

HƯỚNG DẪN BÀI THỰC HÀNH TẠO DẪN XUẤT VÀ THỪA KẾ

1. Xây dựng class Điểm 2 chiều (DIEM):

```
Diem.h  X
DIEM (Global Scope)
1  #pragma once
2  #include<iostream>
3  using namespace std;
4  #ifndef _DIEM
5  class DIEM
6  {
7  protected:
8      double x, y;
9      static int dem;
10 public:
11     ~DIEM();
12     DIEM(double = 0, double = 0);
13     DIEM(const DIEM&);
14     double GetX() const;
15     double GetY() const;
16     static int GetDem();
17     void SetX(double);
18     void SetY(double);
19     void SetXY(double, double);
20     void Nhap();
21     void Xuat() const;
22     void DiChuyen(double, double);
23     bool KiemTraTrung(const DIEM&) const;
24     double TinhKhoangCach(const DIEM&) const;
25     DIEM TimDoiXung() const;
26     bool KiemTraTamGiacHopLe(const DIEM&, const DIEM&) const;
27     double TinhChuViTamGiac(const DIEM&, const DIEM&) const;
28     double TinhDienTichTamGiac(const DIEM&, const DIEM&) const;
29     string PhanLoaiTamGiac(const DIEM&, const DIEM&) const;
30     friend istream& operator>>(istream&, DIEM&);
31     friend ostream& operator<<(ostream&, const DIEM&);
32 };
33 #endif

XuLyDiem.cpp  X
DIEM → DIEM
1  #include "Diem.h"
2  #define epsilon 0.000000001
3  int DIEM::dem = 0;
4  DIEM::~DIEM()
5  {
6      cout << "Da huy 1 diem 2C => Con " << --dem << " diem 2C\n";
7  }
8  DIEM::DIEM(double x, double y): x(x), y(y)
9  {
10     cout << "Da tao 1 diem 2C => Co " << ++dem << " diem 2C\n";
11 }
12 DIEM::DIEM(const DIEM& d) : x(d.x), y(d.y)
13 {
14     cout << "Da tao 1 diem 2C => Co " << ++dem << " diem 2C\n";
15 }
16 double DIEM::GetX() const
17 {
18     return x;
19 }
20 double DIEM::GetY() const
21 {
22     return y;
23 }
24 int DIEM::GetDem()
25 {
26     return dem;
27 }
28 void DIEM::SetX(double x)
29 {
30     this->x = x;
31 }
32 void DIEM::SetY(double y)
33 {
34     this->y = y;
35 }
36 void DIEM::SetXY(double x, double y)
37 {
38     this->x = x;
39     this->y = y;
40 }
```

```

XulyDiem.cpp  X
DIEM  DIEM
41 void DIEM::Nhap()
42 {
43     cout << "Nhap hoành do: ";
44     cin >> x;
45     cout << "Nhap tung do: ";
46     cin >> y;
47 }
48 void DIEM::Xuat() const
49 {
50     cout << x << ", " << y;
51 }
52 void DIEM::DiChuyen(double dx, double dy)
53 {
54     x += dx;
55     y += dy;
56 }
57 bool DIEM::KiemTraTrung(const DIEM& d) const
58 {
59     return x == d.x && y == d.y;
60 }
61 double DIEM::TinhKhoangCach(const DIEM& d) const
62 {
63     return sqrt(pow(x - d.x, 2) + pow(y - d.y, 2));
64 }
65 DIEM DIEM::TimDoiXung() const
66 {
67     return DIEM(-x == 0 ? x : -x, -y == 0 ? y : -y);
68 }
69 bool DIEM::KiemTraTamGiacHopLe(const DIEM& d1, const DIEM& d2) const
70 {
71     double a = this->TinhKhoangCach(d1);
72     double b = d1.TinhKhoangCach(d2);
73     double c = d2.TinhKhoangCach(*this);
74     return a > 0 && b > 0 && c > 0 && a + b > c && b + c > a && a + c > b;
75 }
76 double DIEM::TinhChuViTamGiac(const DIEM& d1, const DIEM& d2) const
77 {
78     double a = this->TinhKhoangCach(d1);
79     double b = d1.TinhKhoangCach(d2);
80     double c = d2.TinhKhoangCach(*this);
81     return a + b + c; //hoặc: return KiemTraTamGiacHopLe(d1, d2) ? a + b + c : 0;

```

```

XulyDiem.cpp  X
DIEM  (Global Scope)
83 double DIEM::TinhDienTichTamGiac(const DIEM& d1, const DIEM& d2) const
84 {
85     double a = this->TinhKhoangCach(d1);
86     double b = d1.TinhKhoangCach(d2);
87     double c = d2.TinhKhoangCach(*this);
88     double p = (a + b + c) / 2; //hoặc: double p = TinhChuViTamGiac(d1, d2) / 2;
89     return sqrt(p * (p - a) * (p - b) * (p - c));
90     //hoặc: return KiemTraTamGiacHopLe(d1, d2) ? sqrt(p * (p - a) * (p - b) * (p - c)) : 0;
91 }
92 string DIEM::PhanLoaiTamGiac(const DIEM& d1, const DIEM& d2) const
93 {
94     double a = this->TinhKhoangCach(d1);
95     double b = d1.TinhKhoangCach(d2);
96     double c = d2.TinhKhoangCach(*this);
97     string chuoikq = "";
98     if (!KiemTraTamGiacHopLe(d1, d2))
99         chuoikq = "TG không hop le";
100     else if (a == b && b == c)
101         chuoikq = "TG deu";
102     else if (a == b || b == c || c == a)
103         if (abs(a * a + b * b - c * c) <= epsilon || abs(a * a + c * c - b * b) <= epsilon || abs(b * b + c * c - a * a) <= epsilon)
104             chuoikq = "TG vuong can";
105         else
106             chuoikq = "TG can";
107     else if (abs(a * a + b * b - c * c) <= epsilon || abs(a * a + c * c - b * b) <= epsilon || abs(b * b + c * c - a * a) <= epsilon)
108         chuoikq = "TG vuong";
109     else
110         chuoikq = "TG thuong";
111     return chuoikq;
112 }
113 istream& operator>>(istream& is, DIEM& d)
114 {
115     //d.Nhap();
116     cout << "Nhap hoành do: ";
117     is >> d.x;
118     cout << "Nhap tung do: ";
119     is >> d.y;
120     return is;
121 }
122 ostream& operator<<(ostream& os, const DIEM& d)
123 {
124     //d.Xuat();
125     os << d.x << ", " << d.y;
126     return os;
127 }

```

2. Xây dựng class Điểm 3 chiều (DIEM3C) kế thừa từ class Điểm 2 chiều (DIEM), bổ sung thuộc tính cao độ z và các phương thức: thiết lập, hủy, set và get giá trị cao độ z, kiểm tra trùng điểm, di chuyển điểm, tính khoảng cách giữa 2 điểm, tìm điểm đối xứng, nhập, xuất, tính chu vi tam giác qua 3 điểm, tính diện tích tam giác qua 3 điểm, overload toán tử >>, overload toán tử <<.

```
Diem3C.h  X
DIEM  (Global Scope)
1  #pragma once
2  #ifndef _DIEM3C
3  #define _DIEM3C
4  #include "Diem.h"
5  class DIEM3C: public DIEM
6  {
7  protected:
8      double z;
9      static int dem;
10 public:
11     ~DIEM3C();
12     /*DIEM3C();
13     DIEM3C(double);
14     DIEM3C(double, double);
15     DIEM3C(double, double, double);*/
16     DIEM3C(double = 0, double = 0, double = 0);
17     DIEM3C(const DIEM3C&);
18     double GetZ() const;
19     void SetZ(double);
20     void SetXYZ(double, double, double);
21     static int GetDem();
22     void Nhap();
23     void Xuat() const;
24     bool KiemTraTrung(const DIEM3C&) const;
25     void DiChuyen(double, double, double);
26     double TinhKhoangCach(const DIEM3C&) const;
27     DIEM3C TimDoiXung() const;
28     friend istream& operator>>(istream&, DIEM3C&);
29     friend ostream& operator<<(ostream&, const DIEM3C&);
30     //SV tự làm thêm 2 phương thức: tính chu vi và diện tích tam giác qua 3 điểm 3 chiều
31 };
```

```
XuLyDiem3C.cpp  X
DIEM  → DIEM3C
1  #include "Diem3C.h"
2  int DIEM3C::dem = 0;
3  DIEM3C::~DIEM3C()
4  {
5      cout << "Da huy 1 diem 3C => Con " << --dem << " diem 3C\n";
6  }
7  //DIEM3C::DIEM3C() :DIEM(0, 0), z(0)
8  //{
9  //    cout << "Da tao 1 diem 3C => Co " << ++dem << " diem 3C\n";
10 //}
11 //DIEM3C::DIEM3C(double x): DIEM(x, 0), z(0)
12 //{
13 //    cout << "Da tao 1 diem 3C => Co " << ++dem << " diem 3C\n";
14 //}
15 //DIEM3C::DIEM3C(double x, double y): DIEM(x, y), z(0)
16 //{
17 //    cout << "Da tao 1 diem 3C => Co " << ++dem << " diem 3C\n";
18 //}
19 //DIEM3C::DIEM3C(double x, double y, double z): DIEM(x, y), z(z)
20 //{
21 //    cout << "Da tao 1 diem 3C => Co " << ++dem << " diem 3C\n";
22 //}
23 DIEM3C::DIEM3C(double x, double y, double z) : DIEM(x, y), z(z)
24 {
25     cout << "Da tao 1 diem 3C => Co " << ++dem << " diem 3C\n";
26 }
27 DIEM3C::DIEM3C(const DIEM3C& d3c) : DIEM(d3c.x, d3c.y), z(d3c.z)
28 {
29     cout << "Da tao 1 diem 3C => Co " << ++dem << " diem 3C\n";
30     //dem++;
31 }
```

```
XuLyDiem3C.cpp ➤ X
DIEM ➔ DIEM3C
32 double DIEM3C::GetZ() const
33 {
34     return z;
35 }
36 void DIEM3C::SetZ(double z)
37 {
38     this->z = z;
39 }
40 void DIEM3C::SetXYZ(double x, double y, double z)
41 {
42     SetXY(x, y);
43     SetZ(z);
44 }
45 int DIEM3C::GetDem()
46 {
47     return dem;
48 }
49 void DIEM3C::Nhap()
50 {
51     DIEM::Nhap();
52     cout << "Nhap cao do: ";
53     cin >> z;
54 }
55 void DIEM3C::Xuat() const
56 {
57     DIEM::Xuat();
58     cout << ", " << z;
59 }
60 bool DIEM3C::KiemTraTrung(const DIEM3C& d3c) const
61 {
62     return x == d3c.x && y == d3c.y && z == d3c.z;
63     //Hoặc: return DIEM::KiemTraTrung(d3c) && z == d3c.z;
64 }
65 void DIEM3C::DiChuyen(double dx, double dy, double dz)
66 {
67     DIEM::DiChuyen(dx, dy);
68     z += dz;
69 }
```

```
XuLyDiem3C.cpp ➤ X
DIEM (Global Scope)
70 double DIEM3C::TinhKhoangCach(const DIEM3C& d3c) const
71 {
72     return sqrt(pow(x - d3c.x, 2) + pow(y - d3c.y, 2) + pow(z - d3c.z, 2));
73 }
74 }
75 DIEM3C DIEM3C::TimDoiXung() const
76 {
77     return DIEM3C(-x == 0 ? x : -x, -y == 0 ? y : -y, -z == 0 ? z : -z);
78 }
79 istream& operator>>(istream& is, DIEM3C& d3c)
80 {
81     d3c.DIEM::Nhap();
82     cout << "Nhap cao do: ";
83     is >> d3c.z;
84     return is;
85 }
86 //Hoặc:
87 // d3c.Nhap();
88 // return is;
89 ostream& operator<<(ostream& os, const DIEM3C& d3c)
90 {
91     d3c.DIEM::Xuat();
92     cout << ", " << d3c.z;
93     return os;
94 }
95 //Hoặc:
96 // d3c.Xuat();
97 // return os;
```

```

main.cpp  X
DIEM (Global Scope)
1  #include "Diem3C.h"
2
3  void main()
4  {
5      //Test Diem 3C:
6      cout << "So diem 3C hien tai la: " << DIEM3C::GetDem() << endl;
7      DIEM3C d1, d2(1), d3(1, 2), d4(1, 2, 3), *d5 = new DIEM3C;
8      cout << "So diem 3C hien tai la: " << DIEM3C::GetDem() << endl;
9      cout << "d1: " << d1 << "\nd2: " << d2 << "\nd3: " << d3 << "\nd4: " << d4 << "\nd5: " << *d5 << endl;
10     cout << "Nhap lai toa do diem d1:\n";
11     cin >> d1;
12     cout << "d1_moi: " << d1 << endl;
13     //-----
14     if (d1.DIEM::KiemTraTrung(d4))
15         cout << "d1 va d4 trung hoan do va tung do\n";
16     else
17         cout << "d1 va d4 khong trung hoan do va tung do\n";
18     //-----
19     if (d1.KiemTraTrung(d4))
20         cout << "d1 va d4 trung hoan do, tung do va cao do\n";
21     else
22         cout << "d1 va d4 khong trung hoan do, tung do va cao do\n";
23     //-----
24     delete d5;
25     cout << "So diem 3C hien tai la: " << DIEM3C::GetDem() << endl;
26 }

```

```

So diem 3C hien tai la: 0
Da tao 1 diem 2C => Co 1 diem 2C
Da tao 1 diem 3C => Co 1 diem 3C
Da tao 1 diem 2C => Co 2 diem 2C
Da tao 1 diem 3C => Co 2 diem 3C
Da tao 1 diem 2C => Co 3 diem 2C
Da tao 1 diem 3C => Co 3 diem 3C
Da tao 1 diem 2C => Co 4 diem 2C
Da tao 1 diem 3C => Co 4 diem 3C
Da tao 1 diem 2C => Co 5 diem 2C
Da tao 1 diem 3C => Co 5 diem 3C
So diem 3C hien tai la: 5
d1: 0, 0, 0
d2: 1, 0, 0
d3: 1, 2, 0
d4: 1, 2, 3
d5: 0, 0, 0
Nhap lai toa do diem d1:
Nhap hoan do: 1
Nhap tung do: 2
Nhap cao do: -3
d1_moi: 1, 2, -3
d1 va d4 trung hoan do va tung do
d1 va d4 khong trung hoan do, tung do va cao do
Da huy 1 diem 3C => Con 4 diem 3C
Da huy 1 diem 2C => Con 4 diem 2C
So diem 3C hien tai la: 4
Da huy 1 diem 3C => Con 3 diem 3C
Da huy 1 diem 2C => Con 3 diem 2C
Da huy 1 diem 3C => Con 2 diem 3C
Da huy 1 diem 2C => Con 2 diem 2C
Da huy 1 diem 3C => Con 1 diem 3C
Da huy 1 diem 2C => Con 1 diem 2C
Da huy 1 diem 3C => Con 0 diem 3C
Da huy 1 diem 2C => Con 0 diem 2C

```

3. Xây dựng class Điểm 3 chiều có màu (DIEMMAU3C) kế thừa từ class Điểm 3 chiều (DIEM3C), bổ sung 3 thuộc tính màu RGB - Red Green Blue (3 số nguyên từ 0 đến 255) và các phương thức: thiết lập, hủy, set và get giá trị màu, kiểm tra trùng màu, nhập, xuất, overload toán tử >>, overload toán tử <<.

```

DiemMau3C.h  + X
DIEM  (Global Scope)
1  #pragma once
2  #ifndef _DIEMMAU3C
3  #define _DIEMMAU3C
4  #include "Diem3C.h"
5  class DIEMMAU3C: public DIEM3C
6  {
7  protected:
8      int r, g, b; //[0, 255]
9      static int dem;
10 public:
11     ~DIEMMAU3C();
12     DIEMMAU3C(double = 0, double = 0, double = 0, int = 0, int = 0, int = 0);
13     int GetR() const;
14     int GetG() const;
15     int GetB() const;
16     void SetR(int);
17     void SetG(int);
18     void SetB(int);
19     void SetRGB(int, int, int);
20     static int GetDem();
21     void Nhap();
22     void Xuat() const;
23     bool KiemTraTrung(const DIEMMAU3C&) const;
24     friend istream& operator>>(istream&, DIEMMAU3C&);
25     friend ostream& operator<<(ostream&, const DIEMMAU3C&);
26 };
27 #endif

```

```

XulyDiemMau3C.cpp  + X  DIEMMAU3C  SetG(i
DIEM
1  #include "DiemMau3C.h"
2  int DIEMMAU3C::dem = 0;
3  DIEMMAU3C::~DIEMMAU3C()
4  {
5      cout << "Da huy 1 diem mau 3C => Con " << --dem << " diem mau 3C\n";
6  }
7  DIEMMAU3C::DIEMMAU3C(double x, double y, double z, int r, int g, int b) : DIEM3C(x, y, z)
8  {
9      SetRGB(r, g, b);
10     cout << "Da tao 1 diem mau 3C => Co " << ++dem << " diem mau 3C\n";
11 }
12 int DIEMMAU3C::GetR() const
13 {
14     return r;
15 }
16 int DIEMMAU3C::GetG() const
17 {
18     return g;
19 }
20 int DIEMMAU3C::GetB() const
21 {
22     return b;
23 }
24 void DIEMMAU3C::SetR(int r)
25 {
26     while (r<0||r>255)
27     {
28         cout << "Nhap lai thong so mau Red tu 0 den 255: ";
29         cin >> r;
30     }
31     this->r = r;
32 }

```

```

XuLyDiemMau3C.cpp -> X
DIEM -> DIEMMAU3C
33 void DIEMMAU3C::SetG(int g)
34 {
35     while (g < 0 || g>255)
36     {
37         cout << "Nhap lai thong so mau Green tu 0 den 255: ";
38         cin >> g;
39     }
40     this->g = g;
41 }
42 void DIEMMAU3C::SetB(int b)
43 {
44     while (b < 0 || b>255)
45     {
46         cout << "Nhap lai thong so mau Blue tu 0 den 255: ";
47         cin >> b;
48     }
49     this->b = b;
50 }
51 void DIEMMAU3C::SetRGB(int r, int g, int b)
52 {
53     SetR(r);
54     SetG(g);
55     SetB(b);
56 }
57 int DIEMMAU3C::GetDem()
58 {
59     return dem;
60 }
61 void DIEMMAU3C::Nhap()
62 {
63     DIEM3C::Nhap();
64     do
65     {
66         cout << "Nhap 3 thong so mau RGB tu 0 den 255: ";
67         cin >> r >> g >> b;
68     } while (r < 0 || r>255 || g < 0 || g>255 || b < 0 || b>255);
69 }

```

```

XuLyDiemMau3C.cpp -> X
DIEM -> DIEMMAU3C -> Nhap()
70 void DIEMMAU3C::Xuat() const
71 {
72     DIEM3C::Xuat();
73     cout << " - Mau RGB(" << r << ", " << g << ", " << b << ")";
74 }
75 bool DIEMMAU3C::KiemTraTrung(const DIEMMAU3C& dm3c) const
76 {
77     return DIEM3C::KiemTraTrung(dm3c) && r == dm3c.r && g == dm3c.g && b == dm3c.b;
78 }
79 istream& operator>>(istream& is, DIEMMAU3C& dm3c)
80 {
81     dm3c.DIEM3C::Nhap();
82     do
83     {
84         cout << "Nhap 3 thong so mau RGB tu 0 den 255: ";
85         is >> dm3c.r >> dm3c.g >> dm3c.b;
86     } while (dm3c.r < 0 || dm3c.r>255 || dm3c.g < 0 || dm3c.g>255 || dm3c.b < 0 || dm3c.b>255);
87     return is;
88 }
89 ostream& operator<<(ostream& os, const DIEMMAU3C& dm3c)
90 {
91     dm3c.DIEM3C::Xuat();
92     os << " - Mau RGB(" << dm3c.r << ", " << dm3c.g << ", " << dm3c.b << ")";
93     return os;
94 }

```

```

So diem mau 3C hien tai la: 0
Da tao 1 diem 2C => Co 1 diem 2C
Da tao 1 diem 3C => Co 1 diem 3C
Da tao 1 diem mau 3C => Co 1 diem mau 3C
Da tao 1 diem 2C => Co 2 diem 2C
Da tao 1 diem 3C => Co 2 diem 3C
Da tao 1 diem mau 3C => Co 2 diem mau 3C
Da tao 1 diem 2C => Co 3 diem 2C
Da tao 1 diem 3C => Co 3 diem 3C
Da tao 1 diem mau 3C => Co 3 diem mau 3C
Da tao 1 diem 2C => Co 4 diem 2C
Da tao 1 diem 3C => Co 4 diem 3C
Da tao 1 diem mau 3C => Co 4 diem mau 3C
Da tao 1 diem 2C => Co 5 diem 2C
Da tao 1 diem 3C => Co 5 diem 3C
Da tao 1 diem mau 3C => Co 5 diem mau 3C
Da tao 1 diem 2C => Co 6 diem 2C
Da tao 1 diem 3C => Co 6 diem 3C
Da tao 1 diem mau 3C => Co 6 diem mau 3C
Da tao 1 diem 2C => Co 7 diem 2C
Da tao 1 diem 3C => Co 7 diem 3C
Nhap lai thong so mau Blue tu 0 den 255: 100
Da tao 1 diem mau 3C => Co 7 diem mau 3C
Da tao 1 diem 2C => Co 8 diem 2C
Da tao 1 diem 3C => Co 8 diem 3C
Da tao 1 diem mau 3C => Co 8 diem mau 3C
So diem mau 3C hien tai la: 8
d1: 0, 0, 0 - Mau RGB(0, 0, 0)
d2: 1, 0, 0 - Mau RGB(0, 0, 0)
d3: 1, 2, 0 - Mau RGB(0, 0, 0)
d4: 1, 2, 3 - Mau RGB(0, 0, 0)
d5: 1, 2, 3 - Mau RGB(0, 0, 0)
d6: 1, 2, 3 - Mau RGB(0, 50, 0)
d7: 1, 2, 3 - Mau RGB(0, 50, 100)
d8: 0, 0, 0 - Mau RGB(0, 0, 0)
Nhap lai toa do diem mau 3C d1:
Nhap hoành do: 1
Nhap tung do: 2
Nhap cao do: 3
Nhap 3 thong so mau RGB tu 0 den 255: 200 250 255
d1_moi: 1, 2, 3 - Mau RGB(200, 250, 255)
d1 va d7 trung hoành do va tung do
d1 va d7 trung hoành do, tung do va cao do
d1 va d7 khong trung hoành do, tung do, cao do va mau
Da huy 1 diem mau 3C => Con 7 diem mau 3C
Da huy 1 diem 3C => Con 7 diem 3C
Da huy 1 diem 2C => Con 7 diem 2C
So diem 3C hien tai la: 7
Da huy 1 diem mau 3C => Con 6 diem mau 3C
Da huy 1 diem 3C => Con 6 diem 3C
Da huy 1 diem 2C => Con 6 diem 2C
Da huy 1 diem mau 3C => Con 5 diem mau 3C
Da huy 1 diem 3C => Con 5 diem 3C
Da huy 1 diem 2C => Con 5 diem 2C
Da huy 1 diem mau 3C => Con 4 diem mau 3C
Da huy 1 diem 3C => Con 4 diem 3C
Da huy 1 diem 2C => Con 4 diem 2C
Da huy 1 diem mau 3C => Con 3 diem mau 3C
Da huy 1 diem 3C => Con 3 diem 3C
Da huy 1 diem 2C => Con 3 diem 2C
Da huy 1 diem mau 3C => Con 2 diem mau 3C
Da huy 1 diem 3C => Con 2 diem 3C
Da huy 1 diem 2C => Con 2 diem 2C
Da huy 1 diem mau 3C => Con 1 diem mau 3C
Da huy 1 diem 3C => Con 1 diem 3C
Da huy 1 diem 2C => Con 1 diem 2C
Da huy 1 diem mau 3C => Con 0 diem mau 3C
Da huy 1 diem 3C => Con 0 diem 3C
Da huy 1 diem 2C => Con 0 diem 2C

```


4. Xây dựng class Điểm 2 chiều có màu (DIEMMAU) kế thừa từ 2 class Điểm 2 chiều (DIEM) và class Màu (MAU) và các phương thức xử lý liên quan.

```
Mau.h  + X
DIEM  (Global Scope)
1  #pragma once
2  #ifndef _MAU
3  #define _MAU
4  #include<iostream>
5  using namespace std;
6  class MAU
7  {
8  private:
9      int r, g, b; //[0, 255]
10     static int dem;
11 public:
12     ~MAU();
13     MAU(int = 0, int = 0, int = 0);
14     MAU(const MAU&);
15     int GetR() const;
16     int GetG() const;
17     int GetB() const;
18     static int GetDem();
19     void SetR(int);
20     void SetG(int);
21     void SetB(int);
22     void SetRGB(int, int, int);
23     void Nhap();
24     void Xuat() const;
25     bool KiemTraTrung(const MAU&) const;
26     friend istream& operator>>(istream&, MAU&);
27     friend ostream& operator<<(ostream&, const MAU&);
28 };
29 #endif
```

```
XuLyMau.cpp  + X
DIEM  MAU
1  #include "Mau.h"
2  int MAU::dem = 0;
3  MAU::~MAU()
4  {
5      cout << "Da huy 1 mau => Con " << --dem << " mau\n";
6  }
7  MAU::MAU(int r, int g, int b)
8  {
9      SetRGB(r, g, b);
10     cout << "Da tao 1 mau => Co " << ++dem << " mau\n";
11 }
12 MAU::MAU(const MAU& m)
13 {
14     SetRGB(m.r, m.g, m.b);
15     cout << "Da tao 1 mau => Co " << ++dem << " mau\n";
16 }
17 int MAU::GetR() const
18 {
19     return r;
20 }
21 int MAU::GetG() const
22 {
23     return g;
24 }
25 int MAU::GetB() const
26 {
27     return b;
28 }
29 int MAU::GetDem()
30 {
31     return dem;
32 }
```

```
XuLyMau.cpp  X
DIEM  MAU
33 void MAU::SetR(int r)
34 {
35     while (r < 0 || r>255)
36     {
37         cout << "Nhap lai thong so mau Red tu 0 den 255: ";
38         cin >> r;
39     }
40     this->r = r;
41 }
42 void MAU::SetG(int g)
43 {
44     while (g < 0 || g>255)
45     {
46         cout << "Nhap lai thong so mau Green tu 0 den 255: ";
47         cin >> g;
48     }
49     this->g = g;
50 }
51 void MAU::SetB(int b)
52 {
53     while (b < 0 || b>255)
54     {
55         cout << "Nhap lai thong so mau Blue tu 0 den 255: ";
56         cin >> b;
57     }
58     this->b = b;
59 }
60 void MAU::SetRGB(int r, int g, int b)
61 {
62     SetR(r);
63     SetG(g);
64     SetB(b);
65 }
```

```
XuLyMau.cpp  X
DIEM  (Global Scope)
66 void MAU::Nhap()
67 {
68     do
69     {
70         cout << "Nhap 3 thong so mau RGB tu 0 den 255: ";
71         cin >> r >> g >> b;
72     } while (r < 0 || r>255 || g < 0 || g>255 || b < 0 || b>255);
73 }
74 void MAU::Xuat() const
75 {
76     cout << " - Mau RGB(" << r << ", " << g << ", " << b << ")";
77 }
78 bool MAU::KiemTraTrung(const MAU& m) const
79 {
80     return r == m.r && g == m.g && b == m.b;
81 }
82 istream& operator>>(istream& is, MAU& m)
83 {
84     do
85     {
86         cout << "Nhap 3 thong so mau RGB tu 0 den 255: ";
87         is >> m.r >> m.g >> m.b;
88     } while (m.r < 0 || m.r>255 || m.g < 0 || m.g>255 || m.b < 0 || m.b>255);
89     return is;
90 }
91 ostream& operator<<(ostream& os, const MAU& m)
92 {
93     os << " - Mau RGB(" << m.r << ", " << m.g << ", " << m.b << ")";
94     return os;
95 }
```

```

DiemMau.h  X
DIEM (Global Scope)
1  #pragma once
2  #ifndef _DIEMMAU
3      #define _DIEMMAU
4  #include "Diem.h"
5  #include "Mau.h"
6  class DIEMMAU :public DIEM, public MAU
7  {
8  private:
9      static int dem;
10 public:
11     ~DIEMMAU();
12     DIEMMAU(double = 0, double = 0, int = 0, int = 0, int = 0);
13     static int GetDem();
14     void Nhap();
15     void Xuat() const;
16     bool KiemTraTrung(const DIEMMAU&) const;
17     friend istream& operator>>(istream&, DIEMMAU&);
18     friend ostream& operator<<(ostream&, const DIEMMAU&);
19 };
20 #endif

```

```

XuLyDiemMau.cpp  X
DIEM (Global Scope)
1  #include "DiemMau.h"
2  int DIEMMAU::dem = 0;
3  DIEMMAU::~DIEMMAU()
4  {
5      cout << "Da huy 1 diem mau => Con " << --dem << " diem mau\n";
6  }
7  DIEMMAU::DIEMMAU(double x, double y, int r, int g, int b) :DIEM(x, y), MAU(r, g, b)
8  {
9      cout << "Da tao 1 mau => Co " << ++dem << " mau\n";
10 }
11 int DIEMMAU::GetDem()
12 {
13     return dem;
14 }
15 void DIEMMAU::Nhap()
16 {
17     DIEM::Nhap();
18     MAU::Nhap();
19 }
20 void DIEMMAU::Xuat() const
21 {
22     DIEM::Xuat();
23     MAU::Xuat();
24 }
25 bool DIEMMAU::KiemTraTrung(const DIEMMAU& dm) const
26 {
27     return DIEM::KiemTraTrung(dm) && MAU::KiemTraTrung(dm);
28 }
29 istream& operator>>(istream& is, DIEMMAU& dm)
30 {
31     dm.Nhap();
32     return is;
33 }
34 ostream& operator<<(ostream& os, const DIEMMAU& dm)
35 {
36     dm.Xuat();
37     return os;
38 }

```

```

main.cpp # X
DIEM (Global Scope)
1 #include "DiemMau.h"
2
3 void main()
4 {
5     //Test Diem mau 2C:
6     cout << "So diem mau 2C hien tai la: " << DIEMMAU::GetDem() << endl;
7     DIEMMAU d1, d2(1, d3(1, 2), d4(1, 2, 100), d5(1, 2, 50, 150), d6(1, 2, 50, 150, 256), *d7 = new DIEMMAU;
8     cout << "So diem mau 2C hien tai la: " << DIEMMAU::GetDem() << endl;
9     cout << "d1: " << d1 << "\nd2: " << d2 << "\nd3: " << d3 << "\nd4: " << d4 << "\nd5: " << d5 << "\nd6: " << d6 << "\nd7: " << *d7 << endl;
10    //-----
11    cout << "Nhap lai toa do diem mau 2C d1:\n";
12    cin >> d1;
13    cout << "d1_moi: " << d1 << endl;
14    //-----
15    if (d1.DIEM::KiemTraTrung(*d7))
16        cout << "d1 va d7 trung hoananh do va tung do\n";
17    else
18        cout << "d1 va d7 khong trung hoananh do va tung do\n";
19    //-----
20    if (d1.MAU::KiemTraTrung(*d7))
21        cout << "d1 va d7 trung mau\n";
22    else
23        cout << "d1 va d7 khong trung mau\n";
24    //-----
25    if (d1.KiemTraTrung(*d7))
26        cout << "d1 va d7 trung hoananh do, tung do va mau\n";
27    else
28        cout << "d1 va d7 khong trung hoananh do, tung do va mau\n";
29    //-----
30    delete d7;
31    cout << "So diem mau 2C hien tai la: " << DIEMMAU::GetDem() << endl;
32 }

```

```

So diem mau 2C hien tai la: 0
Da tao 1 diem 2C => Co 1 diem 2C
Da tao 1 mau => Co 1 mau
Da tao 1 mau => Co 1 mau
Da tao 1 diem 2C => Co 2 diem 2C
Da tao 1 mau => Co 2 mau
Da tao 1 mau => Co 2 mau
Da tao 1 diem 2C => Co 3 diem 2C
Da tao 1 mau => Co 3 mau
Da tao 1 mau => Co 3 mau
Da tao 1 diem 2C => Co 4 diem 2C
Da tao 1 mau => Co 4 mau
Da tao 1 mau => Co 4 mau
Da tao 1 diem 2C => Co 5 diem 2C
Da tao 1 mau => Co 5 mau
Da tao 1 mau => Co 5 mau
Da tao 1 diem 2C => Co 6 diem 2C
Nhap lai thong so mau Blue tu 0 den 255: 255
Da tao 1 mau => Co 6 mau
Da tao 1 mau => Co 6 mau
Da tao 1 diem 2C => Co 7 diem 2C
Da tao 1 mau => Co 7 mau
Da tao 1 mau => Co 7 mau
So diem mau 2C hien tai la: 7
d1: 0, 0 - Mau RGB(0, 0, 0)
d2: 1, 0 - Mau RGB(0, 0, 0)
d3: 1, 2 - Mau RGB(0, 0, 0)
d4: 1, 2 - Mau RGB(100, 0, 0)
d5: 1, 2 - Mau RGB(50, 150, 0)
d6: 1, 2 - Mau RGB(50, 150, 255)
d7: 0, 0 - Mau RGB(0, 0, 0)
Nhap lai toa do diem mau 2C d1:
Nhap hoananh do: 1
Nhap tung do: 1
Nhap 3 thong so mau RGB tu 0 den 255: 0 0 0
d1_moi: 1, 1 - Mau RGB(0, 0, 0)
d1 va d7 khong trung hoananh do va tung do
d1 va d7 trung mau
d1 va d7 khong trung hoananh do, tung do va mau
Da huy 1 diem mau => Con 6 diem mau
Da huy 1 diem 2C => Con 6 diem 2C
So diem mau 2C hien tai la: 6
Da huy 1 diem mau => Con 5 diem mau
Da huy 1 mau => Con 5 mau
Da huy 1 diem 2C => Con 5 diem 2C
Da huy 1 diem mau => Con 4 diem mau
Da huy 1 mau => Con 4 mau
Da huy 1 diem 2C => Con 4 diem 2C
Da huy 1 diem mau => Con 3 diem mau
Da huy 1 mau => Con 3 mau
Da huy 1 diem 2C => Con 3 diem 2C
Da huy 1 diem mau => Con 2 diem mau
Da huy 1 mau => Con 2 mau
Da huy 1 diem 2C => Con 2 diem 2C
Da huy 1 diem mau => Con 1 diem mau
Da huy 1 mau => Con 1 mau
Da huy 1 diem 2C => Con 1 diem 2C
Da huy 1 diem mau => Con 0 diem mau
Da huy 1 mau => Con 0 mau
Da huy 1 diem 2C => Con 0 diem 2C

```

- SV tự xây dựng class Điểm 3 chiều có màu (DIEM3CMAU) kế thừa từ 2 class Điểm 3 chiều (DIEM3C) và class Màu (MAU) và các phương thức xử lý liên quan.