

Класс Дотор Байгуулагч Болон Устгагч Функц Тодорхойлох

Лабораторийн Ажил

(Лаборатори №4)

1. ОРШИЛ/УДИРТГАЛ

Амьдралд байгаа объектыг компьютерийн програмд загварчлахад классыг тодорхойлж, объект үүсгэн ажилчдыг бүртгэх, цаг бүртгэх, цалин тооцох зэрэг үйлдлийг болон гишүүн өгөгдлийг объектод хийнэ. Анхны утгыг байгуулагч функцээр дамжуулж олгоно. Байгуулагч болон устгагч функцийг гаргах үр дүнг туршилтаар харуулна.

2. ЗОРИЛГО

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

1. Классын гишүүн өгөгдлүүдийн төрлийг `private` хандалттайгаар тодорхойлох,
2. Классын байгуулагч болон устгагч функцуудыг тодорхойлох,
3. Бусад гишүүн функцүүдийг тодорхойлох,
4. Объект үүсгэж, боловсруулалт хийх.

3. ОНОЛЫН СУДАЛГАА

3.1 Байгуулагч функц

Байгуулагч функц нь объектын гишүүн өгөгдөлд дамжуулсан утга эсвэл анхны утгыг өгөх зориулалттай. Энэхүү функцийг нэр нь классын нэртэй ижил байна. Объект үүсэх үед `compiler` байгуулагч функцийг автоматаар дуудна. Байгуулагч функцийг 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 Динамик санах ойг байгуулагч, устгагчтай хамтад нь ашиглах

Олон объект байгуулах үед хэт том тэмдэгт хүснэгтийг гишүүн өгөгдөл болгож ашиглах нь санах ойг ихээр зарцуулдаг. Тиймээс яг шаардлагатай тэмдэгт хүснэгтийг нөөцөлж санах ойг үр ашигтай ажиллуулах үүднээс байгуулагч функц дотор new операторыг хэрэглэн санах ойг нөөцөлдөг. Харин устгагч функцээр дамжуулж delete операторын тусламжтай тухайн нөөцөлсөн санах ойг чөлөөлнө. Ингэснээр санах ойн цоорхой үүсэхээс сэргийлнэ. Иймэрхүү байдлаар динамик санах ойг ашигласнаар жижиг тэмдэгт мөрд том санах ойг ашиглахаас сэргийлж тэмдэгт мөрд тохирсон санах ойг хэрэглэнэ.

4. ХЭРЭГЖҮҮЛЭЛТ

Даалгаварт өгсөн загварын дагуу классын гишүүн өгөгдөл, гишүүн функцийг тодорхойлсон. Бүх өгөгдөл болон захиралын цалин бодох функц private хандалттай байсан.

Уг ажилчин класс нь ажилтны дугаар, нэр, албан тушаал, ажилласан цаг гэх өгөгдлүүдийг агуулна. Мөн байгуулагч болон устгагч функцуудээс гадна гараас утга авах, ажилчдын мэдээллийг дэлгэцлэх, цалин бодох, ажилласан цагийг нэмэгдүүлэх гэх мэт функцүүдтэй юм.

Main функц дотор уг классын объектон хүснэгт үүсгээд шинэ ажилчдын мэдээллийг нэмж оруулж болохуйц байдлаар байгуулсан.

Main функц нь давталтаар зохион байгуулагдсан бөгөөд гарах нөхцлийг хангах хүртэл програм ажиллаж ажилчин нэмэх, мэдээллийг оруулах, мэдээллийг харах байдлаар хийсэн.

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

Цалингаар эрэмбэлэх болон main функцууд дуусах үед классын объектуудын устгагч функцууд ажиллаж устлаа гэсэн үгийг дэлгэцлэнэ.

5. ДҮГНЭЛТ

Классыг загварчилснаар өгөгдлийг илүү эмх цэгцтэй мөн хамгаалалт сайтай байдлаар зохион байгуулах боломжийг олгож байна. Гишүүн өгөгдлүүдийг private хандалтай болгосноор утгыг нь логикгүй байдлаар өөрчлөхөөс сэргийлж байна. Байгуулагч функцийг тодорхойлж өгснөөр классын объект үүсэх үед гарааны утгыг шууд оноох боломжтой болгож байна.

6. АШИГЛАСАН МАТЕРИАЛ

1. <https://www.geeksforgeeks.org/c-plus-plus/?ref=shm>

7. ХАВСРАЛТ

```
#include <iostream>
#include <string.h>
using namespace std;
class Ajilchin {
private:
    int dugaar;
    char ner[20];
    char alban_tushaal[20];
    float ajillasan_tsag;
public:
    Ajilchin();
    ~Ajilchin();
    int dugaar_avah();
    void dugaar_ogoh(int);
    void ner_ogoh(char*);
    void alban_tushaal_ogoh(char*);
    char* ner_avah();
    char* alban_tushaal_avah();
    void garaas_utga_avah();
    void medeelel_delgetsleh();
    float tsalin_bodoh();
    bool tsag_nemeh(float);
private:
    float zahiraliin_tsalin();
};
Ajilchin::Ajilchin() {
    dugaar = 0;
    strcpy(ner, "");
    strcpy(alban_tushaal, "ajichin");
    ajillasan_tsag = 0;
}
Ajilchin::~Ajilchin() {
```

```

    cout << "ustlaa\n";
}
void Ajilchin::dugaar_ogoh(int n) {
    dugaar = n;
}
void Ajilchin::ner_ogoh(char* k) {
    strcpy(ner, k);
}
void Ajilchin::alban_tushaal_ogoh(char* k) {
    strcpy(alban_tushaal, k);
}
void Ajilchin::garaas_utga_avah() {
    cout << "ajiltanii dugaariig oruul: ";
    cin >> dugaar;
    cout << "ajiltanii neriig oruul: ";
    gets(ner);
    gets(ner);
    cout << "ajiltanii alban tushaaliig oruul: ";
    gets(alban_tushaal);
}
void Ajilchin::medeelel_delgetsleh() {
    cout << dugaar << " dugaartai ";
    cout << ner << " ajiltan ni ";
    cout << alban_tushaal << " alban tushaaltai buguud ";
    cout << ajillasan_tsag << " tsag ajillasan buguud ";
    cout << tsalin_bodoh() << " tsalintai baina.\n";
}
float Ajilchin::tsalin_bodoh() {
    if(strcmp(alban_tushaal, "zahiral")==0);
        float k = zahiraliin_tsalin();
        return k;
    float tsagiin_huls = 5000;
    return ajillasan_tsag*tsagiin_huls;
}
float Ajilchin::zahiraliin_tsalin() {
    float tsagiin_huls = 10000;
    return ajillasan_tsag*tsagiin_huls;
}
bool Ajilchin::tsag_nemeh(float k) {
    if(k>=0 && k<=24) {
        ajillasan_tsag += k;
        return true;
    }
    return false;
}
int Ajilchin::dugaar_avah() {
    return dugaar;
}

```

```

char* Ajilchin::ner_avah() {
    return ner;
}
char* Ajilchin::alban_tushaal_avah() {
    return alban_tushaal;
}
void erembeleh(Ajilchin* k, int n) {
    for(int j=0; j<n; j++) {
        for(int i=0; i<n; i++) {
            if(k[i].tsalin_bodoh() < k[i+1].tsalin_bodoh()) {
                Ajilchin tmp;
                tmp = k[i];
                k[i] = k[i+1];
                k[i+1] = tmp;
            }
        }
    }
}
int main() {
    int a=0, n=0;
    Ajilchin k[100];
    while(a != 6) {
        cout << "1. Ajiltan nemeh\n2. Ajiltanii medeeler oruulah\n3. Ajilchdiin medeelliig haruulah\n4.
        Ajillasan tsag nemeh\n5. Tsalingaar erembeleh\n6. Garah\n";
        cin >> a;
        if(a==1) {
            n++;
        }
        if(a==2) {
            k[n].garaas_utga_avah();
        }
        if(a==3) {
            for(int i=0; i<=n; i++) {
                k[i].medeeler_delgetsleh();
            }
        }
        if(a==4) {
            cout << "Ajillasan tsagiig nemegduuleh gej bui ajiltanii dugaariig oruul: ";
            int kod;
            float tsag;
            cin >> kod;
            for(int i=0; i<=n; i++) {
                if(k[i].dugaar_avah() == kod) {
                    cout << "nemek tsagaa oruul: ";
                    cin >> tsag;
                    if(k[i].tsag_nemek(tsag) == true) {
                        cout << "Ajiltanii tsag amjilttai nemegdlee.\n";
                    } else {

```

```
        cout << "Oruulsan tsag buruu baina.\n";
    }
}
}
}
if(a==5) {
    erembeleh(k, n);
}
}
return 0;
}
```