# Статик хувьсагч Тодорхойлох Лабораторийн Ажил (Лаборатори №8)

Ц. Гэрэл

ХШУИС, Програм хангамжийн анги-2, ts.gerel.ts@gmail.com

# 1. ОРШИЛ/УДИРТГАЛ

Статик хувьсагчыг компьютерийн програмд ашиглахад C++ хэлийг ашиглан тодорхойлж, хэзээ үүсдэг, яаж зарладаг ба new оператортой жишнэ.

# 2. ЗОРИЛГО

Оюутан нэртэй объектын классыг өгөгдсөн шаардлагын дагуу зарлаж, гишүүн функцуудийг тодорхойлно. Үүний тулд дараах зорилтуудыг тавьж ажилласан:

1. Статик хувьсагчийг тодорхойлох,
2. Статик хувьсагч санах ойд хэзээ үүсэхийг тодорхойлох,
3. Статик функц функц тодорхойлох ба хэрхэн зарладаг, хэрхэн утга оноодогийг тодорхойлох,
4. new оператор ашиглан шинээр обьект үүсгэх,
5. this хувьсагч тодорхойлох,
6. статик болон this хувьсагч ашигласан кодын жишээ бичих.

# 3. ОНОЛЫН СУДАЛГАА

## 3.1 Статик хувьсагч

Статик хувьсагчууд[1] нь функц эсвэл файлыг ажиллахын туршид оршин байдаг хувьсагч юм. Тэдгээр нь функц эсвэл файлын гадна мэдэгддэггүй тул дуудах хооронд утгыг хадгалах боломжтой.

Зарлахад static түлхүүр үг ашиглаж, утга оноохдоо хувьсагч зарлагдсан блокдоо л мэдэгдэх ба өөрийн утгаа хадгална гэдгийг санах хэрэгтэй яагаад гэвэл стэк дээр биш санах ойд хадгалагдана.

## 3.2 Статик хувьсагч санах ойд

[2]Статик хувьсагчууд компайлын үед динамик хувьсагчуудаас өмнө дуудагдана. Анхаарах зүйл нь дуудагдах дараалал нь зарласнаар биш тодорхойлсон дарааллаар явахыг санах хэрэгтэй. Програм дуустал санах ойд үлдэнэ.

Утга оноогдоогүй статик хувьсагчууд BSS сегментэд хадгалагдана, утгатайн Дата сегментэнд байна.

## 3.3 Статик функц

Зөвхөн тодорхойлогдсон файлдаа ашиглагдах фүнкц ба зарлагдахдаа static түлхүүр үг ашиглана.

Класс дотор зарлагдсан үед дуудахад объектээр нь биш, классаар дамжуулан дуудна. Үүнийг классын объект үүсээгүй үед дуудаж дуудах боломжтой байлгахын тулд нэмнэ.

## 3.4 Объектийн хаяг

Объектийн хаяг нь заагчийн хувьсагч байна. Классын тодорхойлолтоос хамааран санах ойд динамик байдлаар зай эзэлдэг учраас объектын эхлэж бүй санах ойн хэсэг рүү заана.

new нь үүсгэсэн хаягийн эхлэлийг буцаана.

## 3.5 this хувьсагч

[3] Энэ заагч нь тухайн гишүүн функцийг дуудсан объектын гишүүний өгөгдөлд хандах боломжтой болгодог заагч юм. Компайлер объектээр дамжуулан дуудагдсан фүнкц, өгөгдөл болгонд this дамжуулж дуудсан объектын хаяг дамжуулна.

## 3.6 статик болон this хувьсагч

Хэрэгжүүлэлт нь “Хэрэгжүүлэлт” бүлэгт тодорхойлогдсон.

# 4. ХЭРЭГЖҮҮЛЭЛТ

3.1

int count **(**int i**)**

**{**

static int c**=**0**;**

**if(**i**)** **return** c**;**

**else** c++;

**return** 0**;**

**}**

Компайлер “с”-г статик санах ойн сегментэнд хийнэ. Утга оноосон тул Дата сегментэнд орно. Дуудагч үүний хаягт 0 оруулна. Програм дуустал санах ойд хувьсагчийн утга хадгалагдана.

3.3

#include <iostream>

class menu

**{**

public**:**

static void stat\_func**();**

**};**

void menu**::**stat\_func**()**

**{**

std**::**cout **<<** "Hello, static"**;**

**}**

int main**()**

**{**

menu**::**stat\_func**();**

**}**

Энд харуулснаар классаар дамжин фүнкц дуудагдана. Std::cout стандартыг оруулахгуй бол код ажиллахгүй.

3.4

int **\*** foo**;**

foo **=** **new** int **[**5**];**

Foo дээр хүснэгт объектын эхний утгын хаяг орно. Үүний араас 20 byte дараалан орно.

3.5

Test**&** Test**::**func**()**

**{**

**return** **\*this;**

**}**

Энд фүнкц классын заалт буцаана.

3.6

void function1**()**

**{**

cout **<<** a **<<** endl**;** // Болно

cout **<<** b **<<** endl**;** // Болно

**}**

static void function2**()**

**{**

cout **<<** a **<<** endl**;** // Болно

cout **<<** b**;** // Алдаа

**}**

int A**::**a **=** 0**;** // Статик хувьсагчийн гарааны утга

int main**()**

**{**

A obj**(**1**,**2**);**

obj**.**function1**();** // Болно

obj**.**function2**();** // Болно

A**.**function1**();** // Алдаа

A**::**function2**();** // Болно

**}**

Статикийн ялгааг гаргахын тулд хоёр фүнкц зарласан.

int A**::**a **=** 0**;** // Статик хувьсагчийн гарааны утга

int main**()**

**{**

A obj**(**1**,**2**);**

obj**.**function1**();** // Болно

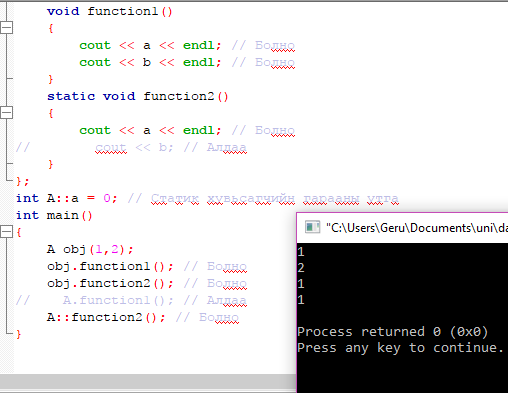
obj**.**function2**();** // Болно

A**.**function1**();** // Алдаа

A**::**function2**();** // Болно

**}**

Объектоор хэрхэн дамжуулж болохыг хауулав



Класс дахь алдаа гаргах хэсэгүүдийг хасаж эхлүүлсэн.

# 5. ДҮГНЭЛТ

Статик фүнкц ба хувьсагч ашишлан өөрийн код сан үүсгэхэд ачтай. 3.6д харсанаар классаар дамжуулан дуудах бол энэхүү код header файлд хийж импортоор ахин ашиглах боломж гарсан.

this хувьсагчаар тодорхой хэрэглэгдэж буй объектын хаягыг авж фүнкц дуудах үед объект дээрх мөр кодыг багасгах боломж гарсан.

# 6. АШИГЛАСАН МАТЕРИАЛ

1. static variables, http://www.c-cpp.ru/books/staticheskie-peremennye

2. static, <http://en.cppreference.com/w/cpp/language/storage_duration#Static_local_variables>

3. this, http://en.cppreference.com/w/cpp/language/this

# 7. ХАВСРАЛТ

#include<iostream>

**using** **namespace** std**;**

class A

**{**

static int a**;**

int b**;**

public**:**

A**(**int a**,** int b**)** // Байгуулагч

**{**

**this->**a **=** a**;** // классын хувьсагч руу заана

**this->**b **=** b**;** // Жирийн объектын гишүүн хувьсагч

**}**

void function1**()**

**{**

cout **<<** a **<<** endl**;** // Болно

cout **<<** b **<<** endl**;** // Болно

**}**

static void function2**()**

**{**

cout **<<** a **<<** endl**;** // Болно

cout **<<** b**;** // Алдаа

**}**

**};**

int A**::**a **=** 0**;** // Статик хувьсагчийн гарааны утга

int main**()**

**{**

A obj**(**1**,**2**);**

obj**.**function1**();** // Болно

obj**.**function2**();** // Болно

A**.**function1**();** // Алдаа

A**::**function2**();** // Болно

**}**