# Байгуулагч, устгагч функц, функц дахин тодорхойлох (Лаборатори №4)

Я.Тэмүүжин

ХШУИС, МКУТ, ПХ 16B1SEAS3369, y.usott@gmail.com

# 1. ОРШИЛ

Об.

# 2. ЗОРИЛГО

Объект, класс ойлголтуудыг судлаж С++ хэлэнд хэрхэн хэрэгжүүлнэ. Объект хандалтад технологийн класс, объект ойлголтуудыг ашиглаж ажилтны цалин бодох програм бичнэ. Үүний тулд дараах зорилтуудыг тавьж ажиллана.

1. Класс ба объектын онолын судалгааг хийнэ.
2. Ажилтан классын гишүүн өгөгдөл, гишүүн функцуудыг зарлаж хандалтын түвшинг тодорхойлж өгнө
3. Ажилтан классын гишүүн функцуудыг тодорхойлно.
4. Ажилтан классын объект үүсгэж хэрэглэгчид хялбар цэсийг хэвлэж харуулна.

# 3. ОНОЛЫН СУДАЛГАА

## Байгуулагч функц гэж юу вэ?

Байгуулагч функц буюу Байгуулагч нь классын объектод анхны гарааны утга олгогч гишүүн функц юм.Объект байгуулахад түүнд зориулан ой бэлдэхээр анхдагч эсвэл хэрэглэгчийн тодорхойлсон байгуудач заавал дуудагдана. C++хэлэнд классын тохиолдол буюу объект үүсхэд байгуулагч дууудагддаг [1]. Байгуулагчын нэр нь классын нэртэй ижил байх бөгөөд ямар ч утга буцаадаггүй ба анхнаасаа void төрлийнх тул void гэж зааж өгөх шаардлага байдаггүй [2]. Байгуулагч функц нь энгин гишүүн функцээс дараах байдлаар ялгарна байгуулагчын нэр классын нэртэй жил байна, байгуулагч буцаах төрөлгүй, объект үүсэх үед байгуулагч автоматаар дуудагддаг, хэрвээ байгуулагч функц зарлаж өгөөгй бол С++ колпайлер биегүй, параметргүй үүсмэл баййгуулагчыг үүсгэдэг гэдгээрэй энгийн фунзцээс ялгарна [3]. Мөн байгуулагчыг дахин тодорхойлж болох ба ингэснээр обьектыг олон янзаар байгуулах боломжото болдог.

Анхдагч байгуулагч

Дээр гарсанчлан хэрвээ классд байгуулагч функц тодорхойлж өгөөгүй бол С++ компайлер өөрөө программд нэмэж тодорхойлдог харин хөгжүүлэгч програмд байгуулагч функцийг үүсгэвэл компайлер анхдагч байгуулагчыг үүсгэдэггүй ба мөн анхдагч байгуулагчыг ил бадлаар үүсгэж болдог. Анхдагч байгуулагч функц нь ямар ч параметргүй байна.

1. // Cpp program to illustrate the   // concept of Constructors
2. #include <iostream>
3. using namespace std;
4. class construct {
5. public:
6. int a, b;
7. // Анхдагч байгуулагч
8. construct()     {
9. a = 10;
10. b = 20;
11. }
12. };
13. int main() {
14. // Объект үүсхэд анхдагч байгуулагч
15. // автоматаар дуудагдана
16. construct c;
17. cout << "a: " << c.a << endl << "b: " << c.b;
18. return 1;
19. }

Үр дүн

a: 10

b: 20

Параметртэй байгуулагч

Анхдагч байгуулагч нь гишүүн өгөгдөлд тогтмол нэг утга оноодог, объектын гарааны утгийг гаднаас авч чададгүй зэрэг ажиллах хүрээ нь нилээд хязгаарлагдмал байдаг. Анхдагч байгуулагчын эдгээр сул талуудыг параметртэй байгуулагч функц ашиглах замаар арилгаж болдог. Параметртэй байгуулагч гэдэг нь аргументтэй байгуулагчыг хэлнэ. Эдгээр аргументууд нь объек үүсэх үед түүнд гарааны утга хангаж өгдөг. Ашиглахдаа анхдагч байгуулагч адилаар бичээд аргумент дамжуулахад л болно. Байгуулагчын бие хэсгийг тодорхойлохдоо параметрийн утгаар объектын гарааны утгыг хангаж өгнө. Мөн байгуулагч функцыг дахин тодорхойлох аргаар олон байгуулагч функцтэйбайж болно.

1. // Параметртэй байгуулагч
2. #include < iostream >
3. using namespace std;
4. class Point {
5. private:
6. int x, y;
7. public:
8. //Параметртэй байгуулагч
9. Point(int x1, int y1)          {
10. x = x1;
11. y = y1;
12. }
13. int getX()                  {
14. return x;
15. }
16. int getY()         {
17. return y;
18. }
19. };
20. int main() {
21. // байгуулагч дууудагдана
23. Point p1(10, 15);
24. // байгуулагчаар олгогдсон гарааны утгад хандаж байна
25. cout << "p1.x = " << p1.getX() << ", p1.y = " << p1.getY();
26. return 0;
27. }

Үр дүн:

p1.x = 10, p1.y = 15

Объект параметртэй байгуулагчаар зарлагдахдаа байгуулагч функруу заавал анхны утгуудыг аргументээр илгээх хэрэгтэй байдаг.

Хуулагч байгуулагч

Хуулагч байгуулагч нь обектыг тухайн объектын үүссэн классаас шинэ объект үүсгэхдээ гарааны утга болгож ашиглах гишүүн функ юм.

## Устгагч функц гэж юу вэ?

Объект үүсхэд заавал түүнд хэрэгтэй ойг нөөцлөх байгуулагч функц дуудагддаж обьектыг үүсгэдэг бол мөн үүнтэй адилаар объек нь устгагч функцгүйгээр устаж чадахгүй [2]. Өөрөөр хэлбэл устгагч функц нь цааш хэрэглэгдэхгүй объект устахад дуудагдаж түүний эзэмшиж байсан ойг чөлөөлж өгдөг. Хэрвээ хөгжүүлэгч устгагч функц бичээгүй бол байгуулагч функцийн нэгэн адилаар С++ компайлар түүнийг нэмэж өгдөг байна. Устгагч функц нь классынхаа гишүүн функц болох бөгөөд функцын нэр нь классын нэртэй ижил ба урдаа ~ тэмдэгт авна. Жишээ нь МУИС() класс байлаа гэж бодход түүний устгагч функц нь ~МУИС(); юм. Устгагч функ нь утга буцаадаггүй, параметр авахгүй, удамшдаггүй ба класс зөвхөн ганц утгагч функцтэй байна. Доор програм нь классын нөөцөлсөн санах ойг устгагч функцээр чөлөөлж байна.

1. class String {
2. private:
3. char \* s;
4. int size;
5. public:
6. String(char \* ); // байгуулагч
8. ~String();      // устгагч
9. };
10. String::String(char \* c) {
11. size = strlen(c);
12. s = new char[size + 1];
13. strcpy(s, c);
14. }
15. String::~String() {
16. delete[] s;
17. }

## Функц дахин тодорхойлох гэж юу вэ? Жишээгээр тайлбарла

C++ хэлний Функц дахин тодорхойлох гэдэг нь ижил нэртэй боловч өөр параметрүүдтэй байхыг хэлнэ. Өөрөөр хэлбэл нэг үйлчлэх хүрээнд нэг функцийн нэрэнд нэгээс олон тодорхойлолт хийж өгөхийг хэлнэ [4]. Функцын тодорхойлолт, төрөл, аргументын тоогоороо бусад функцээс өөр байх хэрэгтэй. Зөвхөн буцаах утгын төрлийг өөрчилж функцыг дахин тодорхойлж чадахгүй. Жишээ нь: A(), A(a), A(a,b),A(a,b,c) нь ижил нэртэй боловч өөр өөр параметруудтэй өөр функцууд юм.

#include <iostream>

using namespace std;

class printData {

public:

void print(int i) {

cout << "Print int: " << i << endl;

}

void print(double f) {

cout << "Print float: " << f << endl;

}

void print(char\* c) {

cout << "Print character: " << c << endl;

}

};

int main(void) {

printData pd;

// бүхэл тоог хэвлэх

pd.print(5);

// Бодит тоог хэвлэх

pd.print(500.263);

// Тэмдэгт мөр хэвлэх

pd.print("Hello C++");

return 0;

}

Дээрх жишээнээс харвал print функцийн нэрэнд 3 өөр тодорхойлолт хийж болж байгаа бөгөөд 3 тодорхойлолт аргументын төрлөөрөө ялгарч байна. Жишээ нь тэмдэг мөр явуулахад print(char\* c) ажиллаж байна харин бүхэл тоо дамжуулахад print(int i) дуудагдаж байгаа харж байна. Үүнээс харвал дамжуулж буй өгөгдлийн төрлөөс хамаарч ижил нэртэй фунзцуудын аль нь ажиллах нь харагдаж байна.

Үр дүн:

Print int: 5

Print float: 500.263

Print character: Hello C++

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

Байгуулагч функц нь объект үүсэх үед түүнд гарааны утга болон хэрэгцээтэй санах ойг нөөцөлдөг ба харин устагч функц объектын үйлчилэх хүрээ дуусаж устгагдхад түүний эзэмшиж байсан санах ойг чөлөөлөхөд ашиглаж ойн цоорхойгоос сэргийлнэ. Жишээ нь класс тэмдэгтийн цуваан заагч тухайн классын нэр гишүүн өгөгдлийг агуулж байгаа бөгөөд түүнд санах ойг нөөцөлж өгөх ба объект үйлчлэх хүрээ дуусаж устахад устгагч функц ажиллаж объектын эзэмшиж буй санах ойг чөлөөлнө.

Жишээ нь доорх жишээ нь анхдагч байгуулагч функцээр анхны утга өгч мөн name, position –д санах ой нөөцөлж байна.

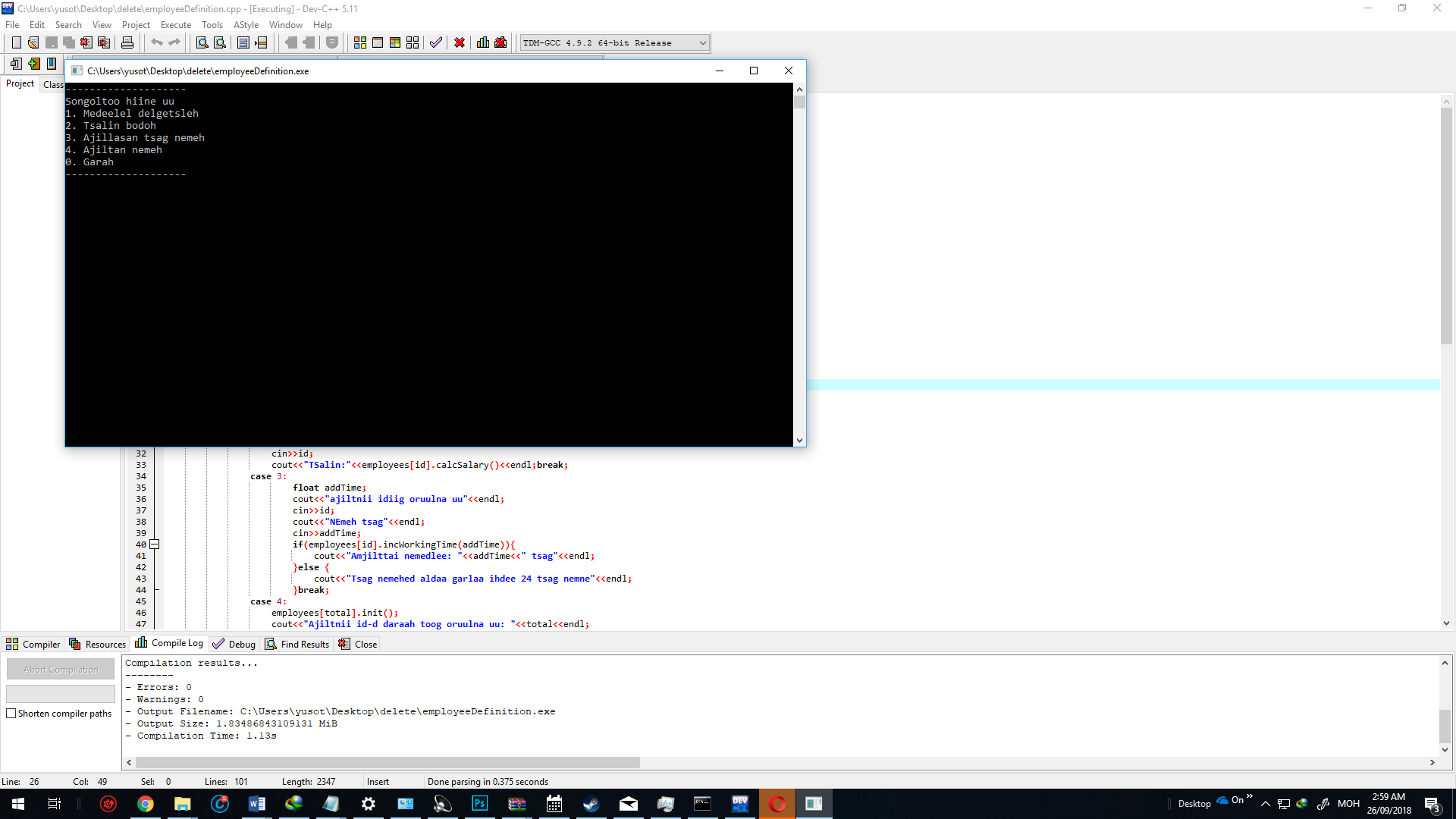
1. EmployeeClass::EmployeeClass() {
2. id = 0;
3. name = new char(strlen("Todorhoiloogui"));
4. position = new char(strlen("Todorhoiloogui"));
5. strcpy(name, "Todorhoiloogui");
6. strcpy(position, "todorhoiloogui");
7. workTime = rand() % 100 + 5;
8. totalEmployee++;
9. }

Харин доорх устгагч функц нь дээр нөөцөлсөн санах ойг чөлөөлнө. Ингэснээс санах ойн цоорхой үүсхээс сэргийлж байна.

1. EmployeeClass::~EmployeeClass() {
2. cout << "Object " << name << " ustlaa" << endl;
3. delete[] name, position;
4. totalEmployee--;
5. }

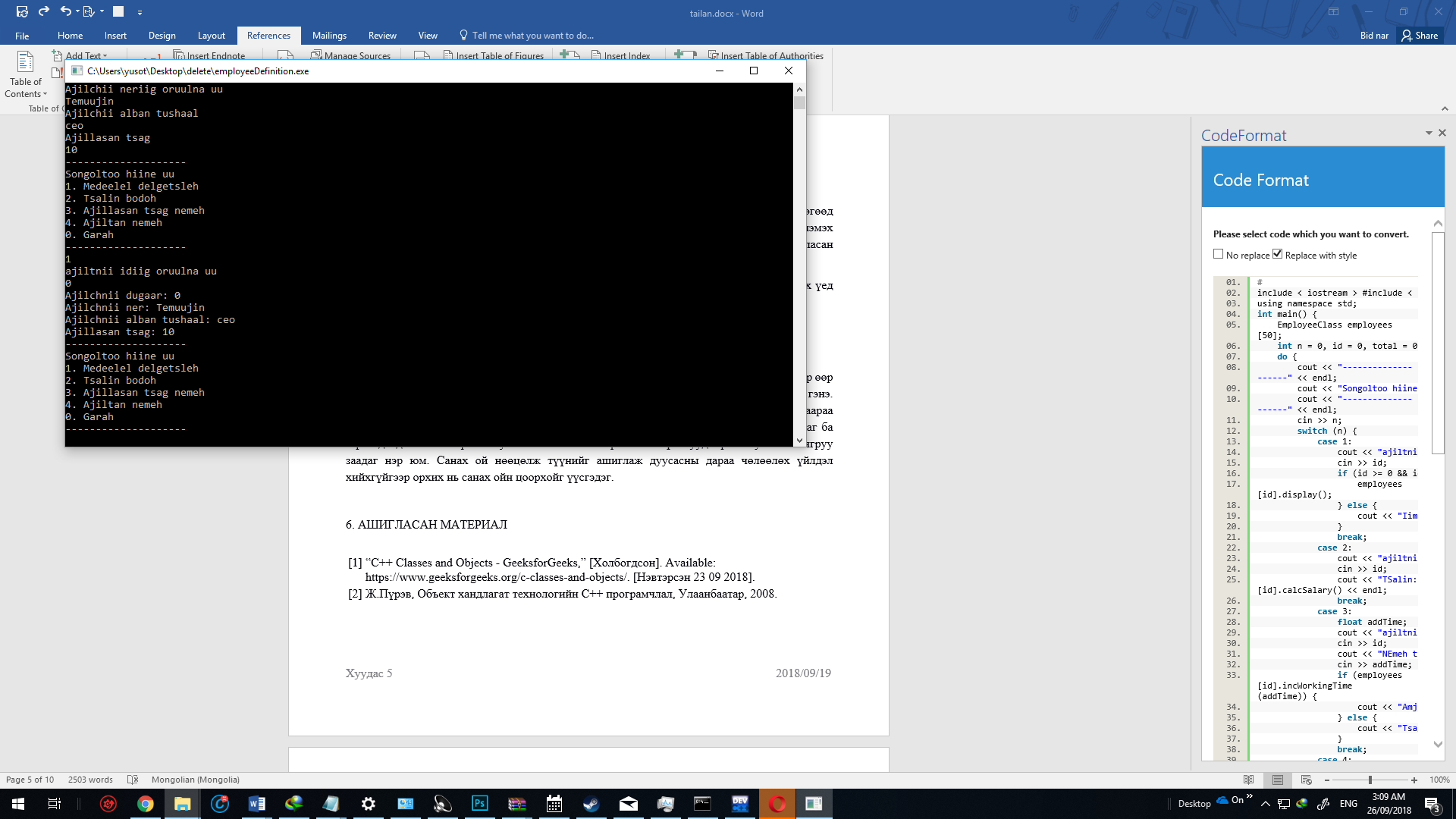
# 4. ХЭРЭГЖҮҮЛЭЛТ

Хавсралт 7.3-н дагуу ажилчин класс болон түүний гишүүн өгөгдөл, гишүүн функцуудыг тодорхойлсон(код хавсралт 7.1 ба 7.2). Тухайн цалингийн програм нь нийт 51 ажилтны мэдээллийг хадгалах бөгөөд мэдээлэл авах, цалин бодох, ажилсан цаг нэмэх, ажилтан нэмэх тухайн функцуудыг дуудах болгонд ажилтны дугаар хэрэг болно. Ажилтны дугаарыг массивын элементийн тоогоор авч явсан. Цалин бодох гишүүн функц нь ажилтаны ажилласан нийт цагийг нэг цагийн хөлсөөр үржүүлж олох бөгөөд үүний дараа ажилтны албан тушаалыг шалгана хэрвээ тухайн ажилтны албан тушаал захирал байвал захиралын цалин бодох функцыг дуудаж түүнээс ирэх хариуг ажилтны цалин бодсон дээр нэмэж захирлын цалинг олно. Хэрэглэгчид амар байх үүднээс программыг сонголттой цэстэй болгосон.

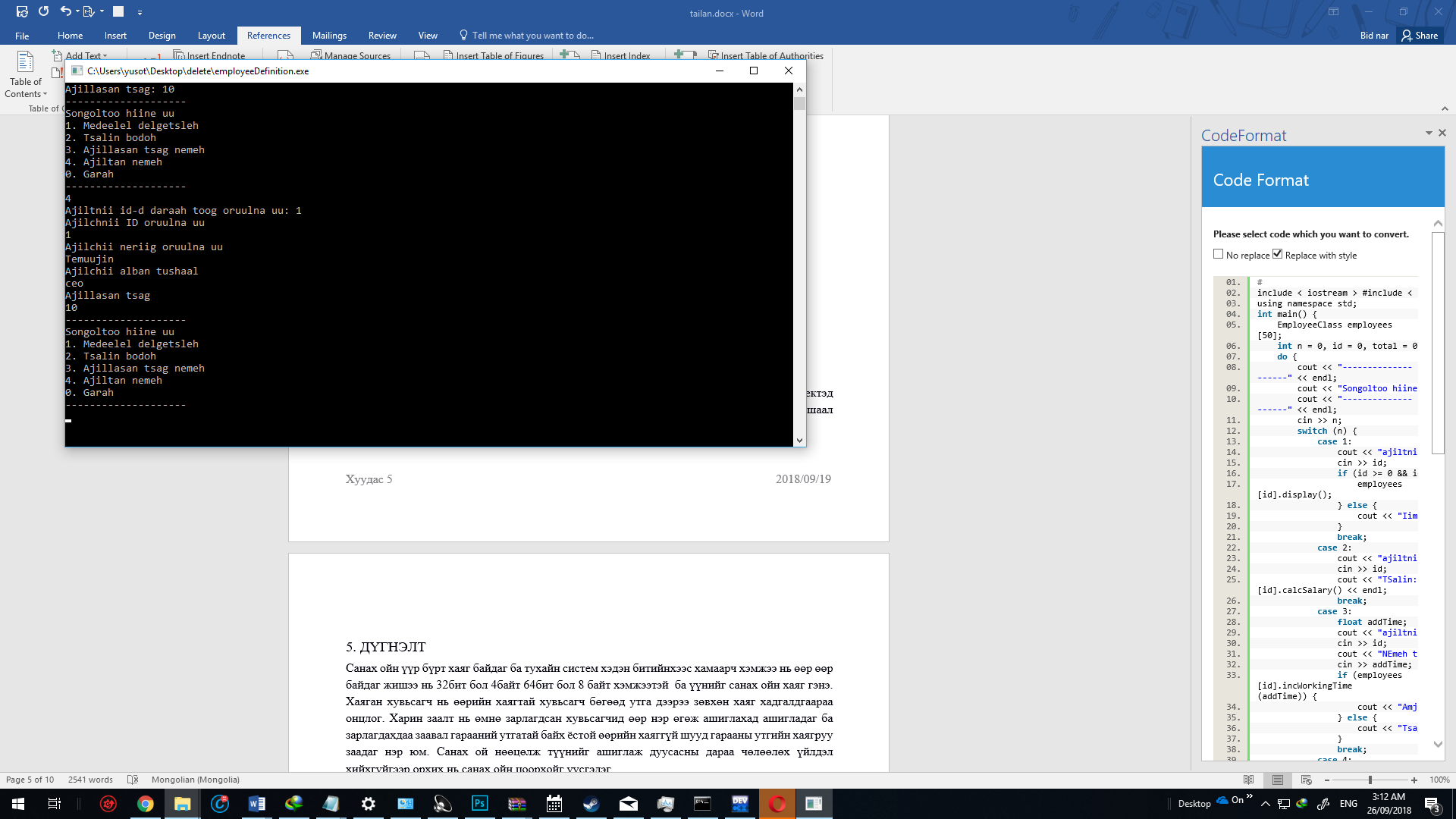


Ажилласан цаг нэмэх функц нь параметрээр нэмэх цагийн хэмжээг агуулах бөгөөд параметрээр авахдаа заалтыг ашигласан нь санах ойг хэмнэж байгаа юм. Ажилласан цаг нэмэх функ нь параметрээр орж ирсэн утга 24-с бага байгаа эсэхийг шалгаах үнэн байвал ажилласан цаг дээр нэмэх гэж буй цагийг нэмээд 1-г буцаана харин 24-с байвал 0-г буцаана.

Мэдээлэл дэлгэцлэх хэсэгт орход мөн ажилтны дугаарыг асуух бөгөөд гараас утга орох үед тухайн индекс дээр буй объектын мэдээллийг харуулна.



Ажилтан нэмэх функц нь өмнө нь нэмэгдсэн n обьектын дараа буюу n+1 индекс дээрх объектэд утга нэмэх ба хэрэглэгчид ямар ажилтны дугаар ашиглахыг зааварлаж өгнө. Мөн албан тушаал дээр ceo гэж бичвэл тухайн объеътын цалин захирлын цалингаар бодогдоно.



# 5. ДҮГНЭЛТ

Класс нь өөрийн гишүүн өгөгдөл, гишүүн функцтэй хэрэгчийн зохионмол өгөдлийн төрөл мөн өөрөөр объектын загвар бөгөөд классыг тодорхойлход санах ой нөөцлөгддөгүй, харин түүний объектыг үүсгэхэд санах ой нөөцлөгдөнө. Объект нь классын нэг удаагийн тохиолдол юм. Объектын гишүүн өгөгдөл, гишүүн функцэд хэрхэн хандах нь түүний хандалтын түвшингээс хамаардаг. Гишүүн функцыг классын тодорхойлолтын дотор болон гадна зарлаж болох бөгөөд гадна зарлахдаа үйлчлэх хүрээний операторыг(::) классын нэр болно функцын нэртэй хамт ашиглана.

6. АШИГЛАСАН МАТЕРИАЛ

|  |  |
| --- | --- |
| [1] | “Constructors in C++,” [Холбогдсон]. Available: https://www.geeksforgeeks.org/constructors-c/. |
| [2] | Ж.Пүрэв, Объект хандлагат технологийн С++ програмчлал, Улаанбаатар, 2008. |
| [3] | “C++ Classes and Objects - GeeksforGeeks,” [Холбогдсон]. Available: https://www.geeksforgeeks.org/c-classes-and-objects/. [Нэвтэрсэн 23 09 2018]. |
| [4] | “Encapsulation in C++ with example,” [Холбогдсон]. Available: https://beginnersbook.com/2017/09/cpp-encapsulation/. [Нэвтэрсэн 23 09 2018]. |

# 7. ХАВСРАЛТ

## 7.1 Классын зарлалт, толгой файл. employeeClass.h

1. #include < iostream >
2. #include < string.h >
3. using namespace std;
4. class EmployeeClass {
5. //Нийт объектын тоог хадгалах статик хувьсагч
6. static int totalEmployee;
7. //Хандалтын түвшинг private байх гишүүн өгөгдлүүдийг тодорхойлж байна
8. private:
9. //ажилчны код
10. int id;
11. //ажилчнын нэр хадгалах гишүүн өгөгдөл
12. char \* name;
13. //албан тушаал хадгалах гишүүн өгөгдөл
14. char \* position;
15. //ажилласан цаг хадгалах гишүүн өгөгдөл
16. float workTime;
17. //хандалтын түвшин public байх функцууд
18. public:
19. //анхдагч байгуулагч функц
20. EmployeeClass();
21. //хуулагч байгуулагч функц
22. //праметрээр ирсэн функцыг дуудаж буй объектын гарааны утга болгоно
23. EmployeeClass(EmployeeClass & emp);
24. //байгуулагч функцийг параметртэйгээр дахин
25. //тодорхойж байна Хэрэглэгчээс объектын бүх гарааны утгийг авах функц
26. EmployeeClass(int id, char \* name, char \* position, float worktime);
27. //Устгагч функц объект устах үед устана
28. ~EmployeeClass();
29. //гишүүн өгөгдлүүдэд гараасутга авах функц
30. void input();
31. //цалин бодох функц
32. float calcSalary();
33. //Объектын мэдээллийг дэлгэцлэх функц
34. void display();
35. //ажилласан цагийг нэмэх функц
36. bool incWorkingTime(float & addTime);
37. // ID гишүүн өгөгдлийн утгыг буцаах функц
38. int getID();
39. // ID гишүүн функцэд утга оноох гишүүн функц
40. void setID(int & n\_id);
41. // Name гишүүн өгөгдлийн утгыг буцаах функц
42. char \* getName();
43. // Name гишүүн функцэд утга оноох гишүүн функц
44. void setName(char \* n\_name);
45. // Position гишүүн өгөгдлийн утгыг буцаах функц
46. char \* getPosition();
47. // Position гишүүн функцэд утга оноох гишүүн функц
48. void setPosition(char \* n\_name);
49. // workTime гишүүн өгөгдлийн утгыг буцаах функц
50. char getWorkTime();
51. // Name гишүүн функцэд утга оноох гишүүн функц
52. void setWorkTime(int a);
53. // Нийт объектын тоог хадгалах     totalEmployee
54. // статик гишүүн өгөгдлийн утгийг буцаах функц
55. static int getTotalEmployee();
56. private:
57. //захиралын цалин бодох функц
58. float calcCeoaSalary();
59. };
60. //Илгээсэн объектон массивыг цалингаар өсөх эрэмбээр эрэмбэлэх
61. //функц employees[] - объектон массив
62. //totalEmployee - нийт ажилчдын тоо
63. void sort(EmployeeClass employees[], int totalEmployee);
64. /\*Ажилчны айдагаар массивын хэддүгээр индекст байгааголж буцаах функц employees[] - объектон массивtotalEmployee - нийт ажилчдын тооid - хайж буй ажилчны код\*/
65. int findById(EmployeeClass employees[], int id, int totalEmployee);

## 

## 7.2 Классын тодорхойлолт. employeeDefinition.cpp

1. #
2. include "employeeClass.h"#
3. include < ctime >
4. /\* Анхдагч байгуулагч объект үүсэх үед дуудагдаж гарааны утга онооноworkTime-д санамсаргүй утга өгсөн нь тест хийх зориулалтаар\*/
5. EmployeeClass::EmployeeClass() {
6. id = 0;
7. name = new char(strlen("Todorhoiloogui"));
8. position = new char(strlen("Todorhoiloogui"));
9. strcpy(name, "Todorhoiloogui");
10. strcpy(position, "todorhoiloogui");
11. workTime = rand() % 100 + 5;
12. totalEmployee++;
13. }
14. /\* Хуулагч байгуулагч   Дуудсан объектод параметрэtotalEmployeeэр орж ирсэн объектын утгийг     гарааны утга болгож өгнө.\*/
15. EmployeeClass::EmployeeClass(EmployeeClass & emp) {
16. id = emp.id;
17. strcpy(name, emp.name);
18. strcpy(position, emp.position); // workTime = emp.workTime;
19. workTime = 0;
20. totalEmployee++;
21. }
22. /\*Параметртэй байгуулагч функцГараас бүх гишүүн өгөгдөлд гарааны утга авна\*/
23. EmployeeClass::EmployeeClass(int n\_id, char \* n\_name, char \* n\_position, float n\_worktime) {
24. id = n\_id;
25. name = new char[strlen(n\_name) + 1];
26. position = new char[strlen(n\_position) + 1];
27. workTime = n\_worktime;
28. totalEmployee++;
29. }
30. /\*Устгагч функцОБъект устах үед дуудагдах бөгөөд санах ойг чөлөөлж ойн цоорхой үүсхээс хамгаалж байна\*/
31. EmployeeClass::~EmployeeClass() {
32. cout << "Object " << name << " ustlaa" << endl;
33. delete[] name, position;
34. totalEmployee--;
35. }
36. /\*Объектын гишүүн өгөгдлүүдэд гараас утга авах функц\*/
37. void EmployeeClass::input() {
38. cout << "Ajilchnii ID oruulna uu" << endl;
39. cin >> id;
40. cout << "Ajilchii neriig oruulna uu" << endl;
41. cin >> name;
42. cout << "Ajilchii alban tushaal" << endl;
43. cin >> position;
44. cout << "Ajillasan tsag" << endl;
45. cin >> workTime;
46. }
47. /\*Цалин бодох функц\*/
48. float EmployeeClass::calcSalary() {
49. float back = workTime \* 2000;
50. if (strcmp(position, "ceo") == 0) {
51. back += calcCeoaSalary();
52. }
53. return back;
54. }
55. /\*Захиралын цалин бодох функц\*/
56. float EmployeeClass::calcCeoaSalary() {
57. return workTime \* 5000;
58. }
59. /\*Дуудсан объектын мэдээллийг харуулах функц\*/
60. void EmployeeClass::display() {
61. cout << "Ajilchnii dugaar: " << id << endl;
62. cout << "Ajilchnii ner: " << name << endl;
63. cout << "Ajilchnii alban tushaal: " << position << endl;
64. cout << "Ajillasan tsag: " << workTime << endl;
65. }
66. /\*Ажилласан цагийг нэмэх функц\*/
67. bool EmployeeClass::incWorkingTime(float & addTime) {
68. if (addTime <= 24) {
69. workTime += addTime;
70. return 1;
71. } else {
72. return 0;
73. }
74. } // ID гишүүн өгөгдлийн утгыг буцаах функц
75. int EmployeeClass::getID() {
76. return id;
77. } // ID гишүүн функцэд утга оноох гишүүн функц
78. void EmployeeClass::setID(int & n\_id) {
79. id = n\_id;
80. } // Name гишүүн өгөгдлийн утгыг буцаах функц
81. char \* EmployeeClass::getName() {
82. return name;
83. } // Name гишүүн функцэд утга оноох гишүүн функц
84. void EmployeeClass::setName(char \* n\_name) {
85. if (name) {
86. delete[] name;
87. }
88. strcpy(name, n\_name);
89. } // Position гишүүн өгөгдлийн утгыг буцаах функц
90. char \* EmployeeClass::getPosition() {
91. return position;
92. } // Position гишүүн функцэд утга оноох гишүүн функц
93. void EmployeeClass::setPosition(char \* n\_pos) {
94. if (position) {
95. delete[] position;
96. }
97. strcpy(name, n\_pos);
98. } // workTime гишүүн өгөгдлийн утгыг буцаах функц
99. char EmployeeClass::getWorkTime() {
100. return workTime;
101. } // setWorkTime гишүүн функцэд утга оноох гишүүн функц
102. void EmployeeClass::setWorkTime(int a) {
103. workTime = a;
104. } //Нийт объектын тоог хадгалах статик хувьсагч
105. int EmployeeClass::totalEmployee = 0; // Нийт объектын тоог хадгалах    totalEmployee  // статик гишүүн өгөгдлийн утгийг буцаах функц
106. int EmployeeClass::getTotalEmployee() {
107. return totalEmployee;
108. } //Илгээсэн объектон массивыг цалингаар өсөх эрэмбээр эрэмбэлэх //функц employees[] - объектон массив //totalEmployee - нийт ажилчдын тоо
109. void sort(EmployeeClass employees[], int totalEmployee) {
110. int i, j;
111. EmployeeClass key;
112. for (i = 1; i < totalEmployee; i++) {
113. key = employees[i];
114. j = i - 1;
115. while (j >= 0 && employees[j].calcSalary() > key.calcSalary()) {
116. employees[j + 1] = employees[j];
117. j -= 1;
118. }
119. employees[j + 1] = key;
120. }
121. }
122. /\*Ажилчны айдагаар массивын хэддүгээр индекст байгааголж буцаах функц employees[] - объектон массивtotalEmployee - нийт ажилчдын тооid - хайж буй ажилчны код\*/
123. int findById(EmployeeClass employees[], int id, int totalEmployee) {
124. for (int i = 0; i < totalEmployee; i++) {
125. if (employees[i].getID() == id) {
126. return i;
127. }
128. }
129. return -1;
130. }

## 7.3 Main функц. employeeMain.cpp

1. # include "employeeClass.h"
2. int main() {
3. //EmployeeClass классын объектон массив үүсгэж байна
4. EmployeeClass employees[10];
5. /\*    n- сонголтын дугаар хадгалах    id - гараас орох ажилчны код    total- нийт ажилчны тоо    \*/
6. int n = 0, id = 0, total = 0;
7. do {
8. cout << "--------------------" << endl;
9. cout << "Songoltoo hiine uu" << endl << "1. Medeelel delgetsleh" << endl << "2. Tsalin bodoh" << endl << "3. Ajillasan tsag nemeh" << endl << "4. Ajiltan nemeh" << endl << "5. Ajiltan hewleh buh" << endl << "0. Garah" << endl;
10. cout << "--------------------" << endl;
11. //гараас сонголтын дугаарыг авч байна
12. cin >> n;
13. //кодоор хайж ажилчны байгаа массивын индексиг заах хувьсагч
14. int response; //сонголтод үндэслэж switch ашиглажбайна
15. switch (n) {
16. case 1:
17. cout << "ajiltnii idiig oruulna uu" << endl;
18. //Мэдээлэл харах ажилчны кодыг авч байна
19. cin >> id;
20. //ажилчны кодоор массивын хэддүгээр индекст байгаа хайж байна
21. response = findById(employees, id, EmployeeClass::getTotalEmployee());
22. //Массивт байгаа эсэхийг шалгаж байна
23. if (response >= 0) {
24. //ажилчны мэдээллийг хэвлэж байна.
25. employees[response].display();
26. } else {
27. cout << "Iim idtai hreglegch bhgui" << endl;
28. }
29. break;
30. case 2:
31. cout << "ajiltnii idiig oruulna uu" << endl;
32. //Цалин бодох ажилчны кодыг авч байна
33. cin >> id;
34. //ажилчны кодоор массивын хэддүгээр индекст байгаа хайж байна
35. response = findById(employees, id, EmployeeClass::getTotalEmployee());
36. //Массивт байгаа эсэхийг шалгаж байна
37. if (response >= 0) {
38. // Ажилчны цалинг бодож байна
39. cout << "Tsalin: " << employees[response].calcSalary() << endl;
40. } else {
41. cout << "Iim idtai hreglegch bhgui" << endl;
42. }
43. break;
44. case 3:
45. float addTime;
46. cout << "ajiltnii idiig oruulna uu" << endl;
47. cin >> id;
48. //ажилчны кодоор массивын хэддүгээр индекст байгаа хайж байна
49. response = findById(employees, id, EmployeeClass::getTotalEmployee());
50. if (response >= 0) {
51. cout << "NEmeh tsag" << endl;
52. //Нэмэх цагиг гараас авч байна
53. cin >> addTime;
54. //Тухайн ажилтанд цагийг нэмэж байна
55. if (employees[response].incWorkingTime(addTime)) {
56. cout << "Amjilttai nemedlee: " << addTime << " tsag" << endl;
57. } else {
58. cout << "Tsag nemehed aldaa garlaa ihdee 24 tsag nemne" << endl;
59. }
60. } else {
61. cout << "Iim idtai hreglegch bhgui" << endl;
62. }
63. break;
64. case 4:
65. // cout<<"Ajiltnii id-d daraah toog oruulna uu: "<<total<<endl;
66. //Дараагийн индекст утга авч байна
67. employees[total].input();
68. //Нийт ажилчнаа нэгээр нэмэгдүүлж байна
69. total++;
70. //Цалингаар өсхөөр эрэмбэлж байна
71. sort(employees, total);
72. break;
73. case 5:
74. //Бүх ажилчны мэдээллийг хэвлэж байна //мөн тэдгээрийн цалинг
75. for (int i = 0; i < total; i++) {
76. cout << "---------------" << endl;
77. employees[i].display();
78. cout << "Tsalin: " << employees[i].calcSalary() << endl;
79. cout << "---------------" << endl;
80. }
81. break;
82. }
83. //n нь 0 биш тохиолдолд давт
84. } while (n != 0);
85. }

## 7.4 Лаботорын ажлын даалгавар

Онолын хэсэг

1. Байгуулагч функц гэж юу вэ?

2. Устгагч функц гэж юу вэ?

3. Функц дахин тодорхойлох гэж юу вэ? Жишээгээр тайлбарал.

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

Хэрэгжүүлэлт

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

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

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

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

Дээрх өөрчлөлтүүдийг оруулаад дараах даалгаврийг гүйцэтгэ.

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

6. Утга оноосон хүснэгтийг цалингаар нь эрэмбэлнэ.