

Классийн удамшил ба байгуулагч функц (Лаборатори №8)

Ү.Мэндсайхан

МУИС, ХШУИС, Компьютерын ухаан, mendsaikhan@gurvanshiidel.mn

1. ОРШИЛ

Энэхүү лаборатороор this хувьсагч гэж юу болох, түүнийг компилятор хэрхэн хэрэглэгдэг талаар, мөн static гишүүн өгөгдөл, static гишүүн функцийн талаарх ойлголтуудыг судалж, хэрэгжүүлсэн.

2. ЗОРИЛГО

Классд түүнээс үүссэн бүх объектийг тоолдог статик хувьсагч нэмж үүссэн объектуудыг тоолох болон өөр хоорондоо ялгаатай дүрсүүдийг талбай болон периметрээр нь эрэмблэх дасгал гүйцэтгэнэ. Үүний тулд дараах зорилтуудыг тавьж ажилласан:

2.1 Гишүүн функцээс гишүүн функц рүү хандахдаа this хувьсагч ашиглах

2.2 Статик гишүүн функц болон статик гишүүн өгөгдлийг тодорхойлох

3. ОНОЛЫН СУДАЛГАА

3.1 This хувьсагч гэж юу вэ?, үүнийг компилятор хэрхэн хэрэглэгдэг вэ?

This хувьсагч нь зөвхөн классын гишүүн функц дотор ашиглагдана. Компилятор гишүүн функцийг дуудахаас өмнө функц дотор хамгийн эхэнд тухайн классын this гэх объектон хаяган хувьсагч зарлаж өгдөг. This хувьсагчийг ашигласнаар программ нь уншууртай ойлгоход хялбар болж өгдөг.

3.2 Static гишүүн өгөгдөл гэж юу вэ? Хэрхэн зарладаг вэ?

Статик гишүүн өгөгдөл нь ямар ч объектой холбогдоогүй, классын гишүүн өгөгдлийг хэлнэ. Статик гишүүн өгөгдлийг объект дунд ерөнхий байдлаар үүсгэдэг. Статик гишүүн өгөгдлийг классын тодорхойлолт дотор статик гэх түлхүүр үг ашиглан зарладаг. Гишүүн гишүүд руу объектоор эсвэл классаар нь дамжин хандаж болно. Статик хувьсагч нь ихэнхдээ тогтмол хувьсагч зарлахад ашиглагддаг.

3.3 Static гишүүн өгөгдлийн амьдралын мөчлөг ямар байдаг вэ?

Статик гишүүн өгөгдлийн амьдралын мөчлөг нь программын амьдралын мөчлөгтэй адил гэж хэлж болно. Програм эхлэх үед үүсэж програм дуусах үед устдаг.

3.4 Static гишүүн өгөгдөлд хандах

Статик гишүүн өгөгдөлд көвхөн классаар нь дамжуулж хандана. Ингэхдээ классын тодорхойлолтын гадна тухайн статик гишүүн өгөгдлийн төрөл классын нэр үйлчлэх хүрээний оператор статик гишүүн өгөгдөл гэх маягаар хандана.

```
class X { static int n; }; // зарлах
int X::n = 1;             // хандах
```

3.5 Static гишүүн өгөгдөл ямар ашигтай вэ?

Статик гишүүн функц глобал үйлдэл гүйцэтгэх эсвэл классын бүх глобал өгөгдөлд хэрэглэдэг. Нэг удаа л зарлагддаг болхоор санах ойд сайн нөлөөтэй.

3.6 Static гишүүн функцийг хэрхэн зарладаг, дууддаг вэ?

Статик функцийг классын тодорхойлолт дотор зарлахдаа static тусгай үгийг ганц удаа хэрэглэнэ. Түүнийг функцийн тодорхойлолтын толгой хэсэгт огт хэрэглэхгүй.

```
class Box {
public:
    static int objectCount; // Статик гишүүн өгөгдөл зарлах

    Box() { // Байгуулагч функц
        objectCount++;
    }

    static int getCount() { // Статик гишүүн функц зарлах
        return objectCount;
    }
};

Box::getCount(); // Статик гишүүн функц дуудах
```

3.7 Функц дахин тодорхойлох гэж юу вэ?

Ижил нэр, төрөл нь ижил боловч параметрийн тоо, төрөл, дараалал өөр байдаг олон функц тодорхойлох үйлийг функц дахин тодорхойлох гэнэ.

```
#include <iostream>
using namespace std;

int sum(int a, int b) {
    return a + b;
}

int sum(int a, int b, int c) {
    return a + b + c;
}

int main() {
    cout << sum(1, 2, 3) << endl; // 6
    cout << sum(1, 2) << endl;    // 3

    return 0;
}
```

3.8 Жинхэнэ хийсвэр функц гэж юу вэ?

Зөвхөн зарласан, тодорхойлогдоогүй функцийг жинхэнэ хийсвэр функц гэнэ. Мөн жинхэнэ хийсвэр функц нь хүүхэд класстаа дахин тодорхойлогдсон байх ёстой.

3.9 Хийсвэр класс гэж юу вэ? Хэрхэн ашигладаг вэ??

Хийсвэр класс гэдэг нь тухайн классаас шууд объект үүсгэн ашиглах боломжгүй, зөвхөн удамшуулан хэрэглэх зориулалттай класс юм.

4. ХЭРЭГЖҮҮЛЭЛТ

4.1 Shape классын байгуулагч функц дуудагдах бүрт статик тоолох хувьсагчийг нэгээр нэмэгдүүлэх зарчмаар үүссэн объектийг тоолсон.

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

public:
    static int shapeCount; // Объект тоолох статик хувьсагч зарлах
    void setName(char *customName) {
        name = new char[strlen(customName) + 1];
        strcpy(name, customName);
    }

    Shape(char *customName) {
        name = new char[strlen(customName) + 1];
        strcpy(name, customName);
        shapeCount++; // Тоолуур нэмэгдүүлэх
        cout << "Created shape!" << endl;
    }

    Shape() {
        shapeCount++; // Тоолуур нэмэгдүүлэх
        cout << "Created shape!" << endl;
    }

    ~Shape() {
        delete[] name;
        shapeCount--; // Тоолуур хорогдуулах
        cout << "Deleted shape!" << endl;
    }

    // Тоолуур буцаах
    static int getCount() {
        return shapeCount;
    }
};
```

4.2 Өмнө зарласан static хувьсагчид утга оноох, утгыг нь авах static гишүүн функц бичих

```
// Тоолуурын утга авах
static int getCount() {
    return objectCount;
}

// Тоолуурын утга нэмэгдүүлэх
static void increaseCount() {
    objectCount++;
}
```

4.3 Классын гишүүн функцууд дотор гишүүн функц руу хандахдаа өмнө нь this хувьсагч ашиглах

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

public:
    static int shapeCount; // Объект тоолох статик хувьсагч зарлах
    void setName(char *customName) {
        this->name = new char[strlen(customName) + 1];
        strcpy(this->name, customName);
    }

    Shape(char *customName) {
        this->name = new char[strlen(customName) + 1];
        strcpy(this->name, customName);
        shapeCount++; // Тоолуур нэмэгдүүлэх
        cout << "Created shape!" << endl;
    }

    Shape() {
        shapeCount++; // Тоолуур нэмэгдүүлэх
        cout << "Created shape!" << endl;
    }

    ~Shape() {
        delete[] name;
        shapeCount--; // Тоолуур хорогдуулах
        cout << "Deleted shape!" << endl;
    }

    // Тоолуур буцаах
    static int getCount() {
        return shapeCount;
    }

    virtual float findP() = 0;
    virtual float findS() = 0;
};
```

4.4 Хоёр хэмжээст дүрс классд приметер олох, талбай олох функцуудыг жинхэнэ хийсвэрээр зарлаж хүүхэд классуудад дахин тодорхойлно

```
class TwoDShape : public Shape {
protected:
    int x, y, r;

public:
    TwoDShape(char *customName, int a, int b, int c) : Shape(customName) {
        this->x = a;
        this->y = b;
        this->r = c;
    }
    TwoDShape(char *customName) : Shape(customName) {}

    virtual float findP() = 0;
    virtual float findS() = 0;
};
```

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

```
int main() {
    // Тойрог
    Circle c1("Toirog1", 10, 10, 5);
    Circle c2("Toirog2", 20, 20, 12);

    // Квадрат
    Square s1("Kvadrat1", 10, 10, 2);
    Square s2("Kvadrat2", 20, 20, 5);

    // Гурвалжин
    Triangle t1("Gurvaljin2", 10, 10, 8);
    Triangle t2("Gurvaljin2", 10, 10, 6);

    int n = 6; // Дүрсийн тоо
    TwoDShape *td[n], *temp;

    // Дүрс бүрийн хаягийг td хаяган хүснэгтэнд өгөх
    td[0] = &c1;
    td[1] = &c2;
    td[2] = &s1;
    td[3] = &s2;
    td[4] = &t1;
    td[5] = &t2;

    // Эрэмблэлт хийхээс өмнөх дүрсийн талбай периметер дэлгэцлэх
    cout << endl << "Before Sorting" << endl << "Perimeters: ";
    for (int i = 0; i < n; i++)
        cout << td[i]->findP() << " ";
    cout << endl << "Areas: ";
    for (int i = 0; i < n; i++)
        cout << td[i]->findS() << " ";
    cout << endl << endl;

    cout << "After Sorting" << endl << "Perimeters: ";

    // Периметрээр эрэмблэлт хийх
    for (int i = 0; i < n; i++)
        for (int j = i + 1; j < n; j++) {
            if (td[i]->findP() > td[j]->findP()) {
                temp = td[i];
                td[i] = td[j];
                td[j] = temp;
            }
        }

    // Эрэмблэлт хийсний дараах дүрсийн периметер дэлгэцлэх
    for (int i = 0; i < n; i++)
        cout << td[i]->findP() << " ";

    // Талбайгаар эрэмблэлт хийх
    for (int i = 0; i < n; i++)
        for (int j = i + 1; j < n; j++) {
            if (td[i]->findS() > td[j]->findS()) {
                temp = td[i];
                td[i] = td[j];
                td[j] = temp;
            }
        }

    // Эрэмблэлт хийсний дараах дүрсийн талбай дэлгэцлэх
    cout << endl << "Areas: ";
    for (int i = 0; i < n; i++)
        cout << td[i]->findS() << " ";
    cout << endl << endl;

    return 0;
}
```

```
mendeelslife@money-maker: ~/Classes/object/lab8
mendeelslife@money-maker:~/Classes/object/lab8$ ./a.out
Created shape!
Created shape!
Created shape!
Created shape!
Created shape!
Created shape!

Before Sorting
Perimeters: 31.4 75.36 8 20 24 18
Areas: 78.5 452.16 4 25 27.7128 15.5885

After Sorting
Perimeters: 8 18 20 24 31.4 75.36
Areas: 4 15.5885 25 27.7128 78.5 452.16

Deleted shape!
Deleted shape!
Deleted shape!
Deleted shape!
Deleted shape!
Deleted shape!
mendeelslife@money-maker:~/Classes/object/lab8$
```

5. ДҮГНЭЛТ

Хийсвэр класс ашигланаар объект үүсэхгүй. Хэрэв объект үүсгэх бол зөвхөн хүүхэд классаар нь дамжуулж үүсгэж болно. This хувьсагч ашигласнаар програм уншууртай болж өгдөг. Хийсвэр классыг сайн хэрэгжүүлснээр ерөнхийлөл хийхэд үр ашигтай.

6. АШИГЛАСАН МАТЕРИАЛ

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

7. ХАВСРАЛТ

Github линк, [Lab 8](#)