Fpt University

FPT POLYTECHNIC

Bài 2:
LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG (OOP)

Nhắc lại bài cũ

- Tìm hiểu khái niệm cấu trúc dữ liệu
  - Dữ liệu, Cấu trúc dữ liệu
  - Các kiểu cấu trúc dữ liệu
- Tìm hiểu khái niệm giải thuật (thuật toán, thuật giải)
  - Khái niệm về giải thuật
  - Biểu diễn giải thuật
  - Độ phức tạp của giải thuật
- Mối liên hệ giữa cấu trúc dữ liệu và giải thuật

FPT POLYTECHNIC

Mục tiêu bài học hôm nay

- Nhắc lại khái niệm lập trình hướng thủ tục (Procedural Programming)
- Tìm hiểu các khái niệm và ý nghĩa Lập trình hướng đối tượng (OOP)
- Lập trình hướng đối tượng trong VB.Net

Tiếp cận vấn đề

- Có thể so sánh kiểu dữ liệu cơ sở so với kiểu dữ liệu có cấu trúc giống như lập trình hướng thủ tục với lập trình hướng đối tượng
- Lập trình hướng thủ tục thường thao tác với dữ liệu phi cấu trúc (biến kiểu dữ liệu cơ sở), trong khi đó Lập trình hướng đối tượng thường thao tác với dữ liệu có cấu trúc đối tượng.
- Cấu trúc dữ liệu đối tượng là loại cấu trúc được sử dụng phổ biến nhất trong lập trình hiện nay.

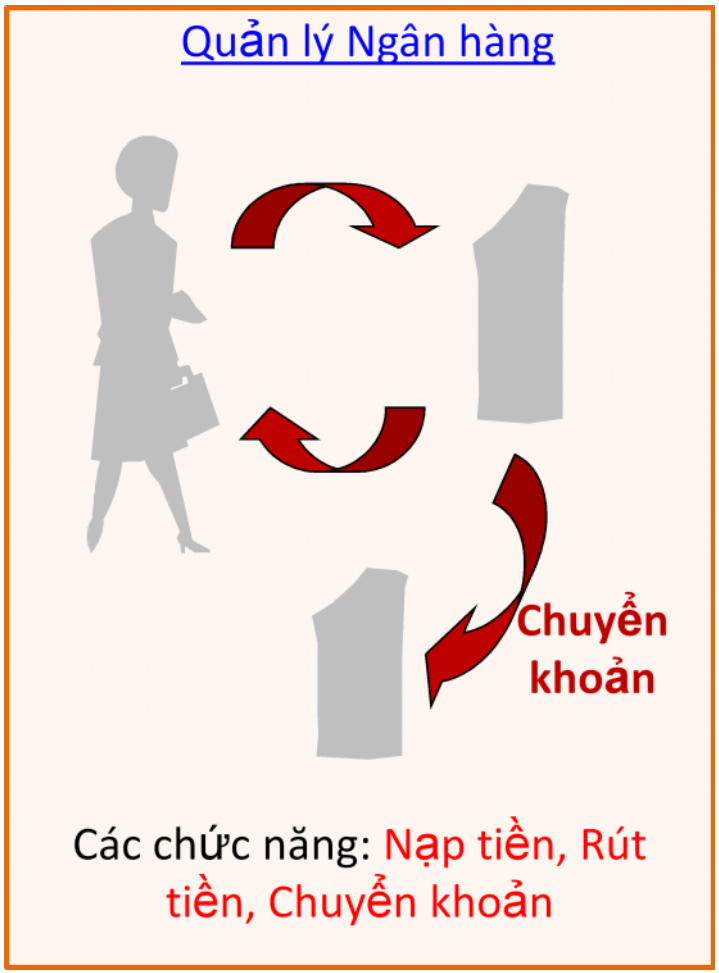
Bài toán ví dụ

Bài toán quản lý giao dịch của Khách hàng với Ngân Hàng:

- Khách hàng có các giao dịch với ngân hàng:
  - Nạp tiền vào tài khoản
  - Rút tiền từ tài khoản
  - Chuyển khoản giữa các tài khoản

Lập trình hướng thủ tục

- Quan tâm đến chức năng (thủ tục) cần thực hiện.
- Chương trình lớn được chia thành các chức năng (thủ tục) nhỏ hơn hoạt động độc lập với nhau.
- Phần lớn các chức năng sử dụng dữ liệu chung.



Lập trình hướng thủ tục

- Các vấn đề gặp phải của lập trình hướng thủ tục với các hệ thống phức tạp:
  - Vấn đề quản lí quá nhiều chức năng
  - Vấn đề về quản lí dữ liệu phức tạp
  - Vấn đề mở rộng chức năng và sử dụng lại module đã viết

-> Phương pháp lập trình hướng đối tượng

Lập trình hướng đối tượng

- Chương trình được chia thành các Đối tượng (Object).
- Mỗi đối tượng chịu trách nhiệm quản lí riêng dữ liệu và các chức năng của nó.
- Các đối tượng tác động và trao đổi thông tin với nhau qua các phương thức (chức năng).

Các khái niệm trong lập trình OOP

- Đối tượng (Object)
- Lớp (Class)
- Thuộc tính (field, attribute)
- Phương thức (Method)

Đối tượng (Object)

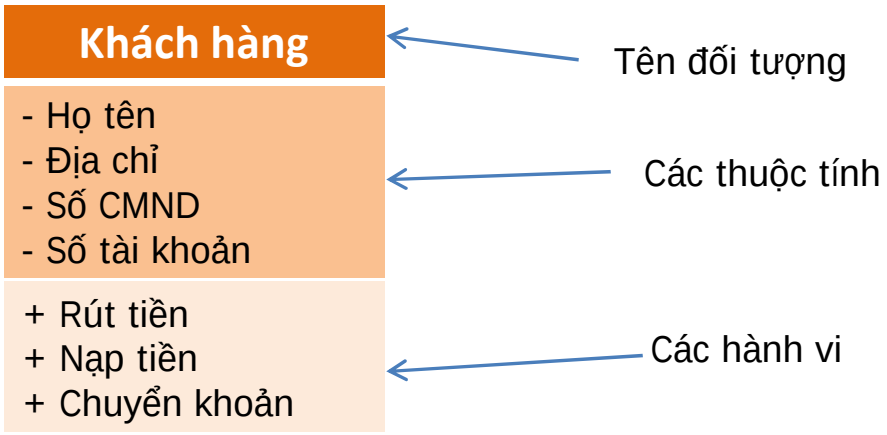
- Biểu diễn 1 đối tượng trong thế giới thực
- Mỗi đối tượng được đặc trưng bởi các thuộc tính và các hành vi riêng của nó
- Ví dụ: Đối tượng Khách hàng

	Thuộc tính	Hành vi
Khách hàng	▪ Họ tên ▪ Địa chỉ ▪ Số CMND ▪ Số tài khoản	▪ Rút tiền từ tài khoản ▪ Nạp tiền vào tài khoản ▪ Yêu cầu chuyển khoản

Đối tượng (Object)

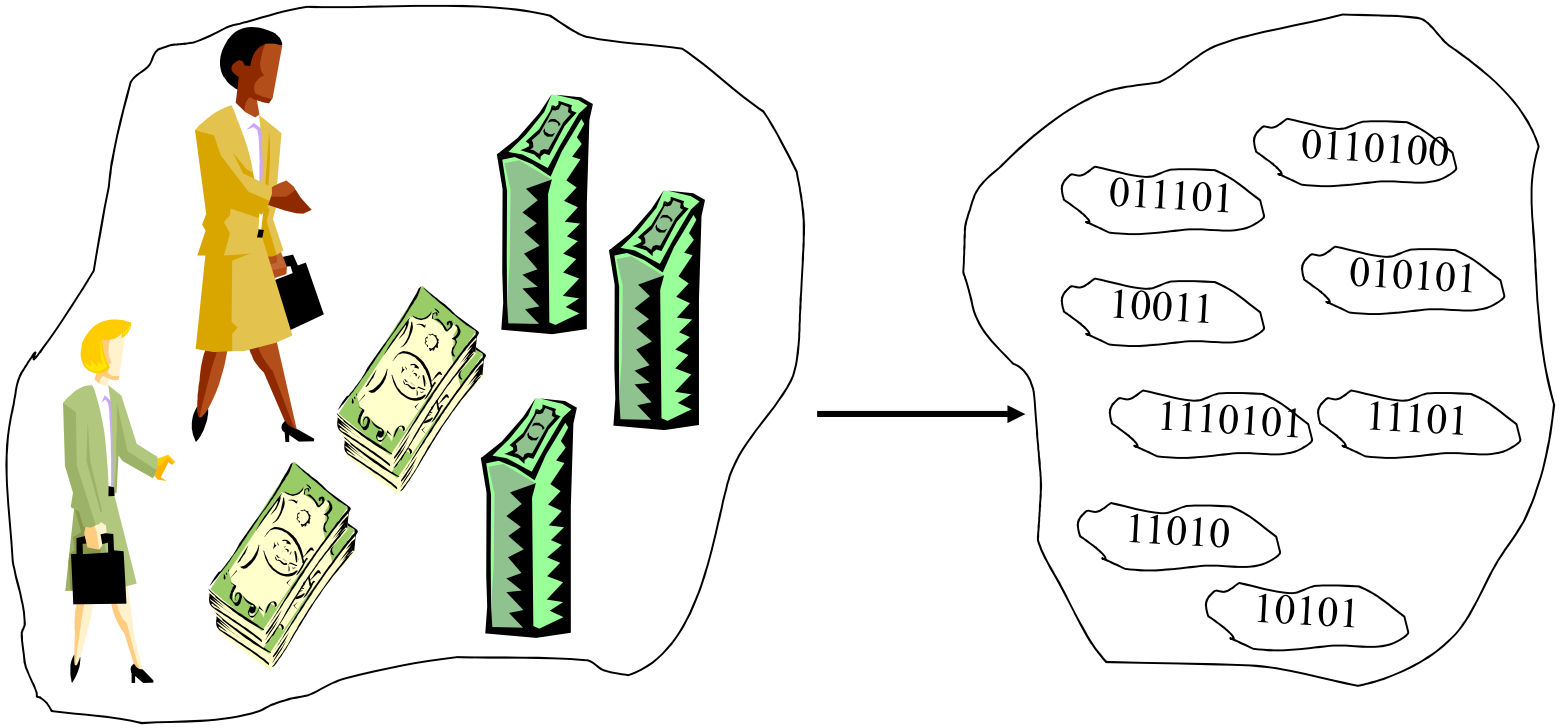
- Đối tượng trong thế giới thực và Đối tượng trong lập trình:

Object trong thế giới thực	Object trong lập trình
Các thuộc tính	Các trường (field)
Các hành vi	Các phương thức (method)



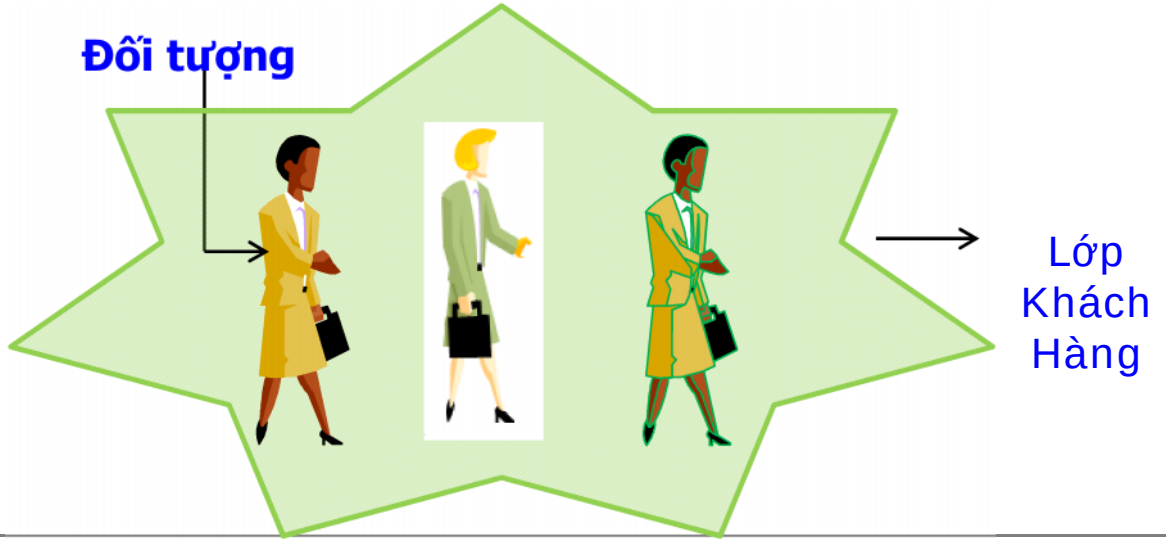
Đối tượng (Object)

- Trong OOP, cần ánh xạ các Đối tượng trong thực tế thành các Đối tượng trong lập trình



Lớp (Class)

- Class là một khuôn mẫu (Prototype) định nghĩa các thuộc tính và các phương thức chung cho tất cả các đối tượng cùng loại.
- Mỗi đối tượng là một thể hiện của một Lớp (hay nói nôm na, đối tượng là sản phẩm được tạo ra từ khuôn là lớp).
- Ví dụ: Lớp các đối tượng Khách hàng



Thuộc tính và phương thức

- Thuộc tính (Property/Attribute) là dữ liệu trình bày đặc trưng các đặc điểm về một Đối tượng.
- Phương thức (Method) tác động lên dữ liệu để thực hiện hành vi của Đối tượng.

Ví dụ OOP trong VB.NET

Định nghĩa của lớp Khách hàng

```
Public Class KhachHang
  'khai báo các thuộc tính
  Private mHoten As String
  Private mDiachi As String
  Private mCMND As String
  Private mSoTK As String

  'khai báo các phương thức
  Public Sub RutTien(ByVal sotien As Integer)
    'mã cài đặt phương thức ở đây
  End Sub
  Public Sub NapTien(ByVal sotien As Integer)
    'mã cài đặt phương thức ở đây
  End Sub
  Public Sub ChuyenKhoan(ByVal nguoinhan As KhachHang, ByVal sotien As
   Integer)
    'mã cài đặt phương thức ở đây
  End Sub
End Class
```

Khai báo Lớp

- Cú pháp khai báo Class:

[Điều khiển truy xuất] [Từ khóa] Class Tên lớp [Implements TenGiaoDien]

‘Khai báo các thuộc tính và phương thức

‘ Thực hiện các phương thức...

End Class

Khai báo Lớp

- Điều khiển truy xuất: định nghĩa khả năng truy cập của class, sử dụng một trong các từ khóa : Public, Private, Protected, Friend,Protected Friend
- Từ khóa: chỉ rõ các lớp có được thừa kế hay không, từ khóa Inherits, NotInheritable hoặc MustInherit.
- Class: đánh dấu bắt đầu một Class
- Tên lớp tên của một Class
- Implements: chỉ rõ class thực thi trên giao diện nào
- Tên Giao Diện: miêu tả tên giao diện.Một class có thể thực thi trên một hoặc nhiều giao diện.
- End Class: đánh dấu kết thúc khai báo của một class

FPT POLYTECHNIC

Bảng các điều khiển truy xuất

Access Modifier	Dùng trong	Mô tả
Public	module, class, structure	Được truy cập từ cùng project, từ project khác hoặc từ thành phần khác
Private	module, class, structure	Chỉ được truy cập trong cùng module, class , structure
Protected	Classes, class member	Được truy cập trong cùng class , hoặc class được kế thừa
Friend	module, class, structure	Truy cập được trong cùng project
Protected Friend	Classes, class member	Truy cập được trong cùng project Và từ các class được kế thừa

Ví dụ khai báo một Class

- Ví dụ khai báo lớp tên là Person

```
Public Class Person
    'Các thành phần khác của class được khai
báo ở đây
End Class
```

Các thành phần của một Class

- Các thành phần chủ yếu của một Class bao gồm:
  - Biến thành viên (Field)
  - Thuộc tính (Property)
  - Phương thức khởi tạo (Constructor)
  - Phương thức (Method)

Biến thành viên

- Biến thành viên (field) là một đặc tính bên trong của lớp.
- Ví dụ khai báo biến thành viên như sau:

```
Public Class Person
    ’ Khai báo 2 biến thành viên
    Private mName As String
    Private mAge As Integer
End Class
```

Thuộc tính

- Thuộc tính (property): cho phép định nghĩa chi tiết thao tác truy cập các biến thành viên
- Ví dụ khai báo thuộc tính như sau:

```
‘ Thuộc tính cho phép đọc và gán
Public Property Name() As String
          Get
         Return mName
          End Get
    Set(ByVal Value As String)
         mName = Value
    End Set
End Property
```

Phương thức khởi tạo (Constructor)

- Được dùng để khởi tạo đối tượng
- Trong VB.Net, thủ tục New chính là phương thức khởi tạo
- Ví dụ:

```
Public Class Person
    ' Khai báo 2 biến thành viên
    Private mName As String
    Private mAge As Integer
    ' Thủ tục khởi tạo
    Public Sub New(ByVal name As String, ByVal age As
Integer)
        Me.mName = name
        Me.mAge = age
    End Sub
End Class
```

Phương thức (Method)

- Có hai kiểu phương thức:
  - Không trả về giá trị
  - Trả về giá trị
- Ví dụ một phương thức không trả về giá trị

```
Public Sub Reset()
    mName = ""
    mAge = 0
End Function
```

- Ví dụ một phương thức trả về giá trị

```
Public Function Age() As Integer
    Return mAge
End Function
```

Tạo Đối tượng

- Khi có định nghĩa Lớp, có thể tạo các Đối tượng từ Lớp (thông qua phương thức khởi tạo)

```
'Định nghĩa đối tượng 1
Dim nguyen_nam_anh As New Person("Nguyễn Nam Anh", 18)

'Định nghĩa đối tượng 2
Dim obj2 As New Person("Lương Gia Thanh", 20)
```

Thao tác với Đối tượng

- Sau khi tạo đối tượng, có thể thực hiện các hành động mà đối tượng cung cấp

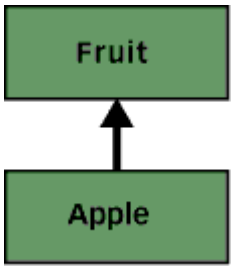
```
‘Định nghĩa đối tượng 1
Dim nguyen_nam_anh As New Person(“Nguyễn Nam Anh”, 18)
‘Định nghĩa đối tượng 2
Dim obj2 As New Person(“Lương Gia Thanh”, 20)

‘Gọi phương thức
Console.Write(obj2.Age());
```

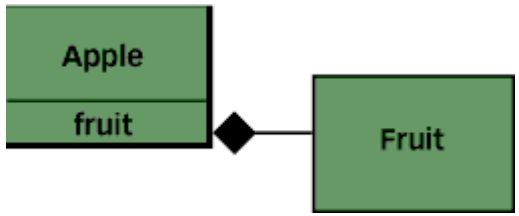
Mở rộng Lớp

- Trong tình huống cần mở rộng chức năng của một Lớp (hay nói cách khác là thêm hành vi cho đối tượng), có một số lựa chọn sau:
  - Nếu có mã nguồn Lớp và muốn thay đổi trực tiếp trên lớp gốc

    -> thay đổi trực tiếp vào mã nguồn lớp gốc
  - Nếu không muốn thay đổi Lớp gốc (hoặc không có mã nguồn), có thể sử dụng hai cách:
    1. Tạo lớp Kế thừa (Inheritance)

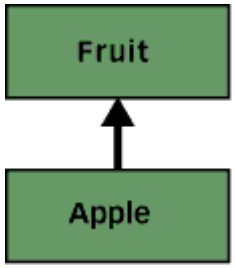


    2. Tạo lớp Tụ hợp (Composition)



Mở rộng Lớp bằng Kế thừa

- Kế thừa là một khả năng quan trọng của OOP
- Một Lớp con (sub-class) có thể được định nghĩa bằng cách thừa kế toàn bộ các đặc tính và hành vi của một lớp cha (super-class).



- Ví dụ trên, Apple cũng là một dạng trái cây (fruit). Do đó để tạo ra lớp Apple, thay vì định nghĩa lại từ đầu, có thể kế thừa từ lớp Fruit và bổ sung thêm các đặc tính và hành vi của quả táo.

Ví dụ Kế thừa

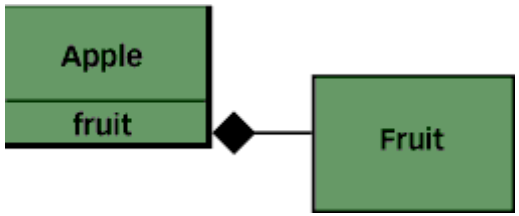
- Ví dụ mã định nghĩa lớp Apple kế thừa từ lớp Fruit

```
Public Class Fruit
'Cài đặt chi tiết cho lớp Fruit
End Class

Public Class Apple Inherits Fruit
'Cài đặt thêm các đặc tính và phương
thức riêng của Apple
End Class
```

Mở rộng Lớp bằng Tụ hợp

- Trong thực tế, tụ hợp được sử dụng nhiều hơn kế thừa bởi tính linh hoạt của nó.
- Lớp mới được tạo ra bằng cách dùng lớp đã có sẵn như là một phần của mình, đồng thời bổ sung thêm các đặc tính và phương thức riêng.



- Ví dụ trên, Apple được tạo ra bằng cách dùng các thuộc tính và phương thức của lớp Fruit (thông qua một đối tượng fruit), đồng thời bổ sung thêm các thành phần của mình.

Ví dụ Tụ hợp

- Ví dụ mã định nghĩa lớp Apple kế thừa từ lớp Fruit

```
Public Class Fruit
'Cài đặt chi tiết cho lớp Fruit
End Class

Public Class Apple
Private fruit As Fruit
'Cài đặt thêm các đặc tính và phương
thức riêng của Apple
End Class
```

OOP và Cấu trúc dữ liệu+giải thuật

- Hiểu rõ về các khái niệm cơ bản của OOP: lớp, đối tượng, thuộc tính, phương thức, quyền truy cập, phương thức khởi tạo, kế thừa, tụ hợp,... sẽ giúp việc tìm hiểu, cài đặt mới hoặc sử dụng các cấu trúc dữ liệu sẵn có (ví dụ trong VB.NET) thuận lợi hơn rất nhiều.

Tổng kết

- Lập trình OOP xử lý với dữ liệu có cấu trúc dạng đối tượng
- Các khái niệm quan trọng trong lập trình OOP:
  - Đối tượng (Object)
  - Lớp (Class)
  - Phương thức
  - Thuộc tính
- Đối tượng được tạo ra thông qua phương thức khởi tạo (constructor)

Tổng kết

- Lớp (Class) là khuôn, Đối tượng (Object) là sản phẩm được tạo ra từ khuôn đó -> các đối tượng được tạo ra từ cùng một khuôn class sẽ giống nhau về đặc tính và hành vi
- Kế thừa (Inheritance) là khả năng định nghĩa lớp con mở rộng từ một lớp cha
- Tụ hợp (Composition) là khả năng tập hợp nhiều lớp khác và các thành phần bổ sung để tạo thành Lớp mới