

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ТЕХНІЧНИЙ КОЛЕДЖ ТНТУ ІМЕНІ ІВАНА ПУЛЮЯ  
ВІДДІЛЕННЯ ЕЛЕКТРОННИХ АПАРАТІВ  
Циклова комісія програмних систем і комплексів

## **ЗВІТ**

про виконання лабораторних робіт з дисципліни:  
**«ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ»**

Студента 3 курсу групи КН-321  
спеціальності 122 «Комп'ютерні науки»

Попович О.В.  
(прізвище та ініціали)

Перевірив: Р.О. Слободян  
(підпис)

Тернопіль – 2020

# Лабораторна робота №2

**Тема:** Робота з лінійними списками. Конструктор і деструктор класу

**Мета:** Навчитись використовувати конструктори і деструктори класів, створювати класи для опису лінійних списків

## Завдання 1.

1 Реалізувати клас згідно варіанту індивідуального завдання, що містить закриті данні, а саме два типа даних: числове значення та рядок, реалізований через вказівник на char (char \*).

2 Реалізувати методи:

- конструктор по замовчуванню;
- конструктор ініціалізації клас(char\*, int);
- конструктор копіювання клас(const клас&);
- деструктор;
- Input() – запит у користувача даних та їх зчитування з клавіатури у поля класу;
- Print() – константний метод виводу даних на екран;
- методи доступу до закритих даних.

3 У функції main() створити декілька екземплярів класу статично і динамічно (із введенням даних із клавіатури користувачем), продемонструвати дію всіх конструкторів і методів.

## КОД ПРОГРАМИ

```
#include <iostream>
using namespace std;

class House
{
private:
    char *Type;
    int Number;
    int Size;
public:
    House() {}

    House(char * Type_, int Number, int Size) {

        this->Number = Number;
        this->Size = Size;

        Type = new char[Size];

        for (int i = 0; i < Size; i++)
        {
            this->Type[i] = Type_[i];
        }
    }
}
```

```

};

House(const House& other) {
    this->Type = new char[other.Number];

    for (int i = 0; i < other.Size; i++)
    {
        this->Type[i] = other.Type[i];
    }
};

void Input() {

    cout << "-----" << endl << "Input Method"
<< endl << "Enter the Number" << endl;
    cin >> this->Number;

    cout << "Enter the size of the string array" << endl;
    cin >> this->Size;

    Type = new char[Size];

    for (int i = 0; i < Size; i++)
    {
        cin >> Type[i];
    }
    cout << "-----";
};

void Print() const {
    cout << "-----" << endl << "Print Method"
" << endl << "char*-->";
    for (int i = 0; i < this->Size; i++)
    {
        cout << Type[i];
    }
    cout << endl;
    cout << "Number --> " << this->Number << endl;
    cout << "-----" << endl;;
};

int getNumber() { return Number; }
void setNumber(int Number) { this->Number = Number; }

void getType()
{
    cout << endl;
    for (int i = 0; i < this->Size; i++)
        cout << Type[i];
}

void setType()
{
    cout << endl << "Enter the size of the string array" << endl;
    int size; cin >> size;
    Type = new char[size];
    for (int i = 0; i < size; i++)
    {
        cin >> Type[i];
    }
}

~House() {
    delete[]Type;
};

};

int main()

```

```

{
    cout << "Enter the size of the string array" << endl;
    int size; cin >> size;

    char *charArr = new char[size];
    for (int i = 0; i < size; i++) {
        cin >> charArr[i];
    }

    cout << "Enter the Number" << endl;
    int Number; cin >> Number;
    House object(charArr, Number, size);
    object.Print();

    cout << "How many objects you want to create" << endl;
    int NumberObject; cin >> NumberObject; NumberObject;

    House *arrObject = new House[NumberObject];

    for (int i = 0; i < NumberObject; i++)
    {
        arrObject[i].Input();
        arrObject[i].Print();
    }

    return 0;
}

```

## РЕЗУЛЬТАТ ВИКОНАННЯ ПРОГРАМИ

```

Enter the size of the string array
5
abcdq
Enter the Number
190
-----
Print Method
char*-->abcdq
Number --> 190
-----
How many objects you want to create
2
-----
Input Method
Enter the Number
111
Enter the size of the string array
3
OOP
-----
Print Method
char*-->OOP
Number --> 111
-----
Input Method
Enter the Number
800
Enter the size of the string array
11
programming
-----
Print Method
char*-->programming
Number --> 800
-----
C:\Users\ASUS\source\repos\ConsoleApplication21\Debug\ConsoleApplication21.exe

```

## Завдання 2.

4 \*Реалізувати клас однозв'язного списку List, який міститиме об'єкти класу, розробленого згідно варіанту індивідуального завдання. Продемонструвати роботу списку, добавивши декілька елементів, після чого вивести на екран увесь список.

### КОД ПРОГРАМИ

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

class House
{
private:
    string Number;
public:
    House() {};

    House(string Number) {

        this->Number = Number;

    };
    void Input() {

        cout << "-----" << endl << "Input Method"
<< endl << "Enter the array char" << endl;
        cin >> this->Number;

    };

    void Print() const {

        cout << "Number --> " << this->Number << endl;
        cout << "-----" << endl;;

    };

    string getNumber() { return Number; }
    void setNumber(int Number) { this->Number = Number; }
};

template<typename T>
class List
{
public:
    List();
    ~List();

    void push_back(T data);
    int GetSize()
    {
        return Size;
    }

    T& operator[](const int index);

private:
    template<typename T>
    class Node
    {
public:
```

```

        Node *pNext;
        T data;
        Node(T data = T(), Node *pNext = nullptr)
        {
            this->data = data;
            this->pNext = pNext;
        }
    };
    int Size;
    Node<T> *head;
};

template<typename T>
List<T>::List()
{
    Size = 0;
    head = nullptr;
}

template<typename T>
List<T>::~~List(){}

template<typename T>
void List<T>::push_back(T data)
{
    if (head == nullptr)
    {
        head = new Node<T>(data);
    }
    else
    {
        Node<T> *current = this->head;

        while (current->pNext != nullptr)
        {
            current = current->pNext;
        }
        current->pNext = new Node<T>(data);
    }
    Size++;
}

template<typename T>
T & List<T>::operator[](const int index)
{
    int counter = 0;
    Node<T> *current = this->head;

    while (current != nullptr)
    {
        if (counter == index)
        {
            return current->data;
        }
        current = current->pNext;
        counter++;
    }
}

int main()
{
    House house1;
    house1.Input();

    House house2;
    house2.Input();

    House house3;

```

```

house3.Input();

House house4;
house4.Input();

House house5;
house5.Input();

List<string>list;

list.push_back(house1.getNumber());
list.push_back(house2.getNumber());
list.push_back(house3.getNumber());
list.push_back(house4.getNumber());
list.push_back(house5.getNumber());

cout << endl;
for (int i = 0; i < list.GetSize(); i++)
{
    cout << "Object" << i + 1 << " value char[] --> " << " " << list[i] << endl;
}

return 0;
}

```

## РЕЗУЛЬТАТ ВИКОНАННЯ ПРОГРАМИ

```

Microsoft Visual Studio Debug Console

-----
Input Method
Enter the array char
Programist
-----
Input Method
Enter the array char
C++
-----
Input Method
Enter the array char
Programming
-----
Input Method
Enter the array char
.net
-----
Input Method
Enter the array char
backend

Object1 value char[] --> Programist
Object2 value char[] --> C++
Object3 value char[] --> Programming
Object4 value char[] --> .net
Object5 value char[] --> backend

C:\Users\ASUS\source\repos\ConsoleApplication21\Debug\ConsoleApplication21.exe

```

**Висновок:** *Навчився використовувати конструктори і деструктори класів, створювати класи для опису лінійних списків.*