

BUỔI 07 OOP: TÍNH KẾ THỪA VÀ ĐA HÌNH (PHẦN 1)

MUC TIÊU

- Hiểu được tính Kế thừa và Đa hình trong Lập trình hướng đối tượng
- Nắm vững cú pháp tạo định nghĩa lớp dẫn xuất và ghi đè các phương thức (method overriding) của lớp cơ sở

Bài tập 1. Class Person, Student và Staff

Person - name: string - birthYear: int - address: string + Person() + Person(string name, int birthYear, string address) + Input(): void + GetAge(): int + ToString(): string

Student

- program: string
- year: int
- + Student()
- Student(string name, int birthYear, string address, string program, int year)
- + Input(): void
- + ToString(): string
- + ChangeProgram(string program): void

Staff

- departmenyt: string
- salary: double
- + Staff()
- Staff(string name, int birthYear, string address, string department, double salary)
- + Input(): void
- + ToString(): string
- + UpdateSalary(double salary): void

Mô tả:

- Class Person
 - + Person(string name, int birthYear, string address): Khởi tạo một đối tượng Person với các thông tin cho trước
 - + Input(): Phương thức để người dùng lần lượt nhập các thông tin cho một Person
 - + GetAge(): Tính tuổi của Person tính đến năm hiện tại
 - + ToString(): Trả về chuỗi biểu diễn một Person

"Person[Name: {name }, Age: {birthYear}, Address: {address}]"



Class Student

- + Student(string name, int birthYear, string address, string program, int year): Khởi tạo tạo một đối tượng Student với các thông tin cho trước (gọi Constructor của class Person)
- + Input(): Overriding phương thức Input() của class Person để người dùng nhập thêm các thông tin program và year
- + ToString(): Overriding phương thức ToString() của class Person để trả về chuỗi biểu diễn một Student

"Student[Person[Name: {name }, Age: {birthYear}, Address: {address}] - Program: {program}, Year: {year}]"

Class Staff

- + Staff(string name, int birthYear, string address, string department, double salary): Khởi tạo tạo một đối tượng Staff với các thông tin cho trước (gọi Constructor của class Person)
- + Input(): Overriding phương thức Input() của class Person để người dùng nhập thêm các thông tin department và salary
- + ToString(): Overriding phương thức ToString() của class Person để trả về chuỗi biểu diễn một Staff

"Staff[Person[Name: {name }, Age: {birthYear}, Address: {address}] - Department: {department}, Salary: {salary}]"

Bài tập 2. Class **Fraction** và **MixedFraction**

Fraction - numerator: int - denominator: int + Fraction() + Fraction(int numerator, int denominator) + Fraction(int numerator) - GCD(int a, int b): void + Simplify(): void + ToString(): string + ToDecimal(): double + Add(Fraction f): Fraction + Subtract(Fraction f): Fraction + Multiply(Fraction f): Fraction + Divide(Fraction f): Fraction

MixedFraction

- + MixedFraction(int wholePart, int numerator, int denominator)
- + MixedFraction(Fraction f)
- + ToString(): string



Mô tả:

Class Fraction

- + Fraction(): Khởi tạo phân số có tử số = 0, mẫu số =1
- + Fraction(int numerator, int denominator): Khởi tạo phân số có tử số = numerator, mẫu số = denominator
- + Fraction(int numerator): Khởi tạo phân số có tử số = numerator, mẫu số = 1
- + ToString(): Trả về chuỗi biểu diễn một phân số có dạng "tử số/mẫu số"

Class MixedFraction

- + MixedFraction(int wholePart, int numerator, int denominator): Khởi tạo hỗn số có phần nguyên = wholePart và phần phân số có tử số = numerator, mẫu số = denominator Gợi ý: Để việc xử lý đơn giản, các hỗn số đều được chuyển về lưu trữ dưới dạng phân số. Do đó constructor sẽ chuyển đổi hỗn số thành phân số và chỉ lưu trữ 2 thuộc tính numerator và denominator mà không cần khai báo thêm thuộc tính wholePart
- + MixedFraction(Fraction f): Khởi tạo hỗn số từ một phân số
- + ToString(): Trả về chuỗi biểu diễn một hỗn số có dạng "phần nguyên [tử số/mẫu số]"