# (Лаборатори №8)

Г. Саруул (20B1NUM2095)

ХШУИС, МКУТ-ийн Компьютерийн ухааны  хөтөлбөрийн 2-р түвшиний оюутан

# **1. ОРШИЛ**

This хувьсагч болон классын static хувьсагч, функцийн талаар судлаж өгөгдсөн асуулт даалгаврыг гүйцэтгэсэн.

# **2. ЗОРИЛГО**

This хувьсагч, Class-ийн static гишүүн өгөгдөл хувьсагчийн талаар дэлгэрэнгүй олж мэдэх, хэрхэн ашиглах талаар судлах, өгөгдсөн даалгавруудыг гүйцэтгэх.

# **3. ОНОЛЫН СУДАЛГАА**

## **3.1 This хувьсагч гэж юу вэ ?**

Тухайн классын буюу объектын гишүүн функцийг хэрэглэж байгаа объектын хаягийг заадаг тусгай зориулалтын хаяган хувьсагчийг this хувьсагч гэнэ.

## **3.2 This хувьсагчийг комплиатор хэрхэн хэрэглэгддэг талаар дэлгэрэнгүй судал**

This хувьсагч бол объеъктон хаяган хувьсагч юм. C++ комплиатор нь object-оор дамжуулаад гишүүн функцийг дуудахад буюу гишүүн функц дуудах үйлдлээс өмнө дараах үйлдлийг хийдэг.

1. Гишүүн функц рүү this хувьсагчийг нэмж оруулах (автоматаар тухайн классын төрлийн this гэдэг нэртэй хаяган хувьсагч нэмж зарладаг.)
2. Тухайн функц дотор (хүрээнд) тэр дуудсан функцийг ажиллаж эхлэхэд this хувьсагчид объектийн хаягийг утга оноож өгнө. (Гишүүн функцийг дуудсан объектийн хаяг)
3. Дуудагдсан функц руу удирдлага шилжинэ.

## **3.3 Static гишүүн өгөгдөл гэж юу вэ ? Хэрхэн зарладаг вэ ?**

Классын гишүүн өгөгдөл статик шинжтэй байхыг статик гишүүн өгөгдөл гэнэ.

Static int number ; гэх мэтчилэн зарладаг.

## **3.4 Static гишүүн өгөгдлийн амьдралын мөчлөг ямар байдаг вэ ?**

Static key ашигласан хувьсагч програм ажлах үед хамгийн түрүүнд санах ой дээр нөөцлөгдөөд, програм ажиллаж дуусахад чөлөөлөгдөнө. Үйлчлэх хүрээ нь тухайн файл, функц, багц коммандын хаалтаар хязгаарлагддаг боловч амьмдрах хугацаа нь програм эхлэхээс төгсөх хүртэл байна.

## **3.5 Static гишүүн өгөгдөлд хэрхэн хандах вэ? Объектоор эсвэл классаар дамжулж хандах**

Static гишүүн өгөгдөл нь классын гишүүн өгөгдөл буюу классын энгийн гишүүн өгөгдлүүд бол object тус бүрээр санах ой нөөцлөгддөг. Харин static нь бүх объектийн дунд буюу shared гэж хэлж болно. (Зөвхөн класс дээрээ үүсдэг). Тийм болохоор static гишүүн өгөгдөлд объектоор нь болон классаар нь дамжуулж хандаж болдог.

Зөвхөн C++ хэл дээр static гишүүн өгөгдөлд классын тодорхойлолтын гадна талд заавал гарааны утга оноохыг шаарддаг. Int class\_name::number = 0 гэх мэт.

Class-ийн гишүүн өгөгдөлд классаар нь class\_name::number гэж хандана.

## **3.6 Static гишүүн өгөгдөл ямар ашигтай вэ ?**

Классийн гишүүн өгөгдөл нь объектын тоогоор үүсэхгүй тул санах ойг үр ашигтайгаар ашигладаг. Мөн тухайн классаас үүссэн объектуудыг тоолох боломжтой.

## **3.7 Static гишүүн функцийг хэрхэн зарладаг, дууддаг вэ ?**

Зөвхөн статик гишиүүн өгөгдөл рүү хандах өгөгдөл дээр боловсруулалт хийх боломжтой функцийг статик гишүүн функц гэнэ. Үүрэг нь зөвхөн статик өгөгдлийг боловсруулах. Мөн static гишүүн өгөгдөл, функц дотор классын this хувьсагч руу хандаж чадахгүй.

Static түлхүүр үгийг функцийн тодорхойлолтын толгой хэсэгт хэрэглэдэггүй.

Функцийнхээ урд static гэдэг түлхүүр үг бичнэ.

Public:

Static void show\_number() ;

Тодорхойлохдоо static түлхүүр үг хэрэглэхгүй. Void employee::show\_number() ;

# **4. ХЭРЭГЖҮҮЛЭЛТ**

**4.1 Лабораторийн ажил 7д хийсэн shape классд түүнээс үүссэн бүх объектийг тоолдог статик хувьсагч нэмж үүссэн объектуудыг тоол.**

class Shape {

    protected:

        char \*name ;

    public:

        static int count ;

        static void getter() ;

        static void setter() ;

        Shape(){

            cout << "Durs uussen" << endl ;

            count++ ;

        }

int Shape::count=0 ;

Объектийг тоолдог тоолуур статик хувьсагчийг public горимтойгоор static int count ; гэж зарлаад байгуулагч функц дотроо count++ гэж shape класс болон түүнээс удамшсан классуудаас үүссэн объеъктийг тоолсон.

**2. Өмнө зарласан static хувьсагчид утга оноох, утгыг нь авах static гишүүн функц бич**

void Shape::getter(){

    cout << "count= " << Shape::count << endl ;

}

void Shape::setter(){

    int x ;

    cout << "Onooh utgaa bichne uu" << endl ;

    cin >> x ;

    Shape::count = x ;

}

Static гишүүн өгөгдөл нь классийн өгөгдөл учир объектийн хаягийг заадаг this хувьсагч хэрэглэхгүй. Тиймээс class\_name::name\_of\_static\_variable гэж хандана.

**3. Классын гишүүн функцууд дотор гишүүн функц руу хандахдаа өмнө нь this хувьсагч ашигла.**

class Circle : public TwoDShape {

    public:

        float findArea(){

            return PI \* this->r \* this->r ;

        }

        float findPerimeter(){

            return 2 \* PI \* this->r ;

        }

class Square : public TwoDShape {

    private:

        int x1, x2, x3, y1, y2, y3 ;

    public:

        float findArea(){

            return this->r \* this->r ;

        }

        float findPerimeter(){

            return 4 \* this->r ;

        }

class Triangle : public TwoDShape {

    private:

        int x1, x2, y1, y2 ;

    public:

        float findArea(){

            return (this->r \* this->r \* sqrt(3)/2 /2) ;

        }

        float findPerimeter(){

            return 3 \* this->r ;

        }

# **5. ДҮГНЭЛТ**

Класс-н гишүүн функц дотор this хаяган хувьсагчийг ашиглах нь кодыг уншихад хялбар буюу уншууртай болгодог бөгөөд мөн классийн static хувьсагч нь зөвхөн класстай хамааралтай байдаг.

# **6. АШИГЛАСАН МАТЕРИАЛ**

[1]. Объект хандлагат технологийн С++ програмчлал, Ж.Пүрэв, 2008, Улаанбаатар.

[2]. [C++ Overloading (Operator and Function) (tutorialspoint.com)](https://www.tutorialspoint.com/cplusplus/cpp_overloading.htm)

# **7. ХАВСРАЛТ**

#include<iostream>

#include<string.h>

#include<math.h>

#define PI 3.14

using namespace std;

class Shape {

    protected:

        char \*name ;

    public:

        static int count ;

        static void getter() ;

        static void setter() ;

        Shape(){

            cout << "Durs uuslee" << endl ;

            count++ ;

        }

        void setName(char \*ner){

            name = new char[strlen(ner)+1];

            strcpy(name, ner);

        }

        Shape(char \*ner){

            name = new char[strlen(ner)+1];

            strcpy(name, ner) ;

            cout << "Durs uussen" << endl ;

            count++ ;

        }

        ~Shape(){

            delete name ;

            cout << "Durs ustgagdlaa" << endl ;

        }

};

int Shape::count=0 ;

void Shape::getter(){

    cout << "count= " << Shape::count << endl ;

}

void Shape::setter(){

    int x ;

    cout << "Onooh utgaa bichne uu" << endl ;

    cin >> x ;

    Shape::count = x ;

}

class TwoDShape : public Shape {

    protected:

        int x, y, r ;

    public:

        TwoDShape(char \*ner, int a, int b, int urt) : Shape(ner){

            this->x = a ;

            this->y = b ;

            this->r = urt ;

        }

        TwoDShape(char \*ner) : Shape(ner){

        }

};

class Circle : public TwoDShape {

    public:

        float findArea(){

            return PI \* this->r \* this->r ;

        }

        float findPerimeter(){

            return 2 \* PI \* this->r ;

        }

        Circle() : TwoDShape("Dugui"){

            this->x = 0 ;

            this->y = 0 ;

        }

        Circle(char \*ner, int a, int b, int urt) : TwoDShape(ner, a, b, urt){

        }

        void setRadius(float a){

            this->r = a ;

        }

        void print(){

            cout << "Ner: " << name << endl ;

            cout << "Toirgiin tuv: " << x << " " << y << endl ;

            cout << "Radius: " << r << endl ;

            cout << "Talbai: " << findArea() << endl ;

            cout << "Toirgiin urt " << findPerimeter() << endl ;

        }

};

class Square : public TwoDShape {

    private:

        int x1, x2, x3, y1, y2, y3 ;

    public:

        float findArea(){

            return this->r \* this->r ;

        }

        float findPerimeter(){

            return 4 \* this->r ;

        }

        Square() : TwoDShape("Kvadrat"){

            x = 0 ;

            y = 0 ;

            r = 1;

            x1 = x + r ;

            y1 = y ;

            x2 = x + r ;

            y2 = y + r ;

            x3 = x ;

            y3 = y + r ;

        }

        Square(char \*ner, int a, int b, int urt) : TwoDShape(ner, a, b, urt){

            x1 = x + r ;

            y1 = y ;

            x2 = x + r ;

            y2 = y + r ;

            x3 = x ;

            y3 = y + r ;

        }

        void setLength(float l){

            x = l ;

            y = l ;

            x1 = x + r ;

            y1 = y ;

            x2 = x + r ;

            y2 = y + r ;

            x3 = x ;

            y3 = y + r ;

        }

        void setA(float a, float b){

            x = a ;

            y = b ;

            x1 = x + r ;

            y1 = y ;

            x2 = x + r ;

            y2 = y + r ;

            x3 = x ;

            y3 = y + r ;

        }

        void print(){

            cout <<"Ner: " << name << endl;

            cout << "Koordinatuud: " << endl;

            cout << "a(" << x << ", " << y << ") ";

            cout << "b(" << x1 << ", " << y1 << ") " << endl;

            cout << "c(" << x2 << ", " << y2 << ") ";

            cout << "d(" << x3 << ", " << y3 << ") " << endl;

            cout << "Urt: " << r << " " << endl;

            cout << "Talbai: " << findArea() << endl;

            cout << "Perimeter: " << findPerimeter() << endl << endl;

        }

};

class Triangle : public TwoDShape {

    private:

        int x1, x2, y1, y2 ;

    public:

        float findArea(){

            return (this->r \* this->r \* sqrt(3)/2 /2) ;

        }

        float findPerimeter(){

            return 3 \* this->r ;

        }

        Triangle() : TwoDShape("Gurvaljin"){

        }

        Triangle(char \*ner, int a, int b, int urt) : TwoDShape(ner, a, b, urt){

            y1 = this->r \* cos(30 \* PI /180.00) + this->y;

            y2 = this->r \* cos(30 \* PI /180.00) + this->y;

            x1 = this->r \* sin(30 \* PI/180.00) + this->x;

            x2 = this->x - this->r \* sin(30 \* PI/180.00);

        }

        void setLength(float l){

            this->r = l ;

        }

        void setA(float a, float b){

            y = b ;

            y1 = this->r \* cos(30 \* PI /180.00) + this->y;

            y2 = this->r \* cos(30 \* PI /180.00) + this->y;

            x1 = this->r \* sin(30 \* PI/180.00) + this->x;

            x2 = this->x - this->r \* sin(30 \* PI/180.00);

        }

        void print(){

        cout <<"Ner: " << name << endl;

        cout << "Koordinatuud: " << endl;

        cout << "a(" << x << ", " << y << ") ";

        cout << "b(" << x1 << ", " << y1 << ") ";

        cout << "c(" << x2 << ", " << y2 << ") " << endl;

        cout << "Urt: " << r << " " << endl;

        cout << "Talbai: " << findArea() << endl;

        cout << "Perimeter: " << findPerimeter() << endl << endl;

        }

};

int main(){

*//Toirog*

    Circle a("Dugui", 10, 10, 5) ;

    a.print() ;

    Square s("Kvadrat", 10, 10, 2);

    s.print() ;

    Triangle c("Gurvaljin", 10, 10, 8) ;

    c.print() ;

    cout << "Niit " << Shape::count << " object shape classaas udamshin uussen baina" << endl ;

    return 0;

}