***Лабораторная работа №3. Абстрактный тип данных: P-ичный процессор.***

Тема: Классы Object Pascal*,*С++

*Цель: Сформировать практические навыки: реализации абстрактного типа данных с помощью классов Object Pasca, С++.*

Задание

1. В соответствии с приведенной ниже спецификацией реализовать абстрактный тип данных “P-ичный процессор”, используя класс
* *Object Pasca,*
* *С++.*
1. Протестировать каждую операцию, определенную на типе данных одним из методов тестирования.

Спецификация типа данных “P-ичный процессор”.

**ADT**TProc

**Данные**

P-ичный процессор (тип TProc) выполняет двухоперандные операции TOprtn = (None, Add, Sub, Mul, Dvd) и однооперандные операции - функции TFunc = (Rev, Sqr). Если операция или функция не может быть выполнена, в поле Error типа String заносится сообщение об ошибке. Левый операнд и результат операции хранится в поле Lop\_Res, правый - в поле Rop. Оба поля имеют тип TPNumber. Процессор может находиться в состоянии “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).

**Операции**

|  |  |
| --- | --- |
| ***Конструктор*** |   |
| Начальные значения: | система счисления (b), точность представления числа (c) |
| Процесс: | Создаёт объект p-ичный процессор типа TProc. Поля Lop\_Res, Rop инициализируются объектами (тип TPNumber) со значением 0 в системе счисления (b), с точность представления числа (c). В поле Error заносится пустая строка. Процессор устанавливается в состояние: “операция не установлена”: (Operation = None). |
|   |   |
| ***СбросПроцессора*** |   |
| Вход: | система счисления (b), точность представления числа (c) |
| Предусловия: | Нет. |
| Процесс: | Поля объекта p-ичный процессор: Lop\_Res, Rop инициализируются объектами (тип TPNumber) со значением 0 в системе счисления (b), с точность представления числа (c). В поле Error заносится пустая строка. 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. Если операция не может быть выполнена, в поле Error заносится сообщение об ошибке. 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. Если операция не может быть выполнена, в поле Error заносится сообщение об ошибке. Состояние объекта не изменяется. |
| Выход: | Нет. |
| Постусловия: | Состояние процессора не меняется. |
|   |   |
| ***ЧитатьЛевыйОперанд*** |   |
| Вход: | Нет. |
| Предусловия: | Нет. |
| Процесс: | Создаёт и возвращает копию объекта, указатель на который хранится в Lop\_Res. |
| Выход: | Объект типа TPNumber. |
| Постусловия: | Состояние процессора не изменяется. |
|   |   |
| ***ЗаписатьЛевыйОперанд*** |   |
| Вход: | Переменная (Operand) типа TPNumber. |
| Предусловия: | Нет. |
| Процесс: | Создаёт копию объекта Operand и заносит указатель на него в поле Lop\_Res. |
| Выход: | Нет. |
| Постусловия: | Состояние процессора не изменяется. |
|   |   |
| ***ЧитатьПравыйОперанд*** |   |
| Вход: | Нет. |
| Предусловия: | Нет. |
| Процесс: | Создаёт и возвращает копию объекта, указатель на который хранится в Rop. |
| Выход: | Объект типа TPNumber. |
| Постусловия: | Состояние процессора не меняется. |
|   |   |
| ***ЗаписатьПравыйОперанд*** |   |
| Вход: | Переменная (Operand) типа TPNumber. |
| Предусловия: | Нет. |
| Процесс: | Создаёт копию объекта Operand и заносит указатель на него в поле Rop. |
| Выход: | Нет. |
| Постусловия: | Состояние процессора не изменяется. |
|   |   |
| ***ЧитатьСостояние*** |   |
| Вход: | Нет. |
| Предусловия: | Нет. |
| Процесс: | Копирует и возвращает значение поля Operation. |
| Выход: | Значение поля Operation. |
| Постусловия: | Состояние процессора не изменяется. |
|   |   |
| ***ЗаписатьСостояние*** |   |
| Вход: | Переменная (Oprtn) типа TOprtn. |
| Предусловия: | Нет. |
| Процесс: | Заносит значение Oprtn в поле Operation. |
| Выход: | Нет. |
| Постусловия: | Состояние процессора изменяется на Oprtn. |
|   |   |
| ***ЧитатьОшибку*** |   |
| Вход: | Нет. |
| Предусловия: | Нет. |
| Процесс: | Копирует и возвращает значение поля Error. |
| Выход: | Значение поля Error. |
| Постусловия: | Состояние процессора не изменяется. |
|   |   |
| ***СбросОшибки*** |   |
| Вход: | Нет. |
| Предусловия: | Нет. |
| Процесс: | Очищает поле Error. |
| Выход: | Нет. |
| Постусловия: | Состояние процессора не изменяется. |

Рекомендации к выполнению

1. Тип данных TProc реализовать, используя класс Object Pascal*,*С++.
2. Число храните как поле типа TPNumber (см. лаб. Раб. Абстрактный тип данных - Р-ичные числа).
3. Для чтения состояния процессора, полей: “левый операнд-результат” (Lop\_Res), “правый операнд” (Rop), “ошибка” ()Error используйте свойство (property).
4. Тип данных реализовать в отдельном модуле UProc.
5. В приведённой ниже таблице показана последовательность изменения состояния процессора при вычислении выражения:

2 + 3 \* 42

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Шаг | Вход | Метод | Rop | Lop\_Res | Operation |
| 0 |   | Конструктор | 0 | 0 | None |
| 1 | 2 |   | 0 | 0 | None |
| 2 | + | Lop\_Res\_Set;OprtnSet | 0 | 2 | Add |
| 3 | 3 |   | 0 | 2 | Add |
| 4 | \* | Rop\_Set;OprtnRun;OprtnSet; | 3 | 2+3 | Mul |
| 5 | 4 |   | 4 | 2+3 | Mul |
| 6 | Sqr | Rop\_Set;FuncRun | 4^2 | 2+3 | Mul |
| 7 | = | OprtnRun | 4^2 | 2+3\* 4^2 | Mul |
| 8 | C | ReSet | 0 | 0 | None |

Содержание отчета

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