# СТАТИК ГИШҮҮН ӨГӨГДӨЛ БА ХИЙСВЭР ФУНКЦ ТЭДГЭЭРИЙН ХЭРЭГЛЭЭ (ЛАБОРАТОРИ №8)

Э.Уранчимэг
ХШИУС, МКУТ, Программ хангамжийн III түвшний оюутан,
21B1NUM0609@stud.num.edu.mn

## 1. ОРШИЛ

Энэ тайлангийн онолын судалгаа хэсэгт this хувьсагч болон классын static хувьсагч, хийсвэр функц, классын талаар судлаж тайлбарласан. Өмнөх лаборатори дээр удамшуулсан дүрсийн класс дээр статик хувьсагч, this хувьсагч, хийсвэр функцуудыг ашиглан нэмэлт сайжруулалт хийн ажиллуулсан хэрэгжүүлэлтийг харуулсан. Хавсралт хэсгээс дэлгэрэнгүй кодыг унших боломжтой.

## 2. ЗОРИЛГО

Энэ лабораторын хүрээнд This хувьсагч, Class-ийн static гишүүн өгөгдөл хувьсагчийн талаар дэлгэрэнгүй олж мэдэх, хэрхэн ашиглах талаар судлах, ба бодлогын хүрээнд хэрэгжүүлнэ. Үүний тулд дараах зорилтыг тавьж ажилласан:

1. Онолын судалгаа хийх,
2. Static гишүүн функц тодорхойлох,
3. This хувьсагч ашиглах,
4. Жинхэнэ хийсвэр функц зарлах,

## 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 хэрэглэдэг 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 амьдрах хугацаа нь програм эхлэхээс төгсөх хүртэл байна.

### *3.5 Static гишүүн өгөгдөлд хэрхэн хандах вэ? Объектоор эсвэл классаар дамжуулж 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 функцийг хэрхэн зарладаг, дууддаг вэ?*

Зөвхөн статик гишүүн 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();

### *3.7 Жинхэнэ хийсвэр функц гэж юу вэ?*

Эх класс дотор зарласан тодорхойлолтгүй буюу тодорхойлолт нь null бүхий функцийг pure abstract function буюу жинхэнэ хийсвэр функц гэнэ.

*3.8 Хийсвэр класс гэж юу вэ? Хэрхэн ашигладаг вэ?*

Жинхэнэ хийсвэр функцийг агуулсан классыг хийсвэр буюу абстракт класс гэнэ. Хийсвэр классаас удамших классыг зохиомжилж объект байгуулж болно.

*3.9 Жинхэнэ хийсвэр функцийг давуу тал буюу яагаад ашиглах шаардлагатай талаар дэлгэрэнгүй бичнэ үү.*

Ерөнхийлөх шаардлагатай бөгөөд тодорхойлох боломжгүй үед ашигладаг. Жинхэнэ хийсвэр функцийн гол давуу тал нь хэрэгжилтийн дэлгэрэнгүй мэдээлэлтэй байх шаардалагагүйгээр олон ангиудад тодорхойлох боломжийг олгодогт оршино.

Жинхэнэ хийсвэр функц нь удамшиж байгаа класс бүртээ ахин тодорхойлж өгөхийг шаарддаг ба ахин тодорхойлоогүй бол тэр класс нь мөн адил хийсвэр класс болоод объект үүсэхгүй. Өмнөх лабораторі дээр эх классаас удамшсан классуудад функцыг ахин тодорхойлоогүй тохиолдолд эх классын функц дуудагддаг байсан бол жинхэнэ хийсвэр функц ашигласнаар заавал хүүхэд класс дотор функыг ахин тодорхойлуулдаг.

## 4. ХЭРЭГЖҮҮЛЭЛТ

Лаб7- д хийсэн төслийг дараах байдлаар өөрчилсөн:

1. Shape классд түүнээс үүссэн бүх объектийг тоолдог статик хувьсагч нэмсэн. Ба зарласан static хувьсагчид утга оноох, утгыг нь авах static гишүүн функц бичсэн.

```
class Shape {
protected:
    char *name;

public:
    static int count ;
    static int getter() ;
    static void setter() ;
    void setName(const char *name);
    char *get_name();
```

```
    Shape();
    Shape(const char *name);
    ~Shape();
 };
```

2. Үүнийг хэвлэж харуулах

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

3. Хоёр хэмжээст дүрс классд приметер олох, талбай олох функцуудыг жинхэнэ хийсвэрээр зарлаж хүүхэд классуудад дахин тодорхойлсон.

```
class TwoDShape : public Shape {
protected:
    int x, y, r ;
    float S;
public:
    TwoDShape(const char *ner, int a, int b, int urt);
    TwoDShape(const char *ner);
    virtual float findArea()=0;
    virtual float findPerimeter()=0;
    virtual void print()=0;
};
```

4. Өөр хоорондоо ялгаатай дүрсүүдийг талбай болон приметерээр нь эрэмбэлнэ. Ингэхдээ приметер болон талбай олох жинхэнэ хийсвэр функцуудыг ашигласан.

```
    for(int i=0; i<6; i++){
        for(int j=i+1; j<6; j++){
            if(shapes[i]->findArea()>shapes[j]->findArea()){
                shp=shapes[i];
                shapes[i]=shapes[j];
                shapes[j]=shp;
            }
        }
    }
    cout<<"Talbaigaar erembelsnii ur dun: \n";
    for(int i=0; i<6; i++){
        shapes[i]->print();
        cout<<endl;
    }
    for(int i=0; i<6; i++){
```

```
        for(int j=i+1; j<6; j++){
            if(shapes[i]->findPerimeter()>shapes[j]->findPerimeter()){
                shp=shapes[i];
                shapes[i]=shapes[j];
                shapes[j]=shp;
            }
        }
    }
    cout<<"Perimetereer erembelsnii ur dun: \n";
    for(int i=0; i<6; i++){
        shapes[i]->print();
        cout<<endl;
    }
```

Хэрэгжүүлэлтийн үр дүн:

```
Dura uussen
Dura uussen
Dura uussen
Dura uussen
Dura uussen
Dura uussen
Talbaigaar erembelsnii ur dun:
Ner: Gurvaljin
Koordinatuud:
a(10, 10) b(10, 10) c(9, 10)
Urt: 1
Talbai: 0.433013
Perimeter: 3


Ner: Gurvaljin
Koordinatuud:
a(10, 10) b(10, 11) c(9, 11)
Urt: 2
Talbai: 1.73205
Perimeter: 6


Ner: Kvadrat
Koordinatuud:
a(10, 10) b(13, 10)
c(13, 13) d(10, 13)
Urt: 3
Talbai: 9
Perimeter: 12


Ner: Kvadrat
Koordinatuud:
a(10, 10) b(14, 10)
c(14, 14) d(10, 14)
Urt: 4
Talbai: 16
Perimeter: 16


Ner: Dugui
Toirgiin tuv: 10 10
Radius: 5
Talbai: 78.5
Toirgiin urt 31.4

Ner: Dugui
Toirgiin tuv: 10 10
Radius: 6
Talbai: 113.04
Toirgiin urt 37.68
```

```
Perimetereer erembelsnii ur dun:
Ner: Gurvaljin
Koordinatuud:
a(10, 10) b(10, 10) c(9, 10)
Urt: 1
Talbai: 0.433013
Perimeter: 3


Ner: Gurvaljin
Koordinatuud:
a(10, 10) b(10, 11) c(9, 11)
Urt: 2
Talbai: 1.73205
Perimeter: 6


Ner: Kvadrat
Koordinatuud:
a(10, 10) b(13, 10)
c(13, 13) d(10, 13)
Urt: 3
Talbai: 9
Perimeter: 12


Ner: Kvadrat
Koordinatuud:
a(10, 10) b(14, 10)
c(14, 14) d(10, 14)
Urt: 4
Talbai: 16
Perimeter: 16


Ner: Dugui
Toirgiin tuv: 10 10
Radius: 5
Talbai: 78.5
Toirgiin urt 31.4

Ner: Dugui
Toirgiin tuv: 10 10
Radius: 6
Talbai: 113.04
Toirgiin urt 37.68

Niit 6 object shape classaas udamshin uussen baina
Durs ustgagdlaa
Durs ustgagdlaa
Durs ustgagdlaa
Durs ustgagdlaa
Durs ustgagdlaa
Durs ustgagdlaa
```

## 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.

Өмнөх лаборатори дээр

## 6. АШИГЛАСАН МАТЕРИАЛ

➢ “Объект хандлагат программчлал лекцийн материал”, М.Золжаргал

## 7. ХАВСРАЛТ

Нэгдсэн кодыг хавсаргав.

shape.h

```
#ifndef __shape__
#define __shape__
#include<iostream>
#include<string.h>
using namespace std;
class Shape {
protected:
    char *name;

Public:
    static int count ;
    static int getter() ;
    static void setter() ;
    void setName(const char *name);
    char *get_name();
    Shape();
    Shape(const char *name);
    ~Shape();
 };
 #endif
```

shape.cpp

```
#include "shape.h"
void Shape::setName(const char *name) {
```

```
        if (this->name != nullptr) {
            delete[] this->name;
        }
       this->name = new char[strlen(name) + 1];
        strcpy(this->name, name);
    }
Shape::Shape() {
        this->name = new char[20];
        strcpy(this->name, "default");
        cout << "Durs uusev" << endl;
    }

Shape::Shape(const char* name) {
        this->name = new char[strlen(name) + 1];
        strcpy(this->name, name);
        cout << "Durs(para) uusev" << endl;
    }

Shape::~Shape() {
        delete[] this->name;
        cout << "Durs ustav" << endl;
    }

char *Shape::get_name() {
        return name;
    }
int Shape::count=0 ;

int Shape::getter(){
    return Shape::count ;
}
void Shape::setter(){
    int x ;
    cout << "Onooh utgaa bichne uu" << endl ;
    cin >> x ;
    Shape::count = x ;
}
```

twoDshape.h

```
#ifndef __twoD__
```

```
#define __twoD__
#include"shape.h"
class TwoDShape : public Shape {
protected:
    int x, y, r ;
    float S;
public:
    TwoDShape(const char *ner, int a, int b, int urt);
    TwoDShape(const char *ner);
    virtual float findArea()=0;
    virtual float findPerimeter()=0;
    virtual void print()=0;
};
#endif
```

twoDshape.cpp

```
#include"twoDshape.h"

TwoDShape::TwoDShape(const char *ner, int a, int b, int urt) : Shape(ner){
    x = a ;
    y = b ;
    r = urt ;
}
TwoDShape::TwoDShape(const char *ner) : Shape(ner){
}
```

circle.h

```
#ifndef __Circle__
#define __Circle__
#include"twoDshape.h"
class Circle : public TwoDShape {
public:
    float findArea();
    float findPerimeter();
    Circle();
    Circle(const char *ner, int a, int b, int urt);
    void setRadius(float a);
    void print();
};
#endif
```

circle.cpp

```
#include "circle.h"
#define Pi 3.14

float Circle::findArea(){
    S=Pi * r * r ;
    return S;
}
float Circle::findPerimeter(){
    return 2 * Pi * r ;
}
Circle::Circle() : TwoDShape("Dugui"){
    x = 0;
    y = 0;
}
Circle::Circle(const char *ner, int a, int b, int urt) : TwoDShape(ner, a,
b, urt){
}
void Circle::setRadius(float a){
    r = a ;
}
void Circle::print(){
    cout << "Ner: " << name << endl ;
    cout << "Toirgiin tuv: " << x << " " << y << endl ;
    cout << "Radius: " << r << endl ;
    cout << "Talbai: " << findArea() << endl ;
    cout << "Toirgiin urt " << findPerimeter() << endl ;
}
```

square.h

```
#ifndef __Square__
#define __Square__
#include"twoDshape.h"
class Square : public TwoDShape {
private:
    int x1, x2, x3, y1, y2, y3 ;
public:
    float findArea();
```

```
    float findPerimeter();
    Square();
    Square(const char *ner, int a, int b, int urt);
    void setLength(float l);
    void setA(float a, float b);
    void print();
};
#endif
```

square.cpp

```
#include "square.h"

float Square::findArea(){
    S=r*r ;
    return S;
}
float Square::findPerimeter(){
    return 4 * r ;
}
Square::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::Square(const 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 Square::setLength(float l){
```

```
    x = l ;
    y = l ;
    x1 = x + r ;
    y1 = y ;
    x2 = x + r ;
    y2 = y + r ;
    x3 = x ;
    y3 = y + r ;
}
void Square::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 Square::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;
}
```

triangle.h

```
#ifndef __Triangle__
#define __Triangle__
#include"twoDshape.h"
class Triangle : public TwoDShape {
private:
    int x1, x2, y1, y2 ;
public:
    float findArea();
```

```
    float findPerimeter();
    Triangle();
    Triangle(const char *ner, int a, int b, int urt);
    void setLength(float l);
    void setA(float a, float b);
    void print();
};
#endif
```

triangle.cpp

```
#include"triangle.h"
#include<math.h>

float Triangle::findArea(){
    S=(r*r*sqrt(3)/2 /2);
    return S;
}
float Triangle::findPerimeter(){
    return 3 * r ;
}
Triangle::Triangle() : TwoDShape("Gurvaljin"){
}
Triangle::Triangle(const char *ner, int a, int b, int urt) :
TwoDShape(ner, a, b, urt){
    y1 = r * cos(30 * PI /180.00) + y;
    y2 = r * cos(30 * PI /180.00) + y;
    x1 = r * sin(30 * PI/180.00) + x;
    x2 = x - r * sin(30 * PI/180.00);
}
void Triangle::setLength(float l){
    r = l ;
}

void Triangle::setA(float a, float b){
    y = b ;
    y1 = r * cos(30 * PI /180.00) + y;
    y2 = r * cos(30 * PI /180.00) + y;
    x1 = r * sin(30 * PI/180.00) + x;
    x2 = x - r * sin(30 * PI/180.00);
}
```

```
void Triangle::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;
}
```

main.cpp

```
#include "circle.h"
#include "triangle.h"
#include "square.h"
#include <iostream>
using namespace std;

int main(){
    TwoDShape *shp;
    Circle a("Dugui", 10, 10, 6) ;
    Circle b("Dugui", 10, 10, 5) ;
    Square c("Kvadrat", 10, 10, 4);
    Square d("Kvadrat", 10, 10, 3);
    Triangle e("Gurvaljin", 10, 10, 2);
    Triangle f("Gurvaljin", 10, 10, 1);
    TwoDShape *shapes[6];
    shapes[0]=&a;
    shapes[1]=&b;
    shapes[2]=&c;
    shapes[3]=&d;
    shapes[4]=&e;
    shapes[5]=&f;

    for(int i=0; i<6; i++){
        for(int j=i+1; j<6; j++){
            if(shapes[i]->findArea()>shapes[j]->findArea()){
                shp=shapes[i];
                shapes[i]=shapes[j];
                shapes[j]=shp;
```

```
            }
        }
    }
    cout<<"Talbaigaar erembelsnii ur dun: \n";
    for(int i=0; i<6; i++){
        shapes[i]->print();
        cout<<endl;
    }
    for(int i=0; i<6; i++){
        for(int j=i+1; j<6; j++){
            if(shapes[i]->findPerimeter()>shapes[j]->findPerimeter()){
                shp=shapes[i];
                shapes[i]=shapes[j];
                shapes[j]=shp;
            }
        }
    }
    cout<<"Perimetereer erembelsnii ur dun: \n";
    for(int i=0; i<6; i++){
        shapes[i]->print();
        cout<<endl;
    }
    cout << "Niit " << Shape::count << " object shape classaas udamshin
uussen baina" << endl ;
    return 0;
}
```