

Задание 1

Создание типа *String* (символьной строки)

При построении программ на C++ каждое объявление класса обычно помещается в *заголовочный файл*, а описание методов этого класса помещается в *файл исходных кодов* с теми же базовыми именами.

```
class String
{
private:
    char *str;
    int len;

public:
    .....
};

#include "String.h"
#include<iostream>
using namespace std;

void main( void ){
//разные способы объявления и инициализации объектов

//объявление объекта s1 и инициализация с помощью конструктора с параметром
    String s1("Privet!!!!");
    s1.show();

    cout << "Press any key..." << endl<<endl;
    system("puase");

//переустановка данных объекта s1 с помощью метода
    s1.Set_str("Minsk!!!!!!");
    s1.show();

    cout << "Press any key..." << endl<<endl;
    system("puase");

//объявление объекта s2, но не инициализация, с помощью конструктора без //параметров
    String s2;

//инициализация данных объекта s1 с помощью метода
    s2.Set_str("12345 7890");
    s2.show();

    cout << "Press any key..." << endl<<endl;
    system("puase");

//объявление объекта s3 и инициализация (конструктор с параметром)
    String s3 = String("Da-Da");
    s3.show();

    cout << "Press any key..." << endl<<endl;
    system("puase");

//объявление объекта s4 и инициализация (конструктор копии)
    String s4 = s3;
    s4.show();

    cout << "Press any key..." << endl<<endl;
    system("puase");
```

```

//динамическое выделение памяти
String *s;
//создание объекта с помощью конструктора без параметров
s = new String;

//косвенный вызов компонентов-функций
s->Set_str("123456");

s->show();

cout << "Press any key..." << endl<<endl;
system("puase");

//динамическое выделение памяти
String *p;
//создание объекта с помощью конструктора с параметром
p = new String("No - No");

//косвенный вызов метода
p->show();

cout << "Press any key..." << endl<<endl;
system("puase");

// чтобы сработали деструкторы
delete s;
delete p;

system("puase");
}

```

Задание 2

Создание типа *Array* (обертка над статическим одномерным массивом)

```

class Array
{
private:
    const int SIZE;
    double *arr;
public:
    Array(int = 10);
    Array(const Array &ob);

    void SetElm(int index, int value);
    int GetElm(int index) const;
    int GetSize() const;

    void ShowMas(void) const;

    void QuickSort(int left, int right);
    void Reverse();

    ~Array();
};

void main()
{
    Array arr;
    arr.ShowMas();
}

```

```

        system("pause");

        for (int i = 0, x; i < arr.GetSize(); ++i)
        {
            cout << "[" << i + 1 << "]=";      cin >> x;
            arr.SetElm(i, x);
        }

        arr.ShowMas();
        system("pause");

        arr.QuickSort(0, arr.GetSize()-1);
        arr.ShowMas();
        system("pause");

        arr.Reverse();
        arr.ShowMas();
        system("pause");

        cout << arr.GetElm(100) << endl;
        system("pause");

        int n;
        cout << "Enter the size of the array: ";
        cin >> n;

        Array arr1(n);

        arr1.ShowMas();
        system("pause");
    }

    system("pause");
}

```

Задание 3

Создание типа **Array2** (обертка над статическим двумерным массивом)