Дєнін Богдан КІТ-120Д 25.05.21

Лабораторна робота №29

Тема: ООП. Шаблонні функції та класи.

Мета: Навчитися використовувати шаблонні функції та класи.

Індивідуальне завдання

Зробити шаблонний клас-список, що має шаблонне поле масиву. Створити наступні методи:

1. Вивід вмісту масиву на екран;
2. Визначити індекс переданого елемента в заданому масиві;
3. Визначити значення мінімального елемента масиву;
4. Додати елемент до кінця масиву;
5. Видалити елемент масиву за індексом.

Хід роботи

1. Зробив шаблонний клас-список, що має шаблонне поле масиву. Код показано на рисунку.

```
template<typename TValue>
struct ListItem
{
    ListItem* next = nullptr;
    ListItem* prev = nullptr;
    TValue value;

    ListItem() = default;
    ListItem(TValue value)
    {
        this->value = value;
    }
};

template<typename TValue>
class List
{
public:
    List() = default;


    void AddItem(const TValue& value);
    void RemoveAt(const size_t index);
    int GetIndexByValue(TValue val);
    TValue GetMinValue();
    void Print() const;


    size_t GetSize() const { return _size; }

    ~List();

    List(const List& source);
    List(List&& source);

    List& operator=(const List& source);
    List& operator=(List&& source);

    void Clear();

private:

    ListItem<TValue>* _head = nullptr;
    ListItem<TValue>* _tail = nullptr;
    size_t _size = 0;

    ListItem<TValue>* GetItemAt(size_t index);


    void Copy(const List& src);
    void Move(List& src);
};
```

Рисунок 1 – Шаблонний клас-список

2. Зробив функцію додавання елемента. Код показано на рисунку.

```
template<typename TValue>
inline void List<TValue>::AddItem(const TValue& value)
{
    ListItem<TValue>* newItem = new ListItem<TValue>(value);

    if (_head == nullptr)
    {
        _head = newItem;
        _tail = newItem;
    }
    else
    {
        _tail->next = newItem;
        newItem->prev = _tail;
    }
    _tail = newItem;
    _size++;
}
```

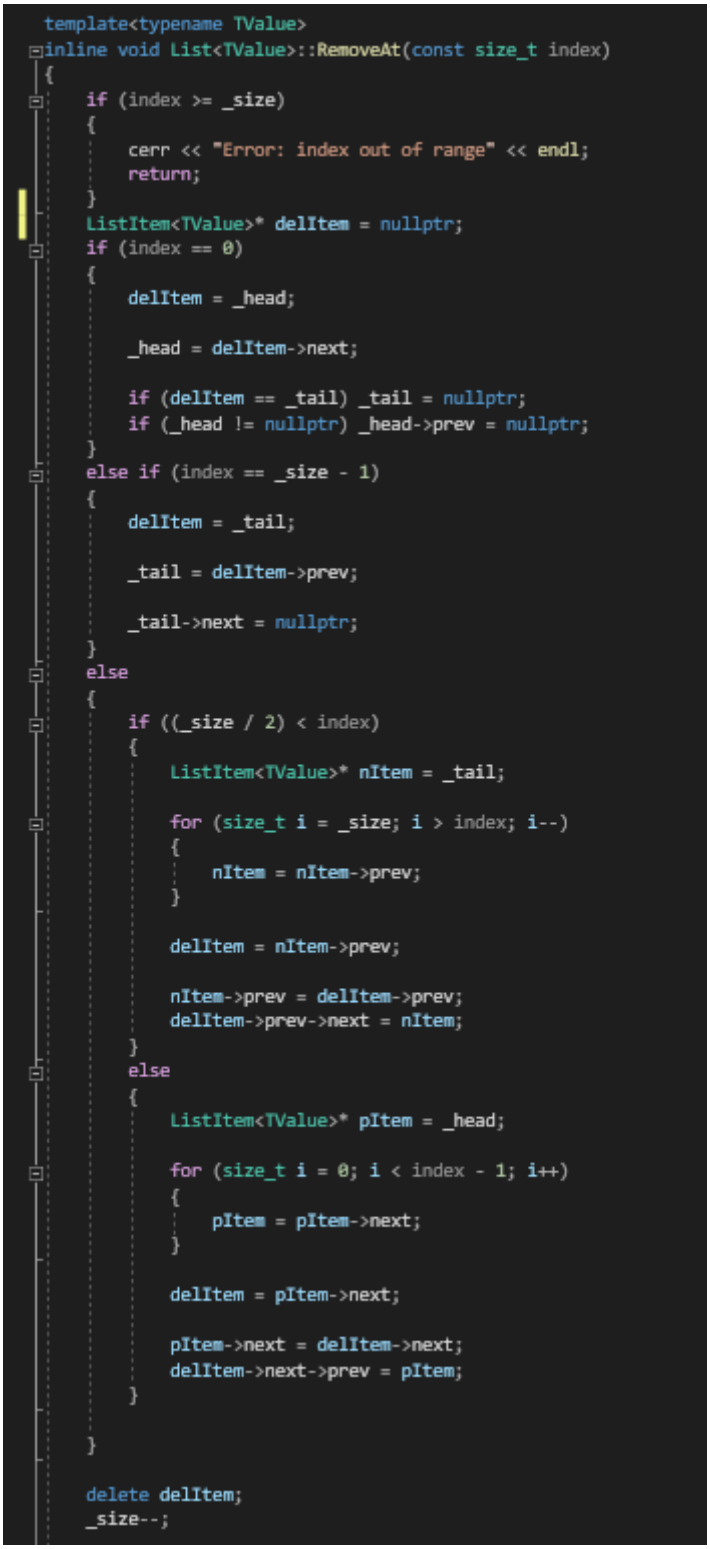
Рисунок 2 – Функція додавання елемента

3. Зробив функцію виводу списку на екран. Код показано на рисунку.

```
template<typename TValue>
inline void List<TValue>::Print() const
{
    ListItem<TValue>* cursor = _head;
    while (cursor != nullptr)
    {
        cout << cursor->value << "\t";
        cursor = cursor->next;
    }
    cout << endl;
}
```

Рисунок 3 – Функція виводу списку на екран

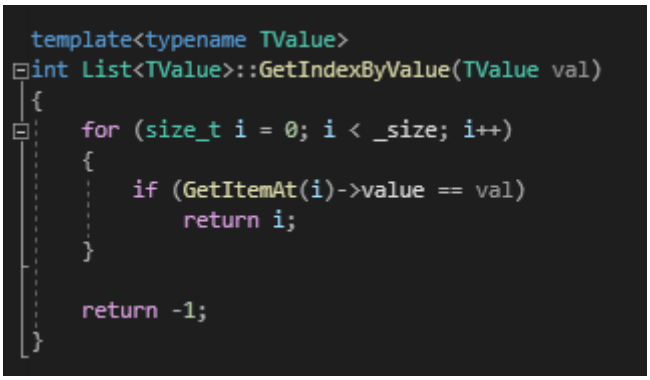
4. Зробив функцію видалення елемента за індексом. Код показано на рисунку.



```
template<typename TValue>
inline void List<TValue>::RemoveAt(const size_t index)
{
    if (index >= _size)
    {
        cerr << "Error: index out of range" << endl;
        return;
    }
    ListItem<TValue>* delItem = nullptr;
    if (index == 0)
    {
        delItem = _head;

        _head = delItem->next;

        if (delItem == _tail) _tail = nullptr;
        if (_head != nullptr) _head->prev = nullptr;
    }
    else if (index == _size - 1)
    {
        delItem = _tail;

        _tail = delItem->prev;

        _tail->next = nullptr;
    }
    else
    {
        if ((_size / 2) < index)
        {
            ListItem<TValue>* nItem = _tail;

            for (size_t i = _size; i > index; i--)
            {
                nItem = nItem->prev;
            }

            delItem = nItem->prev;

            nItem->prev = delItem->prev;
            delItem->prev->next = nItem;
        }
        else
        {
            ListItem<TValue>* pItem = _head;

            for (size_t i = 0; i < index - 1; i++)
            {
                pItem = pItem->next;
            }

            delItem = pItem->next;

            pItem->next = delItem->next;
            delItem->next->prev = pItem;
        }

    }

    delete delItem;
    _size--;
```

Рисунок 4 – Функція видалення елемента

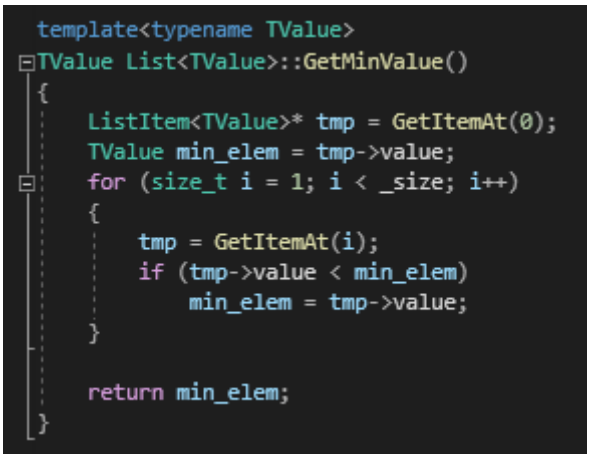
5. Зробив функцію для визначення індексу переданого елемента в заданому масиві. Код показано на рисунку.



```
template<typename TValue>
int List<TValue>::GetIndexByValue(TValue val)
{
    for (size_t i = 0; i < _size; i++)
    {
        if (GetItemAt(i)->value == val)
            return i;
    }

    return -1;
}
```

Рисунок 5 – Функція визначення індексу

6. Зробив функцію для визначення значення мінімального елемента в масиві. Код показано на рисунку.



```
template<typename TValue>
TValue List<TValue>::GetMinValue()
{
    ListItem<TValue>* tmp = GetItemAt(0);
    TValue min_elem = tmp->value;
    for (size_t i = 1; i < _size; i++)
    {
        tmp = GetItemAt(i);
        if (tmp->value < min_elem)
            min_elem = tmp->value;
    }

    return min_elem;
}
```

Рисунок 6 – Функція визначення мінімального елемента

7. Зробив функцію main для демонстрації роботи програми. Код показано на рисунку.

```
int main()
{
    List<int> obj;
    obj.AddItem(5);
    obj.AddItem(4);
    obj.AddItem(6);
    obj.AddItem(2);
    obj.AddItem(1);
    cout << "Array:" << endl;;
    obj.Print();
    cout << endl << "Min value: ";
    cout << obj.GetMinValue() << endl;
    obj.Print();
    cout << endl << "Get index by value(4): ";
    cout << obj.GetIndexByValue(4) << endl;
    cout << endl << "Get index by value(2): ";
    cout << obj.GetIndexByValue(2) << endl;
    obj.Print();
    cout << endl << "Remove by index(2): " << endl;
    obj.RemoveAt(2);
    obj.Print();
    return 0;
}
```

Рисунок 7 – Функція main

8. Приклад роботи програми показано на рисунку.

```
Array:
5       4       6       2       1

Min value: 1
5       4       6       2       1

Get index by value(4): 1

Get index by value(2): 3
5       4       6       2       1

Remove by index(2):
5       4       2       1
Для продолжения нажмите любую клавишу . . .
```

Рисунок 8 – Робота програми

Висновок: Освоїв тему «Шаблонні функції та класи». Навчився використовувати шаблонні функції та класи.