

Contents

Đề_02_LậpTrìnhHướngĐốiTượng.....	2
Đề_02_LậpTrìnhHướngĐốiTượng_ĐápÁn.....	4
code Đề 2.....	9

TRƯỜNG ĐẠI HỌC ĐÔNG Á KHOA CÔNG NGHỆ THÔNG TIN	KỲ THI KẾT THÚC HỌC PHẦN Học phần: Lập trình hướng đối tượng Hình thức thi: Tự luận Thời gian: 90 phút <i>Không được sử dụng tài liệu.</i>
---	--

ĐỀ SỐ: 02

Phần I. LÝ THUYẾT VÀ ĐỌC HIỂU MÃ NGUỒN

Câu 1. (1.0 điểm) Để tạo một lớp con kế thừa từ lớp cha trong java dùng từ khóa nào? Cho ví dụ code đơn giản tạo một lớp con kế thừa từ lớp cha bất kỳ?

Câu 2. (1.0 điểm) Trong lập trình hướng đối tượng java tính đóng gói (encapsulation) được thể hiện qua các từ khóa nào? Nêu các mục đích của tính đóng gói trong lập trình hướng đối tượng java?

Câu 3. (1.0 điểm) **Ghi đè** phương thức (Overriding) là gì? Trong ví dụ bên dưới hãy chỉ ra hàm nào được ghi đè? Nêu ý nghĩa của việc ghi đè phương thức?

```
// Lớp cha
public class Animal {
    public void makeSound() {
        System.out.println("Animal makes a sound");
    }
}

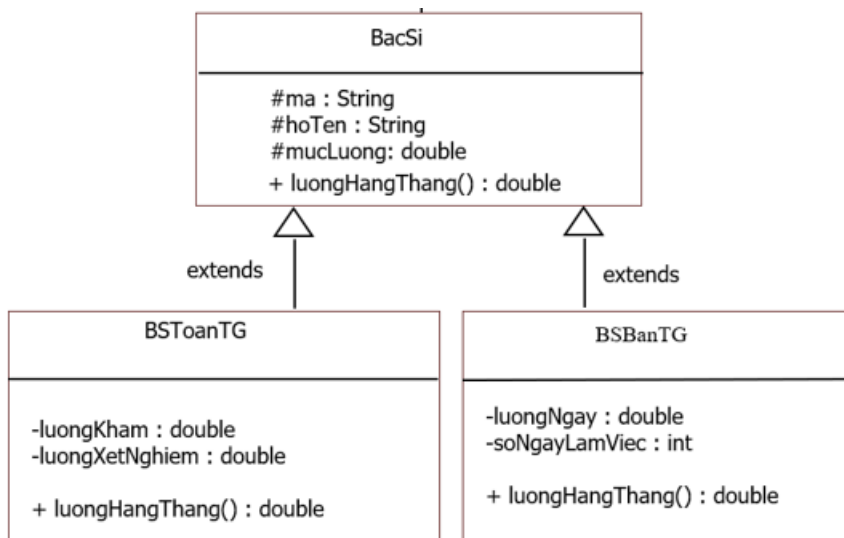
// Lớp con kế thừa từ lớp cha
public class Dog extends Animal {
    @Override
    public void makeSound() {
        System.out.println("Dog gogo...");
    }
}

public class App {
    public static void main(String[] args) {
        Animal myAnimal = new Animal();
        Dog myDog = new Dog

        myAnimal.makeSound(); // Output: Animal makes a sound
        myDog.makeSound();    // Output: Dog barks
    }
}
```

Phần II. VIẾT CHƯƠNG TRÌNH

Cho sơ đồ quan hệ giữa các lớp trong phần mềm quản lý bệnh viện như bên dưới:



Hãy thực hiện các yêu cầu sau:

Câu 4: (1.0 điểm) Định nghĩa lớp **BacSi** với các thuộc tính và phương thức theo sơ đồ trên.

Câu 5: (2.0 điểm) Định nghĩa lớp **BSToanTG** kế thừa (extends) lớp **BacSi** với các thuộc tính và phương thức theo sơ đồ trên. Với `@override` phương thức **luongHangThang()** được tính như sau: lương của **bác sĩ toàn thời gian** tính bằng tổng: **luongKham** + **luongXetNghiem** +

mucLuong.

Giải thích: - **luongKham**: lương khám

- **luongXetNghiem**: lương xét nghiệm

- **mucLuong**: mức lương

Câu 6: (2.0 điểm) Định nghĩa lớp **BSBanTG** kế thừa (extends) lớp **BacSi** với các thuộc tính và phương thức theo sơ đồ trên. Với `@override` phương thức **luongHangThang()** được tính như sau: lương của **bác sĩ bán thời gian** tính bằng tổng: **mucLuong*30%** + **luongNgay*soNgayLamViec**.

Câu 7: Trong hàm **main** chương trình chính (Gợi ý: tạo tệp tin App.java):

7.1 (1.0 điểm) Hãy tạo 2 đối tượng **BSToanTG** và đưa vào ArrayList. Tính tổng lương chỉ cho bác sĩ **toàn** thời gian.

7.2 (1.0 điểm) Hãy tạo 2 đối tượng **BSBanTG** và đưa vào ArrayList. Tính tổng lương chỉ cho bác sĩ **bán** thời gian.

– Hết –

TRƯỜNG ĐẠI HỌC ĐÔNG Á KHOA CÔNG NGHỆ THÔNG TIN	CẤU TRÚC ĐỀ THI KẾT THÚC HỌC PHẦN
	Học phần: Lập trình hướng đối tượng.
	Hình thức thi: Tự luận.
Thời gian thi: 90 phút	<input type="checkbox"/> Được sử dụng tài liệu khi làm bài
<input checked="" type="checkbox"/> Không được sử dụng tài liệu khi làm bài	

PHẦN 1: CÁC YÊU CẦU CỦA CẤU TRÚC ĐỀ THI NHẪM ĐÁP ỨNG CÁC CHUẨN ĐẦU RA CỦA HỌC PHẦN.

Lưu ý: phần này phải thống nhất thông tin với nội dung đã nêu trong đề cương chi tiết của học phần

Ký hiệu CDR	Nội dung CDR	Hình thức kiểm tra đánh giá	Số lượng câu hỏi cần cho 1 đề thi	Điểm số tối thiểu (mức đạt CDR)	Điểm số tối đa
(1)	(2)	(3)	(4)	(5)	(6)
CLO1	Vận dụng kiến thức về ngôn ngữ lập trình để lập trình được các bài toán thực tế.	Bài tập lập trình cụ thể (viết chương trình Java)	1	0.5	1.0
CLO2	Hiểu được các khái niệm, phương pháp về lập trình hướng đối tượng.	Bài kiểm tra lý thuyết về các khái niệm OOP (Object-Oriented Programming) trong Java.	3	1.0	3.0
CLO3	Vận dụng kỹ thuật lập trình hướng đối tượng để giải quyết một số bài toán trong thực tế	Yêu cầu thiết kế và lập trình ứng dụng Java theo mô hình OOP.	2	2.0	4.0
CLO4	Có khả năng giải quyết vấn đề kỹ thuật về lập trình hướng đối tượng.	Có khả năng viết mã code giải quyết các vấn đề kỹ thuật cụ thể trong Java.	1	0.5	2.0
Tổng			7.0	4.0	10.0

Chú thích các cột

1. Ký hiệu của các chuẩn đầu ra (CLO)
2. Nội dung của chuẩn đầu ra tương ứng
3. Hình thức kiểm tra, đánh giá có thể là: tự luận, trắc nghiệm, vấn đáp, thực hành trên máy tính, thực hành tại xưởng, thực hành tại phòng thí nghiệm, dự án, đồ án, báo cáo, thuyết trình, ... Hình thức kiểm tra đánh giá phải thống nhất với đề cương chi tiết của học phần.
4. Số lượng câu hỏi cần thiết cho 1 đề thi để đảm bảo bài đánh giá được CDR đó.
5. Điểm số tối thiểu (mức đạt CDR) là điểm số ít nhất người học cần đạt được để đảm bảo đạt được CDR đó. Tổng cộng điểm số tối thiểu của một đề thi không được dưới 4 điểm.
6. Điểm số tối đa là điểm số tối đa của CDR đó. Tổng cộng điểm số tối đa của một đề thi không được trên 10 điểm .

Đà Nẵng, ngày ... tháng ... năm ...

PHẦN 2: NGÂN HÀNG CÂU HỎI ĐÁNH GIÁ CHUẨN ĐẦU RA CỦA HỌC PHẦN

TRƯỜNG ĐẠI HỌC ĐÔNG Á KHOA CÔNG NGHỆ THÔNG TIN	NGÂN HÀNG CÂU HỎI THI KẾT THÚC HỌC PHẦN Học phần: Kỹ thuật lập trình Hình thức thi: Tự luận
---	--

TT	Nội dung câu hỏi	Điểm	Cấp độ
(1)	(2)	(3)	(4)
Phần I	LÝ THUYẾT VÀ ĐỌC HIỂU MÃ NGUỒN Chuẩn đầu ra CLO2: Hiểu được các khái niệm, phương pháp về lập trình hướng đối tượng.		
Câu 1	Đề tạo một lớp con kế thừa từ lớp cha trong java dùng từ khóa nào? Cho ví dụ code đơn giản tạo một lớp con kế thừa từ lớp cha bất kỳ?	1.0	A
	<i>Đáp án:</i>		
	Ý 1: Trong Java, để tạo một lớp con kế thừa từ một lớp cha, bạn sử dụng từ khóa “ extends ”.	0.5	
	Ý 2: Cho ví dụ đúng về lớp cha và lớp con kế thừa từ lớp cha.	0.5	
Câu 2	Trong lập trình hướng đối tượng java tính đóng gói (encapsulation) được thể hiện qua các từ khóa nào? Nêu các mục đích của tính đóng gói trong lập trình hướng đối tượng java?	1.0	A
	<i>Đáp án:</i>		
	Ý 1: Trong lập trình hướng đối tượng Java, tính đóng gói (encapsulation) được thể hiện qua việc sử dụng các từ khóa truy cập như: private , protected , public và default (không khai báo từ khóa).	0.5	
	Ý 2: trả lời được ít nhất 2 ý Mục đích của tính đóng gói trong lập trình hướng đối tượng Java: - Bảo vệ dữ liệu: Bằng cách sử dụng các từ khóa truy cập như: private, tính đóng gói bảo vệ dữ liệu khỏi việc bị thay đổi hoặc truy cập bất hợp pháp từ bên ngoài. Dữ liệu chỉ có thể được truy cập thông qua các phương thức công khai. - Ẩn chi tiết triển khai: Tính đóng gói giúp ẩn đi các chi tiết triển khai bên trong lớp. Người dùng chỉ cần biết cách sử dụng các phương thức công khai (public methods) mà không cần biết các thuộc tính bên trong hoạt động như thế nào. - Kiểm soát truy cập: Tính đóng gói cho phép kiểm soát cách thức và nơi các thuộc tính và phương thức được truy cập hoặc sửa đổi. Điều này giúp duy trì tính nhất quán và toàn vẹn của dữ liệu. - Dễ dàng bảo trì và nâng cấp: Khi chi tiết triển khai bị ẩn, bạn có thể thay đổi hoặc nâng cấp mã bên trong lớp mà không ảnh hưởng đến mã sử dụng lớp đó. Điều này giúp dễ dàng bảo trì và mở rộng ứng dụng. - Tái sử dụng mã: Tính đóng gói giúp tạo ra các lớp có thể tái sử dụng. Bạn có thể sử dụng lại các lớp mà không cần lo lắng về việc vô tình thay đổi dữ liệu hoặc phương thức bên trong lớp đó.	0.5	
Câu 3	Ghi đè phương thức (Overriding) là gì? Trong ví dụ bên dưới hãy chỉ ra hàm nào được ghi đè? Nêu ý nghĩa của việc ghi đè phương thức?	1.0	A
	<div>//Lớp cha</div>		

	<pre> public class Animal { public void makeSound() { System.out.println("Animal makes a sound"); } } // Lớp con kế thừa từ lớp cha public class Dog extends Animal { @Override public void makeSound() { System.out.println("Dog gogo..."); } } public class App { public static void main(String[] args) { Animal myAnimal = new Animal(); Dog myDog = new Dog myAnimal.makeSound(); // Output: Animal makes a sound myDog.makeSound(); // Output: Dog barks } } </pre>		
	Đáp án:		
	<p>Ý 1: Trong Java, ghi đè phương thức (method overriding) là cho phép một lớp con có thể code lại thân hàm của một phương thức đã được khai báo trong lớp cha của nó với phương thức trong lớp con có cùng tên, cùng kiểu trả về và cùng danh sách tham số như một phương thức trong lớp cha, thì phương thức đó được coi là ghi đè phương thức của lớp cha.</p>	0.5	
	<p>Ý 2: Hàm được ghi đè ở ví dụ trên là hàm public void makeSound() của lớp cha.</p> <pre> public void makeSound() { System.out.println("Dog gogo..."); } </pre>	0.25	
	<p>Ý 3: Trả lời được ít nhất 1 ý</p> <p>Mục đích và lợi ích của ghi đè phương thức:</p> <ul style="list-style-type: none"> - Tính mở rộng: Cho phép lớp con mở rộng hoặc thay đổi hành vi của lớp cha mà không cần thay đổi mã nguồn của lớp cha. - Tính linh hoạt: Cung cấp cho các lớp con khả năng điều chỉnh hành vi của phương thức để phù hợp với nhu cầu cụ thể của chúng. - Tính nhất quán: Giữ cho giao diện của các phương thức nhất quán giữa lớp cha và lớp con, giúp mã nguồn dễ đọc và duy trì hơn. - Đa hình (Polymorphism): Ghi đè phương thức cho phép tính đa hình, nghĩa là cùng một lời gọi phương thức có thể dẫn đến các hành vi khác nhau, tùy thuộc vào đối tượng thực tế. 	0.25	
Phần II	<p>VIẾT CHƯƠNG TRÌNH</p> <p>Chuẩn đầu ra CLO1: Vận dụng kiến thức về ngôn ngữ lập trình để lập trình được các bài toán thực tế.</p>		

	<p>Chuẩn đầu ra CLO3: Vận dụng kỹ thuật lập trình hướng đối tượng để giải quyết một số bài toán trong thực tế.</p> <p>Chuẩn đầu ra CLO4: Có khả năng giải quyết vấn đề kỹ thuật về lập trình hướng đối tượng.</p>		
	<p>Cho sơ đồ quan hệ giữa các lớp trong phần mềm quản lý bệnh viện như bên dưới:</p> <div style="text-align: center;"> <pre> classDiagram class BacSi { #ma : String #hoTen : String #mucLuong : double +luongHangThang() : double } class BSToanTG { -luongKham : double -luongXetNghiem : double +luongHangThang() : double } class BSBanTG { -luongNgay : double -soNgayLamViec : int +luongHangThang() : double } BacSi < -- BSToanTG BacSi < -- BSBanTG </pre> </div> <p>Hãy thực hiện các yêu cầu sau:</p>		B
Câu 4	Định nghĩa lớp BacSi với các thuộc tính và phương thức theo sơ đồ trên.	1.0	B
	<i>Đáp án:</i>		
	<p>Ý 1: Tạo được lớp BacSi gồm:</p> <ul style="list-style-type: none"> - Tên lớp & thuộc tính (0.25đ) - Hàm khởi tạo cho lớp (0.25đ) 	0.5	
	Ý 2: - định nghĩa đúng hàm lương tháng luongHangThang () và trả về kiểu dữ liệu đúng.	0.25	
	Ý 3: - Có hàm các hàm get và set cho từng tham số	0.25	
Câu 5	Định nghĩa lớp BSToanTG kế thừa (extends) lớp BacSi với các thuộc tính và phương thức theo sơ đồ trên. Với <i>@override</i> phương thức luongHangThang() được tính như sau: lương của bác sĩ toàn thời gian tính bằng tổng: luongKham + luongXetNghiem + mucLuong .	2.0	B
	<i>Đáp án:</i>		
	Ý 1: - tạo lớp BSToanTG kế thừa extends từ lớp cha BacSi	0.5	
	Ý 2: - Có hàm khởi tạo cho lớp và có dùng hàm super để gọi hàm khởi tạo lớp cha.	0.5	
	Ý 3: - Có hàm tính toán lương tháng, ghi đè (<i>overwrite</i>) lại hàm luongHangThang() ở lớp cha	0.5	
	Ý 4: - Có hàm get/set cho các tham số.	0.5	
Câu 6	Định nghĩa lớp BSBanTG kế thừa (extends) lớp BacSi với các thuộc tính và phương thức theo sơ đồ trên. Với <i>@override</i> phương thức luongHangThang() được tính như sau: lương của bác sĩ bán thời gian tính bằng tổng: mucLuong*30% + luongNgay*soNgayLamViec .	2.0	B
	<i>Đáp án:</i>		
	Ý 1: - Tạo lớp BSBanTG kế thừa extends từ lớp cha BacSi	0.5	
	Ý 2: - Có hàm khởi tạo cho lớp và có dùng hàm super để gọi hàm khởi tạo lớp cha.	0.5	

	Ý 3: - Có hàm tính toán lương tháng, ghi đè (<i>override</i>) lại hàm luongHangThang() ở lớp cha	0.5	
	Ý 4: - Có hàm get/set cho các tham số.	0.5	
Câu 7	Trong hàm main chương trình chính (<i>Gợi ý</i> : tạo tệp tin App.java): 7.1 (1.0 điểm) Hãy tạo 2 đối tượng BSToanTG và đưa vào ArrayList. Tính tổng lương chi cho bác sĩ toàn thời gian. 7.2 (1.0 điểm) Hãy tạo 2 đối tượng BSBanTG và đưa vào ArrayList. Tính tổng lương chi cho bác sĩ bán thời gian.	2.0	C
	<i>Đáp án:</i>		
	Ý 7.1: - Tạo được 2 đối tượng từ lớp BSToanTG (0.25đ) - add được 2 đối tượng vào arraylist của lớp đối tượng BSToanTG (0.25đ) - Tạo được vòng lặp for tính tổng: Tính tổng lương chi cho bác sĩ toàn thời gian từ arraylist ở trên (0.5đ)	1.0	
	Ý 7.2: - Tạo được 2 đối tượng từ lớp BSBanTG (0.25đ) - add được 2 đối tượng vào arraylist của lớp BSBanTG (0.25đ) - Tạo được vòng lặp for tính tổng: Tính tổng lương chi cho bác sĩ bán thời gian từ arraylist ở trên (0.5đ)	1.0	

Chú thích các cột

1. Số thứ tự của CĐR và câu hỏi trong CĐR

2. Nội dung câu hỏi và đáp án:

- Nội dung câu hỏi phải đầy đủ, dễ hiểu, rõ ràng các ý. Nếu có hình ảnh, bảng biểu, đồ thị phải định dạng bằng hình ảnh có màu sắc tương phản để dễ dàng nhìn thấy khi in trắng đen.

- Nội dung đáp án phải tương ứng với các tiêu chí đánh giá chuẩn đầu ra đó. Đáp án phải đánh giá đầy đủ các ý của câu hỏi và đảm bảo đo lường được.

3. Điểm: là điểm tối đa của câu hỏi/ đáp án

4. Cấp độ: là cấp độ đánh giá của câu hỏi: câu hỏi cấp độ Nhận biết/Thông hiểu ký hiệu: A. Câu hỏi cấp độ Vận dụng, Phân tích:ký hiệu B. Câu hỏi cấp độ Đánh giá, Sáng tạo: ký hiệu C

NGƯỜI DUYỆT

Đà Nẵng, ngày ... tháng ... năm ...
GIẢNG VIÊN/NHÓM BIÊN SOẠN

Lớp BacSi.java

```
public class BacSi {  
    private String maBS;  
    private String hoTen;  
    private double mucLuong;  
  
    public BacSi(String maBS, String hoTen, double mucLuong) {  
        this.maBS = maBS;  
        this.hoTen = hoTen;  
        this.mucLuong = mucLuong;  
    }  
  
    public double luongHangThang() {  
        return mucLuong;  
    }  
  
    public String getMaBS() {  
        return maBS;  
    }  
  
    public void setMaBS(String maBS) {  
        this.maBS = maBS;  
    }  
  
    public String getHoTen() {  
        return hoTen;  
    }  
  
    public void setHoTen(String hoTen) {  
        this.hoTen = hoTen;  
    }  
  
    public double getMucLuong() {  
        return mucLuong;  
    }  
  
    public void setMucLuong(double mucLuong) {  
        this.mucLuong = mucLuong;  
    }  
}
```

Lớp BSToanTG.java

```
public class BSToanTG extends BacSi {
    private double luongKham;
    private double luongXetNghiem;

    public BSToanTG(String maBS, String hoTen, double mucLuong, double
luongKham, double luongXetNghiem) {
        super(maBS, hoTen, mucLuong);
        this.luongKham = luongKham;
        this.luongXetNghiem = luongXetNghiem;
    }

    @Override
    public double luongHangThang() {
        return getMucLuong() + luongKham + luongXetNghiem;
    }

    public double getLuongKham() {
        return luongKham;
    }

    public void setLuongKham(double luongKham) {
        this.luongKham = luongKham;
    }

    public double getLuongXetNghiem() {
        return luongXetNghiem;
    }

    public void setLuongXetNghiem(double luongXetNghiem) {
        this.luongXetNghiem = luongXetNghiem;
    }
}
```

Lớp BSBanTG.java

```
public class BSBanTG extends BacSi {
    private double luongNgay;
    private int soNgayLamViec;
```

```

    public BSBanTG(String maBS, String hoTen, double mucLuong, double
luongNgay, int soNgayLamViec) {
        super(maBS, hoTen, mucLuong);
        this.luongNgay = luongNgay;
        this.soNgayLamViec = soNgayLamViec;
    }

    @Override
    public double luongHangThang() {
        return getMucLuong() * 0.3 + luongNgay * soNgayLamViec;
    }

    public double getLuongNgay() {
        return luongNgay;
    }

    public void setLuongNgay(double luongNgay) {
        this.luongNgay = luongNgay;
    }

    public int getSoNgayLamViec() {
        return soNgayLamViec;
    }

    public void setSoNgayLamViec(int soNgayLamViec) {
        this.soNgayLamViec = soNgayLamViec;
    }
}

```

Lớp App.java (chương trình chính)

```

import java.util.ArrayList;

public class App {
    public static void main(String[] args) {
        // Danh sách bác sĩ toàn thời gian
        ArrayList<BSToanTG> dsToanTG = new ArrayList<>();
        dsToanTG.add(new BSBanTG("BS01", "Nguyen Van A", 1000, 500, 300));
        dsToanTG.add(new BSBanTG("BS02", "Tran Thi B", 1100, 400, 250));

        double tongLuongToanTG = 0;
    }
}

```

```

for (BSToanTG bs : dsToanTG) {
    tongLuongToanTG += bs.luongHangThang();
}

System.out.println("Tổng lương bác sĩ toàn thời gian: " + tongLuongToanTG);

// Danh sách bác sĩ bán thời gian
ArrayList<BSBanTG> dsBanTG = new ArrayList<>();
dsBanTG.add(new BSBanTG("BS03", "Le Van C", 800, 100, 15));
dsBanTG.add(new BSBanTG("BS04", "Pham Thi D", 900, 120, 18));

double tongLuongBanTG = 0;
for (BSBanTG bs : dsBanTG) {
    tongLuongBanTG += bs.luongHangThang();
}

System.out.println("Tổng lương bác sĩ bán thời gian: " + tongLuongBanTG);
}
}

```