# Оператор дахин тодорхойлох (Лаборатори №5)

Я.Тэмүүжин

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

# 1. ОРШИЛ

С++ хэлний операторыг дахин тодорхойлох, түүнийг хэрэглээ ба хэрэгжүүлэлийг матриц дээр хийж гүйцэтгэнэ..

# 2. ЗОРИЛГО

С++ хэлний операторуудыг дахин тодорхойлж тэдгээрийг матриц хооронд нэмэх, хасах, үржих, утга оноох, хөрвүүлэх үйлдэлийг хийдэг байхаар тооцоолж хэрэгжүүлнэ.

Үүний тулд дараах зорилтуудыг тавьж ажиллана.

1. С++ хэлэнд операторыг дахин тодорхойлох арга зам, бичиглэлийг судлана.
2. Матриц гэж юу болох, түүн дээр хийгддэг үйлдлүүдийг судлаж матрицыг классаар хэрхэн илэрхийлэх гишүүн өгөгдөл, функцыг зарлана.
3. Матриц дээр хийгдэх үйлдэл болгонд тохирох операторыг дахин тодорхойлж логик үйлдлүүдийг хийнэ.

# 3. ОНОЛЫН СУДАЛГАА

## Оператор дахин тодорхойлох гэж юу вэ?

С++ хэлний int, float зэрэг дотоод суурь төрлийн өгөгдөл дээр +,-,\*,/,\*= оператороор үйлдэл хийхэд компайлер шууд тайлж уншиж холбогдох үйлдэлийг хийж чаддаг. Харин хэрэглэгчийн зохиосон зохиомол өгөгдлийн төрөл дээр үндсэн операторуудыг шууд ашиглаж болдоггүй. [1] Жишээ нь энэ лабораторын ажилд хэрэгжүүлэх гэж байгаа матрицийг үндсэн + оператороор шууд нэмэж болохгүй. Иймд операторын үйлдлийг дахин тодорхойлох хэрэгтэй болдог ба бидний жишээн дээр + операторыг матриц нэмэх үйлдлийг хийдгээр дахин тодорхойлох юм.

Операторыг дахин тодорхойлохдоо

1. <БуцаахТөрөл> operator <операторТэмдэг>(<Праметр>);

Жишээ нь хоёр матрицийг нэмдэг + операторыг дахин тодорхойлъё

1. Matrix operator +(Matrix &a);

Дээрх зарлалтын тодорхойлолт нь

1. Matrix Matrix::operator +(**float** &a){
2. **int** row = **this**->getRow(),
3. col = **this**->getCol();
4. Matrix temp(row,col);
5. **for**(**int** i=0;i<**this**->getRow();i++){
6. **for**(**int** j=0;j<**this**->getCol();j++){
7. **float** valuesSum= **this**->getValues(&i,&j)+a;
8. temp.setValues(&i,&j,&valuesSum);
9. }
10. }
11. **return** temp;
12. }

## Устгагч функц хэзээ дуудагддаг вэ?

Объект үүсхэд заавал түүнд хэрэгтэй ойг нөөцлөх байгуулагч функц дуудагддаж обьектыг үүсгэдэг бол мөн үүнтэй адилаар объек нь устгагч функцгүйгээр устаж чадахгүй [1]. Өөрөөр хэлбэл устгагч функц нь цааш хэрэглэгдэхгүй объект устахад дуудагдаж түүний эзэмшиж байсан ойг чөлөөлж өгдөг. Устгагч функц нь объект устгагдах үед дуудагддаг. 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. }

## Хуулагч байгуулагч гэж юу вэ? Ач холбогдол нь юу вэ?

Классаас шинэ объект үүсгэхдээ өмнө нь үүсгэсэн объектын утгийг гарааны утга болгон авахдаа хуулаг байгуулагч функцыг ашигладаг. Хуулагч байгуулагч нь объект үүсэх үед л дуудагдаж болно.

Бичиглэл

КлассНэр (const КлассНэр &хуучин\_объект);

Хуулагч байгуулагчын жишээ

1. #include<iostream>
2. **using** **namespace** std;
4. **class** Point
5. {
6. **private**:
7. **int** x, y;
8. **public**:
9. Point(**int** x1, **int** y1) { x = x1; y = y1; }
11. // Хууланч байгуулагч
12. Point(**const** Point &p2) {x = p2.x; y = p2.y; }
14. **int** getX(){**return** x; }
15. **int** getY(){**return** y; }
16. };
18. **int** main()
19. {
20. Point p1(10, 15); // Параметрт байгуулагч
21. Point p2 = p1; // Хуулагч Байгуулагч дуудагдаж байна
23. // дээр үүсгэсэн объектын утгийг харцгаая
24. cout << "p1.x = " << p1.getX() << ", p1.y = " << p1.getY();
25. cout << "\np2.x = " << p2.getX() << ", p2.y = " << p2.getY();
27. **return** 0;
28. }

Үр дүн

p1.x = 10, p1.y = 15

p2.x = 10, p2.y = 15

Хуулагч байгуулагч хэзээ дуудагдах вэ?

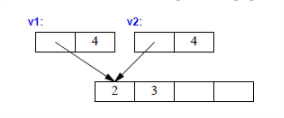
* 1. Объект классын өөр объектийн мэдээлэлтэй ижилхэн үүсгэх үед
  2. Объект параметрээр дамжих үед
  3. Компайлар түр объект үүсгэх үед

## Хуулагч функц гэж юу вэ? Ач холбогдол нь юу вэ? Санах ойн цоорхойгоос хэрхэн сэргийлэх вэ?

Хөгжүүлэгч хуулагч байгуулагч үүсгээгүй бол С++ компайлар заяагдмал хуулагч байгуулагчыг үүсгэдэг. Хэрвээ объект заагч, эсвэл санах ой нөөцөлж байгаа бол түүнд зориулж хуулагч функц үүсгэх хэрэгтэй байдаг. Заяагдмал хуулаг байгуулагч нь доорх жишээгээр утгыг шууд оноодог [2].

1. **void** bad\_copy(Vector v1)
2. {
3. Vector v2 = v1;
4. v1[0] = 2; // v2 ч мөн адил 2!
5. v2[1] = 3; // v1 ч мөн адил 3!
6. }

Үүнийг дүрсэлвэл



Бөгөөд үүнээс үзхэд V2т V1-н утга биш түүний хаяг нь оноогдсон байна. Мөн энэ үйлдэл нь санах ойн цоорхойг үүсгэдэг, мөн өгөгдлийн битүүмжлэлийг алдагдуулдаг байна. Дээрх жишээнээс хархад v1, v2-н зааж буй хаяг нь ижил байна нэг нь мэдээллээ өөрчилхөд нөгөөгийнх мэдээлэл өөрчлөгдөнө гэсэн үг юм. Үүнээс сэргийлэхийн тулд санах ой нөөцөлсөн тохиолдол хэрэглэгчийн хуулагч гишүүн функцыг тодорхойл өгдөг

1. String::String(**const** String& old\_str)
2. {
3. size = old\_str.size;
4. s = **new** **char**[size+1];
5. strcpy(s, old\_str.s);
6. }

Харин дээр жишээ нь орж ирсэн тэмдэгт мөрийн цувааны хэмжээтэй санах ой нөөцөлж түүндээ тэмдэг мөрийг оноож, оноосон утгаа гишүүн функцрүү хийж өгч байна. ингэснээс өмнөх жишээн дуур дурьдсан

## Объектын хаяган хувьсагчийг хэрхэн зарлах вэ? new оператороор санах ой нөөцлөх, хаяган хувьсагчаар дамжуулж объектын гишүүн өгөгдөл, гишүүн функцэд яаж хандах вэ?

Юуны өмнө тухайн объект үүсэж буй классаас хаяган хувьсагч үүсгэх хэрэгтэй үүний тулд

КлассНэр \*ХаяганХувьсагчийнНэр；

Хаяган хувьсагчаа үүсгэсний дараа түүнд харгалзах санах ойг нөөцөлж өгнө. Үүний тулд new түлхүүр үгийн тусламжтайгаар үүсгэж өгнө.

ХаяганХувьсагчийнНэр = **new** КлассНэр();

Ингэснээр бид ХаяганХувьсагчийнНэр хяаган хувьсагчид КлассНэр() классын хэмжээтэй санах ойг нөөцөлж өгнө. Түүний гишүүн өгөгдөлд хандахдаа ХаяганХувьсагчийнНэр**->**гишүүнӨгөгдөл эсвэл ХаяганХувьсагчийнНэр->гишүүнФунц() гэж хандаж болно. Өөрөөр хэлвэл -> операторыг ашиглана. Мэдээж санах нөөцөлж өгсөн бол түүнийг ашиглаж дуусах үедээ чөлөөлөр үйлдэлийг хийх хэрэгтэй үүнийг delete түлхүүр үгийн тусламжтайгаар хийдэг.

DELETE ХаяганХувьсагчийнНэp;

## Объектын хаяган хувьсагчийн хүснэгт үүсгэх жишээ код бичиж ажиллуул.

EmployeeClass \*employees[2];  //Объектын хаяган хүснэгт үүсгэж байна

1. **For**(**int** i=0;i<2;i++){
2. **char** name[20];
3. **int** id;
4. employees[i] = **new** EmployeeClass();
5. cin>>id;
6. **while**(!employees[i]->checkID(id,employees,EmployeeClass::getTotalEmployee())){
7. cout<<"dahij oruul iim id tai ajiltan bna"<<endl;
8. cin>>id;
9. }
10. employees[i]->setID(id);
11. cin>>name;
12. employees[i]->setName(name);
13. cout<<"Neree oruul"<<endl;
14. cin>>name;
15. employees[i]->setPosition(name);
16. cout<<employees[i]->getName()<<" nemegdlee"<<endl;
17. }

Дээрх жишээний эхний хэсэг EmployeeClass классаас \*employees[2];  хэмжээтэй хаяган хүснэгт үүсгэж байна. Харин 1 – 13 мөрөнд үүсгэсэн хаяган хүснэгтэндээ нэг нэгээр хаяг нөөцөлж шинэ объект үүсгэж түүний гишүүн өгөгдөл утга оноож байна.

# 4. ХЭРЭГЖҮҮЛЭЛТ

4.1 Нэрээр эрэмбэлэх функц

Тус функцыг хэрэгжүүлэхдээ сонгон эрэмбэлэлтийн арга ашигласан бөгөөд параметрээр орж ирсэн объектын нэрийг бусад объктын нэртэй strcmp функрээр харьцуулж хаягийн солих замаар эрэмбэлсэн

1. **void** sortName(EmployeeClass \*\*employees, **int** totalEmployee){
2. EmployeeClass \*key;
3. **for**(**int** i=0;i<totalEmployee;i++){
4. **for**(**int** j=i+1;j<totalEmployee;j++){
5. **if**(strcmp(employees[i]->getName(),employees[j]->getName())>0){
6. key = employees[j];
7. employees[j] = employees[i];
8. employees[i] = key;
9. }
10. }
11. }
12. }

4.2 Ажилтны дугаар өмнө нь бүртгэгдсэн эсэхийг шалгах функц

Тус функц нь параметрээр хүлээж авсан ажилтны дугаарыг, мөн параметрээр хүлээж авсан хүснэгтээс харьцуулах засаар хайна. Ингэхдээ хэрвээ тухайн дугаартай тэнцэх дугаар олдвол худал олдоогүй бол үнэн утгыг буцаана.

1. **bool** EmployeeClass:: checkID(**int** id,EmployeeClass \*\*employees,**int** totalEmployee){
2. **for**(**int** i=0;i<totalEmployee;i++){
3. **if**(id==employees[i]->getID()){
4. **return** **false**;
5. }
6. }
7. **return** **true**;
8. }

# 5. ДҮГНЭЛТ

Анхдагч байгуулагч, хуулагч байгуулагч, параметртэй байгуулагч болон бусад дахин тодорхойлсон байгуулагчид нь зөвхөн объект үүсэх үед л дуудагддаг ба үүссэний дараа тэдгээрт хандаж болохгүй. Харин устгагч байгуулагч нь зөвхөн объект устах үед дуудагдах ба түүнийг дахин тодорхойлж болдоггүй. Мөн хуулагч байгуулагч нь санах ойн нөөцөлсөн объектыг хуулахдаа санах ойн цоорхой үүсгэх тул хэрэглэгчийн хуулагч функц бичих замаар шийдвэрлэдэг байна.

6. АШИГЛАСАН МАТЕРИАЛ

|  |  |
| --- | --- |
| [1] | Ж.Пүрэв, Объект хандлагат технологийн С++ програмчлал, Улаанбаатар, 2008. |
| [2] | B. Stroustrup, The C++ Programming Language 4th Edition, 2013. |
| [3] | “C++ Classes and Objects - GeeksforGeeks,” [Холбогдсон]. Available: https://www.geeksforgeeks.org/c-classes-and-objects/. [Нэвтэрсэн 23 09 2018]. |
| [4] | “Constructors in C++,” [Холбогдсон]. Available: https://www.geeksforgeeks.org/constructors-c/. |
| [5] | “C++ Overloading,” [Холбогдсон]. Available: https://www.tutorialspoint.com/cplusplus/cpp\_overloading.htm. [Нэвтэрсэн 23 09 2018]. |

# 7. ХАВСРАЛТ

7.1 employeeMain.cpp

1. #include "employeeClass.h"
3. **int** main(){
4. //EmployeeClass классын объектон массив үүсгэж байна
5. EmployeeClass \*employees[2];
6. **for**(**int** i=0;i<2;i++){
7. **char** name[20];
8. **int** id;
9. employees[i] = **new** EmployeeClass();
10. cin>>id;
11. **while**(!employees[i]->checkID(id,employees,EmployeeClass::getTotalEmployee())){
12. cout<<"dahij oruul iim id tai ajiltan bna"<<endl;
13. cin>>id;
14. }
15. employees[i]->setID(id);
16. cin>>name;
17. employees[i]->setName(name);
18. cout<<"Neree oruul"<<endl;
19. cin>>name;
20. employees[i]->setPosition(name);
21. cout<<employees[i]->getName()<<" nemegdlee"<<endl;
22. }
23. // cout<<"ss"<<EmployeeClass::getTotalEmployee()<<endl;
24. sortName(employees,EmployeeClass::getTotalEmployee());
26. **for**(**int** i=0;i<2;i++){
27. employees[i]->display();
28. }
30. **for**(**int** i=0;i<2;i++){
31. **delete** employees[i];
32. }
33. }

## 7.2 Толгой файл(employeeClass.h)

1. #include <iostream>

## 7.3 Классын тодорхойлолт(employeeDefinition.cpp)

1. #include "employeeClass.h"
2. #include <stdlib.h>
3. #include<ctime>
4. /\*
5. Анхдагч байгуулагч
6. объект үүсэх үед дуудагдаж гарааны утга онооно
7. workTime-д санамсаргүй утга өгсөн нь тест хийх зориулалтаар
8. \*/
9. EmployeeClass::EmployeeClass(){
10. //ажилласан цагийг өөр өөр байлгах үүднээс туршилтан хэрэг болно
11. workTime=rand() % 100 + 5;
12. **char** defaultString[15] = "Todorhoiloogui";
13. setAll(0, defaultString, defaultString, workTime);
14. totalEmployee++;
15. }
17. /\*
18. Хуулагч байгуулагч
19. Дуудсан объектод параметрэtotalEmployeeэр орж ирсэн объектын утгийг
20. гарааны утга болгож өгнө.
21. \*/
22. EmployeeClass::EmployeeClass(EmployeeClass &emp){
23. /\*
24. worktime болон id нь хуулагдаж гарааны утга болох хэрэггүй гэж бодлоо
25. \*/
26. setAll(emp.getID(), emp.getName(), emp.getPosition(), emp.getWorkTime());
27. //Объект
28. totalEmployee++;
29. }
31. /\*
32. Параметртэй байгуулагч функц
33. Гараас бүх гишүүн өгөгдөлд гарааны утга авна
34. \*/
35. EmployeeClass::EmployeeClass(**int** n\_id, **char** \*n\_name, **char** \*n\_position, **float** n\_worktime){
36. setAll(n\_id, n\_name,n\_position, n\_worktime);
37. totalEmployee++;
38. }
40. /\*
41. Устгагч функц
42. ОБъект устах үед дуудагдах бөгөөд санах ойг чөлөөлж
43. ойн цоорхой үүсхээс хамгаалж байна
44. \*/
45. EmployeeClass::~EmployeeClass(){
46. cout<<"Object "<<getName()<<" ustlaa"<<endl;
47. **delete** [] name, position;
48. totalEmployee--;
49. }

52. /\*
53. Параметрт ирсэн бүх утга бүрийг тухайн гишүүн өгөгдлийн өгөгдөлд
54. оноох set функцээр дамжуулж утга онооно.
55. \*/
56. **void** EmployeeClass::setAll(**int** id, **char** \*name, **char** \*position,**float** workTime){
57. setID(id);
58. setName(name);
59. setPosition(position);
60. setWorkTime(workTime);
61. }
63. /\*
64. Цалин бодох функц
65. \*/
66. **float** EmployeeClass::calcSalary(){
67. **float** back =getWorkTime()\*2000;
68. **if**(strcmp(getPosition(),"ceo")==0){
69. back+=calcCeoaSalary();
70. }
71. **return** back;
72. }
74. **void** EmployeeClass::input(){
75. **int** new\_id;
76. cout<<"ID"<<endl;
77. cin>>new\_id;
78. **char** new\_name[20];
79. cout<<"Name"<<endl;
80. cin>>new\_name;
81. **char** new\_position[20];
82. cout<<"position"<<endl;
83. cin>>new\_position;
84. **int** workTime;
85. cout<<"time"<<endl;
86. cin>>workTime;
87. setAll(new\_id,new\_name,new\_position,workTime);
88. }
89. /\*
90. Захиралын цалин бодох функц
91. \*/
92. **float** EmployeeClass::calcCeoaSalary(){
93. **return** getWorkTime()\*5000;
94. }
96. /\*
97. Дуудсан объектын мэдээллийг харуулах функц
98. \*/
99. **void** EmployeeClass::display(){
100. cout<<"Ajilchnii dugaar: "<<getID()<<endl;
101. cout<<"Ajilchnii ner: "<<getName()<<endl;
102. cout<<"Ajilchnii alban tushaal: "<<getPosition()<<endl;
103. cout<<"Ajillasan tsag: "<<getWorkTime()<<endl;
104. }
106. /\*
107. Ажилласан цагийг нэмэх функц
108. \*/
109. **bool** EmployeeClass::incWorkingTime(**float** &addTime){
110. **if**(addTime<=24){
111. addTime+=getWorkTime();
112. setWorkTime(addTime);
113. **return** 1;
114. }**else** {
115. **return** 0;
116. }
117. }
118. // ID гишүүн өгөгдлийн утгыг буцаах функц
119. **int** EmployeeClass::getID(){
120. **return** id;
121. }
122. // ID гишүүн функцэд утга оноох гишүүн функц
123. **void** EmployeeClass::setID(**int** &n\_id){
124. id = n\_id;
125. }
126. // Name гишүүн өгөгдлийн утгыг буцаах функц
127. **char** \*EmployeeClass::getName(){
128. **return** name;
129. }
130. // Name гишүүн функцэд утга оноох гишүүн функц
131. **void** EmployeeClass::setName(**char** \*n\_name){
132. **if**(name){
133. **delete** [] name;
134. }
135. name = **new** **char**[strlen(n\_name)+1];
136. strcpy(name, n\_name);
137. }
138. // Position гишүүн өгөгдлийн утгыг буцаах функц
139. **char** \*EmployeeClass::getPosition(){
140. **return** position;
141. }
142. // Position гишүүн функцэд утга оноох гишүүн функц
143. **void** EmployeeClass::setPosition(**char** \*n\_pos){
144. **if**(position){
145. **delete** [] position;
146. }
147. position = **new** **char**[strlen(n\_pos)+1];
148. strcpy(name, n\_pos);
149. }
150. // workTime гишүүн өгөгдлийн утгыг буцаах функц
151. **char** EmployeeClass::getWorkTime(){
152. **return** workTime;
153. }
154. // setWorkTime гишүүн функцэд утга оноох гишүүн функц
155. **void** EmployeeClass::setWorkTime(**int** a){
156. workTime = a;
157. }
159. **void** EmployeeClass::copy(EmployeeClass \*emp){
161. setAll(emp->getID(), emp->getName(), emp->getPosition(), emp->getWorkTime());
163. }
164. //Нийт объектын тоог хадгалах статик хувьсагч
165. **int** EmployeeClass::totalEmployee=0;
167. // Нийт объектын тоог хадгалах  totalEmployee EmployeeClass
168. // статик гишүүн өгөгдлийн утгийг буцаах функц
169. **int** EmployeeClass::getTotalEmployee(){
170. **return** totalEmployee;
171. }

174. /\*
175. Объектуудын нэрийг өсөх дарааллаар эрэмбэлнэ
176. Сонголтын арга ашигласан
177. \*/
178. **void** sortName(EmployeeClass \*\*employees, **int** totalEmployee){
179. EmployeeClass \*key;
180. **for**(**int** i=0;i<totalEmployee;i++){
181. **for**(**int** j=i+1;j<totalEmployee;j++){
182. **if**(strcmp(employees[i]->getName(),employees[j]->getName())>0){
183. key = employees[j];
184. employees[j] = employees[i];
185. employees[i] = key;
186. }
187. }
188. }
189. }
191. /\*
192. Ажилтны дугаар өмнө нь өөр ажилтанд бүртгэгдсэн эсэхийг шалгана
193. \*/
194. **bool** EmployeeClass:: checkID(**int** id,EmployeeClass \*\*employees,**int** totalEmployee){
195. **for**(**int** i=0;i<totalEmployee;i++){
196. **if**(id==employees[i]->getID()){
197. **return** **false**;
198. }
199. }
200. **return** **true**;
201. }
203. //Илгээсэн объектон массивыг цалингаар өсөх эрэмбээр эрэмбэлэх
204. //функц employees[] - объектон массивsetName
205. //totalEmployee - нийт ажилчдын тооemp
206. **void** sort(EmployeeClass \*\*employees, **int** totalEmployee){
208. EmployeeClass \*key;
209. **for**(**int** i=0;i<totalEmployee;i++){
210. **for**(**int** j=i+1;j<totalEmployee;j++){
211. **if**(employees[i]->calcSalary()>employees[j]->calcSalary()){
212. key = employees[j];
213. employees[j] = employees[i];
214. employees[i] = key;
215. }
216. }
217. }
218. }
219. //    int i, j;
220. //    EmployeeClass \*key;
221. //    for (i = 1; i < totalEmployee; i++)
222. //    {
223. //        key = employees[i];
224. //        j = i-1;
226. //        while (j >= 0 && employees[j]->calcSalary() > key.calcSalary())
227. //        {
228. //            employees[j+1]=employees[j];
229. //            j -= 1;
230. //        }
232. //        employees[j+1]=key;
233. //    }
234. // }
236. /\*
237. Ажилчны айдагаар массивын хэддүгээр индекст байгааг
238. олж буцаах функц employees[] - объектон массив
239. totalEmployee - нийт ажилчдын тоо
240. id - хайж буй ажилчны код
241. \*/
242. **int** findById(EmployeeClass employees[],**int** id, **int** totalEmployee){
243. **for**(**int** i=0;i<totalEmployee;i++){
244. **if**(employees[i].getID()==id){
245. **return** i;
246. }
247. }
248. **return** -1;
249. }

7.4 Хавсралтын кодыг terminal(CMD) дээр туршиж үзэх код

1. g++ -c employeeDefinition.cpp –o employeeDefinition.o && g++ employeeMain.cpp employeeDefinition.o –o main.exe && ./main.exe

## 7.5 Лаботорын ажлын даалгавар

Асуулт:

1. Байгуулагч функц хэзээ дуудагддаг вэ?

2. Устгагч функц хэзээ дуудагдах вэ?

3. Хуулагч байгуулагч гэж юу вэ? Ач холбогдол нь юу вэ?

4. Хуулагч функц гэж юу вэ? Ач холбогдол нь юу вэ? Санах ойн цоорхойгоос хэрхэн сэргийлэх вэ?

5. Объектын хаяган хувьсагчийг хэрхэн зарлах вэ? new оператороор санах ой нөөцлөх, хаяган хувьсагчаар дамжуулж объектын гишүүн өгөгдөл, гишүүн функцэд яаж хандах вэ?

6. Объектын хаяган хувьсагчийн хүснэгт үүсгэх жишээ код бичиж ажиллуул.

Бодлого:

Лаб04 - д тодорхойлсон классын нэр:char[20], албан тушаал:char[10] гэсэн гишүүн өгөгдлийг хаяган хувьсагч болгон өөрчилж гараас өгсөн тэмдэгтийн цуваатай яг ижил урттай санах ой new оператор ашиглан нөөцөлдөг болго. Үүний тулд анхдагч болон параметертэй байгуулагч функцууд тодорхойлж гишүүн өгөгдөлд гарааны утга онооно. Мөн объект устах үед дээрх хоёр гишүүн өгөгдөлд нөөцөлсөн санайх ойг чөлөөлдөг болгож өөрчил.

Өөрчлөлтийг оруулаад дараах даалгаврыг гүйцэтгэ

1. Олон ажилчин бүртгэж ажилчдыг нэрээр нь эрэмбэлэх. Нэрээр эрэмбэлэхдээ объектын хаяган хүснэгт үүсгээд хаягийг нь эрэмбэлбэлнэ.

2. Шинэ ажилчин бүртгэхдээ тухайн ажилчны ID өмнө бүртгэлтэй эсэхийг шалгаж бүртгэлтэй бол бүртгэхгүй. Өөрөөр хэлбэл ID дахин давхцахгүй болго.

Байгуулагч функц болон тэмдэгтийн цуваанд динамик ой нөөцлөх жишээ:

1. #include < iostream >
2. #include < string.h >
3. using namespace std;
4. class student {
5. public: int id;
6. int avg;
7. char \* name;
8. student();
9. student(char \* n, int a, int b);
10. ~student();
11. };
12. student::student() {
13. name = new char[5];
14. strcpy(name, "test");
15. }
16. student::student(char \* n, int a, int b) {
17. id = a;
18. avg = b;
19. name = new char[strlen(n) + 1];
20. strcpy(name, n);
21. }
22. student::~student() {
23. cout << "The object " << name << " is deleted;" << endl;
24. delete name;
25. }
26. int main() {
27. student dorj; {
28. student dulam("dulam", 1234, 4);
29. cout << dulam.name << endl;
30. }
31. cout << dorj.name << endl;
32. }