



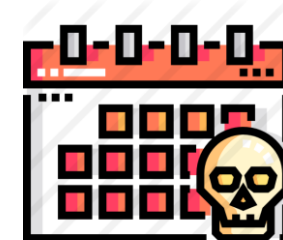
# THIẾT KẾ LỚP CTAMGIAC

WELCOME





NGUYỄN VĂN LINH -20520613  
HỒNG GIA HY -20520561  
NGUYỄN HUY TRÍ DŨNG -20520459  
ĐỖ TRỌNG TÌNH -20520318  
NGUYỄN CÔNG ĐOÀN-20520447



Sit Dolor Amet



# CTAMGIAC

Thiết Kế

## Thiết kế lớp CTamGiac

- Thuộc tính :
  - + CDiem

- Phương thức:
  - + Nhóm phương thức khởi tạo
  - + Nhóm phương thức chung cấp thông tin
  - + Nhóm phương thức cập nhật thông tin
  - + Nhóm phương thức xử lý
  - + Nhóm phương thức kiểm tra

### -Nhóm phương thức khởi tạo:

1. Mặc định
2. Sao chép
3. Khi biết đầy đủ thông tin
4. Phương thức thiết lập mặc định
5. Phương thức thiết lập sao chép
6. Phương thức thiết lập khi biết đầy đủ thông tin
7. Phương pháp nhập
8. Toán tử vào



## Thiết kế lớp CTamGiac

- Nhóm phương thức cung cấp thông tin:

1. Lấy tọa độ điểm A
2. Lấy tọa độ điểm B
3. Lấy tọa độ điểm C
4. Phương thức xuất

## Thiết kế lớp CTamGiac

- Nhóm phương thức cập nhật thông tin:

1. Toán tử gán.

2. Cập nhật điểm A

3. Cập nhật điểm B

4. Cập nhật điểm C

## Thiết kế lớp CTamGiac

### - Nhóm phương thức xử lý thông tin:

+ So sánh bằng

+ So sánh khác

+ Tính chu vi

+ Tính diện tích

+ Phương thức phá hủy



## Thiết kế lớp CTamGiac

- Nhóm phương thức kiểm tra thông tin:
  - + Kiểm tra tam giác đều
  - + Kiểm tra tam giác cân
  - + Kiểm tra tam giác vuông



# CTAMGIAC

Khai báo lớp

## Khai báo lớp CDiem

```
class CDiem
{
private:
    float x;
    float y;

public:
    CDiem()
    {
        x = 0;
        y = 0;
    }
}
```

## Khai báo lớp CDiem

```
CDiem(float x, float y, float z)
{
    this->x = x;
    this->y = y;
}
void KhoiTao()
{
    x = 0;
    y = 0;
}
```



## Khai báo lớp CDiem

```
void KhoiTao(float x, float y, float z)
{
    this->x = x;
    this->y = y;
}
float getX()
{
    return x;
}
float getY()
{
    return y;
}
```

## Khai báo lớp CDiem

```
CDiem &operator=(const CDiem &a)
{
    x = a.x;
    y = a.y;

    return *this;
}
friend istream &operator>>(istream &is, CDiem &a)
{
    cout << "Nhap X: ";
    is >> a.x;
    cout << "Nhap Y: ";
    is >> a.y;
    return is;
}
```

## Khai báo lớp CDiem

```
friend ostream &operator<<(ostream &os, CDiem &a)
{
    os << "(" << a.x << ", " << a.y << ")";
    return os;
};
```

## Khai báo lớp CTamGiac

```
class CTamGiac
{
private:
    CDiem a;
    CDiem b;
    CDiem c;
public:
    // Nhóm phương thức khởi tạo
    // Nhóm phương thức cung cấp thông tin
    // Nhóm phương thức cập nhật thông tin
    // Nhóm phương thức kiểm tra
    // Nhóm phương thức xử lý
```



## Khai báo lớp CTamGiac

// Phương thức khởi tạo

CTamGiac();

CTamGiac(CDiem, CDiem, CDiem);

CTamGiac(CTamGiac &);

void KhoiTao();

void KhoiTao(CDiem, CDiem, CDiem);

void KhoiTao(CTamGiac &);

friend istream &operator>>(istream &, CTamGiac &);

void nhap();

## Khai báo lớp CTamGiac

// Phương thức cung cấp thông tin

CDiem getA();

CDiem getB();

CDiem getC();

friend ostream &operator<<(ostream &, CTamGiac &);

void xuat();

## Khai báo lớp CTamGiac

// Phương thức cập nhập thông tin

```
CTamGiac &operator=(const CTamGiac &);
```

```
void SetA(CDiem &);
```

```
void SetB(CDiem &);
```

```
void SetC(CDiem &);
```

// Phương thức kiểm tra

```
int isCan();
```

```
int isDeu();
```

```
int isVuong();
```

## Khai báo lớp CTamGiac

```
// Phương thức xử lí  
int operator==(CTamGiac);  
int operator!=(CTamGiac);  
float ChuVi();  
float DienTich();  
~CTamGiac();  
};
```





# PHƯƠNG THỨC KHỞI TẠO

CTamGiac

## Nhóm phương thức khởi tạo

// Phương thức thiết lập mặc định

**CTamGiac**::CTamGiac()

{

    a.KhoiTao(0, 0);

    b.KhoiTao(0, 1);

    c.KhoiTao(1, 0);

}

## Nhóm phương thức khởi tạo

// Phương thức thiết lập khi biết đầy đủ thông tin

```
CTamGiac::CTamGiac(CDiem a, CDiem b, CDiem c)
```

```
{
```

```
    this->a = a;
```

```
    this->b = b;
```

```
    this->c = c;
```

```
}
```

// Phương thức thiết lập sao chép

```
CTamGiac::CTamGiac(CTamGiac &x)
```

```
{
```

```
    a = x.a;
```

```
    b = x.b;
```

```
    c = x.c;
```

```
}
```

## Nhóm phương thức khởi tạo

// Phương thức khởi tạo mặc định

```
void CTamGiac::KhoiTao()
```

```
{
```

```
    a.KhoiTao(0, 0);
```

```
    b.KhoiTao(0, 1);
```

```
    c.KhoiTao(1, 0);
```

```
}
```

// Phương thức khởi tạo khi biết đầy đủ thông tin

```
void CTamGiac::KhoiTao(CDiem a, CDiem b, CDiem c)
```

```
{
```

```
    this->a = a;
```

```
    this->b = b;
```

```
    this->c = c;
```

```
}
```



## Nhóm phương thức khởi tạo

// Phương thức khởi tạo sao chép

**void** CTamGiac::KhoiTao(**CTamGiac** &x)

{

    a = x.a;

    b = x.b;

    c = x.c;

}

// Toán tử nhập

**istream** &operator>>(**istream** &is, **CTamGiac** &x)

{

    cout << "Nhap tam giac: \n";

    cout << "Diem A: ";

    is >> x.a;

    cout << "Diem B: ";

    is >> x.b;

    cout << "Diem C: ";

    is >> x.c;

    return is;

}

## Nhóm phương thức khởi tạo

// Phương thức nhập

**void** CTamGiac::nhap()

{

cout << "Nhap tam giac: \n";

cout << "Diem A: ";

cin >> a;

cout << "Diem B: ";

cin >> b;

cout << "Diem C: ";

cin >> c;

}



# PHƯƠNG THỨC CUNG CẤP THÔNG TIN

CTamGiac

## Nhóm phương thức cung cấp thông tin

// Lấy tọa độ điểm A

**CDiem** CTamGiac::getA()

{

    return a;

}

// Lấy tọa độ điểm B

**CDiem** CTamGiac::getB()

{

    return b;

}

## Nhóm phương thức cung cấp thông tin

//Lấy tọa độ điểm C

**CDiem** CTamGiac::getC()

{

    return c;

}

// Toán tử xuất

**ostream** &operator<<(**ostream** &os, **CTamGiac** &X)

{

    os << "Tam giác có: \n";

    os << "Đỉnh A: " << X.a << endl;

    os << "Đỉnh B: " << X.b << endl;

    os << "Đỉnh C: " << X.c << endl;

    return os;

}





# PHƯƠNG THỨC CẬP NHẬT THÔNG TIN

CTamGiac

## Nhóm phương thức cập nhật thông tin

// Toán tử gán

```
CTamGiac &CTamGiac::operator=(const CTamGiac &X)
```

```
{
```

```
    a = X.a;
```

```
    b = X.b;
```

```
    c = X.c;
```

```
    return *this;
```

```
}
```

// Cập nhật điểm A

```
void CTamGiac::SetA(CDiem &x)
```

```
{
```

```
    this->a = x;
```

```
}
```

## Nhóm phương thức cập nhật thông tin

// Cập nhật điểm B

```
void CTamGiac::SetB(CDiem &x)
{
    this->b = x;
}
```

// Cập nhật điểm C

```
void CTamGiac::SetC(CDiem &x)
{
    this->c = x;
}
```



# PHƯƠNG THỨC KIỂM TRA

CTamGiac

## Nhóm phương thức kiểm tra thông tin

// Kiểm tra Tam giác cân

```
int CTamGiac::isCan()  
{  
    float temp, temp1, temp2;  
    temp = a.KhoangCach(b);  
    temp1 = a.KhoangCach(c);  
    temp2 = b.KhoangCach(c);  
    return (temp == temp1 || temp == temp2 || temp1 == temp2);  
}
```

// Kiểm tra Tam giác đều

```
int CTamGiac::isDeu()  
{  
    float temp, temp1, temp2;  
    temp = a.KhoangCach(b);  
    temp1 = a.KhoangCach(c);  
    temp2 = b.KhoangCach(c);  
    return (temp == temp1 && temp == temp2);  
}
```



## Nhóm phương thức kiểm tra thông tin

// Kiểm tra tam giác vuông

```
int CTamGiac::isVuong()
{
    float temp, temp1, temp2;
    temp = a.KhoangCach(b);
    temp1 = a.KhoangCach(c);
    temp2 = c.KhoangCach(b);
    float s1,s2,s3;
    s1 = sqrt(temp * temp + temp1 * temp1);
    s2 = sqrt(temp1 * temp1 + temp2*temp2);
    s3 = sqrt(temp2*temp2 + temp * temp);
    if ((s1 == temp2 || s2 == temp || s3 == temp1))
        return 1;
    else
        return 0;
}
```



# PHƯƠNG THỨC XỬ LÝ

CTamGiac

## Nhóm phương thức xử lý thông tin

// So sánh bằng

```
int CTamGiac::operator==(CTamGiac X)
{
    float temp = this->ChuVi();
    float temp1 = X.ChuVi();
    return (temp == temp1);
}
```

// So sánh khác

```
int CTamGiac::operator!=(CTamGiac X)
{
    float temp = this->ChuVi();
    float temp1 = X.ChuVi();
    return (temp != temp1);
}
```

## Nhóm phương thức xử lý thông tin

// Tính chu vi

**float** CTamGiac::ChuVi()

{

float temp, temp1, temp2;

temp = a.KhoangCach(b);

temp1 = a.KhoangCach(c);

temp2 = b.KhoangCach(c);

return (temp + temp1 + temp2);

}

## Nhóm phương thức xử lý thông tin

// Tính diện tích

**float** CTamGiac::DienTich()

{

**float** temp, temp1, temp2;

temp = a.KhoangCach(b);

temp1 = a.KhoangCach(c);

temp2 = b.KhoangCach(c);

**float** p = (temp + temp1 + temp2) / 2;

return sqrt(p \* (p - temp) \* (p - temp1) \* (p - temp2));

}



## Nhóm phương thức xử lý thông tin

```
// Phương thức phá hủy  
CTamGiac::~~CTamGiac(){  
    return;  
}
```

The background is a dark navy blue. It is decorated with various geometric shapes in orange and red. In the top left, there is a large orange semi-circle. Scattered throughout are smaller shapes: red circles, orange circles, orange triangles, red triangles, orange squares, and orange dashed lines. The word "THANKYOU" is centered in a large, white, sans-serif font.

# THANKYOU