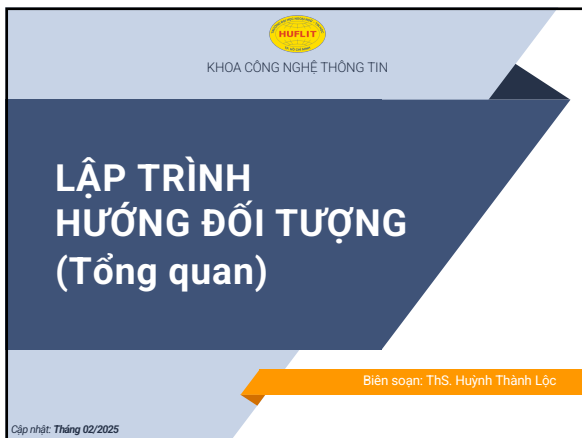
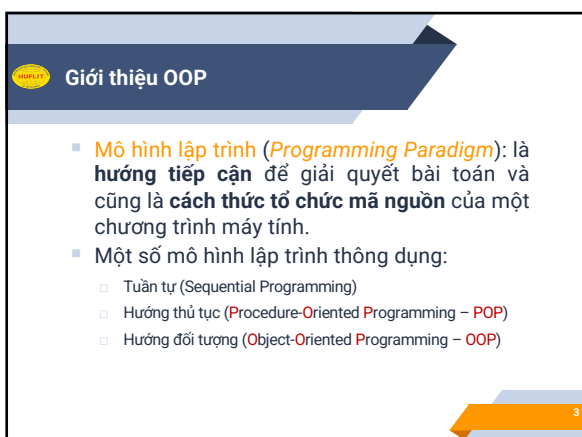




1



2



3

Giới thiệu OOP

💡 **Xét bài toán:** Cho **chiều rộng** và **chiều cao** của hình chữ nhật.

Tính và in ra màn hình **chu vi** và **diện tích** của hình chữ nhật đó.

4

4

Giới thiệu OOP

💡 **Hướng tiếp cận 1:** dùng các biến rời rạc để lưu dữ liệu và viết một **dãy các câu lệnh tuần tự** để giải bài toán

```
static void Main(string[] args)
{
    double w = int.Parse(Console.ReadLine());
    double h = int.Parse(Console.ReadLine());

    double perimeter = (w + h) * 2;
    double area = w * h;

    Console.WriteLine($"Perimeter : {perimeter}");
    Console.WriteLine($"Area : {area}");
}
```

Sequential Programming

5

5

Giới thiệu OOP

💡 **Hướng tiếp cận 2:** định nghĩa các **hàm chức năng** và sử dụng chúng để giải bài toán

```
static double RectanglePerimeter(double width, double height){
    return (width + height) * 2;
}

static double RectangleArea(double width, double height){
    return width * height;
}

static void Main(string[] args){
    double w = int.Parse(Console.ReadLine());
    double h = int.Parse(Console.ReadLine());
    double perimeter = RectanglePerimeter(w, h);
    double area = RectangleArea(w, h);
    Console.WriteLine($"Perimeter : {perimeter}");
    Console.WriteLine($"Area : {area}");
}
```

Procedure-Oriented Programming

6

6

Giới thiệu OOP

💡 **Hướng tiếp cận 3:** tạo các **đối tượng** hình chữ nhật (chứa dữ liệu chiều dài và chiều rộng) và thực hiện các chức năng của đối tượng.

```
//Xây dựng lớp đối tượng Rectangle???
static void Main(string[] args)
{
    double w = int.Parse(Console.ReadLine());
    double h = int.Parse(Console.ReadLine());
    Rectangle rect1 = new Rectangle(w, h);
    double perimeter = rect1.GetPerimeter();
    double area = rect1.GetArea();
    Console.WriteLine($"Perimeter : {perimeter}");
    Console.WriteLine($"Area : {area}");
}
```

Object-Oriented Programming

7

7

Giới thiệu OOP

POP

- Hướng tiếp cận: **xác định các chức năng** cần có
- Mỗi chức năng được định nghĩa thành các thủ tục/hàm
- Giải quyết bài toán bằng cách gọi các hàm

Procedure-Oriented Programming

OOP

- Hướng tiếp cận: **xác định các đối tượng** xuất hiện trong bài toán
- Mỗi loại đối tượng được định nghĩa thành một lớp
- Giải quyết bằng cách tạo các đối tượng và thực hiện các phương thức của đối tượng

Object Oriented Programming

8

8

Giới thiệu OOP

- Lập trình hướng đối tượng (Object-Oriented Programming):** là mô hình lập trình mà **bài toán được phân chia thành các đối tượng**. Mỗi đối tượng sẽ **có dữ liệu riêng** và có khả năng **thực hiện các hành động** để tạo nên kết quả cho bài toán.

9

9


Giới thiệu OOP

 **Bài toán: Quản lý sinh viên**

Viết một chương trình quản lý sinh viên với các tính năng:

- Lưu trữ thông tin một SV: *mã số SV, tên, năm sinh, giới tính, lớp*
- Nhập/xuất thông tin của một SV, tính tuổi của một SV, chuyển lớp cho một SV


10

10


Giới thiệu OOP

 Thay vì tập trung vào các chức năng, hãy phân tích:

- Trong bài toán xuất hiện các **đối tượng** nào?
- Cần lưu trữ **thông tin** gì trong mỗi đối tượng?
- Những **hành động** mà đối tượng cần thực hiện?
- **Mối quan hệ** giữa các đối tượng?



11

11


Giới thiệu OOP

 Bài toán có các đối tượng “sinh viên”:



12

12

Giới thiệu OOP

💡 Trong mỗi đối tượng “sinh viên” cần lưu thông tin:

- Mã số SV
- Tên
- Năm sinh
- Giới tính
- Lớp

13

13

Giới thiệu OOP

💡 Mỗi đối tượng “sinh viên” đều có khả năng thực hiện những hành động:

- Nhập thông tin()
- Xuất thông tin()
- Tính tuổi()
- Thay đổi lớp()

14

14

Giới thiệu OOP

💬 Nhận xét:

- 💡 Lập trình hướng thủ tục giải quyết trực tiếp bài toán bằng cách gọi các hàm
- 💡 Lập trình hướng đối tượng giải quyết bài toán gián tiếp thông qua việc yêu cầu các đối tượng thực hiện các hành động

15

15

Các khái niệm cơ bản

- **Đối tượng (Object):** là đơn vị cơ bản nhất trong OOP. Một đối tượng bao gồm:
 - **Thuộc tính (Attribute/Field):** những thông tin lưu trữ trong đối tượng
 - **Phương thức (Method):** những hành động mà đối tượng có thể thực hiện



The diagram shows an orange circle labeled 'Object' with a bracket pointing to two blue rectangular boxes labeled 'Attributes' and 'Methods'.

16

16

Các khái niệm cơ bản

- **Lớp (Class):** là kiểu dữ liệu do người dùng tự định nghĩa, mô tả bản thiết kế mẫu của các đối tượng cùng loại
 - Nội dung của lớp bao gồm các thuộc tính và phương thức
 - Từ định nghĩa lớp, chúng ta có thể tạo ra các đối tượng có cấu trúc chung nhưng thuộc tính của các đối tượng có thể khác nhau

17

17

Các khái niệm cơ bản

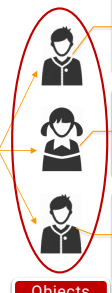
Class Student

Attributes:

- studentId: string
- name: string
- birthYear: int
- gender: bool
- stdClass: string

Methods:

- Input(): void
- PrintInfo(): void
- GetAge(): int
- SetClass(string newStdClass): void



The diagram shows three student icons in a red oval labeled 'Objects'. Arrows point from each icon to a box containing their specific attribute values.

studentId: 21DH110123
name: Nguyễn Ân
birthYear: 2003
gender: false
stdClass: TH2101

studentId: 21DH110456
name: Trần Thị Hoa
birthYear: 2002
gender: true
stdClass: TH2102

studentId: 21DH110789
name: Võ Cường
birthYear: 2001
gender: false
stdClass: TH2101

18

18

Các khái niệm cơ bản

Mỗi **lớp** được biểu diễn bằng một **khối hình chữ nhật** gồm 3 thành phần:

Student

- studentId: string
- name: string
- birthYear: int
- gender: bool
- stdClass: string

- Input(): void
- PrintInfo(): void
- GetAge(): int
- SetClass(string newStdClass): void

Phần trên cùng: tên lớp (Class name)

Phần giữa: các thuộc tính (Attributes)

Phần dưới cùng: các phương thức (Methods)

19

19

Các khái niệm cơ bản

Mỗi **thuộc tính** được viết dưới dạng:

Student

- studentId: string
- name: string
- birthYear: int
- gender: bool
- stdClass: string

- Input(): void
- PrintInfo(): void
- GetAge(): int
- SetClass(string newStdClass): void

Access modifier	Attribute name	Data type
+	studentId	: string

Access modifier:

+ Public

- Private

Protected

(Sẽ được học ở bài sau, tạm thời dùng +)

20

20

Các khái niệm cơ bản

Mỗi **phương thức** được viết dưới dạng:

Student

- studentId: string
- name: string
- birthYear: int
- gender: bool
- stdClass: string

- Input(): void
- PrintInfo(): void
- GetAge(): int
- SetClass(string newStdClass): void

Access modifier	Method name (parameters)	return type
+	Input()	: void

Access modifier:

+ Public

- Private

Protected

(Sẽ được học ở bài sau, tạm thời dùng +)

21

21


Bài tập vận dụng


Bài tập 1: Class **Fraction**
 Hãy xác định các thuộc tính và phương thức cho lớp **Fraction** mô tả các đối tượng là các phân số



22


Bài tập vận dụng


Bài tập 2: Class **Rectangle**
 Hãy xác định các thuộc tính và phương thức cho lớp **Rectangle** mô tả các đối tượng là các hình chữ nhật.


Bài tập 3: Class **BankAccount**
 Hãy xác định các thuộc tính và phương thức cho lớp **BankAccount** mô tả các đối tượng là các tài khoản ngân hàng.



23





Q & A

*"One who never asks
either knows everything or nothing"*
 Malcolm S. Forbes





24
