

## ***Практическая работа. Параметризованный абстрактный тип данных «Память»***

### **Цель**

Сформировать практические навыки реализации параметризованного абстрактного типа данных с помощью шаблона классов C++.

### **Задание**

1. В соответствии с приведенной ниже спецификацией реализовать параметризованный абстрактный тип данных «память», для хранения одного числа – объекта типа T, используя шаблон классов C++.
2. Протестировать каждую операцию, определенную на типе данных, используя средства модульного тестирования.
3. Если необходимо, предусмотрите возбуждение исключительных ситуаций.

### **Спецификация типа данных «память».**

#### **ADT TMemory**

#### **Данные**

Память (тип TMemory, в дальнейшем - память) - это память для хранения «числа» объекта типа T в поле FNumber, и значения «состояние памяти» в поле FState. Объект память - изменяемый. Он имеет два состояния, обозначаемых значениями: «Включена» (\_On), «Выключена» (\_Off). Её изменяют операции: Записать (Store), Добавить (Add), Очистить (Clear).

#### **Операции**

<b>Конструктор</b>	
Начальные значения:	Нет.
Процесс:	Инициализирует поле FNumber объекта «память» (тип TMemory) объектом «число» (тип T) со значением по умолчанию.

	Например для числа типа TFrac со значением 0/1. Память устанавливается в состояние «Выключена», в поле FState «состояние памяти» заносится значение (_Off).
<b>Записать</b>	
Вход:	(E) – объект тип T.
Предусловия:	Нет.
Процесс:	В объект «память» (тип TMemory) в поле FNumber записывается копия объекта E. Память устанавливается в состояние «Включена», в поле FState «состояние памяти» заносится значение (_On).
Выход:	Нет.
Постусловия:	Состояние памяти поле FState – «Включена» (_On).
<b>Взять</b>	
Вход:	Нет.
Предусловия:	Нет.
Процесс:	Создаёт и возвращает копию объекта хранящегося в объекте «память» (тип TMemory) в поле FNumber.
Выход:	Объект типа T.
Постусловия:	Состояние памяти поле FState – «Включена» (_On).
<b>Добавить</b>	
Вход:	(E) – число объект типа T.

Предусловия:	Нет.
Процесс:	В поле FNumber объекта «память» (тип TMemory) записывается объект типа T, полученный в результате сложения числа (E) и числа, хранящегося в памяти в поле FNumber.
Выход:	Нет.
Постусловия:	Состояние памяти поле FState – «Включена» (_On).
<b>Очистить</b>	
Вход:	Нет.
Предусловия:	Нет.
Процесс:	В поле числа (FNumber) объекта «память» (тип TMemory) записывается объект типа T со значением по умолчанию. Например, для простой дроби - 0/1. Память (поле FState) устанавливается в состояние «Выключена» (_Off).
Выход:	Нет.
Постусловия:	Состояние памяти поле FState – «Выключена» (_Off).
<b>Читать Состояние Памяти</b>	
Вход:	Нет.
Предусловия:	Нет.
Процесс:	Копирует и возвращает значение поля FState «состояние памяти» объекта «память» (тип TMemory) в формате строки.

Выход:	Значение поля «состояния памяти» (типа String).
Постусловия:	Нет.
<b>ЧитатьЧисло</b>	
Вход:	Нет.
Предусловия:	Нет.
Процесс:	Копирует и возвращает значение поля «число» (FNumber) объекта «память» (тип TMemory).
Выход:	Объект число (тип T).
Постусловия:	Нет.

*end* TCMemory

### Рекомендации к выполнению

1. Тип данных реализуйте, используя параметризованный класс C++.  

```
template <class T>
```
2. Число храните в поле FNumber типа T.
3. Тип данных реализуйте в отдельном модуле UMemory.

Ниже приведены диаграмма классов и диаграмма состояний для класса «Память».

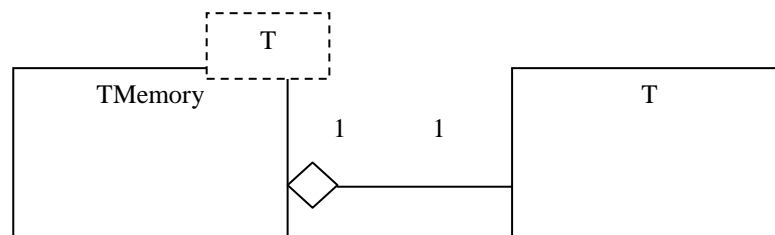


Рис. Диаграмма классов для класса «Память».

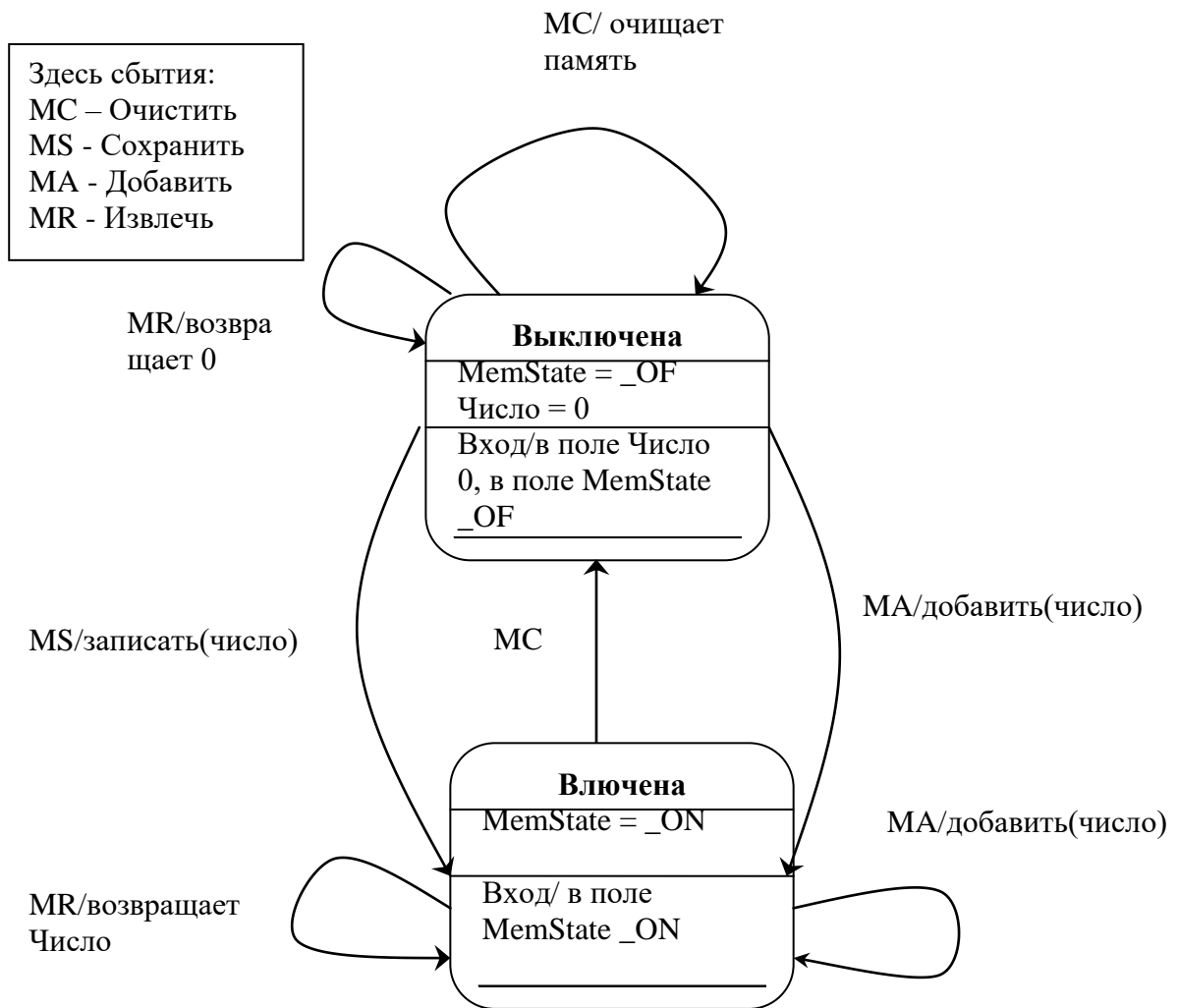


Рис. Диаграмма состояния для класса «Память»

## Содержание отчета

1. Задание.
2. Текст программы.
3. Тестовые наборы данных для тестирования типа данных.

## Контрольные вопросы

1. Когда в классе необходимо явным образом описать конструктор?
2. Что можно использовать в качестве параметров шаблона?
3. Можно ли использовать шаблоны в качестве параметров шаблона?