 Жинхэнэ хийсвэр функц ба статик гишүүн өгөгдөл болон функц

(Лаборатори №8)

Х.Очирсүх

Програм хангамжийн 3-р түвшний оюутан, 20B1NUM0429

# 1. ОРШИЛ

Объект хандалтат програмчлалын лаборатори 8 хүрээнд this хувьсагч, статик гишүүн өгөгдөл болон статик функц, функц дахин тодорхойлох, жинхэнэ хийсвэр функц, хийсвэр класс гэж ямар ойлголт болох талаар судлаж, бодлогонд хэрэглэж хэвших.

# 2. ЗОРИЛГО

Объект хандалгат програмчлалын хэл дээр байдаг this, хийсвэр класс, хийсвэр функц, статик гишүүн өгөгдөл, функц гэх ойлголтыг судлаж түүнтэй холбоотой бүх объектийг тоолох, жинхэнэ хийсвэр функцыг ашиглан талбай болон периметрээр эрэмбэлэх. Зорилгодоо хүрэхийн тулд дараах зорилтуудыг дэвшүүлсэн.

1. Лекцийн материал болон лекцийн үзэж ойлгох
2. Шаардлагатай агуулгадаа бичиж тэмдэглэх
3. Лекцийн тэмдэглэлээ эмх цэгцтэй, ойлгомжтой байдлаар дүрслэх
4. Өөрийн үгээр баяжуулж, нэмэлт материал судлах
5. Объект хандлагат номноос шаардлагатай мэдээллүүдээ цуглуулах

# 3. ОНОЛЫН СУДАЛГАА

## 3.1 This хувьсагч гэж юу бэ?

Гишүүн функцийг хэрэглэж байгаа объектын хаягийг заах тусгай зориулалтын хаяган хувьсагч юм.

## 3.2 This хувьсагчийн комплиатор хэрхэн хэрэглэгдэг талаар дэлгэрэнгүй судал.

C++ компайлер:

|  |
| --- |
| 1. Гишүүн функц руу this хувьсагчийг нэмж оруулах |
| 1. Гишүүн функцийг дуудсан объектон хаягийг this хувьсагч руу хийнэ. |
| 1. Дуудагдсан функц руу удирдлага шилжүүлнэ. |

## Статик гишүүн өгөгдөл гэж юу бэ? Хэрхэн зарладаг бэ?

## 

Нэг классын объектууд бүгд дундаа хэрэглэх боломжтой гишүүн өгөгдөл юм. Уг классаас хэдэн ч объект үүссэн бай бүх объект дунд нэг л статик гишүүн өгөгдөл байна.

Жишээ нь : static int a ;

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

Static гишүүн өгөгдлийн амьдр ах хугацаа нь програм эхлэх үед үүсдэг. Харин үйлчлэх хүрээ, static гишүүн өгөгдлийн амьдралын мөчлөг дуусах хугацаа нь програм дуусах үед байдаг.

## 3.5 Static гишүүн өгөгдөлд хэрхэн хандах вэ? Объектоор эсвэл классаар дамжуулж хандах.

Уг гишүүн өгөгдөлийг л боловсруулдаг функцийг статик гишүүн функц гэнэ. Статик гишүүн функц нь мөн л классын бүх объектуудын дунд нэг байна. Статик гишүүн функц статик биш гишүүн өгөгдлүүд рүү хандаж чадахгүй, мөн this хувьсагч руу хандаж чадахгүй. Статик гишүүн функаар статик гишүүн өгөгдөлд хандахдаа 2 аргаар хандаж болдог. Объектоор хандахдаа тухайн объектын нэр.статик гишүүн функын нэр байна. Жишээ нь

static void show\_number();

employee e1

e1.show\_number();

Харин класс хандахдаа классын нэр үйлчлэх хүрээний опеотор(::) статик гишүүн функцын нэр байна. Жишээ нь employee::show\_number();

## Static гишүүн өгөгдөл ямар ашигтай вэ?

## Статик гишүүн өгөгдөл бүх объект дундаа хэрэглэж болдог. Мөн объектыг тоолох боломжыг олгодог.

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

Зөвхөн статик гишүүн өгөгдөлд хандах боломжтой функцыг статик функц гэнэ.

Зарлахдаа static void show\_number();

Статик гишүүн функцыг дуудахдаа 2 аргаар дуудаж болно. Объектоор болон классаар дуудаж ашиглаж болдог. Объектоор хандахдаа тухайн объектын нэр.статик гишүүн функын нэр байна. Жишээ нь

static void show\_number();

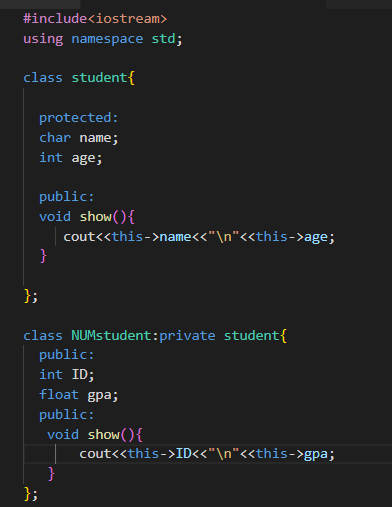
employee e1

e1.show\_number();

Харин класс хандахдаа классын нэр үйлчлэх хүрээний опеотор(::) статик гишүүн функцын нэр байна. Жишээ нь employee::show\_number();

## 3.8 Функц дахин тодорхойлох гэж юу бэ?

Эх классд байгаа функцын бичдэс(функцын нэр, параметр, төрөл) ижлээр хүүхэд классд дахин тодорхойлохыг функц дахин тодорхойлох гэж хэлнэ. Жишээ нь



## 3.9 Жинхэнэ хийсвэр функц гэж юу бэ?

Зөвхөн зарласан их бие нь null утга авдаг, тодорхойлолт байхгүй хийсвэр классд зарлагдсан, virtual түлхүүр үг авж ашигладаг функцыг жинхэнэ хийсвэр функц гэнэ.

## 3.9.1 Хийсвэр класс гэж юу бэ? Хэрхэн ашигладаг бэ?

Жинхэнэ хийсвэр функцыг агуулсан, объект үүсгэж болохгүй классыг хийсвэр класс буюу abstract класс гэнэ. Жишээлбэл shape2D хийсвэр класс үүсгэж, Circle, Square, Triangle зэрэг классыг удамшуулж үүсгэж, жинхэнэ хийсвэр функаар талбай болон перимтр дахин тодорхойлж ашигласан.

# 4. ХЭРЭГЖҮҮЛЭЛТ

Main function

#include "circle.h"

#include "square.h"

#include "triangle.h"

int main()

{

   int i;

   twod \*shapes[9];

   shapes[0] = new circle(1);

   shapes[1] = new circle(5);

   shapes[2] = new circle(10);

   shapes[3] = new square(5);

   shapes[4] = new square(1);

   shapes[5] = new square(9);

   shapes[6] = new triangle(2);

   shapes[7] = new triangle(3);

   shapes[8] = new triangle(10);

  // static

   shapes[0]->show\_number();

  // cout<<"object number: "<<shapes[0]->getNumber();

  cout<<"\nDvrsiin talbainuud\n";

   for (i = 0; i < 9; i++)

      cout << "\n"

           << shapes[i]->get\_s();

   twod \*tmp;

   int k = 1, j;

   for (k = 0; k < 9; k++)

   {

      for (j = 0; j < 9; j++)

      {

         if (shapes[j]->get\_s() > shapes[k]->get\_s())

         {

            tmp = shapes[j];

            shapes[j] = shapes[k];

            shapes[k] = tmp;

         }

      }

   }

   // eremblegdsen

   cout << "\n Talbaigaar eremblegdsen";

   for (i = 0; i < 9; i++)

      cout << "\n"

           << shapes[i]->get\_s();

    // perimeter

      cout<<"\nDvrsiin hvreenii urtuud\n";

   for (i = 0; i < 9; i++)

      cout << "\n"<< shapes[i]->get\_p();

   // eremblegdsen

   for (k = 0; k < 9; k++)

   {

      for (j = 0; j < 9; j++)

      {

         if (shapes[j]->get\_p() > shapes[k]->get\_p())

         {

            tmp = shapes[j];

            shapes[j] = shapes[k];

            shapes[k] = tmp;

         }

      }

   }

   cout << "\n Hvreenii urtaar eremblegdsen";

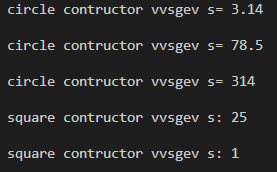
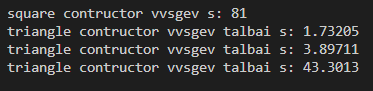
   for (i = 0; i < 9; i++)

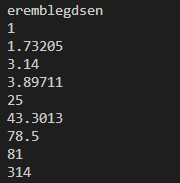
      cout << "\n"

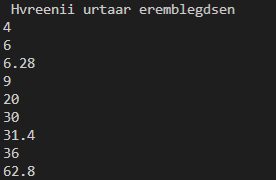
           << shapes[i]->get\_p();

}

Үр дүн



# ДҮГНЭЛТ

Энэхүү лаборатори 8 хүрээнд this хувьсагч, статик гишүүн өгөгдөл болон функц , жинхэнэ хийсвэр функц, хийсвэр класс, функц дахин тодорхойлох ойлголтыг мэдэж авж, кодондоо хэрэгжүүлсэн. This хувьсагч бичихэд программд уншвартай болдог мөн гишүүн функцыг дуудах объектын хаягийг хадгалдаг хаяган хувьсагч байгаа. Статик гишүүн өгөгдлөөр хэдэн объект тухайн классаар үүсэж байгаа мэдээллийг авж чадна. Жинхэнэ хийсвэр функц эх классаас хаяган хувьсагч үүсгээд, хүүхэд классын хаягийг аваад дахин тодорхойлсон функцийг дуудах үед эх классын функц дуудагдаж логикийн алдаа үүсдэг, үүнээс урьдчилан сэргийлэх зорилготой хэрэглэж байдаг. Жинхэнэ хийсвэр функцыг ашиглан периметр болн талбайгаар өөр хоорондоо ялгаатай дүрсүүдээ эрэмблэж чадсан.

# 6. АШИГЛАСАН МАТЕРИАЛ

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

2. Лекц – хийсвэр класс, жинхэнэ хийсвэр функц, статик гишүүн өгөгдөл ба функц, this хувьсагч teams

# ХАВСРАЛТ

Эх класс

#ifndef \_shape\_

#define \_shape\_

#include <iostream>

using namespace std;

class shape

{

protected:

static int number;

    float a;

public:

    // void set\_a(float a);

    shape(float a);

   static void show\_number();

   static int getNumber();

   static void setNumber(const int n);

};

#endif

#include "shape.h"

int shape::number = 0;

shape::shape(float a)

{

      this->a = a;

      number++;

}

void shape::show\_number(){

       cout<<"Objectiin too: "<<number<<endl;

}

int shape::getNumber(){

      return number;

}

void shape::setNumber(const int n){

      number = n;

}

Эх класс удамшсан twod класс

#include "shape.h"

#ifndef \_twod\_

#define \_twod\_

class twod : public shape

{

protected:

  float \*x;

  float \*y;

  float h, s, p;

public:

  virtual float findArea() = 0;

  virtual float findP() = 0;

  float get\_s();

  float get\_p();

  twod(float a, int points);

};

#endif

#include "twod.h"

float twod::get\_s()

{

   return this->s;

}

float twod::get\_p(){

   return this->p;

}

twod::twod(float a, int points) : shape(a)

{

   this->a = a;

   this->x = new float[points];

   this->y = new float[points];

}

Circle

#include "twod.h"

#ifndef \_cicrle\_

#define \_circle\_

#define pi 3.14

class circle : public twod

{

public:

  circle(float a);

  float findArea();

  float findP();

  ~circle();

};

#endif

#include <iostream>

#include "circle.h"

circle::circle(float a) : twod(a, 1)

{

   cout << "\ncircle contructor vvsgev s= " << this->findArea() <<this->findP()<<"\n";

}

float circle::findArea()

{

   this->s = this->a \* this->a \* pi;

   return this->s;

}

float circle::findP(){

    this->p = 2 \* this->a \*pi;

    return this->p;

}

circle::~circle()

{

   delete[] x;

   delete[] y;

}

Triangle

#include "twod.h"

#ifndef \_triangle\_

#define \_triangle\_

class triangle : public twod

{

private:

    float h;

public:

    float findArea();

    float findP();

    void findX();

    void set\_h(float);

    float get\_h();

    triangle(float x);

};

#endif

#include "triangle.h"

#include <math.h>

void triangle::findX()

{

   h = sqrt(pow(a, 2) - pow(a / 2, 2));

   set\_h(h);

}

void triangle::set\_h(float h)

{

   this->h = h;

}

float triangle::get\_h()

{

   return this->h;

}

float triangle::findArea()

{

   findX();

   this->s = (this->a \* get\_h()) / 2;

   return this->s;

}

float triangle::findP(){

   this->p = 3 \* this->a;

   return this->p;

}

triangle::triangle(float x) : twod(x, 3)

{

   cout << "triangle contructor vvsgev talbai s: " << this->findArea()<<this->findP() << "\n";

}

Square

#include "twod.h"

#ifndef \_square\_

#define \_square\_

class square : public twod

{

public:

    float findArea();

    float findP();

    square(float a);

};

#endif

#include "square.h"

float square::findArea()

{

   this->s = this->a \* this->a;

   return this->s;

}

float square::findP(){

   this->p = 4 \* this->a;

   return this->p;

}

square::square(float a) : twod(a, 4)

{

   cout << "\nsquare contructor vvsgev s: " << this->findArea() <<this->findP()<< "\n";

}

Main function

#include "circle.h"

#include "square.h"

#include "triangle.h"

int main()

{

   int i;

   twod \*shapes[9];

   shapes[0] = new circle(1);

   shapes[1] = new circle(5);

   shapes[2] = new circle(10);

   shapes[3] = new square(5);

   shapes[4] = new square(1);

   shapes[5] = new square(9);

   shapes[6] = new triangle(2);

   shapes[7] = new triangle(3);

   shapes[8] = new triangle(10);

  // static

   shapes[0]->show\_number();

  // cout<<"object number: "<<shapes[0]->getNumber();

  cout<<"\nDvrsiin talbainuud\n";

   for (i = 0; i < 9; i++)

      cout << "\n"

           << shapes[i]->get\_s();

   twod \*tmp;

   int k = 1, j;

   for (k = 0; k < 9; k++)

   {

      for (j = 0; j < 9; j++)

      {

         if (shapes[j]->get\_s() > shapes[k]->get\_s())

         {

            tmp = shapes[j];

            shapes[j] = shapes[k];

            shapes[k] = tmp;

         }

      }

   }

   // eremblegdsen

   cout << "\n Talbaigaar eremblegdsen";

   for (i = 0; i < 9; i++)

      cout << "\n"

           << shapes[i]->get\_s();

    // perimeter

      cout<<"\nDvrsiin hvreenii urtuud\n";

   for (i = 0; i < 9; i++)

      cout << "\n"<< shapes[i]->get\_p();

   // eremblegdsen

   for (k = 0; k < 9; k++)

   {

      for (j = 0; j < 9; j++)

      {

         if (shapes[j]->get\_p() > shapes[k]->get\_p())

         {

            tmp = shapes[j];

            shapes[j] = shapes[k];

            shapes[k] = tmp;

         }

      }

   }

   cout << "\n Hvreenii urtaar eremblegdsen";

   for (i = 0; i < 9; i++)

      cout << "\n"

           << shapes[i]->get\_p();

}