

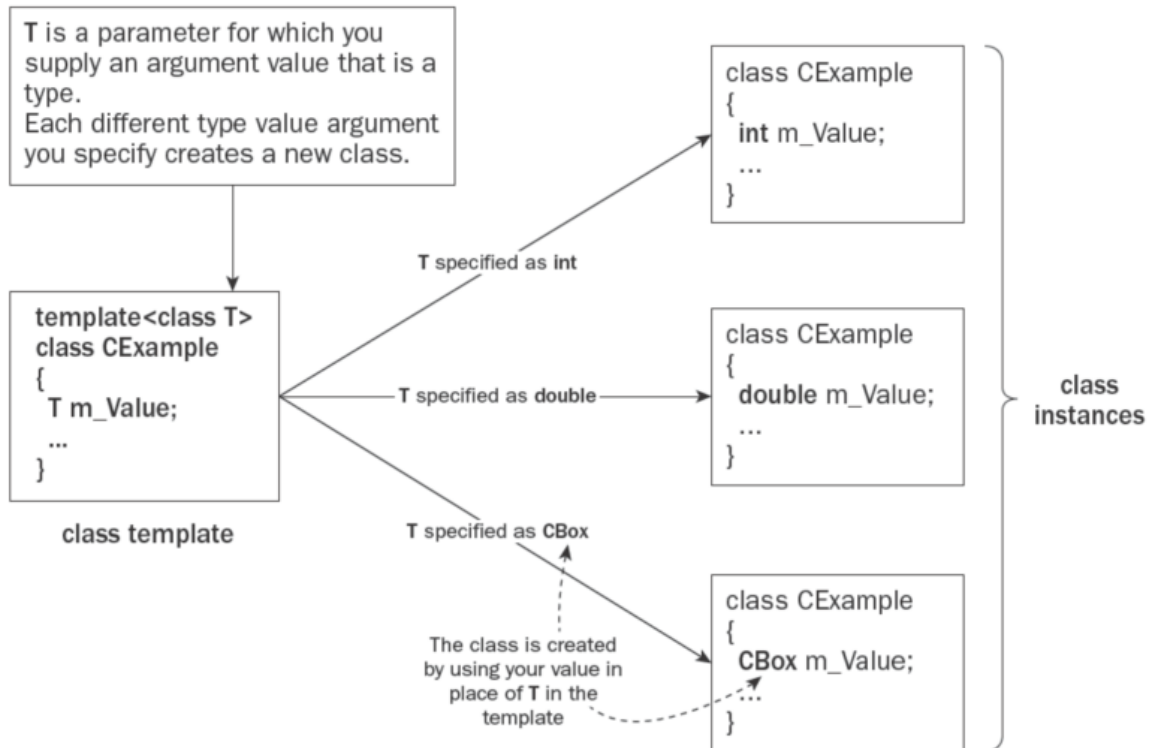
Упражнение 8 – Шаблиони

Шаблионите са директно поддържане на родовото програмиране, т.е. програмиране, в което типовете се използват като параметри. Шаблионът зависи само от характеристиките на параметричния тип, които действително използва, и не изисква различни типове, използвани като аргументи да бъдат свързани по явен път.

Дефиниране на шаблон

```
template<class T>
class CExample
{
    T m_Value;
    ...
}
```

- Използва се служебната дума `template`;
- `<class T>` - типа параметър, който се замества с точно определен тип данна при декларирането на обекта.



Пример:

```

template <class T>
class CSamples
{
    T Max() const;
    // Rest of the template definition...
};

template<class T>
T CSamples<T>::Max() const
{
    T theMax = m_Values[0];
    for(int i = 1; i < m_Free; i++)
        if(m_Values[i] > theMax)
            theMax = m_Values[i];
    return theMax;
}

```

Създаване на обект

Когато се създава обект, трябва да се укаже типа на параметъра:

```
CSamples<double> myData(10.0);
```

Създаване на шаблони клас с повече параметри:

```

template<class T1, class T2>
class CExampleClass
{
    // Class data members
    private:
        T1 m_Value1;
        T2 m_Value2;
    // Rest of the template definition...
};

```

Пример:

```

template <class T, int Size> class CSamples
{
    private:
        T mValues[Size];
        int mFree;
    public:
        CSamples(const T values[], int count)
        {
            mFree = count < Size? count:Size;
            for(int i = 0; i < mFree; i++)
                mValues[i] = values[i];
        }
};

CSamples<double, 10>

```

Задачи за самостоятелна работа:

1. Изтеглете и разгледайте примера Ex8_07.cpp.
2. Реализирайте към CSamples шаблон класа член-функциите:
 - Min() - за намиране на минимален елемент в масив;
 - Average() – за намиране на средно аритметичната стойност на елементите в масива.
3. Демонстрирайте функционалността чрез задаване на int и double тип за обекти от дефинираната шаблон функция.