

Dinamikus memóriefoglalás

- konstruktoroknál → class ... új nylevi elem kell...
 - malloc/ free helyett **new/delete (ezek nem függvényekek hanem operátorok!)**
 - pl.

```
Point * pp = new Point(); /vagy new Point(100, 200); /  
...  
delete pp;
```

- tömbnek hely foglalás:

```
Point * pp = new Point[25]; //itt lehet változó is a 25 helyén...ez csak default  
//konstruktort hív meg!
```

ezek törlése:

amilyen zárójellel létrehoztuk olyannal töröljük:

```
delete[] pp;  
delete pp;
```

pl.:

```
int * pi = new int(3); //minden beépített típusnak van 1 paraméteres konstruktora
```

```
int i = 3;
```

```
Point p = 3; → inicializálás
```

```
int a, b;
```

```
a = b; → értékadás
```

Stack osztály

```
<Stack.h>
```

```
#ifndef STACK_H
```

```
#define STACK_H
```

```
const int NULL = 0;
```

```
class Stack
```

```
{
```

```
    int* pData;
```

```
    unsigned int elemNum;
```

```
public:
```

```
    Stack(){pData = NULL; elemNum = 0;}
```

```

Stack(const Stack &theOther);

//destruktorok
~Stack(){delete[] pData;}

bool isEmpty{return !elemNum;}
bool push(int element);
bool pop(int &element);
};
#endif /*STACK_H*/

bool Stack :: push(int element)
{
    if(elemNum == INT_MAX) return false;
    int *pTemp = new int[elemNum + 1];

    for(unsigned int i = 0; i < elemNum; i++)
    {
        pTemp[i] = pData[i];
    }

    pTemp[elemNum] = element;
    delete[] pData;
    pData = pTemp;
    elemNum++;
}

bool Stack :: pop(int &element)
{
    if(elemNum == 0) return false;
    if(elemNum == 1)
    {
        element = *pData;
        delete[] pData;
        pData = NULL;
        elemNum = 0;
        return true;
    }
    else
    {
        elemNum--;
        int *pTemp = new int[elemNum];
        for(unsigned int i = 0; i < elemNum; i++)
        {
            ptemp[i] = pData[i];
            element = pData[elemNum];
            delete[] pData;
            pData = pTemp;
            return true;
        }
    }
}

```

```
Stack :: Stack(const Stack&theOther)
{
    elemNum = theOther.elemNum;
    if(elemNum == 0){pData = NULL; return; }

    [---]
}
```