

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

ĐỀ THI – ĐÁP ÁN
LẬP TRÌNH HƯỚNG
ĐỐI TƯỢNG
(Bản thảo ngày 19/05/2022)

Nguyễn Tấn Trần Minh Khang
Võ Duy Nguyên

THÀNH PHỐ HỒ CHÍ MINH 2022

LỜI NÓI ĐẦU

Khi bắt đầu học lập trình trên máy tính, học sinh – sinh viên thường gặp khó khăn về tư duy giải quyết bài toán.

Chuyên đề Lưu đồ Thuật toán nhằm giúp cho các bạn học sinh – sinh viên trong việc hình thành Tư duy Giải quyết vấn đề trên máy tính.

Tài liệu này sẽ giúp học sinh – sinh viên có được đáp án đề nghị của tất cả các bài tập được nêu ra trong cuốn sách Chuyên đề Thuật toán.

Trong quá trình biên soạn, chúng tôi đã cố gắng để những sai sót, hạt sạn, các lỗi ít nhất có thể. Tuy nhiên, tập sách này chắc chắn không thể tránh khỏi những thiếu sót, vì thế, rất mong nhận được những góp ý quý báu của bạn đọc để tập sách ngày càng hoàn thiện hơn.

Hy vọng rằng tài liệu này sẽ mang lại những kiến thức bổ ích và những tư duy thiết thực cho học sinh – sinh viên và những người bắt đầu học lập trình trên máy tính.

Chân thành cảm ơn.

Thành phố Hồ Chí Minh, ngày 09 tháng 05 năm 2022.

Các tác giả

TS. Nguyễn Tấn Trần Minh Khang

Th.S Võ Duy Nguyên

MỤC LỤC

CHƯƠNG 01. ĐỀ - LỜI GIẢI ĐỀ NGHỊ HỌC KỲ 1 NĂM HỌC 2013 – 2014 2

01.01 ĐỀ THI.....	2
01.01.01 Câu 1 – 2 điểm.....	2
01.01.02 Câu 2 – 3 điểm.....	2
01.01.03 Câu 2 – 5 điểm.....	4
01.02 LỜI GIẢI ĐỀ NGHỊ.....	4
01.02.01 Lời giải đề nghị câu 1	4
01.02.02 Lời giải đề nghị câu 2	5
01.02.03 Lời giải đề nghị câu 3	9
01.02.03.1 Sơ đồ lớp sơ bộ.....	9
01.02.03.2 Sơ đồ lớp chi tiết.....	9
01.02.03.3 Khai báo lớp	9
01.02.03.4 Định nghĩa phương thức	11

CHƯƠNG 02. ĐỀ - LỜI GIẢI ĐỀ NGHỊ HỌC KỲ 2 NĂM HỌC 2013 - 2014 16

02.01 ĐỀ THI.....	16
02.01.01 Câu 1 – 2 điểm.....	16
02.01.02 Câu 2 – 3 điểm.....	16
02.01.03 Câu 2 – 5 điểm.....	18
02.02 LỜI GIẢI ĐỀ NGHỊ.....	18
02.02.01 Lời giải đề nghị câu 1	18
02.02.02 Lời giải đề nghị câu 2	19
02.02.03 Lời giải đề nghị câu 3	21
02.02.03.1 Sơ đồ lớp sơ bộ.....	21
02.02.03.2 Sơ đồ lớp chi tiết.....	21
02.02.03.3 Khai báo lớp	21
02.02.03.4 Định nghĩa phương thức	22

CHƯƠNG 03. ĐỀ - LỜI GIẢI ĐỀ NGHỊ HỌC KỲ 2 NĂM HỌC 2014 – 2015 26

03.01 ĐỀ THI.....	26
-------------------	----

03.01.01	Câu 1 – 2 điểm.....	26
03.01.02	Câu 2 – 3 điểm.....	26
03.01.03	Câu 1 – 5 điểm.....	26
03.02	ĐÁP ÁN	28
03.02.01	Lời giải đề nghị câu 1	28
03.02.02	Lời giải đề nghị câu 2	29
03.02.03	Lời giải đề nghị câu 3	31
03.02.03.1	Sơ đồ lớp sơ bộ	31
03.02.03.2	Sơ đồ lớp chi tiết.....	31
03.02.03.3	Khai báo lớp	31
03.02.03.4	Định nghĩa phương thức	31

CHƯƠNG 04. ĐỀ - LỜI GIẢI ĐỀ NGHỊ HỌC KỲ 1 NĂM HỌC 2015 – 2016 32

04.01	ĐỀ THI.....	32
04.01.01	Câu 1 – (2 điểm)	32
04.01.02	Câu 2 – (3 điểm)	32
04.01.03	Câu 3 – (5 điểm)	32
04.02	ĐÁP ÁN	34
04.02.01	Lời giải đề nghị câu 1	34
04.02.02	Lời giải đề nghị câu 2	35
04.02.03	Lời giải đề nghị câu 3	38
04.02.03.1	Sơ đồ lớp sơ bộ	38
04.02.03.2	Sơ đồ lớp chi tiết.....	38
04.02.03.3	Khai báo lớp	38
04.02.03.4	Định nghĩa phương thức	38

CHƯƠNG 05. ĐỀ - LỜI GIẢI ĐỀ NGHỊ HỌC KỲ 2 NĂM HỌC 2015 - 2016 39

05.01	ĐỀ THI.....	39
05.01.01	Câu 1 – (2 điểm)	39
05.01.02	Câu 2 – (3 điểm)	39
05.01.03	Câu 3 – (5 điểm)	39
05.02	ĐÁP ÁN	41
05.02.01	Lời giải đề nghị câu 1	41
05.02.02	Lời giải đề nghị câu 2	42
05.02.03	Lời giải đề nghị câu 3	44
05.02.03.1	Sơ đồ lớp sơ bộ	46

05.02.03.2	Sơ đồ lớp chi tiết.....	46
05.02.03.3	Khai báo lớp	46
05.02.03.4	Định nghĩa phương thức	46

CHƯƠNG 06. ĐỀ - LỜI GIẢI ĐỀ NGHỊ HỌC KỲ 1 NĂM HỌC 2016 – 2017 47

06.01	ĐỀ THI.....	47
06.01.01	Câu 1 – (3 điểm).....	47
06.01.02	Câu 2 – (2 điểm).....	47
06.01.03	Câu 3 – (5 điểm).....	48
06.02	ĐÁP ÁN	49
06.02.01	Lời giải đề nghị câu 1	49
06.02.02	Lời giải đề nghị câu 2	50
06.02.03	Lời giải đề nghị câu 3	52
06.02.03.1	Sơ đồ lớp sơ bộ.....	52
06.02.03.2	Sơ đồ lớp chi tiết.....	52
06.02.03.3	Khai báo lớp	52
06.02.03.4	Định nghĩa phương thức	54

CHƯƠNG 07. ĐỀ - LỜI GIẢI ĐỀ NGHỊ HỌC KỲ 2 NĂM HỌC 2016 – 2017 61

07.01	ĐỀ THI.....	61
07.01.01	Câu 1 – (2.5 điểm).....	61
07.01.02	Câu 2 – (2.5 điểm).....	62
07.01.03	Câu 3 – (5 điểm).....	62
07.02	ĐÁP ÁN	63
07.02.01	Lời giải đề nghị câu 1	63
07.02.02	Lời giải đề nghị câu 2	65
07.02.03	Lời giải đề nghị câu 3	68
07.02.03.1	Sơ đồ lớp sơ bộ.....	68
07.02.03.2	Sơ đồ lớp chi tiết.....	68
07.02.03.3	Khai báo lớp	68
07.02.03.4	Định nghĩa phương thức	68

CHƯƠNG 08. ĐỀ - LỜI GIẢI ĐỀ NGHỊ HỌC KỲ 3 NĂM HỌC 2016 - 2017 70

08.01	ĐỀ THI.....	70
08.01.01	Câu 1 – (2 điểm).....	70
08.01.02	Câu 2 – (3 điểm).....	70
08.01.03	Câu 3 – (5 điểm).....	70
08.02	ĐÁP ÁN.....	73
08.02.01	Lời giải đề nghị câu 1.....	73
08.02.02	Lời giải đề nghị câu 2.....	74
08.02.03	Lời giải đề nghị câu 3.....	77
08.02.03.1	Sơ đồ lớp sơ bộ.....	77
08.02.03.2	Sơ đồ lớp chi tiết.....	77
08.02.03.3	Khai báo lớp.....	77
08.02.03.4	Định nghĩa phương thức.....	77

CHƯƠNG 09. ĐỀ - LỜI GIẢI ĐỀ NGHỊ HỌC KỲ 2 NĂM HỌC 2017 – 2018 78

09.01	ĐỀ THI.....	78
09.01.01	Câu 1 – (2 điểm).....	78
09.01.02	Câu 2 – (3 điểm).....	78
09.01.03	Câu 3 – (5 điểm).....	78
09.02	ĐÁP ÁN.....	80
09.02.01	Lời giải đề nghị câu 1.....	80
09.02.02	Lời giải đề nghị câu 2.....	81
09.02.03	Lời giải đề nghị câu 3.....	84
09.02.03.1	Sơ đồ lớp sơ bộ.....	84
09.02.03.2	Sơ đồ lớp chi tiết.....	84
09.02.03.3	Khai báo lớp.....	84
09.02.03.4	Định nghĩa phương thức.....	84

CHƯƠNG 10. ĐỀ - LỜI GIẢI ĐỀ NGHỊ HỌC KỲ 1 NĂM HỌC 2017 - 2018 86

10.01	ĐỀ THI.....	86
10.01.01	Câu 1 – (2 điểm).....	86
10.01.02	Câu 2 – (3 điểm).....	86
10.01.03	Câu 3 – (5 điểm).....	87
10.02	ĐÁP ÁN.....	88
10.02.01	Lời giải đề nghị câu 1.....	88
10.02.02	Lời giải đề nghị câu 2.....	88
10.02.03	Lời giải đề nghị câu 3.....	91

10.02.03.1	Sơ đồ lớp sơ bộ	91
10.02.03.2	Sơ đồ lớp chi tiết.....	91
10.02.03.3	Khai báo lớp	91
10.02.03.4	Định nghĩa phương thức	91

CHƯƠNG 11. ĐỀ - LỜI GIẢI ĐỀ NGHỊ HỌC KỲ 1 NĂM HỌC 2018 – 2019 92

11.01	ĐỀ THI.....	92
11.01.01	Câu 1 – (2.5 điểm).....	92
11.01.02	Câu 2 – (2.5 điểm).....	92
11.01.03	Câu 3 – (5 điểm).....	92
11.02	ĐÁP ÁN.....	94
11.02.01	Lời giải đề nghị câu 1	94
11.02.02	Lời giải đề nghị câu 2	94
11.02.03	Lời giải đề nghị câu 3	98
11.02.03.1	Sơ đồ lớp sơ bộ	98
11.02.03.2	Sơ đồ lớp chi tiết.....	98
11.02.03.3	Khai báo lớp	98
11.02.03.4	Định nghĩa phương thức	98

CHƯƠNG 12. ĐỀ - LỜI GIẢI ĐỀ NGHỊ HỌC KỲ 2 NĂM HỌC 2019 - 2020 99

12.01	ĐỀ THI.....	99
12.01.01	Câu 1 – (2 điểm).....	99
12.01.02	Câu 2 – (3 điểm).....	99
12.01.03	Câu 3 – (5 điểm).....	99
12.02	ĐÁP ÁN.....	101
12.02.01	Lời giải đề nghị câu 1	101
12.02.02	Lời giải đề nghị câu 2	102
12.02.03	Lời giải đề nghị câu 3	104
12.02.03.1	Sơ đồ lớp sơ bộ	104
12.02.03.2	Sơ đồ lớp chi tiết.....	104
12.02.03.3	Khai báo lớp	104
12.02.03.4	Định nghĩa phương thức	104

CHƯƠNG 01. ĐỀ - LỜI GIẢI ĐỀ NGHỊ HỌC KỲ 1 NĂM HỌC 2013 – 2014

01.01 ĐỀ THI

01.01.01 Câu 1 – 2 điểm

a. Phân biệt các phạm vi truy cập **private**, **protected**, **public** và cho ví dụ minh họa.

b. Nêu khái niệm về sự kế thừa và những ưu điểm của kế thừa trong việc lập trình. Cho ví dụ minh họa.

01.01.02 Câu 2 – 3 điểm

a. Xét đoạn chương trình sau:

```
00001. #include <iostream>
00002. using namespace std;
00003. class A
00004. {
00005. public:
00006.     A()
00007.     {
00008.         cout << "Constructing A ";
00009.     }
00010.     ~A()
00011.     {
00012.         cout << "Destructing A ";
00013.     }
00014. };
00015.
00016. class B : public A
00017. {
00018. public:
00019.     B()
00020.     {
00021.         cout << "Constructing B ";
00022.     }
```

```

00023.     ~B()
00024.     {
00025.         cout << "Destructing B ";
00026.     }
00027. };
00028.
00029. int main()
00030. {
00031.     B b1;
00032.     return 0;
00033. }

```

Hãy cho biết kết quả xuất ra màn hình khi thực thi đoạn chương trình trên. Giải thích ngắn gọn tại sao có kết quả đó.

b. Xét đoạn chương trình sau:

```

00034. #include <iostream>
00035. using namespace std;
00036. class A
00037. {
00038. private:
00039.     int x;
00040. public:
00041.     A(int t)
00042.     {
00043.         x = t;
00044.     }
00045.     static void f()
00046.     {
00047.         cout << x;
00048.     }
00049.     int f2()
00050.     {
00051.         return x;
00052.     }
00053. };
00054.
00055. void main()
00056. {
00057.     A a;
00058.     f2(a);
00059. }

```

Cho biết đoạn chương trình trên khi biên dịch có lỗi xảy ra hay không? Nếu có lỗi, hãy chỉ ra các lỗi đó và sửa lỗi để chương trình có thể thực thi được.

01.01.03 Câu 2 – 5 điểm

Giả sử Trường ĐH CNTT TP.HCM đào tạo sinh viên theo 2 hệ là hệ cao đẳng và hệ đại học. Thông tin cần quản lý của một sinh viên cao đẳng bao gồm: mã số sinh viên, họ tên, địa chỉ, tổng số tín chỉ, điểm trung bình, điểm thi tốt nghiệp. Thông tin cần quản lý của một sinh viên đại học bao gồm: mã số sinh viên, họ tên, địa chỉ, tổng số tín chỉ, điểm trung bình, tên luận văn, điểm luận văn.

Cách xét tốt nghiệp của sinh viên mỗi hệ là khác nhau:

- Sinh viên hệ cao đẳng tốt nghiệp khi có tổng số tín chỉ từ 120 trở lên, điểm trung bình từ 5 trở lên và điểm thi tốt nghiệp phải đạt từ 5 trở lên.
- Sinh viên hệ đại học tốt nghiệp khi có tổng số tín chỉ từ 170 trở lên, điểm trung bình từ 5 trở lên và phải bảo vệ luận văn với điểm số đạt được từ 5 điểm trở lên.

Bạn hãy đề xuất thiết kế các lớp đối tượng cần thiết để quản lý danh sách các sinh viên của Trường và hỗ trợ xét tốt nghiệp cho các sinh viên theo tiêu chí đặt ra như trên.

Hãy viết chương trình bằng C++ cho phép thực hiện các yêu cầu sau:

- Nhập vào danh sách sinh viên, có thể sử dụng string cho các chuỗi ký tự.
- Cho biết số lượng sinh viên đủ điều kiện tốt nghiệp?
- Cho biết sinh viên đại học nào có điểm trung bình cao nhất?

01.02 LỜI GIẢI ĐỀ NGHỊ

01.02.01 Lời giải đề nghị câu 1

a. Phân biệt các phạm vi truy cập **private**, **protected**, **public** và cho ví dụ minh họa.

Lời giải đề nghị

- Các thuộc tính và phương thức được khai báo bên trong phạm vi **private** của một lớp thì chỉ được phép truy xuất từ bên trong lớp và không được quyền truy xuất từ bên ngoài lớp. Hơn nữa, các thuộc tính và phương thức này không cho phép kế thừa ở lớp dẫn xuất.

- Các thuộc tính và phương thức được khai báo bên trong phạm vi **protected** của một lớp thì **chỉ được phép truy xuất từ bên trong lớp và không được quyền truy xuất từ bên ngoài lớp**. Hơn nữa, các thuộc tính và phương thức này **cho phép kế thừa ở lớp dẫn xuất**.
- Các thuộc tính và phương thức được khai báo bên trong phạm vi **public** của một lớp thì **được phép truy xuất từ bên trong lớp, bên ngoài lớp**. Hơn nữa, các thuộc tính và phương thức này **cho phép kế thừa ở lớp dẫn xuất**.

b. Nêu khái niệm về sự kế thừa và những ưu điểm của kế thừa trong việc lập trình. Cho ví dụ minh họa.

Lời giải đề nghị

- Kế thừa là một đặc điểm của ngôn ngữ lập trình hướng đối tượng dùng để biểu diễn mối quan hệ đặc biệt hóa – tổng quát hóa giữa các lớp.
- Ưu điểm của kế thừa trong lập trình
 - + Kế thừa cho phép xây dựng lớp mới từ lớp đã có.
 - + Kế thừa cho phép tổ chức các lớp chia sẻ mã chương trình chung, nhờ vậy có thể dễ dàng sửa chữa, nâng cấp hệ thống.
 - + Trong C++, kế thừa còn định nghĩa sự tương thích, nhờ đó ta có cơ chế chuyển kiểu tự động.
- Ví dụ minh họa

```
00060. class CTamGiac
00061. {
00062.     ...
00063. };
00064.
00065. class CTamGiacCan:public CTamGiac
00066. {
00067.     ...
00068. };
```

01.02.02 Lời giải đề nghị câu 2

a. Xét đoạn chương trình sau:

```
00069. #include <iostream>
00070. using namespace std;
00071. class A
00072. {
00073. public:
```

```

00074.    A()
00075.    {
00076.        cout << "Constructing A ";
00077.    }
00078.    ~A()
00079.    {
00080.        cout << "Destructing A ";
00081.    }
00082. };
00083.
00084. class B : public A
00085. {
00086. public:
00087.     B()
00088.     {
00089.         cout << "Constructing B ";
00090.     }
00091.     ~B()
00092.     {
00093.         cout << "Destructing B ";
00094.     }
00095. };
00096.
00097. int main()
00098. {
00099.     B b1;
00100.     return 0;
00101. }

```

Hãy cho biết kết quả xuất ra màn hình khi thực thi đoạn chương trình trên.

Lời giải đề nghị

- Kết quả xuất ra màn hình.

Constructing A Constructing B Destructing B Destructing A

- Giải thích ngắn gọn tại sao có kết quả đó.
 - + Khi chương trình thực hiện đến dòng lệnh **B b1;** thì đối tượng **b1** sẽ gọi thực hiện phương thức thiết lập của lớp đối tượng cơ sở của lớp B trước rồi sau đó mới gọi thực hiện phương thức thiết lập của lớp đối tượng B.
 - + Phạm vi hoạt động của đối tượng **b1** bắt đầu từ dòng 31 kết thúc tại dòng 33. Khi đối tượng **b1** hết phạm vi hoạt

động thì phương thức phá hủy của lớp đối tượng B (lớp dẫn xuất) được gọi thực hiện trước, sau đó phương thức phá hủy của lớp đối tượng A (lớp cơ sở) mới được gọi thực hiện.

b. Xét đoạn chương trình sau:

```
00102. #include <iostream>
00103. using namespace std;
00104. class A
00105. {
00106. private:
00107.     int x;
00108. public:
00109.     A(int t)
00110.     {
00111.         x = t;
00112.     }
00113.     static void f()
00114.     {
00115.         cout << x;
00116.     }
00117.     int f2()
00118.     {
00119.         return x;
00120.     }
00121. };
00122.
00123. void main()
00124. {
00125.     A a;
00126.     f2(a);
00127. }
```

Cho biết đoạn chương trình trên khi biên dịch có lỗi xảy ra hay không? Nếu có lỗi, hãy chỉ ra các lỗi đó và sửa lỗi để chương trình có thể thực thi được.

- Đoạn chương trình trên khi biên dịch có lỗi xảy ra với các lỗi là:
 - + Một phương thức static của một lớp thì không thể truy xuất đến non-static member. Phương thức này được đề nghị định nghĩa lại như sau:

```
00128. void A::f()
00129. {
00130.     cout << x;
```

```
00131. }
```

+ Không có phương thức thiết lập mặc định, ta cần định nghĩa thêm phương thức thiết lập mặc định.

```
00132. A::A()
```

```
00133. {
```

```
00134.     x = 0;
```

```
00135. }
```

+ Câu lệnh `f2(a)` là sai cú pháp và cần được viết lại như sau: `a.f2()`.

– Chương trình hoàn thiện sau khi sửa lỗi

```
00136. #include <iostream>
```

```
00137. using namespace std;
```

```
00138. class A
```

```
00139. {
```

```
00140. private:
```

```
00141.     int x;
```

```
00142. public:
```

```
00143.     A()
```

```
00144.     {
```

```
00145.         x = 0;
```

```
00146.     }
```

```
00147.     A(int t)
```

```
00148.     {
```

```
00149.         x = t;
```

```
00150.     }
```

```
00151.     void f()
```

```
00152.     {
```

```
00153.         cout << x;
```

```
00154.     }
```

```
00155.     int f2()
```

```
00156.     {
```

```
00157.         return x;
```

```
00158.     }
```

```
00159. };
```

```
00160.
```

```
00161. int main()
```

```
00162. {
```

```
00163.     A a;
```

```
00164.     a.f2();
```

```
00165.     return 1;
```

```
00166. }
```

01.02.03 Lời giải đề nghị câu 3

01.02.03.1 Sơ đồ lớp sơ bộ

01.02.03.2 Sơ đồ lớp chi tiết

01.02.03.3 Khai báo lớp

– Khai báo lớp sinh viên

```
00167. class CSinhVien
00168. {
00169.     protected:
00170.         string MaSo;
00171.         string HoTen;
00172.         string DiaChi;
00173.         int TongSoTinChi;
00174.         float DiemTrungBinh;
00175.     public:
00176.         virtual void Nhap();
00177.         virtual void Xuat();
00178.         virtual int ktTotNghiep()==0;
00179.         virtual int ktDaiHoc()==0;
00180.         virtual float GetDiemTrungBinh();
00181.     };
```

– Khai báo lớp sinh viên cao đẳng

```
00182. class CSinhVienCaoDang : public CSinhVien
00183. {
00184.     protected:
00185.         float DiemThiTotNghiep;
00186.     public:
00187.         void Nhap();
00188.         void Xuat();
00189.         int ktTotNghiep();
00190.         int ktDaiHoc();
00191.         float GetDiemTrungBinh();
00192.     };
```

– Khai báo lớp sinh viên đại học

```
00193. class CSinhVienDaiHoc : public CSinhVien
00194. {
00195.     protected:
00196.         string TenLuanVan;
00197.         float DiemLuanVan;
00198.     public:
00199.         void Nhap();
```



```
00200.    void Xuat();
00201.    int ktTotNghiep();
00202.    int ktDaiHoc();
00203.    float GetDiemTrungBinh();
00204.    };
```

– Khai báo lớp trường

```
00205.    class CTruong
00206.    {
00207.    protected:
00208.        CSinhVien* ds[1000];
00209.        int n;
00210.    public:
00211.        void Nhap();
00212.        void Xuat();
00213.        int DemTotNghiep();
00214.        CSinhVien* DaiHocDau();
00215.        CSinhVien* DaiHocGioiNhat();
00216.    };
```

01.02.03.4 Định nghĩa phương thức

– Định nghĩa phương thức lớp sinh viên

```
00217. void CSinhVien::Nhap()
00218. {
00219.     cout << "\nNhap ma so sinh vien: ";
00220.     cin.ignore();
00221.     getline(cin, MaSo);
00222.
00223.     cout << "\nNhap ho ten: ";
00224.     getline(cin, HoTen);
00225.
00226.     cout << "\nNhap dia chi: ";
00227.     getline(cin, DiaChi);
00228.
00229.     cout << "\nNhap tong so tin chi: ";
00230.     cin >> TongSoTinChi;
00231.
00232.     cout << "\nNhap DiemTrungBinh: ";
00233.     cin >> DiemTrungBinh;
00234. }
00235.
00236. void CSinhVien::Xuat()
00237. {
00238.     cout << "\nSinh vien dai hoc.";
00239.     cout << "\nMa so sinh vien: " << MaSo;
00240.     cout << "\nHo ten: " << HoTen;
00241.     cout << "\nDia chi: " << DiaChi;
00242.     cout << "\nTong so tin chi: " << TongSoTinChi;
00243.     cout << "\nDiem trung binh: "
00244.         << DiemTrungBinh;
00245. }
00246.
00247. float CSinhVien::GetDiemTrungBinh()
00248. {
00249.     return DiemTrungBinh;
00250. }
```

– Định nghĩa phương thức lớp sinh viên cao đẳng

```
00251. void CSinhVienCaoDang::Nhap()
00252. {
00253.     CSinhVien::Nhap();
00254.     cout << "\nNhap diem thi tot nghiep: ";
```

```

00255.     cin >> DiemThiTotNghiep;
00256. }
00257.
00258. void CSinhVienCaoDang::Xuat()
00259. {
00260.     CSinhVien::Xuat();
00261.     cout << "\nDiem thi tot nghiep: "
00262.           << DiemThiTotNghiep;
00263.     cout << "\n\n";
00264. }
00265.
00266. int CSinhVienCaoDang::KtTotNghiep()
00267. {
00268.     if(TongSoTinChi >= 120 &&
00269.        DiemTrungBinh >= 5 &&
00270.        DiemThiTotNghiep >= 5)
00271.         return 1;
00272.     return 0;
00273. }
00274.
00275. int CSinhVienCaoDang::KtDaiHoc()
00276. {
00277.     return 0;
00278. }

```

– Định nghĩa phương thức lớp sinh viên đại học

```

00279. void CSinhVienDaiHoc::Nhap()
00280. {
00281.     CSinhVien::Nhap();
00282.     cout << "Nhap ten luan van: ";
00283.     cin.ignore();
00284.     getline(cin, TenLuanVan);
00285.     cout << "Nhap diem luan van: ";
00286.     cin >> DiemLuanVan;
00287. }
00288. void CSinhVienDaiHoc::Xuat()
00289. {
00290.     CSinhVien::Xuat();
00291.     cout << "\nTen luan van: "<< TenLuanVan;
00292.     cout << "\nDiem luan van: "<< DiemLuanVan;
00293. }
00294.
00295. int CSinhVienDaiHoc::ktTotNghiep()

```

```

00296.  {
00297.      if ( TongSoTinChi >= 170 &&
00298.          DiemTrungBinh >= 5 &&
00299.          DiemLuanVan >= 5)
00300.          return 1;
00301.      return 0;
00302.  }
00303.
00304.  int CSinhVienDaiHoc::ktDaiHoc()
00305.  {
00306.      return 1;
00307.  }

```

– Định nghĩa phương thức lớp trường

```

00308. void CTruong::Nhap()
00309. {
00310.     cout << "Nhap so sinh vien: ";
00311.     cin >> n;
00312.     for (int i = 0; i < n; i++)
00313.     {
00314.         int loai;
00315.         cout << "Nhap loai sinh vien
00316.             (1.Cao Dang, 2.Dai Hoc): ";
00317.         cin >> loai;
00318.         switch (loai)
00319.         {
00320.             case 1:
00321.                 ds[i] = new CSinhVienCaoDang;
00322.                 break;
00323.             case 2:
00324.                 ds[i] = new CSinhVienDaiHoc;
00325.                 break;
00326.         }
00327.         ds[i]->Nhap();
00328.     }
00329. }
00330.
00331. void CTruong::Xuat()
00332. {
00333.     cout << "\Danh sach: \n";
00334.     for (int i = 0; i < n; i++)
00335.         ds[i]->Xuat();
00336. }

```

```

00337.
00338. int CTruong::DemTotNghiep()
00339. {
00340.     int dem = 0;
00341.     for (int i = 0; i < n; i++)
00342.         if (ds[i]->ktTotNghiep())
00343.             dem++;
00344.     return dem;
00345. }
00346.
00347. CSinhVien* CTruong::DaiHocDau()
00348. {
00349.     for (int i = 0; i < n; i++)
00350.         if (ds[i]->ktDaiHoc())
00351.             return ds[i];
00352.     return NULL;
00353. }
00354.
00355. CSinhVien* CTruong::DaiHocGioiNhat()
00356. {
00357.     CSinhVien* lc = DaiHocDau();
00358.     if (lc==NULL)
00359.         return NULL;
00360.     for (int i = 0; i < n; i++)
00361.     {
00362.         float x = lc->GetDiemTrungBinh();
00363.         float y = ds[i]->GetDiemTrungBinh();
00364.         if(x<y && ds[i]->ktDaiHoc())
00365.             lc = ds[i];
00366.     }
00367.     return lc;
00368. }

```

– Định nghĩa hàm main

```

00369. int main()
00370. {
00371.     CTruong UIT;
00372.     UIT.Nhap();
00373.     UIT.Xuat();
00374.
00375.     int dem = UIT.DemTotNghiep();
00376.     cout << "\n\nSo sinh vien du dieu kien
00377.         tot nghiep la: " << dem << endl;

```

```
00378.  
00379.    CSinhVien* kq = UIT.DaiHocGioiNhat();  
00380.    if (kq == NULL)  
00381.        cout << "\nKhong co sinh vien  
00382.                dai hoc.";  
00383.    else  
00384.    {  
00385.        cout << "\nSinh vien dai hoc co  
00386.                DiemTrungBinh lon nhat la: ";  
00387.        kq->Xuat();  
00388.    }  
00389.    return 1;  
00390. }
```

CHƯƠNG 02. ĐỀ - LỜI GIẢI ĐỀ NGHỊ HỌC KỲ 2 NĂM HỌC 2013 - 2014

02.01 ĐỀ THI

02.01.01 Câu 1 – 2 điểm

a. Nêu khái niệm Constructor và Destructor. Phân biệt Constructor mặc định và Constructor khác.

b. Phân biệt các kiểu kế thừa **private**, **protected**, **public**.

02.01.02 Câu 2 – 3 điểm

a. Xét lớp phân số được khai báo như sau:

```
00391. class PhanSo
00392. {
00393.     private:
00394.         int ts, ms;
00395.     public:
00396.         PhanSo(int ts = 0, int ms = 1);
00397.         PhanSo operator +(PhanSo);
00398. };
```

Hãy cho biết trong các dòng lệnh sau đây, dòng nào có lỗi xảy ra, giải thích và sửa lỗi nếu có:

```
00399. PhanSo a, b(3, 4), c(2, 5);
00400. a = b + c;
00401. a = b + 3;
00402. a = 5 + c;
```

b. Xét đoạn chương trình sau:

```
00403. #include <iostream>
00404. using namespace std;
00405. class A
00406. {
00407.     public:
```

```

00408.     A()
00409.     {
00410.         cout << "\nHam dung mac dinh lop A ";
00411.     }
00412.     ~A()
00413.     {
00414.         cout << "\nHam huy lop A ";
00415.     }
00416. };
00417.
00418. class B
00419. {
00420. public:
00421.     B()
00422.     {
00423.         cout << "\nHam dung mac dinh lop B ";
00424.     }
00425.     ~B()
00426.     {
00427.         cout << "\nHam huy lop B ";
00428.     }
00429. };
00430.
00431. class C : public A, private B
00432. {
00433. public:
00434.     C()
00435.     {
00436.         cout << "\nHam dung mac dinh lop C ";
00437.     }
00438.     ~C()
00439.     {
00440.         cout << "\nHam huy lop C ";
00441.     }
00442. };
00443.
00444. void main()
00445. {
00446.     C c;
00447. }

```

Hãy cho biết kết quả xuất ra màn hình khi thực thi đoạn chương trình trên. Giải thích ngắn gọn tại sao có kết quả đó.

02.01.03 Câu 2 – 5 điểm

Mùa hè lại đến, công viên văn hóa Đầm Sen hân hoan đón chào các em thiếu nhi, các bạn học sinh, sinh viên và toàn thể quý khách đến tham quan và tham dự các trò chơi kì thú và đầy hấp dẫn. Giả sử trên mỗi chiếc vé mà công viên phát hành, đều có ghi lại mã vé (chuỗi), họ tên người chủ vé (chuỗi), năm sinh của người đó (số nguyên) và số trò chơi mà người đó tham dự (số nguyên). Để phục vụ tối đa cho lợi ích khách hàng, công viên phát hành **2 loại vé là vé trọn gói và vé từng phần**. Giá vé trọn gói là 200.000 VNĐ. Người chơi mua vé trọn gói có thể chơi tất cả 30 trò chơi có trong công viên. Đối với vé từng phần, giá vé là 70.000 VNĐ (giá vé vào cổng), ngoài ra, khi người chơi tham dự một trò chơi nào thì cần trả thêm 20.000 VNĐ cho trò chơi đó.

Hãy viết chương trình bằng C++ cho phép thực hiện các chức năng sau:

- Nhập vào danh sách các vé.
- Tính tổng tiền vé mà công viên thu được.
- Hãy cho biết, có bao nhiêu vé đã bán là vé từng phần.

Yêu cầu:

Sử dụng tính chất kế thừa và đa hình. Vẽ sơ đồ lớp: mô tả các lớp, các thuộc tính, các phương thức và mối liên hệ các lớp (2.0đ) Lập trình các chức năng được yêu cầu (3đ).

02.02 LỜI GIẢI ĐỀ NGHỊ

02.02.01 Lời giải đề nghị câu 1

a. Nêu khái niệm Constructor và Destructor. Phân biệt Constructor mặc định và Constructor khác.

- Khái niệm Constructor: là một phương thức của lớp dùng để tạo dựng một đối tượng mới. Hệ điều hành sẽ cấp phát bộ nhớ cho đối tượng và gọi đến hàm tạo. Hàm tạo sẽ khởi gán giá trị cho các thuộc tính của đối tượng.
- Khái niệm Destructor: Hàm hủy (destructor) là một phương thức của lớp dùng để thực hiện một số công việc có tính “dọn dẹp” trước khi đối tượng được hủy bỏ.
- Phân biệt Constructor mặc định và Constructor khác: Constructor mặc định (default constructor) là constructor được gọi khi thể hiện được khai báo mà không có đối số nào được cung cấp.

Ngược lại, nếu tham số được cung cấp tại khai báo thể hiện, chương trình sẽ gọi constructor khác theo cơ chế overload.

b. Phân biệt các kiểu kế thừa `private`, `protected`, `public`.

	private	protected	public
private			
protected	private	protected	protected
public	private	protected	public

02.02.02 Lời giải đề nghị câu 2

a. Xét lớp phân số được khai báo như sau:

```
00448. class PhanSo
00449. {
00450.     private:
00451.         int ts, ms;
00452.     public:
00453.         PhanSo(int ts = 0, int ms = 1);
00454.         PhanSo operator +(PhanSo);
00455. };
```

Hãy cho biết trong các dòng lệnh sau đây, dòng nào có lỗi xảy ra, giải thích và sửa lỗi nếu có:

```
00456. PhanSo a, b(3, 4), c(2, 5);
00457. a = b + c;
00458. a = b + 3;
00459. a = 5 + c;
```

Chương trình trên dòng lệnh `a=5+c;` có lỗi và sửa như sau:

```
00460. class PhanSo
00461. {
00462.     private:
00463.         int ts, ms;
00464.     public:
00465.         PhanSo(int ts = 0, int ms = 1);
00466.         friend PhanSo operator+(PhanSo,PhanSo);
00467. };
```

b. Xét đoạn chương trình sau:

```
00468. #include <iostream>
00469. using namespace std;
00470. class A
```

```

00471.  {
00472.  public:
00473.      A()
00474.      {
00475.          cout << "\nHam dung mac dinh lop A ";
00476.      }
00477.      ~A()
00478.      {
00479.          cout << "\nHam huy lop A ";
00480.      }
00481.  };
00482.
00483.  class B
00484.  {
00485.  public:
00486.      B()
00487.      {
00488.          cout << "\nHam dung mac dinh lop B ";
00489.      }
00490.      ~B()
00491.      {
00492.          cout << "\nHam huy lop B ";
00493.      }
00494.  };
00495.
00496.  class C : public A, private B
00497.  {
00498.  public:
00499.      C()
00500.      {
00501.          cout << "\nHam dung mac dinh lop C ";
00502.      }
00503.      ~C()
00504.      {
00505.          cout << "\nHam huy lop C ";
00506.      }
00507.  };
00508.
00509.  void main()
00510.  {
00511.      C c;
00512.  }

```

Hãy cho biết kết quả xuất ra màn hình khi thực thi đoạn chương trình trên. Giải thích ngắn gọn tại sao có kết quả đó.

02.02.03 Lời giải đề nghị câu 3

02.02.03.1 Sơ đồ lớp sơ bộ

02.02.03.2 Sơ đồ lớp chi tiết

02.02.03.3 Khai báo lớp

– Khai báo lớp vé

```
00513. class Cve
00514. {
00515.     protected:
00516.         string MaSo;
00517.         string HoTen;
00518.         int NamSinh;
00519.         int SoTroChoi;
00520.     public:
00521.         virtual void Nhap();
00522.         virtual void Xuat();
00523.         virtual int TinhTien()=0;
00524.         virtual int ktVeTungPhan()=0;
00525. };
```

– Khai báo lớp vé trọn gói

```
00526. class CveTronGoi : public Cve
00527. {
00528.     public:
00529.         void Nhap();
00530.         void Xuat();
00531.         int TinhTien();
00532.         int ktVeTungPhan();
00533. };
```

– Khai báo lớp vé từng phần

```
00534. class CveTungPhan : public Cve
00535. {
00536.     public:
00537.         void Nhap();
00538.         void Xuat();
00539.         int TinhTien();
00540.         int ktVeTungPhan();
00541. };
```

– Khai báo lớp công viên

```
00542. class CCongVien
00543. {
00544. protected:
00545.     int n;
00546.     CVe* ds[100];
00547. public:
00548.     void Nhap();
00549.     void Xuat();
00550.     int TongTien();
00551.     int DemVeTungPhan();
00552. };
```

02.02.03.4 Định nghĩa phương thức

– Định nghĩa phương thức lớp vé

```
00553. void CVe::Nhap()
00554. {
00555.     cin.ignore();
00556.     cout << "Nhap ma so ve: ";
00557.     getline(cin, MaSo);
00558.
00559.     cout << "Nhap ho ten nguoi mua: ";
00560.     getline(cin, HoTen);
00561.
00562.     cout << "Nhap nam sinh: ";
00563.     cin >> NamSinh;
00564. }
00565.
00566. void CVe::Xuat()
00567. {
00568.     cout << "\nMa ve: " << MaSo;
00569.     cout << "\nHo ten nguoi mua: " << HoTen;
00570.     cout << "\nNam sinh: " << NamSinh;
00571.     cout << "\nSo tro choi: " << SoTroChoi;
00572. }
```

– Định nghĩa phương thức lớp vé trọn gói

```
00573. void CVeTronGoi::Nhap()
00574. {
00575.     CVe::Nhap();
00576.     SoTroChoi = 30;
00577.     GiaTien = 200000;
00578. }
```

```

00579.
00580. void CVeTronGoi::Xuat()
00581. {
00582.     CVe::Xuat();
00583.     cout << "\n Gia tien: " << GiaTien;
00584.     cout << "\n So tro choi: " << SoTroChoi;
00585. }
00586.
00587. int CVeTronGoi::TinhTien()
00588. {
00589.     return GiaTien;
00590. }
00591.
00592. int CVeTronGoi::ktVeTungPhan()
00593. {
00594.     return 0;
00595. }

```

– Định nghĩa phương thức lớp vé từng phần

```

00596. void CVeTungPhan::Nhap()
00597. {
00598.     CVe::Nhap();
00599.     GiaTien = 70000;
00600.     cout << "\n Nhap so tro choi: ";
00601.     cin >> SoTroChoi;
00602. }
00603.
00604. void CVeTungPhan::Xuat()
00605. {
00606.     CVe::Xuat();
00607.     cout << "\n Gia tien: " << GiaTien;
00608.     cout << "\n So tro choi: " << SoTroChoi;
00609. }
00610.
00611. int CVeTungPhan::TinhTien()
00612. {
00613.     return GiaTien + SoTroChoi * 20000;
00614. }
00615.
00616. int CVeTungPhan::ktVeTungPhan()
00617. {
00618.     return 1;
00619. }

```

– Định nghĩa phương thức lớp công viên

```
00620. void CCongVien::Nhap()
00621. {
00622.     cout << "Nhap so ve ban: ";
00623.     cin >> n;
00624.     for (int i = 0; i < n; i++)
00625.     {
00626.         int loai;
00627.         cout << "Nhap loai ve (1. Ve tron
00628.                 goi, 2. Ve tung phan): ";
00629.         cin >> loai;
00630.         switch (loai)
00631.         {
00632.             case 1:
00633.                 ds[i] = new CVeTronGoi;
00634.                 break;
00635.             case 2:
00636.                 ds[i] = new CVeTungPhan;
00637.                 break;
00638.         }
00639.         ds[i]->Nhap();
00640.     }
00641. }
00642.
00643. void CCongVien::Xuat()
00644. {
00645.     cout << "\nSo ve: " << n;
00646.     for (int i = 0; i < n; i++)
00647.         ds[i]->Xuat();
00648. }
00649.
00650. int CCongVien::TongTien()
00651. {
00652.     int s = 0;
00653.     for (int i = 0; i < n; i++)
00654.         s += ds[i]->TinhTien();
00655.     return s;
00656. }
00657.
00658. int CCongVien::DemVeTungPhan()
00659. {
00660.     int dem = 0;
00661.     for (int i = 0; i < n; i++)
```

```
00662.         if (ds[i]->ktVeTungPhan())
00663.             dem++;
00664.         return dem;
00665.     }
```

– Định nghĩa hàm main

```
00666. int main()
00667. {
00668.     CCongVien a;
00669.     a.Nhap();
00670.     a.Xuat();
00671.
00672.     cout<<"\nTong tien ve:"<<a.TongTien();
00673.     cout << "\nSo ve tung phan ban la: "
00674.         << a.DemVeTungPhan();
00675.     return 1;
00676. }
```


CHƯƠNG 03. ĐỀ - LỜI GIẢI ĐỀ NGHỊ HỌC KỲ 2 NĂM HỌC 2014 – 2015

03.01 ĐỀ THI

03.01.01 Câu 1 – 2 điểm

Phân biệt các kiểu kế thừa private, protected, public

03.01.02 Câu 2 – 3 điểm

Xây dựng lớp đa thức bậc nhất để thể hiện các đa thức bậc nhất có dạng:

$$F(x) = ax + b \text{ (} a \text{ luôn khác } 0\text{)}$$

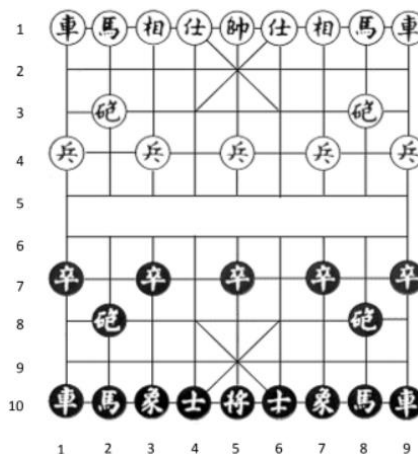
Xây dựng các phương thức:

- a. Phương thức cho phép xác định giá trị của đa thức ứng với $x = x_0$ (tính $F(x_0)$)
- b. Phương thức trả về nghiệm đa thức bậc 1 (nghĩa là $F(x) = 0$)
- c. Phép toán cộng (operator +) để cộng hai đa thức bậc nhất

03.01.03 Câu 1 – 5 điểm

Xây dựng chương trình mô phỏng trò chơi cờ tướng với các mô tả như sau: Bàn cờ là một hình chữ nhật do 9 đường dọc và 10 đường ngang cắt nhau vuông góc tại 90 điểm hợp thành. Một khoảng trống gọi là sông (hay hà) nằm ngang giữa bàn cờ, chia bàn cờ thành hai phần đối xứng bằng nhau.

Mỗi bên có một cung Tướng hình vuông (Cung) do 4 ô hợp thành tại các đường dọc 4, 5, 6 kể từ đường ngang cuối của mỗi bên, trong 4 ô này có vẽ hai đường chéo xuyên qua.



Vị trí các quân ban đầu trên bàn cờ

Tướng trắng (1,5)	Tướng đen (10,5)
Sĩ trắng (1,4) và (1,6)	Sĩ đen (10,4) và (10,6)
Tượng trắng (1,3) và (1,7)	Tượng đen (10,3) và (10,7)
Mã trắng (1,2) và (1,8)	Mã đen (10,2) và (10,8)
Xe trắng (1,1) và (1,9)	Xe đen (10,1) và (10,9)
Pháo trắng (3,2) và (3,8)	Pháo đen (8,2) và (8,8)
Tốt trắng (4,1), (4,3), (4,5), (4,7) và (4,9)	Tốt đen (7,1), (7,3), (7,5), (7,7) và (7,9)

Luật chơi: Quân cờ được di chuyển theo luật sau:

1. Tướng: Đi từng ô một, đi ngang hoặc dọc. Tướng luôn luôn phải ở trong phạm vi cung và không được ra ngoài. Cung tức là hình vuông 2X2 được đánh dấu bằng đường chéo hình chữ X

2. Sĩ: Đi chéo 1 ô mỗi nước. Sĩ luôn luôn phải ở trong cung như Tướng.

3. Tượng: Đi chéo 2 ô (ngang 2 và dọc 2) cho mỗi nước đi. Tượng chỉ được phép ở một bên của bàn cờ, không được di chuyển sang nửa bàn cờ của đối phương. Nước đi của tượng sẽ không hợp lệ khi có một quân cờ nằm chặn giữa đường đi.

4. Xe: Đi ngang hay dọc trên bàn cờ miễn là đừng bị quân khác cản đường từ điểm đi đến điểm đến.

5. Mã: Đi ngang 2 ô và dọc 1 ô (hay dọc 2 ô và ngang 1 ô) cho mỗi nước đi. Nếu có quân nằm ngay bên cạnh mã và cản đường ngang 2 (hay đường dọc 2), mã bị cản không được đi đường đó.

6. Pháo: Đi ngang và dọc giống như xe. Điểm khác biệt là nếu pháo muốn ăn quân, pháo phải nhảy qua đúng 1 quân nào đó. Khi không ăn quân, tất cả những điểm từ chỗ đi đến chỗ đến phải không có quân cản.

7. Tốt: đi một ô mỗi nước. Nếu tốt chưa vượt qua sông, nó chỉ có thể đi thẳng tiến. Khi đã vượt sông rồi, tốt có thể đi ngang 1 nước hay đi thẳng tiến 1 bước mỗi nước.

Áp dụng kiến thức lập trình hướng đối tượng (kế thừa, đa hình) thiết kế sơ đồ chi tiết các lớp đối tượng (1.5đ), khai báo và định nghĩa các lớp gồm thuộc tính và phương thức (1.5đ) để thực hiện các yêu cầu sau:

1. Tạo bàn cờ ban đầu (với các mô tả như trên) (1đ)
2. Yêu cầu người dùng chọn một quân cờ, xuất cách đi của quân cờ tương ứng (1đ)

Lưu ý: Trong trường hợp sinh viên không chơi trò chơi này trước đây thì phải đọc kỹ thông tin về trò chơi trên (các thông tin trên đủ để sinh viên thực hiện các yêu cầu của đề thi) và nghiêm túc làm bài.

03.02 ĐÁP ÁN

03.02.01 Lời giải đề nghị câu 1

Phân biệt các kiểu kế thừa private, protected, public.

Lời giải đề nghị

	private	protected	public
private			
protected	private	protected	protected
public	private	protected	public

- Kế thừa public: Lớp con kế thừa public từ lớp cha thì các thành phần protected của lớp cha trở thành protected của lớp con, các thành phần public của lớp cha trở thành public của lớp con. Nói cách khác mỗi thao tác của lớp cha được kế thừa xuống lớp con. Vì vậy ta có thể sử dụng thao tác của lớp cha cho đối tượng thuộc lớp con.
- Kế thừa protected: Lớp con kế thừa protected từ lớp cha thì các thành phần protected và public của lớp cha trở thành protected của lớp con. Nói cách khác mỗi thao tác của lớp cha đều được lớp con bảo vệ. Vì vậy ta có thể sử dụng thao tác của lớp cha cho đối tượng thuộc lớp con.
- Kế thừa private: Lớp con kế thừa private từ lớp cha thì các thành phần protected và public của lớp cha trở thành private của lớp con. Nói cách khác mỗi thao tác của lớp cha đều bị lớp con che

dầu. Vì vậy trên quan điểm của thế giới bên ngoài lớp con không có các thao tác mà lớp cha có.

03.02.02 Lời giải đề nghị câu 2

Xây dựng lớp đa thức bậc nhất để thể hiện các đa thức bậc nhất có dạng:

$$F(x) = ax + b \text{ (a luôn khác 0)}$$

Xây dựng các phương thức:

a. Phương thức cho phép xác định giá trị của đa thức ứng với $x = x_0$ (tính $F(x_0)$)

b. Phương thức trả về nghiệm đa thức bậc 1 (nghĩa là $F(x) = 0$)

c. Phép toán cộng (operator +) để cộng hai đa thức bậc nhất

Lời giải đề nghị

```
00677. #include <iostream>
00678. #include <iomanip>
00679. using namespace std;
00680.
00681. class CDaThuc
00682. {
00683. private:
00684.     float a;
00685.     float b;
00686. public:
00687.     void Nhap();
00688.     void Xuat();
00689.     //Cau a
00690.     float Tinh(float);
00691.     //Cau b
00692.     int TimX(float&);
00693.     //Cau c
00694.     CDaThuc operator+(CDaThuc);
00695. };
00696.
00697. int main()
00698. {
00699.     CDaThuc p;
00700.     CDaThuc q;
```

```

00701.
00702.     p.Nhap();
00703.     q.Nhap();
00704.
00705.     float x;
00706.     cout << "\nNhap x:";
00707.     cin >> x;
00708.
00709.     //Cau a
00710.     float kq = p.Tinh(x);
00711.     cout << setw(8) << setprecision(5);
00712.     cout << "\nKet qua cua da thuc: " << kq;
00713.
00714.     //Cau b
00715.     int flag = p.TimX(kq);
00716.     if (flag == 0)
00717.         cout << "\nDa thuc vo nghiem";
00718.     else
00719.         if (flag==2)
00720.             cout << "\nDa thuc co vo so nghiem";
00721.         else
00722.             cout << "\nNghiem cua da thuc x: " << kq;
00723.
00724.     //Cau c
00725.     CDaThuc s = p + q;
00726.     cout << "\nTong 2 da thuc: ";
00727.     s.Xuat();
00728.     return 1;
00729. }
00730.
00731. void CDaThuc::Nhap()
00732. {
00733.     cout << "\nNhap he so a:";
00734.     cin >> a;
00735.     cout << "\nNhap he so b:";
00736.     cin >> b;
00737. }
00738.
00739. void CDaThuc::Xuat()
00740. {
00741.     cout << setw(8) << setprecision(5);
00742.     cout << "\nHe so a: " << a;
00743.     cout << setw(8) << setprecision(5);

```

```

00744.      cout << "\nHe so b: " << b;
00745.  }
00746.
00747.  float CDaThuc::Tinh(float x)
00748.  {
00749.      return a * x + b;
00750.  }
00751.
00752.  int CDaThuc::TimX(float& x)
00753.  {
00754.      if (a == 0)
00755.      {
00756.          if (b == 0)
00757.              return 2;
00758.          return 0;
00759.      }
00760.      x = -b / a;
00761.      return 1;
00762.  }
00763.
00764.  CDaThuc CDaThuc::operator+(CDaThuc q)
00765.  {
00766.      CDaThuc temp;
00767.      temp.a = a + q.a;
00768.      temp.b = b + q.b;
00769.      return temp;
00770.  }

```

03.02.03 Lỗi giải đề nghị câu 3

03.02.03.1 Sơ đồ lớp sơ bộ

03.02.03.2 Sơ đồ lớp chi tiết

03.02.03.3 Khai báo lớp

03.02.03.4 Định nghĩa phương thức

```
00771.  as
```

CHƯƠNG 04. ĐỀ - LỜI GIẢI ĐỀ NGHỊ HỌC KỲ 1 NĂM HỌC 2015 – 2016

04.01 ĐỀ THI

04.01.01 Câu 1 – (2 điểm)

- a. Phân biệt các kiểu kế thừa private, protected, public (1 điểm)
- b. Trình bày các đặc điểm quan trọng của lập trình hướng đối tượng (1 điểm)

04.01.02 Câu 2 – (3 điểm)

Xây dựng lớp Thời gian (giờ, phút giây) (1đ). Định nghĩa các phép toán:

- ++ để tăng thời gian thêm 1 giây (1đ)
- >> và << để nhập, xuất dữ liệu thời gian (1đ)

04.01.03 Câu 3 – (5 điểm)

Xây dựng chương trình mô phỏng biên soạn nhạc với các mô tả ký hiệu âm nhạc như sau:

Nốt nhạc: là ký hiệu trong bản nhạc dùng để xác định cao độ (độ cao), trường độ (độ dài, độ ngân vang) của từng âm thanh được vang lên trong bản nhạc.

Có 7 ký hiệu nốt nhạc dùng để xác định cao độ theo thứ tự từ thấp đến cao, đó là Đô (C), Rê (D), Mi (E), Fa (F), Sol (G), La (A), và Si (B)



Để xác định **trường độ** của nốt nhạc có cao độ kể trên, người ta cũng dùng 7 hình nốt để thể hiện, đó là:

Nốt tròn có trường độ tương đương với trường độ của 4 nốt đen

Nốt trắng có trường độ bằng 2 nốt đen

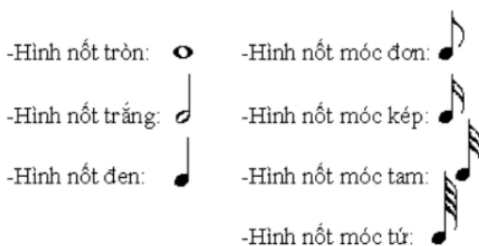
Nốt đen có trường độ bằng 1 phách (đơn vị thời gian trong âm nhạc – ví dụ như 1 bước chân người đi trong không gian)

Nốt móc đơn có trường độ bằng $\frac{1}{2}$ nốt đen

Nốt móc đôi có trường độ bằng $\frac{1}{4}$ nốt đen

Nốt móc tam có trường độ bằng $\frac{1}{8}$ nốt đen

Nốt móc tứ có trường độ bằng $\frac{1}{16}$ nốt đen



Dấu lặng (Z – Zero) là ký hiệu cho biết phải ngưng, không diễn tấu âm thanh (không có cao độ) trong một thời gian nào đó. Các dấu lặng trong thời gian tương ứng (giá trị trường độ) với dạng dấu nhạc nào, thì cũng có tên gọi tương tự.



Trường độ	4	2	1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$
-----------	---	---	---	---------------	---------------	---------------	----------------

Ví dụ: Ký hiệu bản nhạc



C1 - C1/2 - A1/2 - G1/2 - Z1 - D1/2 - C1 - C1 - F2

Trường độ	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	1	1	2
Cao độ	C	C	A	G	Không có (Z)	D	C	C	F
Nốt	Đô đen	Đô móc đơn	La móc đơn	Sol móc đơn	Dấu lặng đen	Rê móc đơn	Đô đen	Đô đen	Fa trắng

Áp dụng kiến thức lập trình hướng đối tượng (kế thừa, đa hình) thiết kế sơ đồ chi tiết các lớp đối tượng (1.5đ) và xây dựng chương trình thực hiện các yêu cầu sau:

1. Soạn một bản nhạc (1.5đ)
2. Tìm và đếm có bao nhiêu dấu lặng đen (Q) trong bản nhạc (1đ)
3. Cho biết nốt nhạc có cao độ cao nhất trong bản nhạc (1đ)

Lưu ý: Trong trường hợp sinh viên không biết về nhạc lý trước đây thì phải đọc kỹ thông tin trên (các thông tin trên đủ để sinh viên thực hiện các yêu cầu của đề thi) và nghiêm túc làm bài. Giám thị coi thi không giải thích gì thêm.

04.02 ĐÁP ÁN

04.02.01 Lời giải đề nghị câu 1

a. Phân biệt các kiểu kế thừa private, protected, public.

Lời giải đề nghị

- Kế thừa public: Lớp con kế thừa public từ lớp cha thì các thành phần protected của lớp cha trở thành protected của lớp con, các thành phần public của lớp cha trở thành public của lớp con. Nói cách khác mỗi thao tác của lớp cha được kế thừa xuống lớp con. Vì vậy ta có thể sử dụng thao tác của lớp cha cho đối tượng thuộc lớp con.
- Kế thừa protected: Lớp con kế thừa protected từ lớp cha thì các thành phần protected và public của lớp cha trở thành protected của lớp con. Nói cách khác mỗi thao tác của lớp cha đều được lớp con bảo vệ. Vì vậy ta có thể sử dụng thao tác của lớp cha cho đối tượng thuộc lớp con.
- Kế thừa private: Lớp con kế thừa private từ lớp cha thì các thành phần protected và public của lớp cha trở thành private của lớp con. Nói cách khác mỗi thao tác của lớp cha đều bị lớp con che dấu. Vì vậy trên quan điểm của thế giới bên ngoài lớp con không có các thao tác mà lớp cha có.

b. Trình bày các đặc điểm quan trọng của lập trình hướng đối tượng (1 điểm).

Lời giải đề nghị

- Các đặc điểm quan trọng của lập trình hướng đối tượng:
 - + Các lớp đối tượng – Classes – Trừu tượng hóa: Cách nhìn khái quát hóa về một tập các đối tượng có chung các đặc

điểm được quan tâm (và bỏ qua những chi tiết không cần thiết).

- + Đóng gói – Encapsulation: Nhóm những gì có liên quan với nhau vào làm một, để sau này có thể dùng một cái tên để gọi đến. Đóng gói để che một số thông tin và chi tiết cài đặt nội bộ để bên ngoài không nhìn thấy
- + Thừa kế – Inheritance: Là cơ chế cho phép một lớp D có được các thuộc tính và thao tác của lớp C, như thể các thuộc tính và thao tác đó đã được định nghĩa tại lớp D.
- + Đa hình – Polymorphism: Là cơ chế cho phép một tên thao tác hoặc thuộc tính có thể được định nghĩa tại nhiều lớp và có thể có nhiều cài đặt khác nhau tại mỗi lớp trong các lớp đó.

04.02.02 Lời giải đề nghị câu 2

Xây dựng lớp Thời gian (giờ, phút giây) (1đ). Định nghĩa các phép toán:

- ++ để tăng thời gian thêm 1 giây (1đ)
 - >> và << để nhập, xuất dữ liệu thời gian (1đ)
- Lời giải đề nghị

Chương trình xây dựng lớp Thời gian hoàn chỉnh:

```
00772. #include <iostream>
00773. using namespace std;
00774.
00775. class CThoiGian
00776. {
00777. private:
00778.     int Gio;
00779.     int Phut;
00780.     int Giay;
00781.
00782. public:
00783.     void Nhap();
00784.     void Xuat();
00785.     friend istream& operator>>(istream&,
00786.                                CThoiGian&);
00787.     friend ostream& operator<<(ostream&,
00788.                                CThoiGian&);
00789.
00790.     CThoiGian& operator++();
```

```

00791.         CThoiGian operator++(int);
00792.     };
00793.
00794.     int main()
00795.     {
00796.         CThoiGian t;
00797.         cin >> t;
00798.         cout << t;
00799.
00800.         CThoiGian kq = t++;
00801.         cout << "\nKet qua kq = ++t:";
00802.         cout << "\nkq: ";
00803.         cout << kq;
00804.         cout << "\nt: ";
00805.         cout << t;
00806.
00807.         kq = ++t;
00808.         cout << "\nKet qua kq = t++:";
00809.         cout << "\nkq: ";
00810.         cout << kq;
00811.         cout << "\nt: ";
00812.         cout << t;
00813.         return 1;
00814.     }
00815.
00816.     void CThoiGian::Nhap()
00817.     {
00818.         cout << "\nNhap gio: ";
00819.         cin >> Gio;
00820.         cout << "Nhap phut: ";
00821.         cin >> Phut;
00822.         cout << "Nhap giay: ";
00823.         cin >> Giay;
00824.     }
00825.
00826.     void CThoiGian::Xuat()
00827.     {
00828.         cout << "\nGio: " << Gio;
00829.         cout << "\nPhut: " << Phut;
00830.         cout << "\nGiay: " << Giay;
00831.     }
00832.
00833.     istream& operator>>(istream&is,CThoiGian&x)

```

```

00834.  {
00835.      cout << "\nNhap gio: ";
00836.      is >> x.Gio;
00837.      cout << "Nhap phut: ";
00838.      is >> x.Phut;
00839.      cout << "Nhap giay: ";
00840.      is >> x.Giay;
00841.      return is;
00842.  }
00843.
00844.  ostream& operator<<(ostream&os,CThoiGian&x)
00845.  {
00846.      cout << "\nGio: " << x.Gio;
00847.      cout << "\nPhut: " << x.Phut;
00848.      cout << "\nGiay: " << x.Giay;
00849.      return os;
00850.  }
00851.
00852.
00853.  CThoiGian& CThoiGian::operator++()
00854.  {
00855.      Giay = Giay + 1;
00856.      if (Giay > 59)
00857.      {
00858.          Phut = Phut + 1;
00859.          if (Phut > 59)
00860.          {
00861.              Gio = Gio + 1;
00862.              if (Gio > 23)
00863.                  Gio = 0;
00864.              Phut = 0;
00865.          }
00866.          Giay = 0;
00867.      }
00868.      return *this;
00869.  }
00870.
00871.  CThoiGian CThoiGian::operator++(int x)
00872.  {
00873.      CThoiGian temp = *this;
00874.      Giay = Giay + 1;
00875.      if (Giay > 59)
00876.      {

```

```

00877.      Phut = Phut + 1;
00878.      if (Phut > 59)
00879.      {
00880.          Gio = Gio + 1;
00881.          if (Gio > 23)
00882.              Gio = 0;
00883.          Phut = 0;
00884.      }
00885.      Giay = 0;
00886.  }
00887.  return temp;
00888.  }

```

04.02.03 Lời giải đề nghị câu 3

04.02.03.1 Sơ đồ lớp sơ bộ

04.02.03.2 Sơ đồ lớp chi tiết

04.02.03.3 Khai báo lớp

04.02.03.4 Định nghĩa phương thức

```
00889.  as
```

CHƯƠNG 05. ĐỀ - LỜI GIẢI ĐỀ NGHỊ HỌC KỲ 2 NĂM HỌC 2015 - 2016

05.01 ĐỀ THI

05.01.01 Câu 1 – (2 điểm)

a. Định nghĩa constructor (phương thức khởi tạo) và default constructor (phương thức khởi tạo mặc định) (1 điểm)

b. Phân biệt các kiểu kế thừa private, protected, public (1 điểm)

05.01.02 Câu 2 – (3 điểm)

Xây dựng lớp Phân số (1đ). Định nghĩa các phép toán

- +, - để thực hiện phép cộng và trừ giữa hai phân số (1đ)
- >> và << để nhập, xuất dữ liệu phân số (1đ)

05.01.03 Câu 3 – (5 điểm)

Xây dựng chương trình mô phỏng game võ lâm truyền kì với các mô tả như sau:

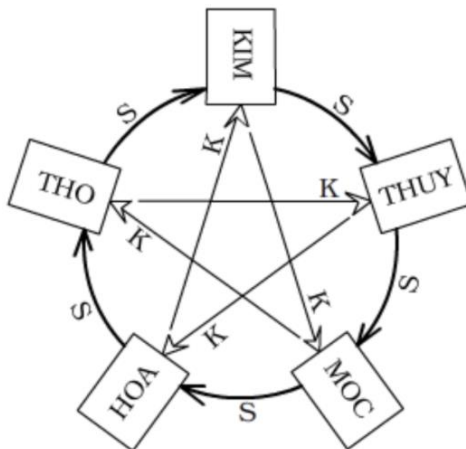
Võ lâm truyền kì là một tựa game theo phong cách nhập vai kiếm hiệp xuất hiện từ những ngày đầu trên thị trường game online Việt Nam. Trong game, người chơi có thể tương tác với nhau để giải trí hoặc tiêu diệt quái vật để phát triển nhân vật của mình.

Nhân vật: Được xem như đại diện cho một người chơi. Mỗi nhân vật thuộc về một môn phái nào đó trong tổng số 10 môn phái của game. Một nhân vật có một giá trị thể hiện cấp độ và mức sát thương. Trong đó Sát thương = Cấp độ x 5

Quái vật: Để gia tăng cấp độ nhân vật của mình, mỗi người chơi sẽ thông qua việc tiêu diệt các quái vật. Có hai loại quái vật : thông thường và đầu lĩnh. Các quái vật cũng sẽ có khả năng tấn công lại người chơi. Quái vật thông thường: Sát thương = Cấp độ x 3. Quái vật đầu lĩnh: Sát thương = Cấp độ x 7

Một nét đặc sắc của game đó là hệ thống ngũ hành tương sinh tương khắc, mỗi một môn phái và quái vật sẽ thuộc về một “hành” nhất định và tương tác giữa các người chơi với nhau, giữa người chơi với quái vật đều dựa trên các quy tắc về ngũ hành.

<p>Hệ Kim:</p> <ul style="list-style-type: none"> • Thiếu Lâm • Thiên Vương bang <p>Hệ Mộc:</p> <ul style="list-style-type: none"> • Ngũ Độc giáo • Đường Môn 	<p>Hệ Thủy:</p> <ul style="list-style-type: none"> • Nga My • Thuyết Yên môn <p>Hệ Hỏa:</p> <ul style="list-style-type: none"> • Cái Bang • Thiên Nhân giáo 	<p>Hệ Thổ:</p> <ul style="list-style-type: none"> • Côn Lôn • Võ Đang
---	---	---



Quy tắc tương sinh: (S) Sát thương gây ra cho người chơi hoặc quái vật + 10 %

Ví dụ: Hỏa sinh thổ, sát thương người chơi (hoặc quái vật) hệ hỏa gây ra cho người chơi (hoặc quái vật) hệ thổ + 10 %

Quy tắc tương khắc: (K) Sát thương gây ra cho người chơi hoặc quái vật ± 20 %

Ví dụ: Mộc khắc thổ, sát thương người chơi (hoặc quái vật) hệ mộc gây ra cho người chơi (hoặc quái vật) hệ thổ + 20 %. Ngược lại, sát thương người chơi (hoặc quái vật) hệ thổ gây ra cho người chơi (hoặc quái vật) hệ mộc - 20 %

Áp dụng kiến thức lập trình hướng đối tượng (kế thừa, đa hình) thiết kế sơ đồ chi tiết các lớp đối tượng (1.5đ) và xây dựng chương trình thực hiện các yêu cầu sau:

1. Tạo và quản lý một danh sách các người chơi và quái vật. (1.5đ)
2. Cho biết phần tử có mức sát thương cao nhất trong danh sách. (1đ)
3. Cho hai phần tử A và B, so sánh giá trị sát thương tác động A lên B và ngược lại. (1đ)

Lưu ý: Trong trường hợp sinh viên không biết về trò chơi này trước đây thì phải đọc kỹ thông tin trên (các thông tin trên đủ để sinh viên thực hiện các yêu cầu của đề thi) và nghiêm túc làm bài. Giám thị coi thi không giải thích gì thêm.

05.02 ĐÁP ÁN

05.02.01 Lời giải đề nghị câu 1

a. Định nghĩa constructor (phương thức khởi tạo) và default constructor (phương thức khởi tạo mặc định) (1 điểm).

Lời giải đề nghị

- Khái niệm Constructor: là một phương thức của lớp dùng để tạo dựng một đối tượng mới. Hệ điều hành sẽ cấp phát bộ nhớ cho đối tượng và gọi đến hàm tạo. Hàm tạo sẽ khởi gán giá trị cho các thuộc tính của đối tượng.
- Phân biệt Constructor mặc định: Constructor mặc định (default constructor) là constructor được gọi khi thể hiện được khai báo mà không có đối số nào được cung cấp.

b. Phân biệt các kiểu kế thừa private, protected, public (1 điểm)

Lời giải đề nghị

- Kế thừa public: Lớp con kế thừa public từ lớp cha thì các thành phần protected của lớp cha trở thành protected của lớp con, các thành phần public của lớp cha trở thành public của lớp con. Nói cách khác mỗi thao tác của lớp cha được kế thừa xuống lớp con. Vì vậy ta có thể sử dụng thao tác của lớp cha cho đối tượng thuộc lớp con.
- Kế thừa protected: Lớp con kế thừa protected từ lớp cha thì các thành phần protected và public của lớp cha trở thành protected của lớp con. Nói cách khác mỗi thao tác của lớp cha đều được lớp con bảo vệ. Vì vậy ta có thể sử dụng thao tác của lớp cha cho đối tượng thuộc lớp con.
- Kế thừa private: Lớp con kế thừa private từ lớp cha thì các thành phần protected và public của lớp cha trở thành private của lớp con. Nói cách khác mỗi thao tác của lớp cha đều bị lớp con che

dấu. Vì vậy trên quan điểm của thế giới bên ngoài lớp con không có các thao tác mà lớp cha có.

05.02.02 Lời giải đề nghị câu 2

Xây dựng lớp Phân số (1đ). Định nghĩa các phép toán

- +, - để thực hiện phép cộng và trừ giữa hai phân số (1đ)
 - >> và << để nhập, xuất dữ liệu phân số (1đ)
- Lời giải đề nghị

Chương trình xây dựng lớp Phân số hoàn chỉnh:

```
00890. #include <iostream>
00891. using namespace std;
00892.
00893. class CPhanSo
00894. {
00895. private:
00896.     int tu;
00897.     int mau;
00898. public:
00899.     void Nhap();
00900.     void Xuat();
00901.     friend istream& operator>>(istream&,
00902.                                CPhanSo&);
00903.     friend ostream& operator<<(ostream&,
00904.                                CPhanSo&);
00905.
00906.     CPhanSo operator+(CPhanSo&);
00907.     CPhanSo operator-(CPhanSo&);
00908. };
00909.
00910. int main()
00911. {
00912.     CPhanSo a;
00913.     cin >> a;
00914.     cout << a;
00915.
00916.     CPhanSo b;
00917.     cin >> b;
00918.     cout << b;
00919.
00920.     cout << "\nTong hai phan so:\n";
```

```

00921.      CPhanSo kq = a + b;
00922.      cout << kq;
00923.      cout << "\nHieu hai phan so:\n";
00924.      kq = a - b;
00925.      cout << kq;
00926.      return 1;
00927.  }
00928.
00929.  void CPhanSo::Nhap()
00930.  {
00931.      cout << "\nNhap tu: ";
00932.      cin >> tu;
00933.      cout << "Nhap mau: ";
00934.      cin >> mau;
00935.  }
00936.
00937.  void CPhanSo::Xuat()
00938.  {
00939.      cout << "\nTu: " << tu;
00940.      cout << "\nMau: " << mau;
00941.  }
00942.
00943.  istream& operator>>(istream& is, CPhanSo&x)
00944.  {
00945.      cout << "\nNhap tu: ";
00946.      is >> x.tu;
00947.      cout << "Nhap mau: ";
00948.      is >> x.mau;
00949.      return is;
00950.  }
00951.
00952.  ostream& operator<<(ostream& os, CPhanSo&x)
00953.  {
00954.      cout << "\nTu: " << x.tu;
00955.      cout << "\nMau: " << x.mau;
00956.      return os;
00957.  }
00958.
00959.  CPhanSo CPhanSo::operator+(CPhanSo& x)
00960.  {
00961.      CPhanSo temp;
00962.      temp.tu = tu * x.mau + mau * x.tu;
00963.      temp.mau = mau * x.mau;

```

```

00964.         return temp;
00965.     }
00966.
00967.     CPhanSo CPhanSo::operator-(CPhanSo& x)
00968.     {
00969.         CPhanSo temp;
00970.         temp.tu = tu * x.mau - mau * x.tu;
00971.         temp.mau = mau * x.mau;
00972.         return temp;
00973.     }

```

05.02.03 Lời giải đề nghị câu 3

Xây dựng chương trình mô phỏng game võ lâm truyền kì với các mô tả như sau:

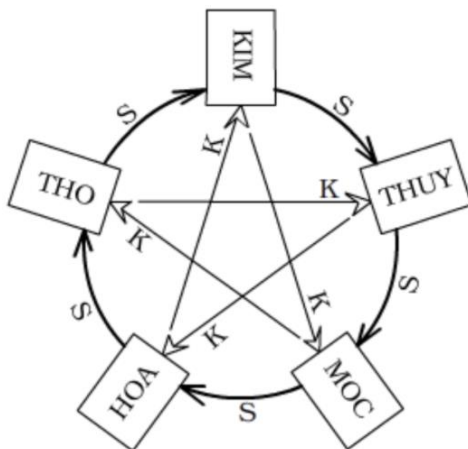
Võ lâm truyền kì là một tựa game theo phong cách nhập vai kiếm hiệp xuất hiện từ những ngày đầu trên thị trường game online Việt Nam. Trong game, người chơi có thể tương tác với nhau để giải trí hoặc tiêu diệt quái vật để phát triển nhân vật của mình.

Nhân vật: Được xem như đại diện cho một người chơi. Mỗi nhân vật thuộc về một môn phái nào đó trong tổng số 10 môn phái của game. Một nhân vật có một giá trị thể hiện cấp độ và mức sát thương. Trong đó Sát thương = Cấp độ x 5

Quái vật: Để gia tăng cấp độ nhân vật của mình, mỗi người chơi sẽ thông qua việc tiêu diệt các quái vật. Có hai loại quái vật : thông thường và đầu lĩnh. Các quái vật cũng sẽ có khả năng tấn công lại người chơi. Quái vật thông thường: Sát thương = Cấp độ x 3. Quái vật đầu lĩnh: Sát thương = Cấp độ x 7

Một nét đặc sắc của game đó là hệ thống ngũ hành tương sinh tương khắc, mỗi một môn phái và quái vật sẽ thuộc về một “hành” nhất định và tương tác giữa các người chơi với nhau, giữa người chơi với quái vật đều dựa trên các quy tắc về ngũ hành.

<p>Hệ Kim:</p> <ul style="list-style-type: none"> Thiếu Lâm Thiên Vương bang <p>Hệ Mộc:</p> <ul style="list-style-type: none"> Ngũ Độc giáo Đường Môn 	<p>Hệ Thủy:</p> <ul style="list-style-type: none"> Nga My Thúy Yên môn <p>Hệ Hỏa:</p> <ul style="list-style-type: none"> Cái Bang Thiên Nhẫn giáo 	<p>Hệ Thổ:</p> <ul style="list-style-type: none"> Côn Lôn Võ Đang
---	---	---



Quy tắc tương sinh: (S) Sát thương gây ra cho người chơi hoặc quái vật + 10 %

Ví dụ: Hoả sinh thổ, sát thương người chơi (hoặc quái vật) hệ hoả gây ra cho người chơi (hoặc quái vật) hệ thổ + 10 %

Quy tắc tương khắc: (K) Sát thương gây ra cho người chơi hoặc quái vật ± 20 %

Ví dụ: Mộc khắc thổ, sát thương người chơi (hoặc quái vật) hệ mộc gây ra cho người chơi (hoặc quái vật) hệ thổ + 20 %. Ngược lại, sát thương người chơi (hoặc quái vật) hệ thổ gây ra cho người chơi (hoặc quái vật) hệ mộc - 20 %

Áp dụng kiến thức lập trình hướng đối tượng (kế thừa, đa hình) thiết kế sơ đồ chi tiết các lớp đối tượng (1.5đ) và xây dựng chương trình thực hiện các yêu cầu sau:

1. Tạo và quản lý một danh sách các người chơi và quái vật. (1.5đ)
2. Cho biết phần tử có mức sát thương cao nhất trong danh sách. (1đ)
3. Cho hai phần tử A và B, so sánh giá trị sát thương tác động A lên B và ngược lại. (1đ)

Lưu ý: Trong trường hợp sinh viên không biết về trò chơi này trước đây thì phải đọc kỹ thông tin trên (các thông tin trên đủ để sinh viên thực hiện các yêu cầu của đề thi) và nghiêm túc làm bài. Giám thị coi thi không giải thích gì thêm.

05.02.03.1 Sơ đồ lớp sơ bộ

05.02.03.2 Sơ đồ lớp chi tiết

05.02.03.3 Khai báo lớp

05.02.03.4 Định nghĩa phương thức

00974. as

CHƯƠNG 06. ĐỀ - LỜI GIẢI ĐỀ NGHỊ HỌC KỲ 1 NĂM HỌC 2016 – 2017

06.01 ĐỀ THI

06.01.01 Câu 1 – (3 điểm)

a. (2 điểm) Nêu khái niệm về sự kế thừa và những ưu điểm của kế thừa trong việc lập trình. Cho ví dụ minh họa

b. (1 điểm) Xét đoạn chương trình sau:

```
00975. class A
00976. {
00977.     A(int i);
00978. };
00979.
00980. void main()
00981. {
00982.     A a1;
00983.     A b2(5);
00984. }
```

Hãy cho biết đoạn chương trình trên có lỗi xảy ra hay không? Nếu có hãy giải thích và chỉ ra cách sửa lỗi.

06.01.02 Câu 2 – (2 điểm)

Cho đoạn chương trình tính toán với phân số như sau:

```
00985. void main()
00986. {
00987.     PhanSo a; // tử: 0; mẫu: 1
00988.     PhanSo b(1, 2); // tử: 1; mẫu: 2
00989.     PhanSo c(3); // tử: 3; mẫu: 1
00990.     a = b + c;
00991.     a.Xuat();
00992. }
```

Hãy khai báo và cài đặt lớp phân số thích hợp để chương trình chạy đúng. Lưu ý rằng không được chỉnh sửa hàm main và sinh viên cần viết các lệnh #include thích hợp.

06.01.03 **Câu 3 – (5 điểm)**

Xét trò chơi Hoàng tử cứu Công chúa với kịch bản như sau:

Công chúa bị Mụ phù thủy giam trong một tòa lâu đài kiên cố có N lớp cổng. Để vào lâu đài cứu Công chúa, Hoàng tử phải vượt qua được tất cả những lớp cổng này. Ở mỗi cổng đều có một người gác cổng. Có 3 loại cổng:

- **Cổng giao thương (Business Gate):** người gác cổng là một tên lái buôn, để qua cổng, Hoàng tử phải mua hàng của tên lái buôn với số tiền = đơn giá * số hàng.
- **Cổng học thuật (Academic Gate):** người gác cổng là một nhà hiền triết, để qua cổng, Hoàng tử phải trả lời được câu hỏi của nhà hiền triết. Câu hỏi có một chỉ số trí tuệ, Hoàng tử cần có chỉ số trí tuệ cao hơn hoặc bằng để trả lời được câu hỏi. Lưu ý: sau khi trả lời câu hỏi, chỉ số trí tuệ của hoàng tử không bị mất đi.
- **Cổng sức mạnh (Power Gate):** người gác cổng là một dũng sỹ, để qua cổng, Hoàng tử phải đánh thắng được dũng sỹ. Dũng sỹ có một chỉ số sức mạnh, Hoàng tử cần có chỉ số sức mạnh cao hơn hoặc bằng để thắng được dũng sỹ. Sau khi chiến thắng, chỉ số sức mạnh của hoàng tử bị hao mòn đi đúng bằng chỉ số sức mạnh của dũng sỹ.

Bảng tóm tắt thông tin các loại cổng như sau:

Loại cổng	Người gác	Điều kiện qua cổng		
		Tiền	Trí tuệ	Sức mạnh
Giao thương	Tên lái buôn	Mất tiền = đơn giá * số hàng	Không	Không
Học thuật	Nhà hiền triết	Không	Trí tuệ >= trí tuệ câu hỏi	Không
Sức mạnh	Dũng sỹ	Không	Không	Mất sức = sức dũng sỹ

Áp dụng kiến thức lập trình hướng đối tượng (kế thừa, đa hình) thiết kế sơ đồ chi tiết các lớp đối tượng (2đ) và xây dựng chương trình để thực hiện các yêu cầu sau:

a. Nhập vào danh sách **N cổng** của lâu đài (1đ)

b. Nhập vào **ba thông số ban đầu** của Hoàng tử là: **số tiền, chỉ số trí tuệ, chỉ số sức mạnh**. Chương trình sẽ cho biết với những thông số này,

Hoàng tử có cứu được Công chúa không. Nếu cứu được thì chương trình tiếp tục cho biết ba thông số còn lại của Hoàng tử (2đ).

06.02 ĐÁP ÁN

06.02.01 Lời giải đề nghị câu 1

a. Nêu khái niệm về sự kế thừa và những ưu điểm của kế thừa trong việc lập trình. Cho ví dụ minh họa.

Lời giải đề nghị

- Kế thừa là một đặc điểm của ngôn ngữ lập trình hướng đối tượng dùng để biểu diễn mối quan hệ đặc biệt hóa – tổng quát hóa giữa các lớp.
- Ưu điểm của kế thừa trong lập trình
 - + Kế thừa cho phép xây dựng lớp mới từ lớp đã có.
 - + Kế thừa cho phép tổ chức các lớp chia sẻ mã chương trình chung, nhờ vậy có thể dễ dàng sửa chữa, nâng cấp hệ thống.
 - + Trong C++, kế thừa còn định nghĩa sự tương thích, nhờ đó ta có cơ chế chuyển kiểu tự động.
- Ví dụ minh họa

```
00993. class CTamGiac
00994. {
00995.     ...
00996. };
00997.
00998. class CTamGiacCan:public CTamGiac
00999. {
01000.     ...
01001. };
```

b. (1 điểm) Xét đoạn chương trình sau:

```
01002. class A
01003. {
01004.     A(int i);
01005. };
01006.
01007. void main()
01008. {
01009.     A a1;
01010.     A b2(5);
01011. }
```


Hãy cho biết đoạn chương trình trên có lỗi xảy ra hay không? Nếu có hãy giải thích và chỉ ra cách sửa lỗi.

- Đoạn chương trình trên khi biên dịch có lỗi xảy ra với các lỗi là:
 - + Chương trình trên sai dòng lệnh **A a1**; Thiếu phương thức thiết lập mặc định.
 - + Chương trình trên thiếu từ khóa khai báo phạm vi truy xuất **public**. Vì trong C++ nếu không khai báo từ khóa phạm vi truy xuất, compiler sẽ hiểu các thuộc tính này có từ khóa phạm vi truy xuất là **private**.
 - + Phương thức này được đề nghị khai báo lại như sau:

```
01012. class A
01013. {
01014. public:
01015.     A();
01016.     A(int i);
01017. };
```

06.02.02 Lời giải đề nghị câu 2

Cho đoạn chương trình tính toán với phân số như sau:

```
01018. void main()
01019. {
01020.     PhanSo a; // tử: 0; mẫu: 1
01021.     PhanSo b(1, 2); // tử: 1; mẫu: 2
01022.     PhanSo c(3); // tử: 3; mẫu: 1
01023.     a = b + c;
01024.     a.Xuat();
01025. }
```

Hãy khai báo và cài đặt lớp phân số thích hợp để chương trình chạy đúng. Lưu ý rằng không được chỉnh sửa hàm main và sinh viên cần viết các lệnh **#include** thích hợp.

- Chương trình cài đặt lớp phân số hoàn chỉnh:

```
01026. #include <iostream>
01027. using namespace std;
01028.
01029. class PhanSo
01030. {
01031. private:
01032.     int tu;
01033.     int mau;
```

```

01034. public:
01035.     PhanSo();
01036.     PhanSo(int);
01037.     PhanSo(int, int);
01038.
01039.     void Xuat();
01040.     PhanSo operator+(PhanSo);
01041. };
01042.
01043. void main()
01044. {
01045.     PhanSo a; // tử: 0; mẫu: 1
01046.     PhanSo b(1, 2); // tử: 1; mẫu: 2
01047.     PhanSo c(3); // tử: 3; mẫu: 1
01048.     a = b + c;
01049.     a.Xuat();
01050. }
01051.
01052. PhanSo::PhanSo()
01053. {
01054.     tu = 0;
01055.     mau = 1;
01056. }
01057.
01058. PhanSo::PhanSo(int tutu)
01059. {
01060.     tu = tutu;
01061.     mau = 1;
01062. }
01063.
01064. PhanSo::PhanSo(int tutu, int maumau)
01065. {
01066.     tu = tutu;
01067.     mau = maumau;
01068. }
01069.
01070. PhanSo PhanSo::operator+(PhanSo x)
01071. {
01072.     PhanSo tong;
01073.     tong.tu = tu * x.mau + mau * x.tu;
01074.     tong.mau = mau * x.mau;
01075.     return tong;
01076. }

```

```

01077.
01078. void PhanSo::Xuat()
01079. {
01080.     cout << "\nTu: " << tu;
01081.     cout << "\nMau: " << mau;
01082. }

```

06.02.03 Lời giải đề nghị câu 3

06.02.03.1 Sơ đồ lớp sơ bộ

06.02.03.2 Sơ đồ lớp chi tiết

06.02.03.3 Khai báo lớp

– Khai báo lớp Công

```

01083. #include<iostream>
01084. using namespace std;
01085.
01086. class CHoangTu;
01087. class CCong
01088. {
01089. public:
01090.     virtual void Nhap() = 0;
01091.     virtual void Xuat() = 0;
01092.     virtual int BiChinhPhuc(CHoangTu&) = 0;
01093. };

```

– Khai báo lớp Hoàng tử

```

01094. #include<iostream>
01095. using namespace std;
01096.
01097. class CCong;
01098. class CLauDai;
01099. class CHoangTu
01100. {
01101. protected:
01102.     int SoTien;
01103.     int TriTue;
01104.     int SucManh;
01105. public:
01106.     void Nhap();
01107.     void Xuat();
01108.
01109.     int GetSoTien();

```

```

01110.    int GetTriTue();
01111.    int GetSucManh();
01112.
01113.    void SetSoTien(int);
01114.    void SetSucManh(int);
01115.
01116.    int ChinhPhuc(CCong*);
01117.    int ChinhPhuc(CLauDai&);
01118. };

```

– Khai báo lớp Lâu Đài

```

01119. class CHoangTu;
01120. class CCong;
01121. class CLauDai
01122. {
01123. protected:
01124.     CCong* ds[1000];
01125.     int n;
01126. public:
01127.     void Nhap();
01128.     void Xuat();
01129.
01130.     int GetSoLuongCong();
01131.     CCong* GetCong(int);
01132.
01133.     int BiChinhPhuc(CHOangTu&);
01134. };

```

– Khai báo lớp cổng giao thương

```

01135. #include "CCong.h"
01136. class CCongGiaoThuong : public CCong
01137. {
01138. protected:
01139.     int SoHang;
01140.     int DonGia;
01141. public:
01142.     void Nhap();
01143.     void Xuat();
01144.     int BiChinhPhuc(CHOangTu&);
01145. };

```

– Khai báo lớp cổng sức mạnh

```

01146. #include "CCong.h"
01147. class CCongSucManh : public CCong
01148. {

```

```

01149.   protected:
01150.       int SucManhCuaDungSy;
01151.   public:
01152.       void Nhap();
01153.       void Xuat();
01154.       int BiChinhPhuc(CHOangTu&);
01155.   };

```

– Khai báo lớp công học thuật

```

01156.   #include "CCong.h"
01157.   class CCongHocThuat : public CCong
01158.   {
01159.   protected:
01160.       int TriTueCauHoi;
01161.   public:
01162.       void Nhap();
01163.       void Xuat();
01164.       int BiChinhPhuc(CHOangTu&);
01165.   };

```

06.02.03.4 Định nghĩa phương thức

– Định nghĩa các phương thức của lớp công giao thương

```

01166.   #include <iostream>
01167.   #include "CCongGiaoThuong.h"
01168.   #include "CHOangTu.h"
01169.
01170.   void CCongGiaoThuong::Nhap()
01171.   {
01172.       cout << "Nhap so hang: ";
01173.       cin >> SoHang;
01174.
01175.       cout << "Nhap don gia: ";
01176.       cin >> DonGia;
01177.   }
01178.
01179.   void CCongGiaoThuong::Xuat()
01180.   {
01181.       cout << "\nCong giao thuong";
01182.       cout << "\nSo hang: " << SoHang;
01183.       cout << "\nDon gia: " << DonGia;
01184.   }
01185.
01186.   int CCongGiaoThuong::BiChinhPhuc(CHOangTu& x)

```

```

01187.  {
01188.      if (DonGia * SoHang > x.GetSoTien())
01189.          return 0;
01190.      x.SetSoTien(x.GetSoTien()-
01191.                  DonGia*SoHang);
01192.      return 1;
01193.  }

```

– Định nghĩa các phương thức của lớp công học thuật

```

01194.  #include <iostream>
01195.  #include "CCongHocThuat.h"
01196.  #include "CHoangTu.h"
01197.
01198.  void CCongHocThuat::Nhap()
01199.  {
01200.      cout << "Nhap tri tue cua cau hoi: ";
01201.      cin >> TriTueCauHoi;
01202.  }
01203.
01204.  void CCongHocThuat::Xuat()
01205.  {
01206.      cout << "\nCong hoc thuat";
01207.      cout << "\nTri tue cua cau hoi: "
01208.              << TriTueCauHoi;
01209.  }
01210.  int CCongHocThuat::BiChinhPhuc(CHoangTu& x)
01211.  {
01212.      if (TriTueCauHoi > x.GetTriTue())
01213.          return 0;
01214.      return 1;
01215.  }

```

– Định nghĩa các phương thức của lớp công sức mạnh

```

01216.  #include "CCongSucManh.h"
01217.  #include "CHoangTu.h"
01218.
01219.  void CCongSucManh::Nhap()
01220.  {
01221.      cout << "Nhap suc manh cua dung sy: ";
01222.      cin >> SucManhCuaDungSy;
01223.  }
01224.
01225.  void CCongSucManh::Xuat()

```

```

01226.  {
01227.      cout << "\nCong suc manh";
01228.      cout << "\nSuc manh cua dung sy: "
01229.          << SucManhCuaDungSy;
01230.  }
01231.
01232.  int CCongSucManh::BiChinhPhuc(CHoangTu&x)
01233.  {
01234.      if (SucManhCuaDungSy > x.GetSucManh())
01235.          return 0;
01236.      x.SetSucManh(x.GetSucManh() -
01237.                  SucManhCuaDungSy);
01238.      return 1;
01239.  }

```

– Định nghĩa các phương thức của lớp

```

01240.  #include <iostream>
01241.  #include "CHoangTu.h"
01242.  #include "CLauDai.h"
01243.  #include "CCong.h"
01244.
01245.  void CHoangTu::Nhap()
01246.  {
01247.      cout << "\nNhap so tien cua hoang tu: ";
01248.      cin >> SoTien;
01249.
01250.      cout << "Nhap tri tue cua hoang tu: ";
01251.      cin >> TriTue;
01252.
01253.      cout << "Nhap suc manh cua hoang tu: ";
01254.      cin >> SucManh;
01255.  }
01256.
01257.  void CHoangTu::Xuat()
01258.  {
01259.      cout << "\nSo tien cua hoang tu: "
01260.          << SoTien;
01261.      cout << "\nTri tue cua hoang tu: "
01262.          << TriTue;
01263.      cout << "\nSuc manh cua hoang tu: "
01264.          << SucManh;
01265.  }
01266.

```

```

01267. int CHoangTu::GetSoTien()
01268. {
01269.     return SoTien;
01270. }
01271.
01272. int CHoangTu::GetSucManh()
01273. {
01274.     return SucManh;
01275. }
01276.
01277. int CHoangTu::GetTriTue()
01278. {
01279.     return TriTue;
01280. }
01281.
01282. void CHoangTu::SetSoTien(int x)
01283. {
01284.     SoTien = x;
01285. }
01286.
01287. void CHoangTu::SetSucManh(int x)
01288. {
01289.     SucManh = x;
01290. }
01291.
01292.
01293. int CHoangTu::ChinhPhuc(CCong* c)
01294. {
01295.     return c->BiChinhPhuc(*this);
01296. }
01297.
01298. int CHoangTu::ChinhPhuc(CLauDai& l)
01299. {
01300.     for (int i=0; i<l.GetSoLuongCong(); i++)
01301.         if (ChinhPhuc(l.GetCong(i)) == 0)
01302.             return 0;
01303.     return 1;
01304. }

```

– Định nghĩa các phương thức của lớp lâu đài

```

01305. #include <iostream>
01306. #include "CLauDai.h"
01307. #include "CHoangTu.h"

```



```

01308. #include "CCongGiaoThuong.h"
01309. #include "CCongHocThuat.h"
01310. #include "CCongSucManh.h"
01311. using namespace std;
01312.
01313. void CLauDai::Nhap()
01314. {
01315.     cout << "\nNhap so luong cong: ";
01316.     cin >> n;
01317.     for (int i = 0; i < n; i++)
01318.     {
01319.         int Loai;
01320.         cout << "\nNhap loai cong
01321.             (0.CongGiaoThuong,
01322.             1.CongHocThuat,
01323.             2.CongSucManh): ";
01324.         cin >> Loai;
01325.         switch (Loai)
01326.         {
01327.             case 0:
01328.                 ds[i] = new CCongGiaoThuong;
01329.                 break;
01330.             case 1:
01331.                 ds[i] = new CCongHocThuat;
01332.                 break;
01333.             case 2:
01334.                 ds[i] = new CCongSucManh;
01335.                 break;
01336.         }
01337.         ds[i]->Nhap();
01338.     }
01339. }
01340.
01341. void CLauDai::Xuat()
01342. {
01343.     cout << "\nSo luong cong: " << n;
01344.     for (int i = 0; i < n; i++)
01345.         ds[i]->Xuat();
01346. }
01347.
01348. int CLauDai::GetSoLuongCong()
01349. {
01350.     return n;

```

```

01351.  }
01352.
01353.  CCong* CLauDai::GetCong(int k)
01354.  {
01355.      return ds[k];
01356.  }
01357.
01358.  int CLauDai::BiChinhPhuc(CHoangTu& x)
01359.  {
01360.      for (int i = 0; i < n; i++)
01361.          if (x.ChinhPhuc(ds[i])==0)
01362.              return 0;
01363.      return 1;
01364.  }

```

– Định nghĩa hàm main

```

01365. #include "CLauDai.h"
01366. #include "CHoangTu.h"
01367. #include "CCongGiaoThuong.h"
01368. #include "CCongHocThuat.h"
01369. #include "CCongSucManh.h"
01370.
01371. int main()
01372. {
01373.     CLauDai x;
01374.     x.Nhap();
01375.     x.Xuat();
01376.
01377.     CHoangTu y;
01378.     y.Nhap();
01379.     y.Xuat();
01380.
01381.     //if (y.ChinhPhuc(x) == 1)
01382.     if(x.BiChinhPhuc(y)==1)
01383.     {
01384.         cout << "\nGiai cuu duoc cong chua";
01385.         cout << "\nChi so cua hoang tu: ";
01386.         y.Xuat();
01387.     }
01388.     else
01389.         cout<<"\nKhong giai cuu duoc ";
01390. }

```


CHƯƠNG 07. ĐỀ - LỜI GIẢI ĐỀ NGHỊ HỌC KỲ 2 NĂM HỌC 2016 – 2017

07.01 ĐỀ THI

07.01.01 Câu 1 – (2.5 điểm)

a. Trình bày của lớp cơ sở trừu tượng (abstract class). Lớp cơ sở trừu tượng được cài đặt trong C++ như thế nào?

b. Xét đoạn chương trình sau:

```
01391. #include <iostream>
01392. using namespace std;
01393. class Array {
01394.     int A[100];
01395.     int n;
01396. public:
01397.     Array(int n)
01398.     {
01399.         //...
01400.     }
01401.     //...
01402. };
01403.
01404. void main()
01405. {
01406.     Array M1;
01407.     Array M2(10);
01408.     cout << M2.A[3];
01409. }
```

Cho biết đoạn chương trình vừa rồi có xảy ra lỗi hay không? Nếu có lỗi, hãy chỉ lỗi các lỗi đó và sửa lỗi để chương trình có thể thực thi được.

07.01.02 Câu 2 – (2.5 điểm)

Xây dựng lớp Thời gian(giờ, phút, giây) (0.5đ). Định nghĩa các phép toán:

- >> và << để nhập, xuất dữ liệu thời gian (1đ)
- -- để thực hiện giảm thời gian đi 1 giây (1đ)

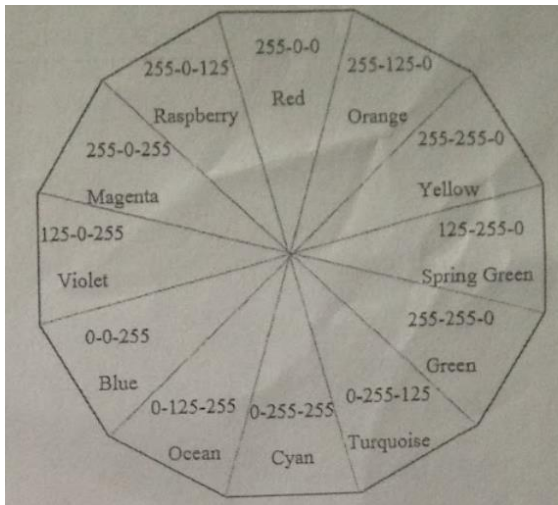
07.01.03 Câu 3 – (5 điểm)

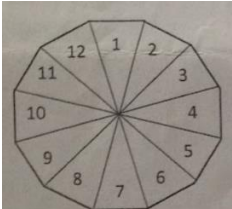

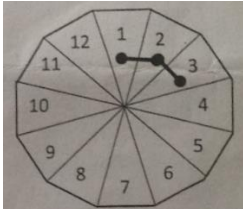
Giao diện website gồm các **thành phần** cơ bản đặc trưng chung bởi các yếu tố về tọa độ (hoành độ, tung độ), kích thước (dài, rộng). Website có 2 thành phần chính:

- **Label** có thêm nội dung text hiển thị, màu chữ và màu nền.
- **Button** có thể hiển thị một hình ảnh hoặc text (màu chữ, màu nền).

Màu sắc trên web được thực hiện bằng cách kết hợp pha trộn của màu đỏ, xanh lá và xanh dương; đây là hình thức phối màu có tên gọi là RGB. Mỗi màu sắc đại diện cho một giá trị số học từ 0 đến 255 và mỗi màu sắc có giá trị tương ứng vs màu đỏ, xanh lá cây và xanh dương.

Phối màu web sẽ giúp chọn các màu khác phù hợp với màu cơ bản, để từ đó có các màu dùng chung cho 1 thiết kế mà đảm bảo tính hài hòa giữa màu sắc. Có rất nhiều phương pháp phối màu, và hầu hết đều dựa trên bánh xe màu để phối. Trong đó đơn giản nhất là 3 cách phối màu sau đây:



<p>Phối màu đơn sắc: Tất cả các thành phần đều có cùng màu nền</p> 	<p>Phối màu bổ túc trực tiếp: là những cặp màu đối xứng với nhau trên bánh xe màu. Vd: 1-7, 2-8, 3-9, 4-10, 5-11, 6-12</p> 	<p>Phối màu tương đồng (thường là 3 màu) các màu liên kề nhau trên bánh xe màu. Vd: 1-2-3, 2-3-4, 11-12-1...</p> 
--	--	---

Xây dựng chương trình hỗ trợ phối màu cho thiết kế web.

Áp dụng kiến thức lập trình hướng đối tượng (kế thừa, đa hình) thiết kế sơ đồ chi tiết các lớp đối tượng (1.5đ) và xây dựng chương trình thực hiện các yêu cầu sau:

1. Nhập danh sách các thành phần có màu trên trang web. (1.5đ)
2. Kiểm tra màu nền và màu chữ của thành phần đầu tiên trong danh sách có phù hợp phối màu bổ túc trực tiếp hay không? (1đ)
3. Kiểm tra màu nền của các thành phần xem phù hợp với quy tắc phối màu nào không? (1đ)

Lưu ý: Trong trường hợp sinh viên không biết các kiến thức về đồ họa này trước đây thì phải đọc kỹ thông tin trên (các thông tin trên đủ để sinh viên thực hiện các yêu cầu của đề thi) và nghiêm túc làm bài. Giám thị coi thi không giải thích gì thêm.

07.02 ĐÁP ÁN

07.02.01 Lời giải đề nghị câu 1

a. Trình bày của lớp cơ sở trừu tượng (abstract class). Lớp cơ sở trừu tượng được cài đặt trong C++ như thế nào?

Lời giải đề nghị

- Trình bày lớp cơ sở trừu tượng
 - + Lớp cơ sở trừu tượng là lớp cơ sở không có đối tượng nào thuộc chính nó.
 - + Ta có thể thay thế cho nội dung không có ý nghĩa bằng phương thức ảo thuần túy. Phương thức ảo thuần túy là phương thức ảo không có nội dung.

- + Khi lớp có phương thức ảo thuần túy, lớp trở thành lớp cơ sở trừu tượng. Ta không thể tạo đối tượng thuộc lớp cơ sở thuần túy.

– Cài đặt của lớp cơ sở trừu tượng được cài đặt trong C++:

```
01410. class Shape    //Abstract
01411. {
01412.     public :
01413.     //Pure virtual Function
01414.     virtual void draw() = 0;
01415. }
01416. class Rectangle : public Shape
01417. {
01418.     public :
01419.     void draw()
01420.     {    // Override Shape::draw()
01421.         ...
01422.     }
01423. }
```

b. Xét đoạn chương trình sau:

```
01424. #include <iostream>
01425. using namespace std;
01426. class Array {
01427.     int A[100];
01428.     int n;
01429. public:
01430.     Array(int n)
01431.     {
01432.         //...
01433.     }
01434.     //...
01435. };
01436.
01437. void main()
01438. {
01439.     Array M1;
01440.     Array M2(10);
01441.     cout << M2.A[3];
01442. }
```

Cho biết đoạn chương trình vừa rồi có xảy ra lỗi hay không? Nếu có lỗi, hãy chỉ lỗi các lỗi đó và sửa lỗi để chương trình có thể thực thi được.

- Đoạn chương trình trên khi biên dịch có lỗi xảy ra với các lỗi là:
 - + Chương trình trên sai dòng lệnh **Array M1**. Thiếu phương thức thiết lập mặc định.
 - + Chương trình trên sai dòng lệnh **cout << M2.A[3]**. Truy xuất thông tin đến thành phần có phạm vi truy xuất private.
 - + Chương trình được đề nghị khai báo lại như sau:

```

01443. #include <iostream>
01444. using namespace std;
01445. class Array {
01446.     int A[100];
01447.     int n;
01448. public:
01449.     Array()
01450.     {
01451.
01452.     }
01453.     Array(int n)
01454.     {
01455.         //...
01456.     }
01457.     //...
01458.     int GetAi(int i)
01459.     {
01460.         return A[i];
01461.     }
01462. };
01463.
01464. void main()
01465. {
01466.     Array M1;
01467.     Array M2(10);
01468.     cout << M2.GetAi(3);
01469. }

```

07.02.02 Lời giải đề nghị câu 2

Xây dựng lớp Thời gian(giờ, phút, giây) (0.5đ). Định nghĩa các phép toán:

- >> và << để nhập, xuất dữ liệu thời gian (1đ)
- -- để thực hiện giảm thời gian đi 1 giây (1đ)

Lời giải đề nghị


```
01470. #include <iostream>
01471. using namespace std;
01472.
01473. class CThoiGian
01474. {
01475.     private:
01476.         int Gio;
01477.         int Phut;
01478.         int Giay;
01479.     public:
01480.         void Nhap();
01481.         void Xuat();
01482.         friend istream& operator>>(istream&, CThoiGian&);
01483.         friend ostream& operator<<(ostream&, CThoiGian&);
01484.
01485.         CThoiGian& operator--();
01486.         CThoiGian operator--(int);
01487. };
01488.
01489. int main()
01490. {
01491.     CThoiGian t;
01492.     cin >> t;
01493.     cout << t;
01494.
01495.     CThoiGian kq = t--;
01496.     cout << "\nKet qua kq = --t:";
01497.     cout << "\nkq: ";
01498.     cout << kq;
01499.     cout << "\nt: ";
01500.     cout << t;
01501.
01502.     kq = --t;
01503.     cout << "\nKet qua kq = t--:";
01504.     cout << "\nkq: ";
01505.     cout << kq;
01506.     cout << "\nt: ";
01507.     cout << t;
01508.     return 1;
01509. }
01510.
```

```

01511. void CThoiGian::Nhap()
01512. {
01513.     cout << "\nNhap gio: ";
01514.     cin >> Gio;
01515.     cout << "Nhap phut: ";
01516.     cin >> Phut;
01517.     cout << "Nhap giay: ";
01518.     cin >> Giay;
01519. }
01520.
01521. void CThoiGian::Xuat()
01522. {
01523.     cout << "\nGio: " << Gio;
01524.     cout << "\nPhut: " << Phut;
01525.     cout << "\nGiay: " << Giay;
01526. }
01527.
01528. istream& operator>>(istream& is, CThoiGian& x)
01529. {
01530.     cout << "\nNhap gio: ";
01531.     is >> x.Gio;
01532.     cout << "Nhap phut: ";
01533.     is >> x.Phut;
01534.     cout << "Nhap giay: ";
01535.     is >> x.Giay;
01536.     return is;
01537. }
01538.
01539. ostream& operator<<(ostream& os, CThoiGian& x)
01540. {
01541.     cout << "\nGio: " << x.Gio;
01542.     cout << "\nPhut: " << x.Phut;
01543.     cout << "\nGiay: " << x.Giay;
01544.     return os;
01545. }
01546.
01547.
01548. CThoiGian& CThoiGian::operator--()
01549. {
01550.     Giay--;
01551.     if (Giay < 0)
01552.     {
01553.         Phut--;

```

```

01554.         if (Phut < 0)
01555.         {
01556.             Gio--;
01557.             if (Gio < 0)
01558.                 Gio = 23;
01559.             Phut = 59;
01560.         }
01561.         Giay = 59;
01562.     }
01563.     return *this;
01564. }
01565.
01566. CThoiGian CThoiGian::operator--(int x)
01567. {
01568.     CThoiGian temp = *this;
01569.     Giay--;
01570.     if (Giay < 0)
01571.     {
01572.         Phut--;
01573.         if (Phut < 0)
01574.         {
01575.             Gio--;
01576.             if (Gio < 0)
01577.                 Gio = 23;
01578.             Phut = 59;
01579.         }
01580.         Giay = 59;
01581.     }
01582.     return temp;
01583. }
01584.

```

07.02.03 Lỗi giải đề nghị câu 3

07.02.03.1 Sơ đồ lớp sơ bộ

07.02.03.2 Sơ đồ lớp chi tiết

07.02.03.3 Khai báo lớp

07.02.03.4 Định nghĩa phương thức

01585. as

CHƯƠNG 08. ĐỀ - LỜI GIẢI ĐỀ NGHỊ HỌC KỲ 3 NĂM HỌC 2016 - 2017

08.01 ĐỀ THI

08.01.01 Câu 1 – (2 điểm)

- Phân biệt private, protected, public (1 điểm)
- Trình bày về constructor (1 điểm)

08.01.02 Câu 2 – (3 điểm)

Xây dựng lớp đa thức bậc hai (1 điểm) để thể hiện các đa thức bậc hai có dạng:

$$F(x) = ax^2 + bx + c \text{ (a luôn khác 0)}$$

Xây dựng các phương thức: (2 điểm)

- Phương thức cho phép xác định giá trị của đa thức ứng với $x = x_0$ (tính $F(x_0)$)
- Phép toán cộng (operator +) để cộng hai đa thức bậc hai

08.01.03 Câu 3 – (5 điểm)

Xây dựng chương trình mô phỏng sáng tác thơ với các mô tả như sau:

Thơ là một loại hình nghệ thuật của ngôn từ, âm thanh của thơ có vần có điệu nhịp nhàng. Lời lẽ của thơ ngắn gọn, hàm chứa, súc tích. Về hình thức, thơ có nhiều thể loại, có thể kể đến như : Lục Bát, Song Thất Lục Bát, Đường Luật Thất Ngôn Bát Cú,...

Luật thơ của thể thơ là toàn bộ những quy tắc về số câu, số tiếng, cách gieo vần, phép hài thanh, ngắt nhịp... được khái quát theo một kiểu mẫu nhất định. Ở đây, chỉ tạm xét đến số câu, số tiếng và cách gieo vần.

Thể thơ lục bát (còn gọi là thể sáu - tám)

- Số tiếng: Mỗi cặp lục bát gồm hai dòng: dòng lục (6 tiếng), dòng bát (8 tiếng). Bài thơ lục bát gồm nhiều cặp câu như thế

- Vần: Vần lưng hiệp vần ở tiếng thứ 6 của hai dòng và giữa tiếng thứ 8 của dòng bất với tiếng thứ 6 của dòng lục.

Ví dụ:

Trăm năm trong cõi người **ta**.

Chữ tài chữ mệnh khéo **là** ghét **nhau**

Trải qua một cuộc bề **dâu**

Những điều trông thấy mà **đau** đốn lòng.

(Nguyễn Du, Truyện Kiều)

Thể thơ song thất lục bát (còn gọi là gián thất hay song thất)

- Số tiếng: cặp song thất (7 tiếng) và cặp lục bát (6 - 8 tiếng) luân phiên kế tiếp nhau trong toàn bài.
- Vần: gieo vần lưng ở mỗi cặp (lục - mọc, buồn - khôn); cặp song thất có vần trắc, cặp lục bát có vần bằng. Giữa cặp song thất và cặp lục bát có vần liền (non - buồn).

Ví dụ:

Ngồi đầu cầu nước trong như **lọc**,

Đường bên cầu cỏ **mọc** còn **non**.

Đưa chàng lòng dặc dặc **buồn**,

Bộ khôn bằng ngựa, thủy **khôn** bằng thuyền.

(Chinh phụ ngâm)

Thể thơ Đường luật Thất ngôn bát cú

- Số câu: 8, số tiếng trong mỗi câu: 7
- Vần: Các tiếng cuối các câu 1, 2, 4, 6, 8 hiệp vần bằng với nhau.
- Nội dung về đối thanh, đối nghĩa không xét đến trong yêu cầu đề thi này.

Ví dụ:

Bước tới đèo Ngang bóng xế **tà**

Cỏ cây chen đá, lá chen **hoa**

Lom khom dưới núi, tiều vài chú

Lác đác bên sông, chợ mấy **nhà**

Nhớ nước đau lòng con quốc quốc

Thương nhà mỗi miệng cái gia **gia**

Dừng chân đứng lại, trời non nước

Một mảnh tình riêng ta với **ta**.

(Bà Huyện Thanh Quan, Qua Đèo Ngang)

Áp dụng kiến thức lập trình hướng đối tượng (kế thừa, đa hình) thiết kế sơ đồ chi tiết các lớp đối tượng (1.5đ) và xây dựng chương trình thực hiện các yêu cầu sau:

1. Soạn một tập thơ (bao gồm nhiều bài thơ thuộc các thể loại khác nhau) (1.5đ)

2. Cho biết bài thơ dài nhất (có nhiều câu nhất) trong tập thơ (1đ)

3. Kiểm tra các bài thơ trong tập thơ có phù hợp với luật thơ không? (1đ)

Lưu ý: Trong trường hợp sinh viên không biết về luật thơ trước đây. thì phải đọc kỹ thông tin trên (các thông tin trên đủ để sinh viên thực hiện các yêu cầu của đề thi) và nghiêm túc làm bài. Giám thị coi thi không giải thích gì thêm.

Giả sử đã có hàm kiểm tra gieo vần như bên dưới và sinh viên có thể sử dụng hàm này và không cần định nghĩa lại:

```
01586. //kiểm tra hai tiếng có vần với nhau hay không, nếu
có trả về 1, nếu không trả về 0
01587. int ktgieovan(char a[], char b[])
01588. {
01589.     int i;
01590.     int check = 0;
01591.     ...
01592.     return check;
01593. }
```

Ví dụ sử dụng hàm ktgieovan để kiểm tra các câu thơ có phù hợp với luật thơ lục bát:

```
01594. //gia su bai tho co 4 cau nhu sau
01595. char* str1[] = {
"tram","nam","trong","coi","nguoi","ta" };
01596. char* str2[] = { "chu", "tai", "chu", "menh",
"kheo","la", "ghet", "nhau" };
```

```

01597. char* str3[] = { "trai","qua", "mot","cuoc", "be",
"dau" };
01598. char* str4[] = { "nhung","dieu", "trong","thay",
"ma","dau","don", "long" };
01599. int kt = 1;
01600. //kiem tra gieo van tieng thu 6 cua cau luc voi
tieng thu 6 cua cau bat(ta, la)
01601. if (ktgieovan(str1[5], str2[5]) == 0)
01602.     kt = 0;
01603. //kiem tra gieo van tieng thu 8 cua cau bat voi
tieng thu 6 cua cau luc tiep theo(nhau, dau).
01604. if (ktgieovan(str2[7], str3[5]) == 0)
01605.     kt = 0;
01606. //kiem tra gieo van tieng thu 6 cua cau luc voi
tieng thu 6 cua cau bat(dau, dau)
01607. if (ktgieovan(str3[5], str4[5]) == 0)
01608.     kt = 0;
01609. if (kt == 1) cout << "luat tho luc bat";

```

08.02 ĐÁP ÁN

08.02.01 Lời giải đề nghị câu 1

a. Phân biệt private, protected, public (1 điểm)

Lời giải đề nghị

- Các thuộc tính và phương thức được khai báo bên trong phạm vi private của một lớp thì chỉ được phép truy xuất từ bên trong lớp và không được quyền truy xuất từ bên ngoài lớp. Hơn nữa, các thuộc tính và phương thức này không cho phép kế thừa ở lớp dẫn xuất.
- Các thuộc tính và phương thức được khai báo bên trong phạm vi protected của một lớp thì chỉ được phép truy xuất từ bên trong lớp và không được quyền truy xuất từ bên ngoài lớp. Hơn nữa, các thuộc tính và phương thức này cho phép kế thừa ở lớp dẫn xuất.
- Các thuộc tính và phương thức được khai báo bên trong phạm vi public của một lớp thì được phép truy xuất từ bên trong lớp, bên ngoài lớp. Hơn nữa, các thuộc tính và phương thức này cho phép kế thừa ở lớp dẫn xuất.

b. Định nghĩa constructor (phương thức khởi tạo)

Lời giải đề nghị

- Khái niệm Constructor: là một phương thức của lớp dùng để tạo dựng một đối tượng mới. Hệ điều hành sẽ cấp phát bộ nhớ cho đối tượng và gọi đến hàm tạo. Hàm tạo sẽ khởi gán giá trị cho các thuộc tính của đối tượng.

08.02.02 Lời giải đề nghị câu 2

Xây dựng lớp đa thức bậc hai (1 điểm) để thể hiện các đa thức bậc hai có dạng:

$$F(x) = ax^2 + bx + c \text{ (a luôn khác 0)}$$

Xây dựng các phương thức: (2 điểm)

a. Phương thức cho phép xác định giá trị của đa thức ứng với $x = x_0$ (tính $F(x_0)$)

b. Phép toán cộng (operator +) để cộng hai đa thức bậc hai

Lời giải đề nghị

Chương trình xây dựng lớp đa thức bậc hai hoàn chỉnh:

```
01610. #include <iostream>
01611. #include <iomanip>
01612. using namespace std;
01613.
01614. class CDaThuc
01615. {
01616. private:
01617.     float a;
01618.     float b;
01619.     float c;
01620. public:
01621.     void Nhap();
01622.     void Xuat();
01623.     //Cau a
01624.     float Tinh(float);
01625.     //Cau b
01626.     int TimX(float &, float &);
01627.     //Cau c
01628.     CDaThuc operator+(CDaThuc);
01629. };
01630.
01631. int main()
01632. {
```

```

01633.      CDaThuc p;
01634.      CDaThuc q;
01635.
01636.      p.Nhap();
01637.      q.Nhap();
01638.
01639.      float x;
01640.      cout << "\nNhap x:";
01641.      cin >> x;
01642.
01643.      //Cau a
01644.      float kq = p.Tinh(x);
01645.      cout << setw(8) << setprecision(5);
01646.      cout << "\nKet qua cua da thuc bac hai: " << kq;
01647.
01648.      //Cau b
01649.      float kq1, kq2;
01650.      int flag = p.TimX(kq1, kq2);
01651.
01652.      cout << "\nNghiem cua da thuc bac hai: ";
01653.      if (flag == 0)
01654.      {
01655.          cout << setw(8) << setprecision(5);
01656.          cout << "\nDa thuc co nghiem duy nhat x = " <<
kq1;
01657.      }
01658.      else
01659.          if (flag == -1)
01660.              cout << "\nDa thuc vo nghiem";
01661.          else
01662.          {
01663.              cout << "\nDa thuc co nghiem kep:";
01664.              cout << setw(8) << setprecision(5);
01665.              cout << "\nx1 = " << kq1;
01666.              cout << "\nx2 = " << kq2;
01667.          }
01668.
01669.      //Cau c
01670.      CDaThuc s = p + q;
01671.      cout << "\nTong 2 da thuc bac hai: ";
01672.      s.Xuat();
01673.      return 1;
01674.  }

```

```

01675.
01676. void CDaThuc::Nhap()
01677. {
01678.     cout << "\nNhap he so a:";
01679.     cin >> a;
01680.     cout << "\nNhap he so b:";
01681.     cin >> b;
01682.     cout << "\nNhap he so c:";
01683.     cin >> c;
01684. }
01685.
01686. void CDaThuc::Xuat()
01687. {
01688.     cout << setw(8) << setprecision(5);
01689.     cout << "\nHe so a: " << a;
01690.     cout << setw(8) << setprecision(5);
01691.     cout << "\nHe so b: " << b;
01692.     cout << setw(8) << setprecision(5);
01693.     cout << "\nHe so c: " << c;
01694. }
01695.
01696. float CDaThuc::Tinh(float x)
01697. {
01698.     return a * x * x + b * x + c;
01699. }
01700.
01701. int CDaThuc::TimX(float &x1, float& x2)
01702. {
01703.     float d = b * b - 4 * a * c;
01704.     if (d <= 0)
01705.     {
01706.         if (d == 0)
01707.         {
01708.             x1 = x2 = (-b) / (2 * a);
01709.             return 0;
01710.         }
01711.         return -1;
01712.     }
01713.     x1 = (-b + sqrt(d) / (2 * a));
01714.     x2 = (-b - sqrt(d) / (2 * a));
01715.     return 1;
01716. }
01717.

```

```
01718.  CDaThuc CDaThuc::operator+(CDaThuc x)
01719.  {
01720.      CDaThuc tong;
01721.      tong.a = a + x.a;
01722.      tong.b = b + x.b;
01723.      tong.c = c + x.c;
01724.      return tong;
01725.  }
```

08.02.03 Lời giải đề nghị câu 3

08.02.03.1 Sơ đồ lớp sơ bộ

08.02.03.2 Sơ đồ lớp chi tiết

08.02.03.3 Khai báo lớp

08.02.03.4 Định nghĩa phương thức

```
01726.  as
```

CHƯƠNG 09. ĐỀ - LỜI GIẢI ĐỀ NGHỊ HỌC KỲ 2 NĂM HỌC 2017 – 2018

09.01 ĐỀ THI

09.01.01 Câu 1 – (2 điểm)

- a. Hàm thuần ảo là gì? Lớp trừu tượng là gì? Cho ví dụ minh họa. (1đ)
- b. Hãy nêu các đặc điểm quan trọng của lập trình hướng đối tượng. (1đ)

09.01.02 Câu 2 – (3 điểm)

Xây dựng lớp Đa thức bậc n với các toán tử $>>$, $<<$, $+$ (3đ)

09.01.03 Câu 3 – (5 điểm)

Đầu những năm 1900, dựa trên sự hiện diện của các kháng nguyên trên màng hồng cầu, các nhà khoa học đã xác định rằng con người có 4 nhóm máu khác nhau: **O**, **A**, **B** và **AB**. Hệ thống phân loại nhóm máu này (gọi là hệ thống nhóm máu **ABO**) cung cấp cho bác sĩ các thông tin quan trọng để lựa chọn nhóm máu phù hợp trong việc truyền máu. Và đồng thời có thể tiên đoán được nhóm máu tương đối của người con dựa trên nhóm máu của cha mẹ theo cơ chế di truyền học.

Nhóm máu của người con khi biết được nhóm máu của cha và mẹ:

		Nhóm máu người cha				
		A	B	AB	O	
Nhóm máu người mẹ	A	A hoặc O	A, B, AB hoặc O	A, B hoặc AB	A hoặc O	Dự đoán khả năng nhóm máu người con

	B	A, B, AB hoặc O	B hoặc O	A, B hoặc AB	B hoặc O	
	AB	A, B hoặc AB	A, B hoặc AB	A, B hoặc AB	A hoặc B	
	O	A hoặc O	B hoặc O	A hoặc B	O	

Ngoài ra còn có thêm hệ thống phân loại Rh (Rhesus)

Căn cứ vào sự khác biệt khi nghiên cứu về sự vận chuyển oxy của hồng cầu thì các hồng cầu có thể mang ở mặt ngoài một protein gọi là Rhesus. Nếu có kháng nguyên D thì là nhóm Rh+ (dương tính), nếu không có là Rh- (âm tính). Các nhóm máu **A, B, O, AB** mà Rh- thì được gọi là âm tính **A-, B-, O-, AB-**. Nhóm máu Rh- chỉ chiếm 0,04% dân số thế giới. Đặc điểm của nhóm máu Rh này là chúng chỉ có thể nhận và cho người cùng nhóm máu, đặc biệt phụ nữ có nhóm máu Rh- thì con rất dễ tử vong.

Người có nhóm máu Rh+ chỉ có thể cho người cũng có nhóm máu Rh+ và nhận người có nhóm máu Rh+ hoặc Rh-

Người có nhóm máu Rh- có thể cho người có nhóm máu Rh+ hoặc Rh- nhưng chỉ nhận được người có nhóm máu Rh- mà thôi.

Trường hợp người có nhóm máu Rh- được truyền máu Rh+, trong lần đầu tiên sẽ không có bất kỳ phản ứng tức thì nào xảy ra, nhưng nếu tiếp tục truyền máu Rh+ lần thứ 2 sẽ gây ra những hậu quả nghiêm trọng do tai biến truyền máu. Tương tự với trường hợp mẹ Rh- sinh con (lần đầu và lần thứ hai trở đi).

Khả năng tương thích:

✓: Có thể cho - nhận.

✗: Không thể cho - nhận.

Bảng khả năng tương thích hồng cầu								
Người nhận	Người cho							
	O-	O+	A-	A+	B-	B+	AB-	AB+

O-	✓	X	X	X	X	X	X	X
O+	✓	✓	X	X	X	X	X	X
A-	✓	X	✓	X	X	X	X	X
A+	✓	✓	✓	✓	X	X	X	X
B-	✓	X	X	X	✓	X	X	X
B+	✓	✓	X	X	✓	✓	X	X
AB-	✓	X	✓	X	✓	X	✓	X
AB+	✓	✓	✓	✓	✓	✓	✓	✓

Áp dụng kiến thức lập trình hướng đối tượng (kế thừa, đa hình) thiết kế sơ đồ chi tiết các lớp đối tượng (1.5đ) và xây dựng chương trình thực hiện các yêu cầu sau:

1. Nhập danh sách các nhóm máu của một nhóm người. (1đ).
2. Cho một bộ 3 nhóm máu của 3 người là cha, mẹ, con. Hãy kiểm tra và đưa ra kết quả nhóm máu có phù hợp với quy luật di truyền hay không? (1đ)
3. Chọn một người X trong danh sách. Hãy liệt kê tất cả các người còn lại trong danh sách có thể cho máu người X này. (1đ)

Lưu ý: Trong trường hợp sinh viên không biết về nhóm máu và di truyền học trước đây thì phải đọc kỹ thông tin trên (các thông tin trên đủ để sinh viên thực hiện các yêu cầu của đề thi) và nghiêm túc làm bài. Giám thị coi thi không giải thích gì thêm.

09.02 ĐÁP ÁN

09.02.01 Lời giải đề nghị câu 1

a. Hàm thuần ảo là gì? Lớp trừu tượng là gì? Cho ví dụ minh họa. (1đ)

- Khái niệm hàm thuần ảo:
 - + Hàm thuần ảo (Phương thức ảo thuần túy) là phương thức ảo không có nội dung.

- Khái niệm hàm thuần ảo:
 - + Lớp cơ sở trừu tượng là lớp cơ sở không có đối tượng nào thuộc chính nó.
 - + Khi lớp có phương thức ảo thuần túy, lớp trở thành lớp cơ sở trừu tượng. Ta không thể tạo đối tượng thuộc lớp cơ sở thuần túy.
- Cài đặt của lớp cơ sở trừu tượng được cài đặt trong C++:

```

01727. class Shape      //Abstract
01728. {
01729.     public :
01730.     //Pure virtual Function
01731.     virtual void draw() = 0;
01732. }
01733. class Rectangle : public Shape
01734. {
01735.     public :
01736.     void draw()
01737.     {      // Override Shape::draw()
01738.         ...
01739.     }
01740. }

```

b. Trình bày các đặc điểm quan trọng của lập trình hướng đối tượng (1 điểm).

Lời giải đề nghị

- Các lớp đối tượng – Classes – Trừu tượng hóa: Cách nhìn khái quát hóa về một tập các đối tượng có chung các đặc điểm được quan tâm (và bỏ qua những chi tiết không cần thiết).
- Đóng gói – Encapsulation: Nhóm những gì có liên quan với nhau vào làm một, để sau này có thể dùng một cái tên để gọi đến. Đóng gói để che một số thông tin và chi tiết cài đặt nội bộ để bên ngoài không nhìn thấy
- Thừa kế – Inheritance: Là cơ chế cho phép một lớp D có được các thuộc tính và thao tác của lớp C, như thể các thuộc tính và thao tác đó đã được định nghĩa tại lớp D.
- Đa hình – Polymorphism: Là cơ chế cho phép một tên thao tác hoặc thuộc tính có thể được định nghĩa tại nhiều lớp và có thể có nhiều cài đặt khác nhau tại mỗi lớp trong các lớp đó.

09.02.02 Lời giải đề nghị câu 2

Xây dựng lớp Đa thức bậc n với các toán tử >>, <<, + (3đ)


```
01741. #include <iostream>
01742. #include <iomanip>
01743. using namespace std;
01744.
01745. class CDaThuc
01746. {
01747. private:
01748.     int n;
01749.     float a[500];
01750. public:
01751.     CDaThuc();
01752.     CDaThuc(int);
01753.     void Nhap();
01754.     void Xuat();
01755.     friend istream& operator>>(istream&, CDaThuc&);
01756.     friend ostream& operator<<(ostream&, CDaThuc&);
01757.
01758.     int BacLonNhat(CDaThuc);
01759.     CDaThuc operator+(CDaThuc&);
01760. };
01761.
01762. int main()
01763. {
01764.     CDaThuc p;
01765.     CDaThuc q;
01766.
01767.     cin >> p;
01768.     cin >> q;
01769.
01770.     cout << p;
01771.     cout << q;
01772.
01773.     CDaThuc kq = p + q;
01774.     cout << "\nKet qua:";
01775.     cout << kq;
01776.     return 1;
01777. }
01778.
01779. CDaThuc::CDaThuc()
01780. {
```

```

01781.     n = 0;
01782. }
01783.
01784. CDaThuc::CDaThuc(int nn)
01785. {
01786.     n = nn;
01787. }
01788.
01789. void CDaThuc::Nhap()
01790. {
01791.     cout << "\nNhap bac n:";
01792.     cin >> n;
01793.     for (int i = 0; i < n + 1; i++)
01794.     {
01795.         cout << "\nNhap he so thu " << i << ": ";
01796.         cin >> a[i];
01797.     }
01798. }
01799.
01800. void CDaThuc::Xuat()
01801. {
01802.     cout << "\nDa thuc bac " << n << " co dang:\n";
01803.     for (int i = n; i > 0; i--)
01804.         cout << a[i] << "x^" << i << " + ";
01805.     cout << a[0];
01806. }
01807.
01808. istream& operator>>(istream& is, CDaThuc& x)
01809. {
01810.     cout << "\nNhap bac n:";
01811.     is >> x.n;
01812.     for (int i = 0; i < x.n + 1; i++)
01813.     {
01814.         cout << "\nNhap he so thu " << i << ": ";
01815.         is >> x.a[i];
01816.     }
01817.     return is;
01818. }
01819.
01820. ostream& operator<<(ostream& os, CDaThuc& x)
01821. {
01822.     os << "\nDa thuc bac " << x.n << " co dang:\n";
01823.     for (int i = x.n; i > 0; i--)

```

```

01824.         os << x.a[i] << "x^" << i << " + ";
01825.         os << x.a[0];
01826.         return os;
01827.     }
01828.
01829.     int CDaThuc::BacLonNhat(CDaThuc x)
01830.     {
01831.         if (n > x.n)
01832.             return n;
01833.         return x.n;
01834.     }
01835.
01836.     CDaThuc CDaThuc::operator+(CDaThuc& x)
01837.     {
01838.         int mm, nn;
01839.         mm = this->BacLonNhat(x);
01840.         nn = n + x.n - mm;
01841.
01842.         CDaThuc temp;
01843.         temp.n = mm;
01844.         for (int i = 0; i < mm + 1; i++)
01845.         {
01846.             if (i > nn)
01847.                 if (mm == n)
01848.                     x.a[i] = 0;
01849.             else
01850.                 a[i] = 0;
01851.             temp.a[i] = a[i] + x.a[i];
01852.         }
01853.         return temp;
01854.     }

```

09.02.03 Lời giải đề nghị câu 3

09.02.03.1 Sơ đồ lớp sơ bộ

09.02.03.2 Sơ đồ lớp chi tiết

09.02.03.3 Khai báo lớp

09.02.03.4 Định nghĩa phương thức

```
01855.     as
```


CHƯƠNG 10. ĐỀ - LỜI GIẢI ĐỀ NGHỊ HỌC KỲ 1 NĂM HỌC 2017 - 2018

10.01 ĐỀ THI

10.01.01 Câu 1 – (2 điểm)

- Phân biệt các phạm vi truy cập private, protected và public.
- Cho biết ý nghĩa và mục đích của các hàm get/set trong một lớp

10.01.02 Câu 2 – (3 điểm)

Cho đoạn chương trình tính toán với lớp đối tượng ngày tháng năm (cNgày) như sau:

```
01856. void main()
01857. {
01858.     cNgày ng1;
01859.     // ng1 sẽ có giá trị là ngày 1 tháng 1 năm 1
01860.
01861.     cNgày ng2(2017, 1);
01862.     // ng2 sẽ có giá trị là ngày 1 tháng 1 năm 2017
01863.     cNgày ng3(2017, 1, 7);
01864.     // ng3 sẽ có giá trị là ngày 7 tháng 1 năm 2017
01865.     cin >> ng1;
01866.     cout << ng1;
01867.     if (ng1 < ng2)
01868.         cout << "Ngày 1 truoc ngay 2" << endl;
01869.     else
01870.         cout << "Ngày 1 khong truoc ngay 2" << endl;
01871. }
```

Hãy định nghĩa lớp cNgày thích hợp để chương trình không bị lỗi biên dịch và chạy đúng. Lưu ý rằng không được chỉnh sửa hàm main và sinh viên cần viết cả các lệnh #include thích hợp

10.01.03 Câu 3 – (5 điểm)

Công ty quản lý ca sỹ XYZ cần quản lý các thông tin để tính lương cho các ca sỹ thuộc công ty. Giả sử công ty XYZ chia các ca sỹ thành 2 nhóm: **ca sỹ “chưa” nổi tiếng** và **ca sỹ nổi tiếng**. Thông tin chung của cả 2 nhóm bao gồm:

- Họ tên ca sỹ.
- Số năm làm việc cho công ty.
- Số đĩa đã bán được.
- Số buổi trình diễn đã tham gia.

Ngoài ra, ca sỹ nổi tiếng được mời tham gia nhiều Gameshow nên còn có thêm thông tin: số gameshow tham gia.

Công ty quy định cách tính và trả lương cho ca sỹ như sau:

- Với ca sỹ “chưa” nổi tiếng:

$Lương = 3.000.000 + 500.000 * \text{số năm làm việc} + 1.000 * \text{số đĩa bán được} + 200.000 * \text{số buổi trình diễn}$.

- Với ca sỹ nổi tiếng:

$Lương = 5.000.000 + 500.000 * \text{số năm làm việc} + 1.200 * \text{số đĩa bán được} + 500.000 * \text{số buổi trình diễn} + 500.000 * \text{số Gameshow}$.

Bạn hãy đề xuất thiết kế các lớp đối tượng cần thiết (vẽ sơ đồ lớp chi tiết) để quản lý danh sách các ca sỹ của Công ty và hỗ trợ tính lương cho ca sỹ theo quy định như trên (3 đ)

Hãy viết chương trình bằng C++ cho phép thực hiện các yêu cầu sau:

1. Nhập danh sách ca sỹ (lưu trữ trong một mảng duy nhất) (1 đ)
2. Tìm ca sỹ có lương cao nhất trong công ty. Nếu có nhiều ca sỹ có cùng mức lương cao nhất, chỉ cần trả về 1 ca sỹ trong số đó (1 đ)

Lưu ý:

- Sử dụng tính chất kế thừa và đa hình
- Sử dụng string để lưu chuỗi.
- Vẽ sơ đồ lớp: mô tả các lớp, các thuộc tính, các hàm và mối liên hệ các lớp (1.5đ)
- Khai báo và định nghĩa chi tiết các lớp (1.5đ)

10.02 ĐÁP ÁN

10.02.01 Lời giải đề nghị câu 1

a. Phân biệt các phạm vi truy cập **private**, **protected**, **public** và cho ví dụ minh họa.

Lời giải đề nghị

- Các thuộc tính và phương thức được khai báo bên trong phạm vi **private** của một lớp thì **chỉ được phép truy xuất từ bên trong lớp và không được quyền truy xuất từ bên ngoài lớp**. Hơn nữa, các thuộc tính và phương thức này **không cho phép kế thừa ở lớp dẫn xuất**.
- Các thuộc tính và phương thức được khai báo bên trong phạm vi **protected** của một lớp thì **chỉ được phép truy xuất từ bên trong lớp và không được quyền truy xuất từ bên ngoài lớp**. Hơn nữa, các thuộc tính và phương thức này **cho phép kế thừa ở lớp dẫn xuất**.
- Các thuộc tính và phương thức được khai báo bên trong phạm vi **public** của một lớp thì **được phép truy xuất từ bên trong lớp, bên ngoài lớp**. Hơn nữa, các thuộc tính và phương thức này **cho phép kế thừa ở lớp dẫn xuất**.

b. Cho biết ý nghĩa và mục đích của các hàm **get/set** trong một lớp

- Ngoài việc bảo vệ các nguyên tắc đóng gói, ta cần kiểm tra xem giá trị mới cho thành viên dữ liệu có hợp lệ hay không.
- Sử dụng phương thức truy vấn cho phép ta thực hiện việc kiểm tra trước khi thực sự thay đổi giá trị của thành viên.
- Chỉ cho phép các dữ liệu có thể truy vấn hay thay đổi mới được truy cập đến.

10.02.02 Lời giải đề nghị câu 2

Cho đoạn chương trình tính toán với lớp đối tượng ngày tháng năm (cNgày) như sau:

```
01872. void main()
01873. {
01874.     cNgày ng1;
01875.     // ng1 sẽ có giá trị là ngày 1 tháng 1 năm 1
01876.
01877.     cNgày ng2(2017, 1);
01878.     // ng2 sẽ có giá trị là ngày 1 tháng 1 năm 2017
01879.     cNgày ng3(2017, 1, 7);
```

```

01880. // ng3 sẽ có giá trị là ngày 7 tháng 1 năm 2017
01881.     cin >> ng1;
01882.     cout << ng1;
01883.     if (ng1 < ng2)
01884.         cout << "Ngày 1 truoc ngay 2" << endl;
01885.     else
01886.         cout << "Ngày 1 khong truoc ngay 2" << endl;
01887. }

```

Hãy định nghĩa lớp cNgày thích hợp để chương trình không bị lỗi biên dịch và chạy đúng. Lưu ý rằng không được chỉnh sửa hàm main và sinh viên cần viết cả các lệnh #include thích hợp

Lời giải đề nghị

Chương trình xây dựng lớp cNgày hoàn chỉnh:

```

01888. #include <iostream>
01889. using namespace std;
01890.
01891. class cNgày
01892. {
01893. private:
01894.     int Ngay;
01895.     int Thang;
01896.     int Nam;
01897. public:
01898.     cNgày();
01899.     cNgày(int, int);
01900.     cNgày(int, int, int);
01901.
01902.     friend istream& operator>>(istream&, cNgày&);
01903.     friend ostream& operator<<(ostream&, cNgày&);
01904.
01905.     int operator<(cNgày);
01906. };
01907.
01908. void main()
01909. {
01910.     cNgày ng1;
01911.     // ng1 sẽ có giá trị là ngày 1 tháng 1 năm 1
01912.
01913.     cNgày ng2(2017, 1);
01914.     // ng2 sẽ có giá trị là ngày 1 tháng 1 năm 2017
01915.     cNgày ng3(2017, 1, 7);

```



```

01916.    // ng3 sẽ có giá trị là ngày 7 tháng 1 năm 2017
01917.    cin >> ng1;
01918.    cout << ng1;
01919.    if (ng1 < ng2)
01920.        cout << "Ngày 1 truoc ngay 2" << endl;
01921.    else
01922.        cout << "Ngày 1 khong truoc ngay 2" << endl;
01923.    }
01924.
01925.    cNgay::cNgay()
01926.    {
01927.        Ngay = 1;
01928.        Thang = 1;
01929.        Nam = 1;
01930.    }
01931.
01932.    cNgay::cNgay(int NamNam, int ThangThang)
01933.    {
01934.        Ngay = 1;
01935.        Thang = ThangThang;
01936.        Nam = NamNam;
01937.    }
01938.
01939.    cNgay::cNgay(int NamNam, int ThangThang, int
NgayNgay)
01940.    {
01941.        Ngay = NgayNgay;
01942.        Thang = ThangThang;
01943.        Nam = NamNam;
01944.    }
01945.
01946.    istream& operator>>(istream& is, cNgay& x)
01947.    {
01948.        cout << "\nNhap ngay: ";
01949.        is >> x.Ngay;
01950.        cout << "Nhap thang: ";
01951.        is >> x.Thang;
01952.        cout << "Nhap nam: ";
01953.        is >> x.Nam;
01954.        return is;
01955.    }
01956.
01957.    ostream& operator<<(ostream& os, cNgay& x)

```

```

01958.  {
01959.      cout << "\nNgày: " << x.Ngày;
01960.      cout << "\nThang: " << x.Thang;
01961.      cout << "\nNam: " << x.Nam;
01962.      cout << endl;
01963.      return os;
01964.  }
01965.
01966.  int cNgày::operator<(cNgày x)
01967.  {
01968.      if (Nam < x.Nam)
01969.          return 1;
01970.      if (Thang < x.Thang)
01971.          return 1;
01972.      if (Ngày < x.Ngày)
01973.          return 1;
01974.      return 0;
01975.  }

```

10.02.03 Lỗi giải đề nghị câu 3

10.02.03.1 Sơ đồ lớp sơ bộ

10.02.03.2 Sơ đồ lớp chi tiết

10.02.03.3 Khai báo lớp

10.02.03.4 Định nghĩa phương thức

```

01976.  as

```

CHƯƠNG 11. ĐỀ - LỜI GIẢI ĐỀ NGHỊ HỌC KỲ 1 NĂM HỌC 2018 – 2019

11.01 ĐỀ THI

11.01.01 Câu 1 – (2.5 điểm)

a. Phân biệt khái niệm lớp và đối tượng trong lập trình hướng đối tượng. (1đ)

b. Trình bày khái niệm đa hình trong lập trình hướng đối tượng (1 đ). Cho ví dụ minh họa. (0.5 đ)

11.01.02 Câu 2 – (2.5 điểm)

Định nghĩa lớp CDate biểu diễn khái niệm ngày, tháng, năm (0.5 đ) với phép toán ++ (thêm bớt một ngày) theo dạng prefix ++a và postfix a++ (1 đ). Phép toán <<, >> đề xuất, nhập dữ liệu loại CDate (1 đ).

11.01.03 Câu 3 – (5 điểm)

Như chúng ta đã biết, sức khỏe tốt là nền tảng góp phần cho một cuộc sống hạnh phúc và việc tập luyện hàng ngày chính là một biện pháp cực kỳ hiệu quả để nâng cao cả về sức khỏe tinh thần và thể chất. Năm bắt xu thế quan tâm sức khỏe của người dân, ngày 26/10/2018 vừa qua, trung tâm thể hình MTK của thị xã Dĩ An đã được khai trương và bắt đầu đón nhận khách hàng.

Mỗi khách hàng tham gia CLB của trung tâm có thể tùy chọn một trong 3 gói dịch vụ khác nhau là Premium (nâng cao), Basic (cơ bản), Non-member (không thành viên). Đây là phương thức mà trung tâm tiến hành để từng loại đối tượng khách hàng có thể dễ dàng tiếp cận các dịch vụ tương ứng.

Bên cạnh thể hình là dịch vụ mặc định, tùy gói dịch vụ mà khách hàng có thể đăng kí tham gia các lớp học được cung cấp xuyên suốt các thời gian trong ngày như: Yoga, Aerobic, Boxing, múa bụng, body combat..., dịch vụ xông hơi và hỗ trợ từ huấn luyện viên cá nhân (PT) cũng sẽ được cung cấp.

Mỗi khách hàng khi đăng kí sẽ phải điền thông tin họ tên và số CMND và lựa chọn gói dịch vụ cũng như thời gian muốn sử dụng dịch vụ đó (theo tháng).

Chi tiết từng gói dịch vụ trong **một tháng** được mô tả theo tháng theo bảng sau. Đơn vị: Nghìn đồng.

	Premium	Basic	Non-member
Phí cơ bản	1 000	500	200
Chi phí lớp học	Miễn phí	100 / lớp	Không có
Dịch vụ xông hơi	Miễn phí	Không có	Không có
Hỗ trợ PT	Miễn phí	100	200

Lưu ý:

- **Miễn phí:** Khách hàng không chi trả thêm tiền khi đăng kí tính năng này.
- **Không có:** Khách hàng sẽ không được cung cấp tính năng này.

Sinh viên hãy ứng dụng kiến thức về lập trình hướng đối tượng để thực hiện các yêu cầu sau:

1. Xây dựng sơ đồ phân lớp kế thừa. (1 đ)
2. Cài đặt các lớp thích hợp. (2 đ)
3. Quản lí việc nhập xuất danh sách khách hàng. (1 đ)

4. Trung tâm cần thông tin của những khách hàng đã chi tiêu nhiều nhất để tiến hành tư vấn gói dịch vụ thích hợp cho họ. Hãy viết phương thức thực hiện tính năng này. (1 đ)

11.02 ĐÁP ÁN

11.02.01 Lời giải đề nghị câu 1

a. Phân biệt khái niệm **lớp** và **đối tượng** trong lập trình hướng đối tượng. (1đ)

Lời giải đề nghị

- Phân biệt khái niệm lớp và đối tượng:
 - + Lớp là một mô tả trừu tượng của nhóm các đối tượng cùng bản chất, ngược lại mỗi một đối tượng là một thể hiện cụ thể cho những mô tả trừu tượng đó.

b. Trình bày khái niệm đa hình trong lập trình hướng đối tượng (1 đ). Cho ví dụ minh họa. (0.5 đ)

Lời giải đề nghị

- Khái niệm đa hình:
 - + Đa hình: Là hiện tượng các đối tượng thuộc các lớp khác nhau có khả năng hiểu cùng một thông điệp theo các cách khác nhau.
- Ví dụ minh họa:
 - + Nhận được cùng một thông điệp “nhảy”, một con kangaroo và một con cóc nhảy theo hai kiểu khác nhau: chúng cùng có hành vi “nhảy” nhưng các hành vi này có nội dung khác nhau.

11.02.02 Lời giải đề nghị câu 2

Định nghĩa lớp CDate biểu diễn khái niệm ngày, tháng, năm (0.5 đ) với phép toán ++ (thêm bớt một ngày) theo dạng prefix ++a và postfix a++ (1 đ). Phép toán <<, >> đề xuất, nhập dữ liệu loại CDate (1 đ).

Lời giải đề nghị

Chương trình xây dựng lớp CDate hoàn chỉnh:

```
01977. #include <iostream>
01978. using namespace std;
01979.
01980. class CDate
01981. {
01982. private:
01983.     int Ngay;
01984.     int Thang;
01985.     int Nam;
```

```

01986. public:
01987.     CDate();
01988.     CDate(int,int,int);
01989.
01990.     friend istream& operator>>(istream&, CDate&);
01991.     friend ostream& operator<<(ostream&, CDate&);
01992.
01993.     int SoNgayToiDaTrongThang();
01994.     int SoNgayToiDaTrongNam();
01995.     int SoThuTuTrongNam();
01996.     int SoThuTu();
01997.     friend CDate TimNgay(int, int);
01998.     friend CDate TimNgay(int);
01999.     int ktNhuan();
02000.
02001.     CDate& operator++();
02002.     CDate operator++(int);
02003. };
02004.
02005. void main()
02006. {
02007.     CDate ng1;
02008.     cin >> ng1;
02009.     cout << ng1;
02010.
02011.     cout << "\nNgay ke tiep";
02012.     ng1++;
02013.     cout << ng1;
02014. }
02015.
02016. CDate::CDate()
02017. {
02018.     Ngay = 1;
02019.     Thang = 1;
02020.     Nam = 1;
02021. }
02022.
02023. CDate::CDate(int NgayNgay, int ThangThang, int
NamNam)
02024. {
02025.     Ngay = NgayNgay;
02026.     Thang = ThangThang;
02027.     Nam = NamNam;

```

```

02028.     }
02029.
02030.     istream& operator>>(istream& is, CDate& x)
02031.     {
02032.         cout << "\nNhap ngay: ";
02033.         is >> x.Ngay;
02034.         cout << "Nhap thang: ";
02035.         is >> x.Thang;
02036.         cout << "Nhap nam: ";
02037.         is >> x.Nam;
02038.         return is;
02039.     }
02040.
02041.     ostream& operator<<(ostream& os, CDate& x)
02042.     {
02043.         cout << "\nNgay: " << x.Ngay;
02044.         cout << "\nThang: " << x.Thang;
02045.         cout << "\nNam: " << x.Nam;
02046.         cout << endl;
02047.         return os;
02048.     }
02049.
02050.     int CDate::ktNhuan()
02051.     {
02052.         if (Nam % 4 == 0 && Nam % 100 != 0)
02053.             return 1;
02054.         if (Nam % 400 == 0)
02055.             return 1;
02056.         return 0;
02057.     }
02058.
02059.     int CDate::SoNgayToiDaTrongThang()
02060.     {
02061.         int Ngaythang[12] = { 31, 28, 31, 30, 31, 30,
02062.                               31, 31, 30, 31, 30, 31 };
02063.         if (this->ktNhuan())
02064.             Ngaythang[1] = 29;
02065.         return Ngaythang[Thang - 1];
02066.     }
02067.
02068.     int CDate::SoNgayToiDaTrongNam()
02069.     {
02070.         if (this->ktNhuan())

```

```

02071.         return 366;
02072.     return 365;
02073. }
02074.
02075. int CDate::SoThuTuTrongNam()
02076. {
02077.     int stt = 0;
02078.     for (int i = 1; i <= Thang - 1; i++)
02079.     {
02080.         CDate temp = { 1,i, Nam };
02081.         stt += temp.SoNgayToiDaTrongThang();
02082.     }
02083.     return (stt + Ngay);
02084. }
02085.
02086. int CDate::SoThuTu()
02087. {
02088.     int stt = 0;
02089.     for (int i = 1; i <= Nam - 1; i++)
02090.     {
02091.         CDate temp = { 1,1,i };
02092.         stt = stt + temp.SoNgayToiDaTrongNam();
02093.     }
02094.     return (stt + this->SoThuTuTrongNam());
02095. }
02096.
02097. CDate TimNgay(int nam, int stt)
02098. {
02099.     CDate temp = { 1,1,nam };
02100.     temp.Thang = 1;
02101.     while (stt - temp.SoNgayToiDaTrongThang() > 0)
02102.     {
02103.         stt = stt - temp.SoNgayToiDaTrongThang();
02104.         temp.Thang++;
02105.     }
02106.     temp.Ngay = stt;
02107.     return temp;
02108. }
02109.
02110.
02111. CDate TimNgay(int stt)
02112. {
02113.     int nam = 1;

```



```

02114.    int sn = 365;
02115.    while (stt - sn > 0)
02116.    {
02117.        stt = stt - sn;
02118.        nam++;
02119.        CDate temp(1, 1, nam);
02120.        sn = temp.SoNgayToiDaTrongNam();
02121.    }
02122.    return TimNgay(nam, stt);
02123. }
02124.
02125. CDate& CDate::operator++()
02126. {
02127.     int stt = this->SoThuTu();
02128.     stt = stt + 1;
02129.     *this = TimNgay(stt);
02130.     return *this;
02131. }
02132.
02133. CDate CDate::operator++(int x)
02134. {
02135.     CDate temp;
02136.     temp = *this;
02137.     int stt = this->SoThuTu();
02138.     stt = stt + 1;
02139.     *this = TimNgay(stt);
02140.     return temp;
02141. }

```

11.02.03 Lời giải đề nghị câu 3

11.02.03.1 Sơ đồ lớp sơ bộ

11.02.03.2 Sơ đồ lớp chi tiết

11.02.03.3 Khai báo lớp

11.02.03.4 Định nghĩa phương thức

```
02142.  as
```

CHƯƠNG 12. ĐỀ - LỜI GIẢI ĐỀ NGHỊ HỌC KỲ 2 NĂM HỌC 2019 - 2020

12.01 ĐỀ THI

12.01.01 Câu 1 – (2 điểm)

a. Hãy trình bày những đặc điểm của tính đóng gói (encapsulation) trong lập trình hướng đối tượng. Trường hợp nào có thể vi phạm tính đóng gói? Cho ví dụ minh họa (1đ).

b. Hãy trình bày những ưu điểm của kế thừa trong việc lập trình hướng đối tượng và cho ví dụ minh họa (1đ).

12.01.02 Câu 2 – (3 điểm)

Cho lớp phân số (CPhanSo). Hãy khai báo và định nghĩa các phương thức cần thiết để các đối tượng thuộc lớp CPhanSo có thể thực hiện được các câu lệnh sau:

```
02143. CPhanSo a(5, 3);  
02144. CPhanSo b, c, kq;  
02145. cin >> b >> c;  
02146. kq = a + b + 5 + c;  
02147. cout << "Kết quả là: " << kq;  
02148. if (a == b)  
02149.     cout << "Phan so a bang phan so b" <<endl;
```

12.01.03 Câu 3 – (5 điểm)

Trước hết phải khẳng định, đất đai là nguồn tài nguyên vô cùng quý giá, là tài sản quan trọng của quốc gia, là tư liệu sản xuất, ... Đặc biệt, đất đai là điều kiện cần cho mọi hoạt động sản xuất và đời sống. Ở nước ta, khi còn nhiều sống nhờ vào nông nghiệp, thì đất đai càng trở thành nguồn lực rất quan trọng.

Muốn phát huy tác dụng của nhiều nguồn lực đất đai, ngoài việc bảo vệ quỹ đất của quốc gia, còn phải quản lý đất đai hợp lý, nâng cao hiệu quả

sử dụng đất sao cho vừa bảo đảm được lợi ích trước mắt, vừa tạo điều kiện sử dụng đất hiệu quả lâu dài để phát triển bền vững đất nước.

Hiện nay, ở Việt Nam đất đai được phân chia thành 2 loại chính sau:

- Đất nông nghiệp
- Đất phi nông nghiệp (đất ở)

Quan điểm nhất quán của Đảng, Nhà nước và nhân dân ta đã được xác định từ năm 1980 đến nay là đất đai thuộc sở hữu toàn dân, do Nhà nước đại diện chủ sở hữu và thống nhất quản lý. Để góp phần nâng cao hiệu quả quản lý nhà nước về đất đai, mỗi thửa đất được nhà nước quản lý và **cấp quyền sử dụng** cho một hoặc nhiều người dân (nhà nước cho phép nhiều người dân có thể đồng sở hữu quyền sử dụng đất) có nhu cầu sử dụng (**Giấy chứng nhận quyền sử dụng đất** hay còn được gọi là **Sổ hồng**).

- Với các **thửa đất nông nghiệp**, thông tin cần quản lý gồm: sổ giấy chứng nhận (chuỗi), người sở hữu quyền sử dụng đất (gồm họ và tên, năm sinh, CMND, địa chỉ thường trú), sổ thửa đất, sổ tờ bản đồ, địa chỉ thửa đất, diện tích (m²), thời hạn sử dụng (được sử dụng đến năm nào), ngày cấp, đơn giá thuế phải đóng cho nhà nước hàng năm/1m².
- Với các **thửa đất phi nông nghiệp (đất ở)**, thông tin cần quản lý gồm: sổ giấy chứng nhận (chuỗi), người sở hữu quyền sử dụng đất (gồm họ và tên, năm sinh, CMND, địa chỉ thường trú), sổ thửa đất, sổ tờ bản đồ, địa chỉ thửa đất, diện tích (m²), ngày cấp, đơn giá thuế phải đóng cho nhà nước hàng năm/1m².

Áp dụng kiến thức lập trình hướng đối tượng (kế thừa, đa hình) thiết kế sơ đồ chi tiết các lớp đối tượng (**1.0đ**) và khai báo các lớp (**1.0đ**) để xây dựng chương trình thực hiện các yêu cầu sau:

1. Tạo danh sách các giấy chứng nhận quyền sử dụng đất mà nhà nước đã cấp cho người dân. (**1.0đ**)
2. Tính tiền thuế mà người sử dụng đất phải đóng cho nhà nước và cho biết thửa đất nào (thông tin thửa đất) có tiền thuế phải đóng nhiều nhất. (**1.0đ**)
3. Xuất ra màn hình thông tin các thửa đất nông nghiệp đã hết thời hạn sử dụng (năm sử dụng < năm hiện tại). (**1.0 đ**)

Lưu ý: Các thông tin trong đề chỉ mô phỏng các thông tin với mục tiêu để sinh viên vận dụng kiến thức lập trình hướng đối tượng. Do vậy, các thông tin trong đề **KHÔNG** nhất thiết phải đúng hoặc khớp với các thông tin hiện tại trong thế giới thực. Sinh viên cần bám sát các mô tả trong đề thi để làm bài.

12.02 ĐÁP ÁN

12.02.01 Lời giải đề nghị câu 1

a. Hãy trình bày những đặc điểm của tính đóng gói (encapsulation) trong lập trình hướng đối tượng. Trường hợp nào có thể vi phạm tính đóng gói? Cho ví dụ minh họa (1đ).

- Đặc điểm của tính đóng gói
 - + Đóng gói: Nhóm những gì có liên quan với nhau vào làm một, để sau này có thể dùng một cái tên để gọi đến.
 - + Che dấu thông tin: đóng gói để che một số thông tin và chỉ tiết cái đặt nội bộ để bên ngoài không nhìn thấy.
- Trường hợp có thể vi phạm tính đóng gói: Khai báo các thuộc tính trong lớp trong phạm vi từ khóa private.
- Ví dụ minh họa:

```
02150. class PhanSo
02151. {
02152. public:
02153.     int ts, ms;
02154. };
```

b. Hãy trình bày những ưu điểm của kế thừa trong việc lập trình hướng đối tượng và cho ví dụ minh họa (1đ).

- Ưu điểm của kế thừa trong lập trình
 - + Kế thừa cho phép xây dựng lớp mới từ lớp đã có.
 - + Kế thừa cho phép tổ chức các lớp chia sẻ mã chương trình chung, nhờ vậy có thể dễ dàng sửa chữa, nâng cấp hệ thống.
 - + Trong C++, kế thừa còn định nghĩa sự tương thích, nhờ đó ta có cơ chế chuyển kiểu tự động.
- Ví dụ minh họa

```
02155. class CTamGiac
02156. {
02157.     ...
02158. };
02159.
02160. class CTamGiacCan : public CTamGiac
02161. {
02162.     ...
02163. };
```

12.02.02 Lời giải đề nghị câu 2

Cho lớp phân số (CPhanSo). Hãy khai báo và định nghĩa các phương thức cần thiết để các đối tượng thuộc lớp CPhanSo có thể thực hiện được các câu lệnh sau:

```
02164. CPhanSo a(5, 3);
02165. CPhanSo b, c, kq;
02166. cin >> b >> c;
02167. kq = a + b + 5 + c;
02168. cout << "Kết quả là: " << kq;
02169. if (a == b)
02170.     cout << "Phan so a bang phan so b" <<endl;
```

Lời giải đề nghị

Chương trình xây dựng lớp CPhanSo hoàn chỉnh:

```
02171. #include <iostream>
02172. using namespace std;
02173.
02174. class CPhanSo
02175. {
02176. private:
02177.     int tu;
02178.     int mau;
02179. public:
02180.     CPhanSo();
02181.     CPhanSo(int);
02182.     CPhanSo(int, int);
02183.
02184.     friend istream& operator >> (istream&, CPhanSo&);
02185.     friend ostream& operator << (ostream&, CPhanSo&);
02186.
02187.     CPhanSo operator+(CPhanSo);
02188.     CPhanSo operator-(CPhanSo&);
02189.     int operator==(CPhanSo&);
02190. };
02191.
02192. int main()
02193. {
02194.     CPhanSo a(5, 3);
02195.     CPhanSo b, c, kq;
02196.     cin >> b >> c;
02197.     kq = a + b + 5 + c;
```

```

02198.         cout << "Kết quả là: " << kq;
02199.         if (a == b)
02200.             cout << "Phan so a bang phan so b" << endl;
02201.         return 1;
02202.     }
02203.
02204.     CPhanSo::CPhanSo()
02205.     {
02206.         tu = 0;
02207.         mau = 0;
02208.     }
02209.
02210.     CPhanSo::CPhanSo(int tutu)
02211.     {
02212.         tu = tutu;
02213.         mau = 1;
02214.     }
02215.
02216.     CPhanSo::CPhanSo(int tutu, int maumau)
02217.     {
02218.         tu = tutu;
02219.         mau = maumau;
02220.     }
02221.
02222.     istream& operator>>(istream& is, CPhanSo& x)
02223.     {
02224.         cout << "\nNhap tu: ";
02225.         is >> x.tu;
02226.         cout << "Nhap mau: ";
02227.         is >> x.mau;
02228.         return is;
02229.     }
02230.
02231.     ostream& operator<<(ostream& os, CPhanSo& x)
02232.     {
02233.         cout << "\nTu: " << x.tu;
02234.         cout << "\nMau: " << x.mau;
02235.         return os;
02236.     }
02237.
02238.     CPhanSo CPhanSo::operator+(CPhanSo x)
02239.     {
02240.         CPhanSo temp;

```

```

02241.      temp.tu = tu * x.mau + mau * x.tu;
02242.      temp.mau = mau * x.mau;
02243.      return temp;
02244.  }
02245.
02246.  CPhanSo CPhanSo::operator-(CPhanSo& x)
02247.  {
02248.      CPhanSo temp;
02249.      temp.tu = tu * x.mau - mau * x.tu;
02250.      temp.mau = mau * x.mau;
02251.      return temp;
02252.  }
02253.
02254.  int CPhanSo::operator==(CPhanSo& x)
02255.  {
02256.      CPhanSo temp = *this - x;
02257.      if (temp.tu == 0)
02258.          return 1;
02259.      return 0;
02260.  }

```

12.02.03 Lời giải đề nghị câu 3

12.02.03.1 Sơ đồ lớp sơ bộ

12.02.03.2 Sơ đồ lớp chi tiết

12.02.03.3 Khai báo lớp

12.02.03.4 Định nghĩa phương thức

```
02261.  as
```