(Лаборатори №5)

Э.Багабанди

ХШУИС - ийн ПХ хөтөлбөрийн

3-р түвшиний оюутан, 19B1NUM0700@num.edu.mn

1. **ОРШИЛ/УДИРТГАЛ**

Уг лабораторийн хүрээнд өгөгдсөн бодлогуудыг тайлбарлахад ашиглагдсан C++ хэлний онолын ойлголт /байгуулагч функц, устгагч функцын хэрэгжүүлэлт, хуулагч функц, объектын хаяган хувьсагч/ -уудыг судалж, эзэмшин хэрэгжүүлсэн болно.

2. **ЗОРИЛГО**

Энэхүү лабораторийн ажлын хүрээнд C++ хэлний байгуулагч функц, устгагч функцын хэрэгжүүлэлт, түүний шинжлэх, хуулагч функц, түүний ач холбогдол болон хэрэгжүүлэлт, обьектын хаяган хувьсагч гэх онолын ойлголтуудыг судалж, үүнийг ашиглан лабораторт өгөгдсөн бодлого шийдвэрлэж, асуултуудад хариулна.

3. **ОНОЛЫН СУДАЛГАА**

3.1 Байгуулагч функц хэзээ дуудагддаг вэ?

- **Байгуулагч функц**
  - Байгуулагч функцийг объект үүсгэх үед C++ компайлер автоматаар дуудах бөгөөд нэг объект байгуулах үед нэг л удаа дуудагдана.

3.2 Устгагч функц хэзээ дуудагдах вэ?

- **Устгагч функц**
  - Устгагч функцийг ямар нэг объект дахин ашиглагдахгүй болж, устах үед зөвхөн C++ компайлераар дуудагдана.

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

- **Хуулагч функц, түүний ач холбогдол**
  - Хуулагч байгуулагч бол шинэ объект нь түүний өмнөх аль нэгэн объектын сүүлийн утгыг хуулбарлаж авах боломжийг олгодог бөгөөд тухайн үйлдлээр байгуулагдах шинэ объект ба утга нь хуулбарлагдах объект хоорондоо нэрээрээ ялагаатай. Иймд хуулагч байгуулагч нь объект хуулбарлах зориулалтай функц гэж үзэж болох бөгөөд ийм байгуулагч нь нэг параметр авах ба тэр нь класс нэртэй хуулагдах объектын заалт байна. Хуулагч байгуулагч функц нь аргумент болох объект руу заалтаар нь хандаж түүний хуулбар объектыг нэг л удаа үүсгэнэ.

3.4 Хуулагч функц санах ойн цоорхойгоос хэрхэн сэргийлэх вэ?

- **Хуулагч функц санах ойн цоорхой үүсэх нь**
  - Хуулагч функц ашиглан шинэ объект үүсгэх үед тухайн объектын классд ямар нэгэн заагч хувьсагч зарласан бол тухайн объектын заагч төрлийн өгөгдлийг хуулагч функц дотор устгаж өгөх шаардлагатай болдог. Учир нь тухайн объект аль хэдийн байгуулагдчихсан ба түүний заагч төрлийн өгөгдөл нь санах ой нөөцөлж үүссэн байна. Иймд түүнийг устгаж өгөх шаардлагатай.

```
class Employee {
private:
    int empid; // Ажилтаны дугаарыг private төлөвтэйгээр зарлав.
    char *name; //Ажилтны нэрийг заагч төрлөөр, ptivate төлөвтэйгээр зарлав.
    char *position;
    float wtime; //Ажилтны ажилласан цагийг зарлав.
    float ceoCalcSalary();
public:
    Employee();
    Employee(char *n, char *p, int e, float wt);
    Employee(Employee &); //Хуулагч байгуулагч зарлав.
    void copy(Employee &); //Хуулагч функц зарлав.
    void setData(); //гишүүн өгөгдөлд гаднаас хандаж, өөрчлөлт оруулах функц
    void getData(); //гишүүн өгөгдлийн утгыг гаднаас хандаж, авах функц
    float calcSalary(); //Цалин бодох функцийг зарлав.
    int incWorkTime(float inc);
};
Employee::Employee(Employee &e){
    name = new char(strlen(e.name) + 1);
    strcpy(name, e.name);
    position = new char(strlen(e.position) + 1);
    strcpy(position, e.position);
    empid = e.empid;
    wtime = e.wtime;
}
void Employee::copy(Employee &e){
    name = new char(strlen(e.name) + 1);
    strcpy(name, e.name);
    position = new char(strlen(e.position) + 1);
    strcpy(position, e.position);
    empid = e.empid;
    wtime = e.wtime;
}
```

- **Шийдэл болон сэргийлэлт**

- Хуулагч функц дотор хуулах гэж буй объектын заагч өгөгдөлд санах ой нөцөөлөхийн өмнө хуулагч функцийг дуудсан буюу хуулж буй обьектын өмнө нөөцлөгдсөн санах ойг delete оператор ашиглан чөлөөлнө.

```
void Employee::copy(Employee &e){
    delete name;
    delete position;
    name = new char(strlen(e.name) + 1);
    strcpy(name, e.name);
    position = new char(strlen(e.position) + 1);
    strcpy(position, e.position);
    empid = e.empid;
    wtime = e.wtime;
}
```

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

- **Объектын хаяган хувьсагчийг хэрхэн зарлах**
  - Объектын хаяган хувьсагчийг тухайн объектын классын нэрний араас заагч хувьсагч бичих байдлаар зарлана.

```
class_name *Object_pointer_name;
```

- **new операторoор санах ой нөөцлөх, хаяган хувьсагчаар дамжуулж объектын гишүүн өгөгдөл, гишүүн функцэд яаж хандах**
  - Зарласан объект төрлийн хаяган хувьсагчид new операторoор санах ой нөөцлөхдөө new араас классын байгуулагч функцыг дуудна. Харин тухайн объектын хаягаар дамжуулж, түүний гишүүн өгөгдөл болон гишүүн функцэд хандахдаа объект төрлийн хаяган хувьсагчийн өмнө **/*/** тавих бөгөөд ингэснээр тухайн хаяг дээр объектэд шууд **/./** – ээр хандах боломжтой болно. Харин хаяган хувьсагчийн өмнө ямар нэг оператор ашиглахгүйгээр хандах үед **/->/** буюу суман оператор ашиглан тухайн хаяг дах объектын гишүүн өгөгдөл болон функцүүдэд хадгалах боломжтой болно.
  - **New** оператор ашиглан тухайн объектэд санах ой нөөцлөж байгаа учир тухайн обьектыг ашиглагдахгүй болох үед нь **delete** операторoос түүний эзэмшиж байсан санах ойг чөлөөлөх шаардлагатай. Ингэснээр санах ойн цоорхойноос урьдчилан сэргийлэх бөгөөд санах ойг үр ашигтай ашиглаж буй хэлбэр юм.

```
class_name *Object_pointer_name;
Object_pointer_name = new class_name(parameter_list);
```

```
(*Object_pointer_name).function_name(parameter_list);

(*Object_pointer_name).var_name; // var_name нь public гишүүн өгөгдөл байх үед боломжтой.
```

```
Object_pointer_name->function_name(parameter_list);

Object_pointer_name->var_name; // var_name нь public гишүүн өгөгдөл байх үед бо
ломжтой.
```

```
delete example;
```

3.6 Объектын хаяган хувьсагчийн хүснэгт үүсгэх жишээ код бичиж ажиллуул.

- **Объектын хаяган хувьсагчийн хүснэгт**
  - Жишээ код.

```
    int n;

    cout<<"Enter count of employee: ";

    cin>>n;

    Employee *Employees[n]; //n хэмжээтэй Employee төрлийн хаяган хувьсагчийн хүснэ
гт зарлав

    Employee *EmpPtr; //Employee төрлийн хаяган хувьсагч зарлав

    cout<<"Enter data of each employee: "<<endl;

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

    {

        cout<<"\tEmployee No"<<i + 1<<":\n";

        EmpPtr = new Employee; //Employee төрлийн объектын хаягийг нөөцөлж, нөөцөлс
өн хаягийг EmpPtr хадгалав

        EmpPtr->setData(); // EmpPtr хаяг дээр байх объектэд оруулав

        Employees[i] = EmpPtr; // EmpPtr хаягийг Employees хаяган хувьсагчийн хүснэ
гтийн I индекс дээр хадгалав

    }
```

4. **ХЭРЭГЖҮҮЛЭЛТ**

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.

```
class Employee {
private:
    int empid;
    char *name;
    char *position;
    float wtime;
    float ceoCalcSalary();
public:
    Employee(); //Андагч байгуулагч функц зарлав.
    Employee(char *n, char *p, int e, float wt);
    ~Employee(); // Устгагч функц зарлав
    void setData(); //гишүүн өгөгдөлд гаднаас хандаж, өөрчлөлт оруулах функц
    void getData(); //гишүүн өгөгдлийн утгыг гаднаас хандаж, авах функц
    float calcSalary(); //Цалин бодох функцийг зарлав.
    int incWorkTime(float inc); //Ажилласан цагийг нэмэгдүүлэх функцийг зарлав.
};
Employee::Employee(){
    empid = 0; //empid - д анхны утга оноов.
    name = new char(1); // name хувьсагчид char төрлийн 1 байт ой нөөцлөв.
    strcpy(name, ""); // 1 byte урттай нөөцөлсөн ойд хоосон буюу null утга оноов.
    position = new char(9);
    strcpy(position, "Employee");
    wtime = 0;
}
Employee::Employee(char* n, char* p, int e, float wt){
    name = new char(strlen(n) + 1);
    strcpy(name, n); // Нөөцөлсөн ойд n параметрийн утгыг олгов.
    position = new char(strlen(p) + 1);
    strcpy(position, p);
    empid = e;
    wtime = wt;
}
Employee::~Employee() //устгагч функцийг тодорхойлов.{
    cout<<"\n deleted " << name <<endl;
    delete name;
    delete position;
}
```

Өөрчлөлтийг оруулаад дараах даалгаврыг гүйцэтгэ

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

```
    int n;
    cout<<"Enter count of employee: ";
    cin>>n;
    Employee *emps[n]; //n хэмжээтэй Employee төрлийн хаяган хувьсагчийн хү
снэгт зарлав
    Employee *EmpPtr; //Employee төрлийн хаяган хувьсагч зарлав
    cout<<"Enter data of each employee: "<<endl;
    ...

    for(int i = 1; i < n; i++)
    {
        Employee *key = emps[i];
        char *name = new char(strlen(emps[i]->getName()) + 1);
        strcpy(name, emps[i]->getName());
        int j = i - 1;
        while(j >= 0 && strcmp(name, emps[j]->getName()) < 0)
        {
            emps[j + 1]->copy(*emps[j])
            j--;
        }

        emps[j + 1]->copy(*key);
        delete name;
    }
```

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

```
int checkId(int emps[], int n, int empid){
    for(int i = 0; i < n; i++){
        if(emps[i] == empid) return 0;
    }
    return 1;
}
int main()
{
    int n;
    cout<<"Enter count of employee: ";
    cin>>n;
    Employee *Employees[n];
    Employee *EmpPtr;
    int empsid[n] = {0};
    cout<<"Enter data of each employee: "<<endl;
    for(int i = 0; i < n; i++){
        cout<<"\tEmployee No"<<i + 1<<":\n";
        EmpPtr = new Employee();
        char name[20], pos[20];
        int empid, wt;
        cout<<"Enter empid: "<<endl;
        cin>>empid;
        int avail = checkId(empsid, n, empid);
        while(avail == 0){
            cout<<"Enter empid again /repeated/: "<<endl;
            cin>>empid;
            avail = checkId(empsid, n, empid);
        }
        EmpPtr->setEmpid(empid);
        empsid[i] = empid;
        ...
    }
```

5. ДҮГНЭЛТ

Энэхүү лабораторын ажлын асуултуудын хүрээнд Employee буюу ажилтан гэх классыг илүү сайжруулж, түүнийг ашигласан жижиг хэмжээний програм хөгжүүлсэн. Тухайн классын сайжруулахдаа классын нэр:char[20], албан тушаал:char[10] гэсэн гишүүн өгөгдлийг хаяган хувьсагч болгон өөрчилж, гараас өгсөн тэмдэгтийн цуваатай яг ижил урттай санах ойг new оператор ашиглан нөөцөлдөг болгосон. Энэ нь санах ойг үр ашигтай ашиглаж байгаа хэрэг бөгөөд гараас ямар ч хэмжээтэй тэмдэгтийн цуваag оруулах боломжийг олгох юм. Мөн жижиг хэмжээний эрэмбэлэлтийн програм хөгжүүлэх дээ объектын хаяган хүснэгт үүсгэж, хаягийг ашигласан эрэмбэлсэн. Учир нь объектын хаяган хүснэгт ашигласнаар тухайн объектуудын зөвхөн хаягийг ашиглан тэдгээрийг боловсруулах боломжийг олгож байна.

6. АШИГЛАСАН МАТЕРИАЛ

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

7. ХАВСРАЛ

```
1.  #include <iostream>
2.  #include <string.h>
3.  #define pricePerHour 5000
4.  using namespace std;
5.
6.
7.  class Employee {
8.  private:
9.      int empid;
10.     char *name;
11.     char *position;
12.     float wtime;
13.     float ceoCalcSalary();
14. public:
15.     Employee(); //Андагч байгуулагч функц зарлав.
16.     Employee(char *n, char *p, int e, float wt); //Параметртэй байгуулагч функц зарлав.
17.     ~Employee(); // Устгагч функц зарлав
18.     Employee(Employee &); //Хуулагч байгуулагч зарлав.
19.     void empCopy(Employee &); //Хуулагч функц зарлав.
20.     void setEmpid(int);
21.     void setPosition(char*);
22.     void setWtime(int);
23.     void setName(char*);
24.     char* getName(); //гишүүн өгөгдлийн утгыг гаднаас хандаж, авах функц
25.     char* getPosition();
26.     int getEmpid();
27.     int getWtime();
28.     float calcSalary(); //Цалин бодох функцийг зарлав.
29.     int incWorkTime(float inc); //Ажилласан цагийг нэмэгдүүлэх функцийг зарлав.
30. };
31.
32. void Employee::setName(char* name) // set фунцкцийг тодорхойлов.
33. {
34.     this->name = name;
35. }
36.
```

```
37. void Employee::setPosition(char* pos) // set фунцкцийг тодорхойлов.
38. {
39.     position = pos;
40. }
41.
42. void Employee::setWtime(int wt) // set фунцкцийг тодорхойлов.
43. {
44.     wtime = wt;
45. }
46.
47. void Employee::setEmpid(int empid) // set фунцкцийг тодорхойлов.
48. {
49.     this->empid = empid;
50. }
51.
52. int Employee::getEmpid() // get функцийг тодорхойлов.
53. {
54.     return empid;
55. }
56. char* Employee::getName() // get функцийг тодорхойлов.
57. {
58.     return name;
59. }
60. char* Employee::getPosition() // get функцийг тодорхойлов.
61. {
62.     return position;
63. }
64. int Employee::getWtime() // get функцийг тодорхойлов.
65. {
66.     return wtime;
67. }
68.
69. float Employee::calcSalary() //цалин бодох функцийг тодорхойлов.
70. {
71.     if(strcmp(position, "CEO") == 0){
72.         return wtime*pricePerHour + ceoCalcSalary();
73.     }
74.     return wtime*pricePerHour;
75. }
76.
77. float Employee::ceoCalcSalary() // захирлын цалин бодох функцийг тодорхойлов.
78. {
79.     return wtime*pricePerHour;
80. }
81.
82. Employee::Employee()
83. {
84.     empid = 0; //empid - д анхны утга оноов.
85.     name = new char(1); // name хувьсагчид char төрлийн 1 байт ой нөөцлөв.
86.     strcpy(name, ""); // 1 byte урттай нөөцөлсөн ойд хоосон буюу null утга оноов.
87.
88.     position = new char(9);
```

```
89.     strcpy(position, "Employee");
90.     wtime = 0;
91. }
92.
93. Employee::Employee(char* n, char* p, int e, float wt)
94. {
95.     name = new char(strlen(n) + 1); // name хувьсагчид n урт дээр нэмэх 1 ойг нөөцлөв.
96.     strcpy(name, n); // Нөөцөлсөн ойд n параметрийн утгыг олгов.
97.     position = new char(strlen(p) + 1);
98.     strcpy(position, p);
99.     empid = e;
100.        wtime = wt;
101.    }
102.
103.    Employee::Employee(Employee &e)
104.    {
105.        name = new char(strlen(e.name) + 1);
106.        strcpy(name, e.name);
107.        position = new char(strlen(e.position) + 1);
108.        strcpy(position, e.position);
109.        empid = e.empid;
110.        wtime = e.wtime;
111.    }
112.
113.    void Employee::empCopy(Employee &e)
114.    {
115.        delete name;
116.        delete position;
117.        name = new char(strlen(e.name) + 1);
118.        strcpy(name, e.name);
119.        position = new char(strlen(e.position) + 1);
120.        strcpy(position, e.position);
121.        empid = e.empid;
122.        wtime = e.wtime;
123.    }
124.
125.    Employee::~Employee() //устгагч функцийг тодорхойлов.
126.    {
127.        cout<<"\n deleted " << name <<endl;
128.        delete name;
129.        delete position;
130.    }
131.
132.    int Employee::incWorkTime(float inc) //ажилласан цаг нэмэгдүүлэх функцийг
    тодорхойлов.
133.    {
134.
135.        if(0 < inc && inc <= 24){
136.            wtime += inc;
137.            return 1;
138.        }
139.
```

```
    return 0;
}

int checkId(int emps[], int n, int empid){
    for(int i = 0; i < n; i++){
        if(emps[i] == empid) return 0;
    }

    return 1;
}

int main()
{
    int n;
    cout<<"Enter count of employee: ";
    cin>>n;

    Employee *Employees[n];
    Employee *EmpPtr;
    int empsid[n] = {0};
    cout<<"Enter data of each employee: "<<endl;
    for(int i = 0; i < n; i++)
    {
        cout<<"\tEmployee No"<<i + 1<<":\n";
        EmpPtr = new Employee();
        char name[20], pos[20];
        int empid, wt;

        cout<<"Enter empid: "<<endl;
        cin>>empid;

        int avail = checkId(empsid, n, empid);
        while(avail == 0){
            cout<<"Enter empid again /repeated/: "<<endl;
            cin>>empid;
            avail = checkId(empsid, n, empid);
        }

        EmpPtr->setEmpid(empid);

        empsid[i] = empid;

        cout<<"Enter name: "<<endl;
        cin>>name;
        EmpPtr->setName(name);

        cout<<"Enter position: "<<endl;
        cin>>pos;
        EmpPtr->setPosition(pos);

        cout<<"Enter wtime: "<<endl;
        cin>>wt;
```

```
192.          EmpPtr->setWtime(wt);
193.          Employees[i] = EmpPtr;
194.       }
195.
196.
197.       for(int i = 1; i < n; i++)
198.       {
199.          Employee *key = Employees[i];
200.          char *key_name = new char(strlen(Employees[i]->getName()) + 1);
201.          strcpy(key_name, Employees[i]->getName());
202.          int j = i - 1;
203.          char *j_name = new char(strlen(Employees[j]->getName()) + 1);
204.          strcpy(j_name, Employees[j]->getName());
205.          while(j >= 0 && strcmp(key_name, j_name) < 0)
206.          {
207.             Employees[j + 1]->empCopy(*Employees[j]);
208.             j--;
209.          }
210.          Employees[j + 1]->empCopy(*key);
211.          delete key_name;
212.          delete j_name;
213.       }
214.
215.       cout<<"\n\nEmloyees were sorted by name: "<<endl;
216.       for(int i = 0; i < n; i++)
217.       {
218.          cout<<endl;
219.          cout<<Employees[i]->getEmpid()<<endl;
220.          cout<<Employees[i]->getName()<<endl;
221.          cout<<Employees[i]->getPosition()<<endl;
222.          cout<<Employees[i]->getWtime()<<endl;
223.       }
224.
225.
226.       for(int i = 0; i < n; i++)
227.       {
228.          delete Employees[i];
229.       }
230.
231.       return 0;
232.    }
```