**Параметрический полиморфизм** - еще одно из проявлений принципа полиморфизма, которое позволяет многократно использовать один и тот же код применительно к разным типам. Реализация функции или класса осуществляется один раз, но при этом тип указывается как параметр этой функции или класса. При использовании этой функции или класса параметра типа заменяется на встроенный или абстрактный тип данных и универсальных код, не зависящий от типа, работает с указанным типом.

Шаблоны реализуют в языке С++ такое отношение между классами, как инстанцирование. При создании шаблона класса тип одного или нескольких его переменных членов задаются в качестве параметра. При использовании этого шаблона необходимо будет указать, какой тип использовать в качестве параметра. Это и называется инстанцированием.

Процесс генерации объявления класса по шаблону и аргументу называется инстанцированием шаблона.

Пример:

template <class T> // это шаблон класса с T вместо фактического (передаваемого) типа данных

class Array

{

private:

int m\_length;

T \*m\_data;

public:

Array()

{

m\_length = 0;

m\_data = nullptr;

}

Array(int length)

{

m\_data = new T[length];

m\_length = length;

}

~Array()

{

delete[] m\_data;

}

};

int main()

{

Array<int> intArray(10);

Array<double> doubleArray(10);

//……

return 0;

}