

Практическая работа. Параметризованный абстрактный тип данных «Процессор».

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 «Процессор», 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 «Процессор».

ADT TProc

Данные

Процессор (тип TProc) выполняет двухоперандные операции TOprtn = (None, Add, Sub, Mul, Dvd) и однооперандные операции - функции TFunc = (Rev, Sqr) над значениями типа T. Левый операнд и результат операции хранится в поле Lop_Res, правый - в поле Rop. Оба поля имеют тип T. Процессор может находиться в состояниях: «операция установлена» - поле Operation не равно None (значение типа TOprtn) или в состоянии «операция не установлена» - поле Operation = None. Значения типа TProc - изменяемые. Они изменяются операциями: «Сброс операции» (OprtnClear), «Выполнить операцию» (OprtnRun), «Вычислить функцию» (FuncRun), «Установить операцию» (OprtnSet), «Установить левый операнд» (Lop_Res_Set), «Установить правый операнд» (Rop_Set), «Сброс калькулятора» (ReSet). На значениях типа T должны быть определены указанные выше операции и функции.

Операции

Конструктор	
Начальные значения:	Нет
Процесс:	Инициализирует поля объекта процессор типа TProc. Поля Lop_Res, Rop инициализируются объектами (тип T) со значениями по умолчанию. Например, для простых дробей - 0/1. Процессор устанавливается в состояние: «операция не установлена»: (Operation = None).
СбросПроцессора	
Вход:	Нет.
Предусловия:	Нет.
Процесс:	Поля объекта процессор: Lop_Res, Rop инициализируются объектами (тип T) со значениями по умолчанию. Например, для простых дробей - 0/1. Процессор устанавливается в состояние: «операция не установлена»: (Operation = None).
Выход:	Нет.
Постусловия:	Состояние процессора – «операция сброшена» (Operation = None).
СбросОперации	
Вход:	Нет.
Предусловия:	Нет.
Процесс	Процессор устанавливается в состояние:

	«операция не установлена»: (Operation = None).
Выход:	Нет.
Постусловия:	Состояние процессора – «операция сброшена» (Operation = None).
ВыполнитьОперацию	
Вход:	Нет.
Предусловия:	Нет.
Процесс	Вызывает выполнение текущей операции (записанной в поле Operation). Операция (Operation) выполняется над значениями, хранящимися в полях Rop и Lop_Res. Результат сохраняется в поле Lop_Res. Если Operation = None, никакие действия не выполняются. Состояние объекта не изменяется.
Выход:	Нет.
Постусловия:	Состояние процессора не изменяется.
ВычислитьФункцию	
Вход:	Вид функции (Func: TFunc).
Предусловия:	Нет.
Процесс	Вызывает выполнение текущей функции (Func). Функция (Func) выполняется над значением, хранящимся в поле Rop. Результат сохраняется в нём же. Состояние объекта не изменяется.
Выход:	Нет.

Постусловия:	Состояние процессора не меняется.
<i>ЧитатьЛевыйОперанд</i>	
Вход:	Нет.
Предусловия:	Нет.
Процесс	Создаёт и возвращает копию объекта, который хранится в поле Lop_Res.
Выход:	Объект типа T.
Постусловия:	Состояние процессора не изменяется.
<i>ЗаписатьЛевыйОперанд</i>	
Вход:	Переменная Operand типа T.
Предусловия:	Нет.
Процесс	Создаёт копию объекта Operand и заносит её в поле Lop_Res.
Выход:	Нет.
Постусловия:	Состояние процессора не изменяется.
<i>ЧитатьПравыйОперанд</i>	
Вход:	Нет.
Предусловия:	Нет.
Процесс	Создаёт и возвращает копию объекта, который хранится в Rop.
Выход:	Объект типа T.
Постусловия:	Состояние процессора не меняется.
<i>ЗаписатьПравыйОперанд</i>	
Вход:	Переменная Operand типа T.
Предусловия:	Нет.

Процесс	Создаёт копию объекта Operand и заносит её в поле Rop.
Выход:	Нет.
Постусловия:	Состояние процессора не изменяется.
<i>ЧитатьСостояние</i>	
Вход:	Нет.
Предусловия:	Нет.
Процесс	Копирует и возвращает значение поля Operation.
Выход:	Значение поля Operation.
Постусловия:	Состояние процессора не изменяется.
<i>ЗаписатьСостояние</i>	
Вход:	Переменная Oprtn типа TOprtn.
Предусловия:	Нет.
Процесс	Заносит значение Oprtn в поле Operation.
Выход:	Нет.
Постусловия:	Состояние процессора изменяется на Oprtn.

Рекомендации к выполнению

1. Тип данных TProc реализовать, используя шаблон классов C++, template <class T>.
2. Числа хранить как поля типа T.
3. Для чтения состояния процессора, полей: «левый операнд-результат» (Lop_Res), «правый операнд» (Rop), используйте свойство (property).
4. Тип данных реализовать в отдельном модуле UProc.
5. В приведённой ниже таблице показана последовательность изменения состояния процессора, если $T = TProc$, при вычислении выражения:

$$2/1 + 3/1 * (4/1)^2$$

Шаг	Вход	Метод	Rop	Lop_Res	Operation
0		Create	0/1	0/1	None
1	2		0/1	0/1	None
2	+	Lop_Res_Set; OprtnSet	0/1	2/1	Add
3	3		0/1	2/1	Add
4	*	Rop_Set; OprtnRun; OprtnSet;	3/1	2/1+3/1	Mul
5	4		4/1	2/1+3/1	Mul
6	Sqr	Rop_Set; FuncRun	$(4/1)^2$	2/1+3/1	Mul
7	=	OprtnRun	$(4/1)^2$	2/1+3/1* $(4/1)^2$	Mul
8	C	ReSet	0/1	0/1	None

Ниже приведена диаграмма классов для класса память.

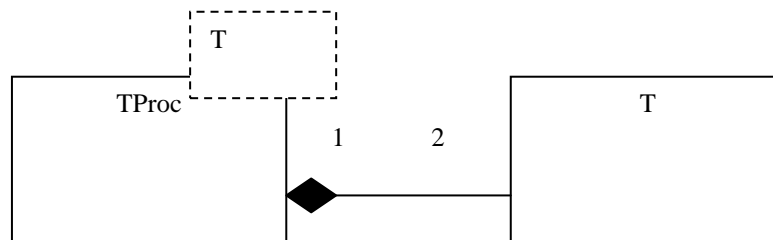


Рис. Диаграмма классов Процессор.

Содержание отчета

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

Контрольные вопросы

1. Как использовать тип в качестве параметра шаблона?
2. Как использовать переменную в качестве параметра шаблона?

3. Какие существуют отношения между классами?