# Байгуулагч, Устгагч функц (Лаборатори №4)

Д.Самдан (16B1SEAS2873)

ХШУИС, ПХ-ийн 3-р түвшин, pri.samdan@gmail.com

# **1. ОРШИЛ**

Ажилчин нэртэй объектод хуулагч байгуулагч функц нэмж, түүнийгээ ашиглан хоёр ажилчингийн мэдээллийг сольсон. Мөн ажилчин объектын хаяган хүснэгт үүсгэж, ажилчдыг нэрээр нь эрэмбэлсэн. Ажилчингийн байрыг нь солиход хаягийг нь ашиглан солино.

# **2. ЗОРИЛГО**

Ажилчин нэртэй объектын хуулагч байгуулагч нэмэх. Мөн объектын хаяган хүснэгт үүсгэж, түүнийгээ эрэмбэлэх. Үүний тулд дараах зорилтуудыг тавьж ажилласан:

* Хуулагч байгуулагч функц, класс ба ойн менежментийн талаар онолын судалгаа хийх
* Хуулагч байгуулагч функц зарлах, тодорхойлох
* Класс ба ойн менежмент
  + new оператор ашиглан шинэ ажилчин үүсгэх
  + delete оператор ашиглан ажилчин устгах
* Объектын хаяган хүснэгт үүсгэх
* Объектын хүснэгтээ эрэмбэлэх

# **3. ОНОЛЫН СУДАЛГАА**

**Хуулагч Байгуулагч Функц**

Хуулагч байгуулагч бол шинэ объектод түүний өмнө аль нэг объектын сүүлийн утгыг хуулбарлаж авах боломжийг олгох функц юм. Ийм маягаар байгуулагдах шинэ объект ба утга нь хуулбарлагдах объект хоорондоо зөвхөн нэрээрээ ялгаатай. Тийм учраас хуулагч байгуулагч нь объект хуулбарлах функц юм[1].

Зарлах:

class\_name(const class\_name &temp);

Тодорхойлох:

class\_name::class\_name (const class\_name &temp) { /\* Some code \*/ }

Ашиглах:

employee emp1(emp2);

**Класс ба Ойн Менежмент**

1.**Объектын хаяг**

Объектын хаяг хадгалах хаяган хувьсагчийг бусад хаяган хувьсагчтай адил зарлана.

class\_name \*pointer\_name;

Дээрх байдалаар үүссэн хувьсагчийг объектон буюу классын хаяган хувьсагч гэнэ. Объектын эхлэх хаягийг (&) хаягийн оператор ашиглан авч болно. Түүнийгээ класс ижил хаяган хувьсагчид хадгалж болно[1].

class\_name object\_name;

pointer\_name = &object\_name;

Объектын хаяган хувьсагчаас классын гишүүн рүү хоёр янзаар хандаж болно.

* Шууд сонголтын цэг (.) оператор

(\*pptr).member\_function()

* Шууд бус сонголтын сум (->) оператор

pptr->member\_function()

new болон delete-ийг объект үүсгэж устгахад ашиглах.

class\_name \*p\_name;

p\_name = new class\_name();

delete p\_name;

**4. Хэрэгжүүлэлт**

Хавсралт employee.h-н 42-р мөрөнд хуулагч байгуулагч функцийг зарлаж өгсөн. Хавсралт employee.cpp-н 16-22-р мөрүүдэд хуулагч байгуулагч функцийн тодорхойлолт байна. Энд параметрээр орж ирсэн ажилчингийн мэдээллийг үүсгэж буй ажилчинд set функц ашиглан хуулна.

Хавсралт employee.h-н 204-р мөрөнд ажилчин хүснэгтийг нэрээр нь эрэмбэлэх функцийн зарлал байна. Хавсралт employee.cpp-н 175-202-р мөрүүдэд ажилчин хүснэгтийг нэрээр нь эрэмбэлэх функцийн тодорхойлолт байна. Энэ функц нь selection sort ашиглан ажилчин хүснэгтийг нэрээр нь эрэмбэлж байна. 180-191-р мөрүүдэд хамгийн эхэнд байх нэрийг олж 194-200-р мөрүүдэд хаягийг нь ашиглан өмнө орох ажилчингийн байрыг сольж байна.

Хавсралт main.cpp-н 7-р мөрөнд ажилчин объектын хаяган хүснэгт үүсгэж байна. Хавсралт main.cpp-н 10-12-р мөрүүдэд 3 ажилчинг new оператор ашиглан үүсгэж, анхны утгыг параметртэй байгуулагч функц ашиглан оноож өгсөн. 14-р мөрөн дээр дурдагдсан sort\_employee\_by\_name функцийг дуудан зарласан 3 ажилчинг эрэмбэлж байна. Сүүлд нь 16-21-р мөрүүдэд эрэмбэлсэн ажилчдыг print функц дуудан дэлгэцэнд хэвлэж байна. Ажилчинг хэвлэсний дараа delete оператор ашиглан устгаж байна.

# **5. ДҮГНЭЛТ**

# Хуулагч байгуулагч функц ажилчин объектод шинээр нэмж өгсөний үр дүнд нэг объектын мэдээллийг шинээр үүсэж буй объектод хуулах боломжтой болсон. Үүний үр дүнд объектын хүснэгтийн хоёр объектын байрыг солиход хялбар болсон. Түүний дараа объектон хаяган хүснэгт үүсгэж ажилчдыг нэрээр нь эрэмбэлсэн. Хаяган хүснэгт учир объектын байрыг нь солиход объектын утгыг нь биш хаягийг нь солих учир санах ой ашиглалт нь маш буурсан.

# **6. АШИГЛАСАН МАТЕРИАЛ**

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

# **7. ХАВСРАЛТ**

**employee.h**

#ifndef EMPPLOYEE\_H

#define EMPPLOYEE\_H

#include <iostream>

using namespace std;

class employee

{

public:

// Ажилчны нийт тоо

static int id\_number;

private:

    // Гишүүн ажилчны дугаар

    int id;

    // Гишүүн ажилчны нэр

    char \*name;

    // Гишүүн ажилчны албан тушаал

    char \*position;

    // Гишүүн ажилчны ажилсан цаг

    float worked\_hour;

public:

    /\*

        Анхдагч байгуулагч функц

        name-т Default

        position-т Employee

        worked\_hour-т 0.0

        зэрэг утгуудыг оноож, санах ой нөөцлөнө

        Params:

        void

    \*/

    employee();

/\*

Хуудагч байгуулагч нэг ажилчингий мэдээлэлийг нөгөө ажилчинд өгнө

Param:

employee &emp - Хуулах ажилчингийн хаяг

\*/

employee(const employee &emp);

    /\*

        Параметртэй байгуулагч функц. Параметрийн дагуу объектд анхны утгууд оноож, санах ой нөөцлөнө

        Params:

        int e\_id - шинээр үүсгэж буй ажилчингийн дугаар

        char e\_name[] - шинээр үүсгэж буй ажилчингийн нэр

        char e\_position[] - шинээр үүсгэж буй ажилчингийн албан тушаал

    \*/

    employee(const char e\_name[], const char e\_position[], float e\_worked\_hour = 0.0);

    /\*

        Устгагч функц

        ажилчингийн new гээр нөөцөлсөн name, position-г delete оператор ашиглан чөлөөлнө.

    \*/

    ~employee();

/\*

Ажилчдын тоог буцааж өгнө

        Params:

        void

        Return:

        void

\*/

static int get\_id\_number();

    /\*

        Гишүүн ажилчингийн дугаарыг буцааж өгнө

        Params:

        void

        Return:

        int - Гишүүн ажилчингийн дугаар

    \*/

    int get\_id();

    /\*

        Гишүүн ажилчингийн нэрийг буцааж өгнө

        Params:

        void

        Return:

        char\* - Гишүүн ажилчингийн нэр

    \*/

    char \*get\_name();

    /\*

        Гишүүн ажилчингийн албан тушаалыг буцааж өгнө

        Params:

        void

        Return:

        char\* - Гишүүн ажилчингийн албан тушаал

    \*/

    char \*get\_position();

    /\*

        Гишүүн ажилчингийн ажилласан цагийг буцааж өгнө

        Params:

        void

        Return:

        float - Гишүүн ажилчингийн ажилласан цаг

    \*/

    float get\_worked\_hour();

    /\*

        Гишүүн ажилчингийн нэрийг өөрчилнө

        Params:

        int e\_id - шинээр өөрчлөх гишүүн ажилчингийн нэр

        Return:

        void

    \*/

    void set\_name(const char e\_name[]);

    /\*

        Гишүүн ажилчингийн албан тушаалыг өөрчилнө

        Params:

        int e\_id - шинээр өөрчлөх гишүүн ажилчингийн албан тушаал

        Return:

        void

    \*/

    void set\_position(const char e\_position[]);

    /\*

        Гишүүн ажилчингийн ажилласан цагийг өөрчилнө

        Params:

        int e\_id - шинээр өөрчлөх гишүүн ажилчингийн ажилласан цаг

        Return:

        void

    \*/

    void set\_worked\_hour(float e\_worked\_hour);

    /\*

        Гишүүн ажилчны мэдээлэлийг дэлгэцэнд хэвлэнэ

        Params:

        void

        Return:

        void

    \*/

    void print();

    /\*

        Гишүүн ажилчны цалинг ажилласан цагаар нь бодож буцаана.

        Params:

        void

        Return:

        void

    \*/

    float calculate\_salary();

    /\*

        Гишүүн ажилчны ажилласан цагийг нэмнэ

        Params:

        float hour - Ажилчингын ажилсн цаг

        Return:

        bool - Зөв утга уруулсан үед 1 бусад үед 0

    \*/

    bool add\_worked\_hour(float hour);

    /\*

        Ажилчин хүснэгтийг цалингаар нь багаас ихруу эрэмблэх

        Params:

        employee employee\_array[] - Эрэмблэх ажилчин хүснэгт

int emp\_array\_length - эрэмблэх ажилчин хүснэгтийн урт

        Return:

void

    \*/

    void sort\_employee\_by\_salary(employee \*emp\_array[], int emp\_array\_lenght);

/\*

        Ажилчин хүснэгтийг нэрээр нь эрэмблэх

        Params:

        employee employee\_array[] - Эрэмблэх ажилчин хүснэгт

int emp\_array\_length - эрэмблэх ажилчин хүснэгтийн урт

        Return:

void

\*/

void sort\_employee\_by\_name(employee \*\*emp\_array, int emp\_array\_length);

private:

    /\*

        Захирлын цалинг бодох

        Params:

        void

        Return:

        void

    \*/

    float calculate\_salary\_ceo();

/\*

       Ажилчидын тоог нэгээр ихэсгэх

        Params:

        void

        Return:

        void

\*/

static void inc\_id\_number();

    /\*

        Гишүүн ажилчны дугаарыг өөрчилнө

        Params:

        int e\_id - шинээр өөрчлөх гишүүн ажилчингийн дугаар

        Return:

        void

    \*/

    void set\_id(int e\_id);

};

#endif

**employee.cpp**

**#include "./employee.h"**

#include <string.h>

// Static ажилчингийн тоонд анхны утга оноож байна

int employee::id\_number = 0;

employee::employee()

{

employee::inc\_id\_number();

this->set\_id(id\_number);

this->set\_name("Default");

    this->set\_position("Employee");

this->set\_worked\_hour(0.0);

}

employee::employee(const employee &emp)

{

this->set\_id(emp.id);

this->set\_name(emp.name);

this->set\_position(emp.position);

this->set\_worked\_hour(emp.worked\_hour);

}

employee::employee(const char e\_name[], const char e\_position[], float e\_worked\_hour)

{

employee::inc\_id\_number();

this->set\_id(id\_number);

this->set\_name(e\_name);

this->set\_position(e\_position);

this->set\_worked\_hour(e\_worked\_hour);

}

employee::~employee()

{

    delete[] name;

    delete[] position;

}

void employee::inc\_id\_number()

{

id\_number++;

}

int employee::get\_id()

{

    return id;

}

char \*employee::get\_name()

{

    return name;

}

char \*employee::get\_position()

{

    return position;

}

float employee::get\_worked\_hour()

{

    return worked\_hour;

}

void employee::set\_id(int e\_id)

{

    id = e\_id;

}

void employee::set\_name(const char e\_name[])

{

if (name != nullptr)

{

delete[] name;

}

    if (e\_name[strlen(e\_name) != '\0'])

    {

        name = new char[strlen(e\_name) + 1];

    }

    else

    {

        name = new char[strlen(e\_name)];

    }

    strcpy(name, e\_name);

}

void employee::set\_position(const char e\_position[])

{

if (position != nullptr)

{

delete[] position;

}

    if (e\_position[strlen(e\_position) != '\0'])

    {

        position = new char[strlen(e\_position) + 1];

    }

    else

    {

        position = new char[strlen(e\_position)];

    }

    strcpy(position, e\_position);

}

void employee::set\_worked\_hour(float e\_worked\_hour)

{

    worked\_hour = e\_worked\_hour;

}

void employee::print()

{

    cout << "Id: " << id << endl;

    cout << "Name: " << name << endl;

    cout << "Position: " << position << endl;

    cout << "Wordked hours: " << worked\_hour << endl;

    cout << endl;

}

float employee::calculate\_salary()

{

    float salary = worked\_hour \* 100;

    if (strcmp(position, "CEO") == 0)

        return salary + calculate\_salary\_ceo();

    return salary;

}

float employee::calculate\_salary\_ceo()

{

    return worked\_hour \* 50;

}

bool employee::add\_worked\_hour(float hour)

{

    if (hour > 0 && hour < 24)

    {

        worked\_hour += hour;

        return true;

    }

    return false;

}

void employee::sort\_employee\_by\_salary(employee \*emp\_array[], int emp\_array\_lenght)

{

for (int emp\_index = 0; emp\_index < emp\_array\_lenght - 1; emp\_index++)

{

// Хамгийн бага цалинтай ажилчингийн индекс, цалингийн хэмжээг хадгалах хувьсагч зарласан

float min\_salary = (emp\_array[emp\_index])->calculate\_salary();

int min\_index = emp\_index;

// Ажилчин хүснэгтээс хамгийн бага цалитай ажилчинг олож индекс болон цалингийн хэмжээг хадаглана

for(int emp\_cmp\_index = emp\_index + 1; emp\_cmp\_index < emp\_array\_lenght; emp\_cmp\_index++)

{

if (min\_salary > emp\_array[emp\_cmp\_index]->calculate\_salary() )

{

min\_salary = emp\_array[emp\_cmp\_index]->calculate\_salary();

min\_index = emp\_cmp\_index;

}

}

// Хамгийн бага цалинтай ажилчиг эхнээс нь байрандан оруулна

// Хамгийн бага цалинтай ажилчин болон байранд оруулах индекс адилхаг үед сольхгүй

if (min\_index != emp\_index)

{

employee \*temp = emp\_array[min\_index];

emp\_array[min\_index] = emp\_array[emp\_index];

emp\_array[emp\_index] = temp;

}

}

}

void employee::sort\_employee\_by\_name(employee \*emp\_array[], int emp\_array\_lenght)

{

for (int emp\_index = 0; emp\_index < emp\_array\_lenght - 1; emp\_index++)

{

// Өмнө нь байх нэрий, индексийг хадаглах хувьсагч зарласан

char \*min\_name = emp\_array[emp\_index]->get\_name();

int min\_index = emp\_index;

// Өмнө нь байх нэрийг олож дээрэх хувьсагчид хадгална

for(int emp\_cmp\_index = emp\_index + 1; emp\_cmp\_index < emp\_array\_lenght; emp\_cmp\_index++)

{

if ( strcmp(min\_name, emp\_array[emp\_cmp\_index]->get\_name()) > 0 )

{

min\_name = emp\_array[emp\_cmp\_index]->get\_name();

min\_index = emp\_cmp\_index;

}

}

// Өмнө нь орох нэртэй ажилчинг байранд нь оруулж, солигдож буй ажилчид өөрийн утгаа өгнө

if (min\_index != emp\_index)

{

employee \*temp = emp\_array[min\_index];

emp\_array[min\_index] = emp\_array[emp\_index];

emp\_array[emp\_index] = temp;

}

}

}

**main.cpp**

**#include "./employee.h"**

#define NUMBER\_OF\_EMPLOYEE 3

int main()

{

employee \*emp\_array[NUMBER\_OF\_EMPLOYEE];

// Shine ajilchingiin haygiig emp\_index-d hadgalj baina

employee \*new\_emp\_0 = new employee("Sam", "CEO", 10.0);

employee \*new\_emp\_1 = new employee("Dan", "EMP", 21.0);

employee \*new\_emp\_2 = new employee("Anna", "EMP", 11.0);

emp\_array[0] = new\_emp\_0;

emp\_array[1] = new\_emp\_1;

emp\_array[2] = new\_emp\_2;

emp\_array[0]->sort\_employee\_by\_salary(emp\_array, NUMBER\_OF\_EMPLOYEE);

for (int emp\_index = 0; emp\_index < NUMBER\_OF\_EMPLOYEE; emp\_index++)

{

emp\_array[emp\_index]->print();

}

return 0;

}