

Загвар функц (Лаборатори №10)

О. Ихбаяр

ХШУИС, Програм хангамж, 3-р түвшин, 17b1num2575@stud.num.edu.mn

1. ОРШИЛ

Загвар функц гэж юу болох түүнийг хэрхэн хэрэглэх талаар авч үзэх болно.

2. ЗОРИЛГО

Загвар функцийг таньж мэдэн зөв зохистой хэрэглэж сурах зорилготой.

3. ОНОЛЫН СУДАЛГАА

3.1 Загвар функц

C++ дээр загвар функцийг ашигладаг бөгөөд өгөгдлийн төрлийг параметр хэлбэрээр дамжуулах боломжийг олгодог C++ - ийн загвар юм. Зарлахдаа `template` түлхүүр үгийг ашигладаг. Жишээ нь: `template < typename T >`;

```
T max(T a, T b){  
    Return a>b?a:b;  
}
```

Дуудахдаа: `a=2,5; b=3.2; үед max <float> (a,b);` Эсвэл `max(a,b)` гээд дуудаж болно.

4. ХЭРЭГЖҮҮЛЭЛТ

4.1 Өгөгдөл хадгалах Struct тодорхойлох

```
//Ugugdul hadgalah  
template <class T>  
struct Node {  
    T val;  
    Node<T> *next;  
};
```

Тодорхойлолт: `struct` байгуулж түүндээ `T` төрлийн хувьсагчийг `val` - д хадгална. `Next` нь жагсаалтынхаа дараагийн элементийн хаягийг зааж өгнө.

```
Ehlead: 1, 21, 24, 4, 6,  
3-r bairlal deerh ugugdul ni: 24  
3-r bairlal deer 10-g nemhed: 1, 21, 10, 24, 4, 6,  
4-r bairlal deerh elementiig ustgah: 1, 21, 10, 4, 6,  
Jagsaaltiin urt ni: 5  
Char torliin ogogdol hadgal: a, b,  
urt ni: 2
```

4.2 Жагсаалтын төгсгөлд нэмэх

//Жагсаалтын төгсгөлд утга нэмнэ

```
template <class T>
void List<T>::add(T t){
    Node<T>* newNode = new Node<T>;
    newNode->val = t;
    newNode->next = NULL;
    Node<T>* temp = first;
    if(temp == NULL){
        first = newNode;
    }else{
        while(temp!=NULL){
            if(temp->next==NULL){
                temp->next=newNode;
                last = temp->next;
                break;
            }
            temp=temp->next;
        }
    }
}
```

Т төрлийн t хувьсагчийг жагсаалтын төгсгөлд хийнэ. Ингэхийн тулд t хувьсагчаа хадгалах newNode хаягийг бэлдэнэ. t хувьсагчаа хамгийн сүүлд хийхийн тулд жагсаалтаа төгсгөл хүртэл нь гүйлгээд хамгийн сүүлийн элемент дээр нэмэгдэж буй элементийн хаягийг зааж өгснөөр шинэ элемент нэмэгдэнэ.

5. ДҮГНЭЛТ

Энэхүү лабораторын ажиллаар загвар буюу template Функцийг хэрхэн хэрэглэх талаар болон түүний үүргүүдийн талаар тодорхой хэмжээний ойлголттой болсон.

6. АШИГЛАСАН МАТЕРИАЛ

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

7. ХАВСРАЛТ

```
#include <iostream>

using namespace std;

//Ugugdul hadgalah

template <class T>

struct Node {

    T val;

    Node<T> *next;
```

```
};
```

```
template <class T>
```

```
class List {
```

```
private:
```

```
    //Жагсаалтын эхлэлийг заана
```

```
    Node<T> *first;
```

```
    //Жагсаалтын төгсгөлийг заана
```

```
    Node<T> *last;
```

```
public:
```

```
    //Жагсаалтын төгсгөлд утга нэмнэ
```

```
    void add(T t);
```

```
    //index байрлалд элемент нэмнэ
```

```
    void insert(T t, int index);
```

```
    //index байрлал дахь элемент буцаана
```

```
    T get(int index);
```

```
    //index байрлал дахь элемент устгана.
```

```
    void deletes(int index);
```

```
    //жагсаалтын уртыг буцаана
```

```
    int size();
```

```
    //жагсаалтыг хэвлэх
```

```
    void print();
```

```
    List();
```

```
    ~List();
```

```
};
```

```
//Гарааны утга онооно
```

```
template <class T>
```

```
List<T>::List(){
```

```

    first = NULL;

    last = NULL;
}

//Нөөцөлсөн үүрнүүдээ устгана
template <class T>
List<T>::~~List(){
    Node<T>* temp = first;
    while(temp != NULL){
        temp = temp->next;
        delete(first);
        first = temp;
    }
}

//Жагсаалтын төгсгөлд утга нэмнэ
template <class T>
void List<T>::add(T t){
    Node<T>* newNode = new Node<T>;
    newNode->val = t;
    newNode->next = NULL;
    Node<T>* temp = first;
    if(temp == NULL){
        first = newNode;
    }else{
        while(temp!=NULL){
            if(temp->next==NULL){
                temp->next=newNode;
                last = temp->next;
                break;
            }
        }
    }
}

```

```

    }
    temp=temp->next;
}
}
}

```

```

template <class T>
void List<T>::insert(T t, int index){
    int too=0;
    Node<T>* p = first, *newNd = new Node<T>;
    while(p!=NULL){
        too++;
        if(too == index-1){
            newNd->val = t;
            newNd->next = p->next;
            p->next = newNd;
            break;
        }
        p=p->next;
    }
}

```

```

template <class T>
T List<T>::get(int index){
    int too=0;
    Node<T>* p = first;
    while(p!=NULL){
        too++;

```

```

        if(too == index){
            return p->val;
        }
        p=p->next;
    }
}

```

```

template <class T>
void List<T>::deletes(int index){
    int i;
    Node<T>* p = first;
    for(i=1;i<index-1;i++){
        p=p->next;
    }
    p->next=p->next->next;
};

```

//Жагсаалтын хэмжээг буцаана

```

template <class T>
int List<T>::size(){
    int siz=0;
    Node<T>* temp = first;
    if(temp == NULL){
        return siz;
    }
    if(temp->next == NULL){
        siz++;
    }else{
        while(temp != NULL){

```

```

        temp = temp->next;
        siz++;
    }
}
return siz;
}

```

```

template <class T>
void List<T>::print(){
    Node<T>* temp = first;
    if(temp == NULL){
        cout<<"";
    }
    if(temp->next == NULL){
        cout<<temp->val;
    }else{
        while(temp != NULL){
            cout<<temp->val;
            temp = temp->next;
            cout<<" ";
        }
    }
}

```

```

main(){
    //int төрлийн жагсаалт үүсгэх
    List <int> p;
    p.add(1);
}

```

```

p.add(21);
p.add(24);
p.add(4);
p.add(6);
cout<<"Ehlead: ";
p.print();
cout<<"\n3-r bairlal deerh ugugdul ni: "<<p.get(3);
cout<<"\n3-r bairlal deer 10-g nemhed: ";
p.insert(10,3);
p.print();
cout<<"\n4-r bairlal deerh elementiiг ustgah: ";
p.deletes(4);
p.print();
cout<<"\nJagsaaltiin urt ni: "<<p.size();
//char төрлийн жагсаалт үүсгэх
cout<<"\nChar torliin ogogdol hadgal: ";
List <char> l;
l.add('a');
l.add('b');
l.print();
cout<<"\nurt ni: "<<l.size();
}

```