**БАЙГУУЛЛАГЧ, УСТГАГЧ ФУНКЦ БА**

**ХАНДАЛТЫН ТҮВШИНИЙ ТАЛААРХ О**

**(ЛАБОРАТОРИ №4)**

Э.Уранчимэг

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

1. **ОРШИЛ**

Энэ тайлангийн онолын судалгаа хэсэгт байгуулагч функц, анхдагч байгуулагч, устгагч функцын талаар дэлгэрэнгүй тайлбарласан. Мөн функцийг дахин тодорхойлох, болон түүнийг жишээгээр тайлбарласан тайлбар байх ба хандалтын түвшин түүний төрлийн талаар судлан өгүүллээ. Динамик санах ой болон байгуулагч, устгагч функцийг хэрхэн хамтад нь байгуулахыг тайлбарласан. Хэрэгжүүлэлт хэсэгт тодорхойлсон класст анхдагч болон параметртэй байгууллагч тодорхойлсон ба устгагч функц тодорхойлж, устгагч функц хэзээ дуудагдаж байгааг туршилтад үндэслэн харуулсан. Бас гишүүн функц дотроос гишүүн функц дуудах үйлдлийг хийж, set болон get функц бичиж утга оноож, утгыг буцааж авч, тга оноосон хүснэгтийг эрэмбэлж үзүүлсэн. Хавсралт хэсгээс дэлгэрэнгүй кодыг унших боломжтой.

1. **ЗОРИЛГО**

Объектыг байгуулах үед гишүүн функцүүдэд гарааны утга оноож, санах ой нөөцлөж, объектын эзэмшиж байгаа ойг чөлөөлж, функцыг дахин тодорхойлон жишээгээр дуудан хэрэгжүүлсэн. Үүний тулд дараах зорилтуудыг тавьж ажилласан:

1. Классыг тодорхойлох,

2. Хандалтын түвшин бичих,

3. Анхдагч байгууллагч функц тодорхойлох,

4. Параметртэй байгууллагч функц нэмж тодорхойлох,

5. Класст устгагч функц тодорхойлох,

6. Set функц бичиж утга оноох,

7. Get функц бичиж оноосон утгыг буцаах,

1. **ОНОЛЫН СУДАЛГАА**

*3.1 Байгуулагч функц гэж юу вэ? Анхдагч байгуулагч гэж юу вэ?*

Байгууллагч функц гэдэг нь объектыг байгуулах үед гишүүн функцүүдэд гарааны утга оноох, санах ой нөөцлөх зорилготой тусгай зориулалтын функц юм.

Онцлог:

1. Утга буцаахгүй бас void биш.
2. Нэгэнт байгуулагдсан объектоор дамжуулж дуудахгүй.
3. Объект байгуулах үед автоматаар нэг удаа дуудагддаг.
4. Дахин тодорхойлж болно
5. Нэр нь классын нэртэй яг ижил байна.

Байгуулагч функцийг параметртэй болон параметргүй (анхдагч ) гэж 2 ангилдаг. Анхдагч байгуулагч нь хэрэглэгчээс тодорхойлогдоогүй үед c++ компайлэр автоматаар дуудаж ажилдаг. Хэрэглэгчээс тодорхойлсон үед гишүүн өгөгдөлүүдэд гарааны утгыг оноох зарчмаар ажилдаг. Бүх объектыг адилхан байгуулах үед хэрэглэнэ​.

*3.2 Устгагч функц гэж юу вэ?*

Устгагч функц гэдэг нь объектын эзэмшиж байгаа ойг чөлөөлөх үүрэгтэй тусгай зориулалтын функц юм.

Онцлог:

1. Утга буцаахгүй бас void биш.
2. Нэгэнт байгуулагдсан объектоор дамжуулж дуудаж болохгүй.
3. Автоматаар объект устах үед нэг удаа дуудагддаг.
4. Дахин тодорхойлж болохгүй.
5. Классын нэртэй ижил байх ба урд нь ~ тэмдэгийг тавина.

Классад устгагч функцийг илээр тодорхойлж өгөөгүй бол хамгийн бага хэмжээ бүхий анхдагч устгагч функцийг C++ compiler үүсгэж дуудна.

*3.3 Функц дахин тодорхойлох гэж юу вэ? Жишээ тайлбар.*

Ижил нэртэй хэдий ч функцын параметрийг тоо, дараалал өөр байж болдог ба үүнийг функцыг дахин тодорхойлох гэдэг. Аргумент нь төрөл, дараалал, авах параметрийн тоо зэргээс шалтгаалан өөрчлөгдөнө.

Жишээлбэл

Энэ функц нь 2 тоог жиших ба a, b хувьсагч нь бүхэл тоо үед дуудагддаг.

int maxx(int a, int b){

return a>b ? a : b;

}

Энэ функц нь 3 тоог жиших гэж байгаа үед дуудагддаг.

int maxx(int a, int b, int c=6){

int d = a>b ? a : b;

return d>c ? d : c;

}

Энэ функц нь 2 тоог жиших ба a, b хувьсагч нь рациональ үед дуудагддаг.

int maxx(float a, float b){

return a>b ? a : b;

}

Ийм 3 ижил нэртэй ч өөр, өөр аргумент, төрлөөс шалтгаалан дуудагдах байдал нь өөр функцуудыг бичиж болдог нь функц дахин тодорхойлох юм.

*3.4 Хандалтын түвшин гэж юу вэ? Ямар ямар хандалтын түвшин байдаг вэ?*

Хандалтын түвшин гэдэг нь классын гишүүн өгөгдөлд классын хүрээний гадна талаас хандах хандалтыг зохицуулдаг ойлголт юм. Энэ нь 3 төрөлтэй байдаг ба тэдгээр нь: Private, Protected, Public юм.

Private, Protected → Эдгээрийн шинж чанар төстэй бөгөөд удамшлын үед ялгаа нь гарч ирдэг.

* Private - Классын хамрах хүрээний гадна талаас буюу өөр класс эсвэл өөр функц нь гишүүн өгөгдөл, гишүүн функцд объектоор нь дамжуулж хандахыг хориглодог.
* Public - Классын хамрах хүрээний гадна талаас объектоор нь дамжуулж хандаж болдог.

*3.5 Динамик санах ой (new, delete оператор ашигласан) болон байгуулагч, устагч функцийг хэрхэн хамтад нь ашиглах вэ?*

Байгуулагчаар функцаар гишүүн өгөгдөлд динамикаар санах ой нөөцлөн гарааны утга оноосон бол устгагч функцыг тодорхойлохдоо тухайн объектийн динамикаар нөөцөлж өгсөн гишүүн өгөгдөлийн санах ойг чөлөөлөхөөр ашиглана.

1. **ХЭРЭГЖҮҮЛЭЛТ**

1. Lab03-д тодорхойлсон класст анхдагч болон параметртэй байгуулагч нэмж тодорхойлох ба тодорхойлохдоо байгуулагч функцийн үүргийг сайтар хэрэгжүүлсэн байх шаардлагатай.

public:

// primary constructor

employee()

{

name = new char[10];

strcpy(name, " ");

time = 0;

number = 0;

position = new char[10];

strcpy(position, " ");

}

// parameter constructor

employee(int number, char \*name, float time, char \*position)

{

this->number = number;

this->name = new char[strlen(name) + 1];

strcpy(this->name, name);

this->time = time;

this->position = new char[strlen(position) + 1];

strcpy(this->position, position);

}

2. Lab03-д тодорхойлсон класст устгагч функц тодорхойлж устгагч функц хэзээ дуудагдаж байгааг туршилтад үндэслэн хариул.

~employee()

{

cout << "ustgagch function ajillaa" << endl;

delete[] name;

delete[] position;

}

Туршилт:

1 urana zahiral 2.4

32400

194400

2 taivan ajiltan 2.5

8750

50750

3 orgil ajiltan 2.8

9800

51800

ustgagch function ajillaa

ustgagch function ajillaa

1 urana zahiral 14.4

tsalin: 194400

3 ☻ ajiltan 14.8

tsalin: 51800

2 X§· ajiltan 14.5

tsalin: 50750

ustgagch function ajillaa

Байгуулагч функц ажиллаж дуусах үед нөөцөлсөн санах ойгоо чөлөөлж, устгагч функц ажиллаж байна.

3. Lab03-д тодорхойлсон классын захирлын цалин бодох функцийг private хандалтын түвшинтэй болгож өөрчил.

private: /\*Private buyu nuutslaltai handaltiin tuvshind gishuun ugugdul

bolon zahirliin tsalin gishuun functsiig todorhoilson\*/

char \*name; /\*Char buyu temdegt ugugdliin turultei

ner haygan huvisagch zarlasan\*/

int number;/\*Integer buyu buhel toon ugugdliin turultei

dugaar huvisagch zarlasan\*/

float time;/\*Float buyu rational toon ugugdliin turultei

tsag huvisagch zarlasan\*/

char \*position;/\*Char buyu temdegt ugugdliin turultei

alban tushaal haygan huvisagch zarlasan\*/

float directorSalary(void) /\*Zahirliin tsalin bodoh functiong todorhoilson\*/

{

return time \* 10000;

}

Дараа нь цалин бодох функц дотор албан тушаал нь захирал байвал захирлын цалин бодох функцийг цалин бодох функц дотор дуудаж захирлын цалинг бодно. Энд гишүүн функц дотроос гишүүн функц дуудах үйлдлийг хийнэ.

float employee::getSalary()

{

if (strcmp(position, "zahiral") == 0)

{

return 3500 \* time + directorSalary();

}

return 3500 \* time;

}

4. Мөн гишүүн өгөгдөл бүрийг private хандалтын түвшинтэй болгож лекц дээр заасан set болон get функц бичиж утга оноож, утгыг буцааж авна.

// getter setter functions

char \*getName()

{

return name;

}

int getNumber()

{

return number;

}

float getTime()

{

return time;

}

char \*getPosition()

{

return position;

}

char setName(char \*name)

{

delete this->name;

this->name = new char[strlen(name) + 1];

strcpy(this->name, name);

}

void setNumber(int number)

{

this->number = number;

}

void setPosition(char \*position)

{

delete this->position;

this->name = new char[strlen(position) + 1];

strcpy(this->position, position);

}

void setTime(float time)

{

this->time = time;

}

5. Ажилчин классын хүснэгт үүсгээд гараас хэд хэдэн ажилчны утга онооно.

employee arr[n];

6. Утга оноосон хүснэгтийг цалингаар нь эрэмбэлнэ.

int j;

for (int i = 1; i < n; i++)

{

employee temp = arr[i];

for (j = i - 1; j >= 0; j--)

{

if (arr[j].getSalary() < temp.getSalary())

arr[j + 1] = arr[j];

else

break;

}

arr[j + 1] = temp;

}

Программ ажилуулсан туршилтын үр дүн:

ajilchdiin toogoo oruulna uu

3

Anhdagch baiguulagch function ajillaa

Anhdagch baiguulagch function ajillaa

Anhdagch baiguulagch function ajillaa

number oruul: 1

ner oruul: urana

tsag oruul: 2.4

position oruul: zahiral

number oruul: 2

ner oruul: taivan

tsag oruul: 2.5

position oruul: ajiltan

number oruul: 3

ner oruul: orgil

tsag oruul: 2.8

position oruul: ajiltan

1 urana zahiral 2.4

32400

194400

2 taivan ajiltan 2.5

8750

50750

3 orgil ajiltan 2.8

9800

51800

ustgagch function ajillaa

ustgagch function ajillaa

1 urana zahiral 14.4

tsalin: 194400

3 ☻ ajiltan 14.8

tsalin: 51800

2 X§· ajiltan 14.5

tsalin: 50750

ustgagch function ajillaa

1. **ДҮГНЭЛТ**

Энэ лабораторын ажилд хандалтын түвшин болон байгуулагч, устгагч функцын талаар дэлгэрэнгүй судаллаа. Хандалтын түвшин private буюу өгөгдлийн битүүмжлэлээр өөр классын гишүүн функц болон функцуудаас тухайн классын өгөгдөлд хандахаас сэргийлэх ба зөвхөн уг классын функцууд нь гишүүн өгөгдөлд хандаж болох горим. Энэ нь санамсаргүй байдлаар классын гишүүн өгөгдөлд өөрчлөлт гарахаас сэргийлдэг. Харин байгуулагч болон устгагч функц нь программ бичигч өөрөө гараас утга оноох гэж байгаа үед хэрэглэх тусгай зориулалтын функцүүд байдаг.

1. **АШИГЛАСАН МАТЕРИАЛ**

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

1. **ХАВСРАЛТ**

Нэгдсэн кодыг хавсаргав.

#include <iostream>

#include <cstring>

using namespace std;

class employee /\*Ajilchin nertei class todorhoiloh\*/

{

private: /\*Private buyu nuutslaltai handaltiin tuvshind gishuun ugugdul

bolon zahirliin tsalin gishuun functsiig todorhoilson\*/

char \*name; /\*Char buyu temdegt ugugdliin turultei

ner haygan huvisagch zarlasan\*/

int number;/\*Integer buyu buhel toon ugugdliin turultei

dugaar huvisagch zarlasan\*/

float time;/\*Float buyu rational toon ugugdliin turultei

tsag huvisagch zarlasan\*/

char \*position;/\*Char buyu temdegt ugugdliin turultei

alban tushaal haygan huvisagch zarlasan\*/

float directorSalary(void) /\*Zahirliin tsalin bodoh functiong todorhoilson\*/

{

return time \* 10000;

}

public:

// primary constructor

employee()

{

cout << "Anhdagch baiguulagch function ajillaa" << endl;

name = new char[10];

strcpy(name, " ");

time = 0;

number = 0;

position = new char[10];

strcpy(position, " ");

}

// parameter constructor

employee(int number, char \*name, float time, char \*position)

{

cout << "Para baiguulagch function ajillaa" << endl;

this->number = number;

this->name = new char[strlen(name) + 1];

strcpy(this->name, name);

this->time = time;

this->position = new char[strlen(position) + 1];

strcpy(this->position, position);

}

// destructor function

~employee()

{

cout << "ustgagch function ajillaa" << endl;

delete[] name;

delete[] position;

}

// getter setter functions

char \*getName()

{

return name;

}

int getNumber()

{

return number;

}

float getTime()

{

return time;

}

char \*getPosition()

{

return position;

}

char setName(char \*name)

{

delete this->name;

this->name = new char[strlen(name) + 1];

strcpy(this->name, name);

}

void setNumber(int number)

{

this->number = number;

}

void setPosition(char \*position)

{

delete this->position;

this->name = new char[strlen(position) + 1];

strcpy(this->position, position);

}

void setTime(float time)

{

this->time = time;

}

// other functions

void getValue(void);

void showData(void);

float getSalary(void);

bool workedTime(float);

};

void employee::getValue()

{

cout << "number oruul: ";

cin >> number;

name = new char[20];

cout << "ner oruul: ";

cin >> name;

cout << "tsag oruul: ";

cin >> time;

cout << "position oruul: ";

position = new char[10];

cin >> position;

}

float employee::getSalary()

{

if (strcmp(position, "zahiral") == 0)

{

return 3500 \* time + directorSalary();

}

return 3500 \* time;

}

void employee::showData()

{

cout << number << " " << name << " " << position << " " << time << endl;

}

bool employee::workedTime(float time)

{

if (time >= 0 && time <= 24)

{

this->time += time;

return 1;

}

return 0;

}

int main()

{

int n;

cout << "ajilchdiin toogoo oruulna uu" << endl;

cin >> n;

employee arr[n];

for (int i = 0; i < n; i++)

{

arr[i].getValue();

}

for (int i = 0; i < n; i++)

{

arr[i].showData();

cout << arr[i].getSalary() << endl;

arr[i].workedTime(12);

cout << arr[i].getSalary() << endl;

}

int j;

for (int i = 1; i < n; i++)

{

employee temp = arr[i];

for (j = i - 1; j >= 0; j--)

{

if (arr[j].getSalary() < temp.getSalary())

arr[j + 1] = arr[j];

else

break;

}

arr[j + 1] = temp;

}

for (int i = 0; i < n; i++)

{

arr[i].showData();

cout << "tsalin: " << arr[i].getSalary() << endl;

}

}