệm : AddToFile()

+) Chuyển tiền vào Quỹ Tiết kiệm : Update()

+) Xóa Quỹ tiết kiệm : Delete()

+) Rút tiền khỏi Quỹ Tiết Kiệm : Update()

+) Thống kê Thu-> Statistics()

+) Thống kê Chi-> Statistics()

+) Thống kê Saving-> Statistics()

+) Hiển thị các hoạt động giao dịch đã làm trong lần chạy app hiện tại -> Show()

+) Thoát chương trình