

Класс Дотор Санах Ойг Динамикаар Үүсгэж Өгөгдлийг Хадгалах Лабораторийн Ажил (Лаборатори №5)

1. ОРШИЛ/УДИРТГАЛ

Амьдралд байгаа объектыг компьютерийн програмд загварчлахад классыг тодорхойлж, объект үүсгэн ажилчдыг бүртгэх, цаг бүртгэх, цалин тооцох, нэрээр нь эрэмбэлэх зэрэг үйлдлийг хийнэ. Анхны утгыг байгуулагч функцээр дамжуулж олгоно. Тэмдэгт мөрийг хадгалахдаа динамикаар санах ойг нөөцөлж хадгална.

2. ЗОРИЛГО

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

1. Гишүүн өгөгдлүүдийн төрлийг тодорхойлох,
2. Гишүүн функц тодорхойлохдоо тэмдэгт мөр хадгалах гэж байгаа бол санах ойг нөөцлөх,
3. Байгуулагч функц тодорхойлох,
4. Устгагч функц тодорхойлох,
5. Ажилчин классын хүснэгт үүсгэж ажилчдыг бүртгэж боловсруулалт хийх.

3. ОНОЛЫН СУДАЛГАА

3.1 Байгуулагч функц

Байгуулагч функц нь объектын гишүүн өгөгдөлд дамжуулсан утга эсвэл анхны утгыг өгөх зориулалттай. Энэхүү функцийн нэр нь классын нэртэй ижил байна. Объект үүсэх үед compiler байгуулагч функцийг автоматаар дуудна. Байгуулагч функцийн буцаах утгын төрлийг тодорхойлдоггүй. Хэрэв классд байгуулагч функцийг тодорхойлоогүй бол compiler автоматаар хоосон байгуулагчийг үүсгэнэ.

Мөн класст нэгээс олон байгуулагч функц тодорхойлж өгч болно. Энэ нь дахин тодорхойлогдож байна гэсэн үг болно. Хэд хэдэн байгуулагч функцтай класс зарлах үед тэдгээр байгуулагч функцууд нь хоорондоо параметрийнхээ төрлөөр ялгагдаж байх ёстой. Ингээгүй тохиолдолд compiler байгуулагч функцуудын алийг нь дуудахаа тодорхойлж чадахгүй байдалд хүрч алдаа өгнө.

```
классын_нэр() {  
...  
}  
классын_нэр(төрөл параметрын_нэр, ...) {  
...  
}
```

3.2 Устгагч функц

Устгагч функц нь объектын динамик өгөгдлийн санах ойг чөлөөлөхөд хэрэглэгдэнэ. Класс ганц устгагч функцтэй байна. Хэрэв класст устгагч функц байхгүй бол compiler объектийн өгөгдлийн нөөцөлсөн санах ойг автоматаар чөлөөлнө. Устгагч функцийг дуудаж ажиллуулдаггүй бөгөөд үйлчлэх хүрээ дуусах үед compiler автоматаар дуудаж ажиллуулна. Утга буцаахгүй бөгөөд параметр авахгүй.

```
~классын_нэр() {  
...  
}
```

3.3 Хуулагч байгуулагч

Хуулагч байгуулагч функцийг тусламжтайгаар аль хэдийн үүссэн объектын гишүүн өгөгдлүүдтэй ижил гишүүн өгөгдөлтэй объектыг байгуулна.

```
классын_нэр (классын_нэр &объектын_нэр) {  
...  
}
```

```
объектын_нэр = объектын_нэр;
```

Дээрх үйлдлийн тусламжтайгаар эхний объектод дараагийн объектын бүх гишүүн өгөгдлийг хуулж тавина. Гэхдээ класс дотор хаяган хувьсагч байгаа тохиолдолд дээрх үйлдэл тохиромжгүй юм. Учир нь нэг санах ойд хоёр хувьсагч хоёул хандаж эхэлнэ. Мөн санах ойн цоорхойг үүсгэнэ. Санах ойн цоорхойгоос сэргийлэхийн тулд динамикаар нөөцөлсөн санах ойг хэрэглэж дууссаны дараа устгах хэрэгтэй.

4. ХЭРЭГЖҮҮЛЭЛТ

Хамгийн түрүүнд классыг тодорхойлсон

```
class Ajilchin {  
private:  
    int dugaar;  
    float ajillasan_tsag;  
    char* ner;  
    char* alban_tushaal;  
    float zahirliin_tsalin_bodoh();  
public:  
    Ajilchin();  
    ~Ajilchin();  
    Ajilchin(Ajilchin&);  
    Ajilchin(int,char*,char*,float);  
    void set_dugaar(int);  
    void set_ner(char*);  
    void set_alban_tushaal(char*);  
    void set_ajillasan_tsag(float);  
    bool ajillasan_tsag_nemegduuleh(float);  
    int get_dugaar();
```

```

char* get_ner();
char* get_alban_tushaal();
float get_ajillasan_tsag();
float tsalin_bodoh();
};

```

Түүний дараа гишүүн функцуудыг тодорхойлсон.

Тэгээд нэрээр эрэмбэлдэг болон дугаар давхацсан эсэхийг шалгадаг хэрэглэгчийн функцийг тодорхойлсон.

```

int dugaar_davhtssan_eseh(Ajilchin a[], int n, int id) {
    int k=0;
    for(int i=0; i<n; i++) {
        if(a[i].get_dugaar() == id) {
            k++;
        }
    }
    if(k==0)
        return 0;
    return 1;
}

void nereer_erembeleh(Ajilchin a[], int n) {
    int sav1, sav2;
    char sav3[20], sav4[20];
    float sav5, sav6;
    for(int j=0; j<=n; j++) {
        for(int i=0; i<n; i++) {
            if(strcmp(a[i].get_ner(), a[i+1].get_ner())>0) {
                sav1 = a[i].get_dugaar();
                sav2 = a[i+1].get_dugaar();
                a[i].set_dugaar(sav2);
                a[i+1].set_dugaar(sav1);

                strcpy(sav3, a[i].get_ner());
                strcpy(sav4, a[i+1].get_ner());
                a[i].set_ner(sav4);
                a[i+1].set_ner(sav3);

                strcpy(sav3, a[i].get_alban_tushaal());
                strcpy(sav4, a[i+1].get_alban_tushaal());
                a[i].set_alban_tushaal(sav4);
                a[i+1].set_alban_tushaal(sav3);

                sav5 = a[i].get_ajillasan_tsag();
                sav6 = a[i+1].get_ajillasan_tsag();
                a[i].set_ajillasan_tsag(sav6);
                a[i+1].set_ajillasan_tsag(sav5);
            }
        }
    }
}

```

Main функц дотор уг классын объектон хүснэгт үүсгэсэн бөгөөд Main функц нь давталтаар зохион байгуулагдсан бөгөөд гарах нөхцлийг хангах хүртэл програм ажиллаж ажилчин нэмэх, мэдээллийг оруулах, мэдээллийг харах, нэрсээр нь эрэмбэлэх үйлдлүүдийг хийнэ.

```
main() {
    int m=0, n=-1;
    int a;
    char b[20], c[20];
    float d;
    Ajilchin k[30];
    while(m != 6) {
        cout << "1. Ajiltan nemeh\n";
        cout << "2. Ajiltnii medeelliig oruulah\n";
        cout << "3. Ajilchdiin medeelliig delgetsleh\n";
        cout << "4. Ajilchdiig nereer ni erembeleh\n";
        cout << "5. Ajillasan tsag nemegduuleh\n";
        cout << "6. Duusgah\n";
        cin >> m;
        if(m==1) {
            n++;
        }
        if(m==2) {
            cout << "dugaar oruul: ";
            cin >> a;
            if(dugaar_davhtssan_eseh(k, n, a)==0) {
                k[n].set_dugaar(a);
                cout << "ner oruul: ";
                cin >> b;
                k[n].set_ner(b);
                cout << "alban tushaal oruul: ";
                cin >> c;
                k[n].set_alban_tushaal(c);
            } else {
                cout << "Iim id dugaartai ajiltan burtgeltei baina.\n";
            }
        }
        if(m==3) {
            for(int i=0; i<=n; i++) {
                cout << k[i].get_dugaar() << " dugaartai ajiltan ";
                cout << k[i].get_ner() << " ni ";
                cout << k[i].get_alban_tushaal() << " alban tushaaltai buguud ";
                cout << k[i].get_ajillasan_tsag() << " tsag ajillasan ba ";
                cout << k[i].tsalin_bodoh() << " tsalintai baina.\n";
            }
        }
        if(m==4) {
            nereer_erembeleh(k, n);
        }
        if(m==5) {
            int h=0;
            cout << "Tsagiig ni nemeh ajiltnii dugaariig oruul: ";
            cin >> a;
```

```

        for(int i=0; i<=n; i++) {
            if(k[i].get_dugaar()==a) {
                h++;
            }
        }
        for(int i=0; i<=n; i++) {
            if(k[i].get_dugaar()==a) {
                cout << "Nemeh tsagaa oruul: ";
                cin >> d;
                if(k[i].ajillasan_tsag_nemegduuleh(d)==1) {
                    cout << "Ajiltnii tsag amjilttai nemegdlee.\n";
                } else {
                    cout << "Buruu tsag oruulsan baina.\n";
                }
            }
        }
        if(h==0) {
            cout << "Tanii oruulsan dugaartai ajiltan baihgui baina\n";
        }
    }
}

```

Ажилтны цалинг ажилласан цаг дээр үндэслэн бодож ажилчдын мэдээллийг дэлгэцлэхэд харагдана. Захиралын цалинг цагийн 10000, бусад ажилчдын цалинг цагийн 5000 гэж үзсэн.

```

float Ajilchin::tsalin_bodoh() {
    if(strcmp(alban_tushaal, "zahirai")==0) {
        return zahirliin_tsalin_bodoh();
    }
    int tsagiin_huls = 5000;
    return tsagiin_huls * ajillasan_tsag;
}

float Ajilchin::zahirliin_tsalin_bodoh() {
    int tsagiin_huls = 10000;
    return tsagiin_huls * ajillasan_tsag;
}

```

Main функц дуусах үед объектын устгагч функц автоматаар дуудагдаж нөөцөлсөн санах ойг устгаад устлаа гэдэг үгийг дэлгэцлэнэ.

5. ДҮГНЭЛТ

Байгуулагч функцийг тодорхойлж өгснөөр классын объект үүсэх үед гарааны утгыг шууд оноох боломжтой болгож байна. Динамик байдлаар санах ойг нөөцөлж ашиглах нь санах ойн замбараагүй хэрэглээг багасгах сайн талтай хэдий ч хэрэв нөөцөлсөн санах ойгоо устгахаа мартвал санах ойн цоорхой үүсэх эрсдэлтэй.

6. АШИГЛАСАН МАТЕРИАЛ

1. <https://www.geeksforgeeks.org/>

7. ХАВСРАЛТ

```
#include <iostream>
#include <string.h>
using namespace std;

class Ajilchin {
private:
    int dugaar;
    float ajillasan_tsag;
    char* ner;
    char* alban_tushaal;
    float zahirliin_tsalin_bodoh();
public:
    Ajilchin();
    ~Ajilchin();
    Ajilchin(Ajilchin&);
    Ajilchin(int, char*, char*, float);
    void set_dugaar(int);
    void set_ner(char*);
    void set_alban_tushaal(char*);
    void set_ajillasan_tsag(float);
    bool ajillasan_tsag_nemegduuleh(float);
    int get_dugaar();
    char* get_ner();
    char* get_alban_tushaal();
    float get_ajillasan_tsag();
    float tsalin_bodoh();
};

int dugaar_davhtssan_eseh(Ajilchin a[], int n, int id) {
    int k=0;
    for(int i=0; i<n; i++) {
        if(a[i].get_dugaar() == id) {
            k++;
        }
    }
    if(k==0)
        return 0;
    return 1;
}

void nereer_erembeleh(Ajilchin a[], int n) {
    int sav1, sav2;
    char sav3[20], sav4[20];
    float sav5, sav6;
    for(int j=0; j<=n; j++) {
        for(int i=0; i<n; i++) {
            if(strcmp(a[i].get_ner(), a[i+1].get_ner())>0) {
                sav1 = a[i].get_dugaar();
            }
        }
    }
}
```

```

        sav2 = a[i+1].get_dugaar();
        a[i].set_dugaar(sav2);
        a[i+1].set_dugaar(sav1);

        strcpy(sav3, a[i].get_ner());
        strcpy(sav4, a[i+1].get_ner());
        a[i].set_ner(sav4);
        a[i+1].set_ner(sav3);

        strcpy(sav3, a[i].get_alban_tushaal());
        strcpy(sav4, a[i+1].get_alban_tushaal());
        a[i].set_alban_tushaal(sav4);
        a[i+1].set_alban_tushaal(sav3);

        sav5 = a[i].get_ajillasan_tsag();
        sav6 = a[i+1].get_ajillasan_tsag();
        a[i].set_ajillasan_tsag(sav6);
        a[i+1].set_ajillasan_tsag(sav5);
    }
}

}

Ajilchin::Ajilchin() {
    dugaar = 0;
    ner = new char[4];
    strcpy(ner, "bat");
    alban_tushaal = new char[9];
    strcpy(alban_tushaal, "ajilchin");
    ajillasan_tsag = 0;
}

Ajilchin::~Ajilchin() {
    delete ner;
    delete alban_tushaal;
    cout << "ustlaa.\n";
}

Ajilchin::Ajilchin(Ajilchin& staff) {
    dugaar = staff.dugaar;
    delete ner;
    ner = new char[strlen(staff.ner)+1];
    strcpy(ner, staff.ner);
    delete alban_tushaal;
    alban_tushaal = new char [strlen(staff.alban_tushaal)+1];
    strcpy(alban_tushaal, staff.alban_tushaal);
    ajillasan_tsag = staff.ajillasan_tsag;
}

Ajilchin::Ajilchin(int a, char* b, char* c, float d) {
    dugaar = a;
    delete ner;
    ner = new char[strlen(b)+1];
    strcpy(ner, b);

```

```

        delete alban_tushaal;
        alban_tushaal = new char[strlen(c)+1];
        strcpy(alban_tushaal, c);
        ajillasan_tsag = d;
    }

void Ajilchin::set_dugaar(int a) {
    this->dugaar = a;
}

void Ajilchin::set_ner(char* a) {
    delete ner;
    ner = new char[strlen(a)+1];
    strcpy(ner, a);
}

void Ajilchin::set_alban_tushaal(char* a) {
    delete alban_tushaal;
    alban_tushaal = new char[strlen(a)+1];
    strcpy(alban_tushaal, a);
}

void Ajilchin::set_ajillasan_tsag(float a) {
    ajillasan_tsag = a;
}

bool Ajilchin::ajillasan_tsag_nemegduuleh(float a) {
    if(a>=0 && a<=24) {
        ajillasan_tsag += a;
        return true;
    }
    return false;
}

int Ajilchin::get_dugaar() {
    return dugaar;
}

char* Ajilchin::get_ner() {
    return ner;
}

char* Ajilchin::get_alban_tushaal() {
    return alban_tushaal;
}

float Ajilchin::get_ajillasan_tsag() {
    return ajillasan_tsag;
}

float Ajilchin::tsalin_bodoh() {
    if(strcmp(alban_tushaal, "zahiral")==0) {
        return zahirliin_tsalin_bodoh();
    }
}

```



```

    int tsagiin_huls = 5000;
    return tsagiin_huls * ajillasan_tsag;
}

float Ajilchin::zahirliin_tsalin_bodoh() {
    int tsagiin_huls = 10000;
    return tsagiin_huls * ajillasan_tsag;
}

main() {
    int m=0, n=-1;
    int a;
    char b[20], c[20];
    float d;
    Ajilchin k[30];
    while(m != 6) {
        cout << "1. Ajiltan nemeh\n";
        cout << "2. Ajiltnii medeelliig oruulah\n";
        cout << "3. Ajilchdiin medeelliig delgetsleh\n";
        cout << "4. Ajilchdiig nereer ni erembeleh\n";
        cout << "5. Ajillasan tsag nemegduuleh\n";
        cout << "6. Duusgah\n";
        cin >> m;
        if(m==1) {
            n++;
        }
        if(m==2) {
            cout << "dugaar oruul: ";
            cin >> a;
            if(dugaar_davhtssan_eseh(k, n, a)==0) {
                k[n].set_dugaar(a);
                cout << "ner oruul: ";
                cin >> b;
                k[n].set_ner(b);
                cout << "alban tushaal oruul: ";
                cin >> c;
                k[n].set_alban_tushaal(c);
            } else {
                cout << "Iim id dugaartai ajiltan burtgeltei baina.\n";
            }
        }
        if(m==3) {
            for(int i=0; i<=n; i++) {
                cout << k[i].get_dugaar() << " dugaartai ajiltan ";
                cout << k[i].get_ner() << " ni ";
                cout << k[i].get_alban_tushaal() << " alban tushaal tai buguud ";
                cout << k[i].get_ajillasan_tsag() << " tsag ajillasan ba ";
                cout << k[i].tsalin_bodoh() << " tsalintai baina.\n";
            }
        }
        if(m==4) {
            nereer_erembeleh(k, n);
        }
        if(m==5) {

```

```

int h=0;
cout << "Tsagiig ni nemeh ajiltnii dugaariig oruul: ";
cin >> a;
for(int i=0; i<=n; i++) {
    if(k[i].get_dugaar()==a) {
        h++;
    }
}
for(int i=0; i<=n; i++) {
    if(k[i].get_dugaar()==a) {
        cout << "Nemeh tsagaa oruul: ";
        cin >> d;
        if(k[i].ajillasan_tsag_nemegduuleh(d)==1) {
            cout << "Ajiltnii tsag amjilttai nemegdlee.\n";
        } else {
            cout << "Buruu tsag oruulsan baina.\n";
        }
    }
}
if(h==0) {
    cout << "Tanii oruulsan dugaartai ajiltan baihgui baina\n";
}
}
}
}
}

```