



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG-HCM
ĐỀ THI KẾT THÚC HỌC PHẦN
Học kỳ 2 – Năm học 2022-2023

MÃ LƯU TRỮ
(do phòng KT-ĐBCL ghi)

Tên học phần: Phương pháp lập trình hướng đối tượng Mã HP: CSC10003

Thời gian làm bài: 100 phút Ngày thi: _____

Ghi chú: Sinh viên được phép sử dụng sử dụng tài liệu giấy khi làm bài.

Họ tên sinh viên: MSSV: STT:

Câu 1

Khai báo lớp đối tượng (class) Mobile, mô tả thông tin của 1 điện thoại, gồm 3 thuộc tính (attribute): mã nhận dạng (code), tên dòng điện thoại (name), nhãn hiệu (brand), ngày sản xuất (manufacturing date). Cài đặt toán tử (operator) nhập, toán tử xuất và phương thức (method) so sánh 2 điện thoại (so sánh theo ngày sản xuất).

Câu 2

Giải thích ngắn gọn và cho biết kết quả in ra màn hình của đoạn code sau.

```
#include <iostream>
using namespace std;

class Base
{
public:
    Base()
    {
        cout << "Base::Base()" << endl;
    }
    Base(int t)
    {
        cout << "Base::Base(int)" << endl;
    }
    Base(int t, int k)
    {
        cout<<"Base::Base(int,int)" << endl;
    }
    Base(const Base& src)
    {
        cout<<"Base::Base(const Base&)" << endl;
    }
    Base& operator=(const Base& src)
    {
        cout << "Base::operator=()" << endl;
        return *this;
    }
    virtual void doSth(int x)
    {
        cout << "Base::doSth(int)" << endl;
    }
    void doSthE(int x)
    {
        cout << "Base::doSthE(int)" << endl;
    }
    virtual ~Base()
    {
        cout << "Base::~~Base()" << endl;
    }
};
```

```
class Derived: public Base
{
public:
    Derived()
    {
        cout<<"Derived::Derived()" << endl;
    }
    Derived(int t): Base(t)
    {
        cout<<"Derived::Derived(int)" << endl;
    }
    Derived(int t, int k)
    {
        cout<<"Derived::Derived(int, int)" << endl;
    }
    Derived(const Derived& src): Base(src)
    {
        cout<<"Derived::Derived(const Derived &)" << endl;
    }
    Derived& operator=(const Derived& src)
    {
        cout<<"Derived::operator=()" << endl;
        return *this;
    }
    void doSth(int x)
    {
        cout << "Derived::doSth(int)" << endl;
    }
    void doSthE(int x)
    {
        cout << "Derived::doSthE(int)" << endl;
    }
    ~Derived()
    {
        cout << "Derived::~~Derived()" << endl;
    }
};
```

Họ tên người ra đề/MSCB:
Họ tên người duyệt đề:

Chữ ký:
Chữ ký:

(Đề thi gồm 4 trang)
[Trang 1/4]

<pre>}; }</pre>	<pre>};</pre>
<pre>void testing(Base& first, Derived second) { cout << "Testing 3: " << endl; Derived x; x = second; first.doSth(1); first.doSth(2); cout << "Testing 4: " << endl; x.doSth(3); x.doSthE(4); }</pre>	<pre>int main() { cout << "Testing 1: " << endl; Derived x(5), y(3, 7); cout << "Testing 2: " << endl; testing(x, y); cout << "Testing 5: " << endl; return 0; }</pre>

Câu 3

Công ty AlphaTech mở cuộc thi Hackathon nội bộ cho các nhân viên trong công ty mỗi tháng. Có 2 loại dự án mà nếu chiến thắng sẽ được tài trợ kinh phí như sau:

- Dự án dạng **ý tưởng**: tùy theo số **tháng** nhóm cần để phát triển mà được tài trợ tiền tương ứng. Ban đầu công ty quyết định mỗi tháng sẽ tài trợ **20** triệu.
- Dự án dạng **sản phẩm tối thiểu**: Dự án loại này đã được phát triển tính năng tối thiểu vừa đủ (MVP - Minimal viable product). Tiền tài trợ sẽ bao gồm hai chi phí là **marketing** để quảng bá và chi phí **vận hành** sản phẩm thử nghiệm với số lượng người dùng ban đầu có được từ việc quảng bá. Chi phí cho mỗi tháng sẽ được tài trợ là **15** triệu và là như nhau cho cả hai mục đích quảng bá cũng như vận hành.

Bạn được yêu cầu xây dựng ứng dụng giúp công ty quản lý các dự án được tài trợ theo phương pháp lập trình hướng đối tượng.

Biết hàm main chính của chương trình có mã nguồn như sau:

```
int main()
{
    cout << "Chương trình hỗ trợ Hackathon của AlphaTech tháng 6/2023" << endl;
    vector<Project*> projects = { // Bỏ qua tiềm ẩn rò rỉ vùng nhớ
        new Idea("SuperTech", 6),
        new Mvp("FutureCar", 6, 3),
        new Idea("VirtualHouse", 7),
        new Idea("NightVision", 4),
        new Mvp("SmartKey", 4, 5)
    };
    cout << "Số dự án: " << projects.size() << endl;
    int sum = 0;
    for(int i = 0; i < projects.size(); i++) {
        sum += projects[i]->grantMoney();
        cout << i + 1 << ".";
        projects[i]->print();
        cout << endl;
    }
    cout << "Tổng tiền tài trợ:" << sum;
    return 0;
}
```

Kết quả mong đợi của hàm main sẽ có dạng như sau:



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG-HCM
ĐỀ THI KẾT THÚC HỌC PHẦN
Học kỳ 2 – Năm học 2022-2023

MÃ LƯU TRỮ
(do phòng KT-ĐBCL ghi)

Chương trình hỗ trợ Hackathon của AlphaTech tháng 6/2023

Số dự án: 5

1. Ý tưởng: SuperTech. Phát triển: 6 tháng. Kinh phí: 120000000
2. Sản phẩm: FutureCar. Vận hành: 6 tháng. Marketing: 3 tháng. Kinh phí: 135000000
3. Ý tưởng: VirtualHouse. Phát triển: 7 tháng. Kinh phí: 140000000
4. Ý tưởng: NightVision. Phát triển: 4 tháng. Kinh phí: 80000000
5. Sản phẩm: SmartKey. Vận hành: 4 tháng. Marketing: 5 tháng. Kinh phí: 135000000

Tổng tiền tài trợ: 610000000

Câu 3.1

Vẽ sơ đồ lớp, khai báo và cài đặt các lớp phục vụ chương trình trên.

Chú ý: Một giải pháp giải quyết vấn đề theo hướng đối tượng sẽ cân nhắc các yếu tố về tính tiến hóa, dễ mở rộng, dễ bảo trì và mã nguồn có khả năng tái sử dụng cao.

Câu 3.2

Câu 3.2.a: Trong trường hợp ứng dụng muốn nâng cấp để hiển thị số tiền không phải dạng thô mà có kí tự phân cách hàng ngàn (thousand separator) là dấu chấm (“.”) thì bạn sẽ cải tiến chương trình của mình ra sao? Ngoài ra nếu kí tự phân cách hàng ngàn là dấu phẩy (“,”) thì sao?

Ví dụ: Thay vì hiển thị 1200000000 thì sẽ hiển thị 120.000.000 hoặc 120,000,000

Hãy trình bày ý tưởng và **vẽ lại sơ đồ lớp** đã được cập nhật, không cần cài đặt.

Câu 3.2.b: Trong trường hợp ứng dụng muốn hiển thị dạng tiền tệ không chỉ VND mà hỗ trợ tiền \$ thì sao?

Ví dụ: Nếu là tiền tệ của Việt Nam: 120.000.000 đ

Nếu là tiền tệ của Mỹ: \$ 120,000,000

Hãy trình bày ý tưởng và **vẽ lại sơ đồ lớp** đã được cập nhật để hỗ trợ cách thức hiển thị mới này, không cần cài đặt.

Câu 3.2.c: Thay vì hiển thị từng dòng, ứng dụng muốn **cung cấp thêm** cách hiển thị ở dạng bảng như thế này thì sao:

Loại dự án	Tên dự án	Thông tin	Kinh phí
Ý tưởng	SuperTech	Phát triển: 6 tháng	120.000.000
Sản phẩm	FutureCar	Vận hành: 6 tháng. Marketing: 3 tháng	135.000.000
Ý tưởng	VirtualHouse	Phát triển: 7 tháng	140.000.000
Ý tưởng	NightVision	Phát triển: 4 tháng	80.000.000
Sản phẩm	SmartKey	Vận hành: 4 tháng. Marketing: 5 tháng	135.000.000

Hãy trình bày ý tưởng và **vẽ lại sơ đồ lớp** đã được cập nhật, không cần cài đặt.

Câu 3.2.d: Nếu sau này công ty thay đổi chiến lược tài trợ, chẳng hạn:

- Dự án sẽ được tài trợ vừa là tiền pháp định (VND) và tiền mã hóa crypto cũng như tài trợ văn phòng (ví dụ địa chỉ là 227 Nguyễn Văn Cừ P4 Q5, theo sau là tên phòng được cấp)



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG-HCM
ĐỀ THI KẾT THÚC HỌC PHẦN
Học kỳ 2 – Năm học 2022-2023

MÃ LƯU TRỮ
(do phòng KT-ĐBCL ghi)

- Dự án kéo dài quá 6 tháng thì từ tháng thứ 7 sẽ nhận thêm 20% tiền pháp định mỗi tháng.

Bạn sẽ thay đổi thiết kế như thế nào để hỗ trợ chiến lược tài trợ mới này? Hãy trình bày ý tưởng và **vẽ lại sơ đồ lớp** đã được cập nhật để hỗ trợ thay đổi này, không cần cài đặt.

===== **HẾT** =====

Câu 1

```
#include <iostream>
#include <string>
using namespace std;

class Date {
private:
    int day;
    int month;
    int year;
public:
    friend istream& operator>>(istream&, Date&);
    friend ostream& operator<<(ostream&, Date);
    bool operator>(Date);
};

istream& operator>>(istream& is, Date& d) {
    cout << "Day: ";
    is >> d.day;
    cout << "Month: ";
    is >> d.month;
    cout << "Year: ";
    is >> d.year;
    return is;
}

ostream& operator<<(ostream& os, Date d) {
    os << d.day << "/" << d.month << "/" << d.year;
    return os;
}

bool Date::operator>(Date b) {
    if (year > b.year) {
        return true;
    }
    if (year == b.year && month > b.month) {
        return true;
    }
    if (year == b.year && month == b.month && day > b.day) {
        return true;
    }
    return false;
}

//----- Dưới đây là lớp Mobile -----
class Mobile {
private:
    string code;
    string name;
    string brand;
    Date manufacturingDate;
public:
    friend istream& operator>>(istream&, Mobile&);
    friend ostream& operator<<(ostream&, Mobile);
    bool operator>(Mobile b);
};

istream& operator>> (istream& is, Mobile& m) {
    cout << "Code: ";
    is >> m.code;
    cout << "Name: ";
```

```

        is >> m.name;
        cout << "Brand: ";
        is >> m.brand;
        cout << "Input Date" << endl;
        is >> m.manufacturingDate;
        return is;
    }

ostream& operator<< (ostream& os, Mobile m) {
    os << "Code: " << m.code << endl;
    os << "Name: " << m.name << endl;
    os << "Brand " << m.brand << endl;
    os << "Date: " << m.manufacturingDate << endl;
    return os;
}

bool Mobile::operator>(Mobile b) {
    if (manufacturingDate > b.manufacturingDate) {
        return true;
    }
    return false;
}

int main() {
    Mobile a, b;
    cout << "Mobile a: ";
    cin >> a;
    cout << "Mobile b: ";
    cin >> b;
    if (a > b) {
        cout << "Ngày sản xuất a > b";
    }
    else {
        cout << "Ngày sản xuất b > a";
    }
}

```

Câu 2

Đoạn mã nguồn trên sẽ in ra các thông tin như sau:

Testing 1:

Dòng Derived x(5) sẽ in ra 2 thông tin, vì Derived có 1 tham số đầu vào có kế thừa từ lớp Base (có tham số đầu vào)

Base::Base(int)

Derived::Derived(int)

Dòng Derived y(3, 7) sẽ in 2 thông tin bên dưới

Base::Base()

Derived::Derived(int, int)

Do tham số đầu vào của testing(x,y) có y là tham trị nên một phương thức khởi tạo sao chép sẽ được tạo ra cho biến y, nên kết quả in là

Testing 2:

Base::Base(const Base&)

Derived::Derived(const Derived &)

Testing 3:

Derived::Derived() // một phương thức khởi tạo mặc định được tạo ra

Derived::operator=() // operator gán được in ra

Derived::doSthE(int) // do lớp cha sử dụng từ khóa virtual nên phương thức doSth(int) của lớp cha không được in mà in ở lớp con (Derived)

Base::doSthE(int) // phương thức doSthE(int) của lớp cha sẽ được gọi

Testing 4:

```
Derived::doSth(int) // phương thức doSth(int) của lớp Derived được gọi
Derived::doSthE(int) // phương thức doSthE(int) của lớp Derived được gọi
Xóa 2 biến class Derived trong hàm testing, xóa con trước rồi xóa cha
Derived::~~Derived()
Base::~~Base()
Derived::~~Derived()
Base::~~Base()
```

Testing 5:

```
Xóa 2 biến x và y trong main, xóa con trước rồi xóa cha
Derived::~~Derived()
Base::~~Base()
Derived::~~Derived()
Base::~~Base()
```

Câu 3

```
#include <iostream>
#include <vector>
#include <string>

using namespace std;

class Project {
protected:
    string name;
    //string office;
    //float crypto;
public:
    Project();
    virtual long grantMoney();
    virtual void print() = 0;
};

Project::Project() {
    name = " ";
}

long Project::grantMoney() {
    // Để cho có
    return 1;
}

//-----Lớp Idea-----
class Idea : public Project{
private:
    int growth;
public:
    Idea(string, int);
    long grantMoney();
    void print();
};

Idea::Idea(string nam, int gro) {
    name = nam;
    growth = gro;
}

long Idea::grantMoney() {
    return growth * 20000000;
}
```

```

void Idea::print() {
    cout << "Y tuong: " << name << "." << "Phat trien: " << growth << " thang. ";
    cout << "Kinh phi: " << grantMoney();
    cout << endl;
}

//-----Lớp Mvp-----
class Mvp : public Project {
private:
    int marketing;
    int operation;
public:
    Mvp(string, int, int);
    long grantMoney();
    void print();
};

Mvp::Mvp(string nam, int ope, int mar) {
    name = nam;
    operation = ope;
    marketing = mar;
}

long Mvp::grantMoney() {
    return (marketing + operation) * 15000000;
}

void Mvp::print() {
    cout << "San pham: " << name << "." << "Van hanh: " << operation << " thang. ";
    cout << "Marketing: " << marketing << " thang. ";
    cout << "Kinh phi: " << grantMoney();
    cout << endl;
}

int main()
{
    vector<Project*> projects = {
        new Idea("SuperTech", 6),
        new Mvp("FutureCar", 6, 3),
        new Idea("VirtualHouse", 7),
        new Idea("NightVision", 4),
        new Mvp("Smartkey", 4, 5)
    };

    cout << "So du an: " << projects.size() << endl;
    int sum = 0;
    for (int i = 0; i < projects.size(); i++) {
        sum += projects[i]->grantMoney();
        cout << i + 1 << ". ";
        projects[i]->print();
        cout << endl;
    }
    cout << "Tong tien tai tro: " << sum << endl;

    return 0;
}

```

Trình bày ý tưởng câu 3.2a

Bước 1: Chuyển kiểu dữ liệu số tiền từ long -> string. Tên chuỗi s.

120.000.000

Bước 2: Dùng vòng lặp để chạy từ dưới lên của chuỗi s.

Nếu biến đếm count = 3 chèn một dấu chấm.

Xét thêm nếu count = 3, nhưng biến chạy = 0. Lúc này đã về đầu chuỗi.

Thì chúng ta cũng dừng vòng lặp.
 Nếu biến đếm không đủ 3. Thì ngưng không làm nữa.
 Bước 3: Kết thúc vòng dừng chương trình, in ra số tiền cần chuẩn hóa.

Trình bày ý tưởng câu 3.2b

Giả sử ban đầu số tiền đang hiển thị là: 120.000.000 thì cách để chuyển sang VND như sau:

- Bước 1: Chèn ký tự tại `s[size] = " "` (dấu khoảng cách)
 Bước 2: Sau đó chèn ký tự tại `s[size+1] = "đ"` (dấu VND)
 Bước 3: Dừng chương trình hiển thị lại số tiền.

Nếu chuyển sang đô la mỹ thì làm như sau:

- Bước 1: Dùng vòng lặp chạy từ đầu đến cuối chuỗi
 Thay các dấu '.' thành dấu phẩy ',' (kết quả sau bước 1: 120,000,000)
 Bước 2: Dùng vòng lặp chạy từ cuối chuỗi dời các ký tự về 2 ô (Chèn hình)
 Bước 3: Chèn `s[0] = "$"` và chèn `s[1] = " "` (dấu khoảng cách)

Trình bày ý tưởng câu 3.2c

- Bước 1: Hiển thị các tiêu đề của bảng bằng cách dùng hàm `cout`
 Loại dự án | Tên dự án | Thông tin | Kinh phí

- Bước 2: Tiếp tục in thông tin của Project là Idea
`"Y tuong: " (dấu tab) << "|" << name << (dấu tab) << "|" << " Phát triển: " << growth << << " thang "`
`(dấu tab) << "|" grantMoney();`
 Bước 3: (viết theo kiểu Bước 2)

Trình bày ý tưởng câu 3.2d

- Bước 1: Thêm thuộc tính `office` (kiểu chuỗi) và `crypto` (kiểu số thực) vào lớp `Project`
 Bước 2: Khi dự án chạy quá 6 tháng. Thì công thức tính tiền tài trợ sẽ thay đổi như sau:
 Idea: Tiền tài trợ mới = tiền tài trợ cũ + $(\text{growth}-6) * (20\% * \text{tiền tài trợ cũ})$
 Mvp: Tiền tài trợ mới = tiền tài trợ cũ + $((\text{operation}-6) + (\text{marketing}-6)) * (20\% * \text{tiền tài trợ cũ})$
 Bước 3: Chạy lại số liệu để cập nhật cho thông tin mới ở trên

