

Практическая работа. Абстрактный тип данных (ADT) p -ичное число

Тема: Классы Object Pascal, C++

Цель: Сформировать практические навыки: реализации абстрактного типа данных с помощью классов C++.

Задание

1. Реализовать абстрактный тип данных « p -ичное число», используя класс, в соответствии с приведенной ниже спецификацией.
2. Протестировать каждую операцию, определенную на типе данных, одним из методов тестирования.
3. Протестировать тип данных в целом.

Спецификация типа данных « p -ичное число».

ADT TPNumber**Данные**

P -ичное число TPNumber - это действительное число (n) со знаком в системе счисления с основанием (b) (в диапазоне 2..16), содержащее целую и дробную части. Точность представления числа – ($c \geq 0$). P -ичные числа изменяемые.

Операции

Операции могут вызываться только объектом p -ичное число (тип TPNumber), указатель на который в них передаётся по умолчанию. При описании операций этот объект называется «само число».

Конструктор Число

Начальные значения:	Вещественное число (a) во внутреннем формате, система счисления (b), точность представления числа (c)
Процесс:	Создаёт p -ичное число: система счисления (b), точность представления (c). В поле (n) созданного числа заносится (a).

Например:

NCreate($a, 3, 3$) = число a в системе счисления 3 с тремя разрядами после троичной точки.

NCreate($a, 3, 2$) = число a в системе счисления 3 с двумя разрядами после троичной точки.

КонструкторСтрока

Начальные значения:	Строковое представление p -ичного числа (a), система счисления (b), точность представления числа (c)
Процесс:	Создаёт p -ичное число: система счисления (b), точность представления (c). В поле (n) созданного числа заносится результат преобразования строки (a) в числовое представление. b -ичное число (a) и основание системы счисления (b) представлены в формате строки.

Например:

$S\text{Create}('20', '3', '6') = 20$ в системе счисления 3, точность 6 знаков после запятой.

$S\text{Create}('0', '3', '8') = 0$ в системе счисления 3, точность 8 знаков после запятой.

Копировать:

Вход:	Нет.
Предусловия:	Нет.
Процесс:	Создаёт копию самого числа (тип <code>TPNumber</code>).
Выход:	p -ичное число.
Постусловия:	Нет.

Сложить

Вход:	p -ичное число d с основанием и точностью такими же, как у самого числа.
Предусловия:	Нет.
Процесс:	Создаёт и возвращает p -ичное число (тип <code>TPNumber</code>), полученное сложением полей (n) самого числа и числа d .
Выход:	p -ичное число.
Постусловия:	Нет

Умножить

Вход:	p -ичное число d с основанием и точностью такими же, как у самого числа.
Предусловия:	Нет.
Процесс:	Создаёт и возвращает p -ичное число (тип <code>TPNumber</code>), полученное умножением полей (n) самого числа и числа d .
Выход:	p -ичное число (тип <code>TPNumber</code>).
Постусловия:	Нет.

Вычитать

Вход:	p -ичное число d с основанием и точностью такими же, как у самого числа.
Предусловия:	Нет.
Процесс:	Создаёт и возвращает p -ичное число (тип <code>TPNumber</code>), полученное вычитанием полей (n) самого числа и числа d .
Выход:	p -ичное число (тип <code>TPNumber</code>).
Постусловия:	Нет.

Делить

Вход:	p -ичное число d с основанием и точностью такими же, как у самого числа.
Предусловия:	Поле (n) числа (d) не равно 0.
Процесс:	Создаёт и возвращает p -ичное число (тип <code>TPNumber</code>), полученное делением полей (n) самого числа на поле (n) числа d .
Выход:	p -ичное число (тип <code>TPNumber</code>).
Постусловия:	Нет.

Обратить

Вход:	Нет.
Предусловия:	Поле (n) самого числа не равно 0.
Процесс:	Создаёт p -ичное число, в поле (n) которого заносится значение, полученное как $1/(n)$ самого

	числа.
Выход:	P -ичное число (тип <code>TPNumber</code>).
Постусловия:	Нет.

Квадрат

Вход:	Нет.
Предусловия:	Нет.
Процесс:	Создаёт p -ичное число, в поле (n) которого заносится значение, полученное как квадрат поля (n) самого числа.
Выход:	P -ичное число (тип <code>TPNumber</code>).
Постусловия:	Нет.

Взять P Число

Вход:	Нет.
Предусловия:	Нет.
Процесс:	Возвращает значение поля (n) самого числа.
Выход:	Вещественное значение.
Постусловия:	Нет.

Взять P Строка

Вход:	Нет.
Предусловия:	Нет.
Процесс:	Возвращает p -ичное число (q) в формате строки, изображающей значение поля (n) самого числа в системе счисления (b) с точностью (c).
Выход:	Строка.
Постусловия:	Нет.

ВзятьОснованиеЧисло

Вход:	Нет.
Предусловия:	Нет.
Процесс:	Возвращает значение поля (b) самого числа (q).
Выход:	Целочисленное значение
Постусловия:	Нет.

ВзятьОснованиеСтрока

Вход:	Нет.
Предусловия:	Нет.
Процесс:	Возвращает значение поля (b) самого числа в формате строки, изображающей (b) в десятичной системе счисления.
Выход:	Строка.
Постусловия:	Нет.

ВзятьТочностьЧисло

Вход:	Нет.
Предусловия:	Нет.
Процесс:	Возвращает значение поля (c) самого числа .
Выход:	Целое значение.
Постусловия:	Нет.

ВзятьТочностьСтрока

Вход:	Нет.
Предусловия:	Нет.
Процесс:	Возвращает значение поля (c) самого числа в формате строки, изображающей (c) в десятичной системе счисления.
Выход:	Строка.
Постусловия:	Нет.

УстановитьОснованиеЧисло

Вход:	Целое число (newb).
Предусловия:	$2 \leq \text{newb} \leq 16$.
Процесс:	Устанавливает в поле (b) самого числа значение (newb).
Выход:	Нет.
Постусловия:	Нет.

УстановитьОснованиеСтрока

Вход:	Строка (bs), изображающая основание (b) р-ичного числа в десятичной системе счисления.
-------	--

Предусловия:	Допустимый диапазон числа, изображаемого строкой (bs) - 2,,16.
Процесс:	Устанавливает значение поля (b) самого числа значением, полученным в результате преобразования строки (bs).
Выход:	Строка.
Постусловия:	Нет.

Установить Точность Число

Вход:	Целое число (newc).
Предусловия:	$newc \geq 0$.
Процесс:	Устанавливает в поле (c) самого числа значение (newc).
Выход:	Нет.
Постусловия:	Нет.

Установить Точность Строка

Вход:	Строка (newc).
Предусловия:	Строка (newc) изображает десятичное целое ≥ 0 .
Процесс:	Устанавливает в поле (c) самого числа значение, полученное преобразованием строки (newc).
Выход:	Нет.
Постусловия:	Нет.

end TPNumber**Рекомендации к выполнению**

1. Тип данных реализовать, используя класс C++.
2. Число храните как поле вещественного типа.
3. Основание системы счисления храните как поле целочисленного типа.
4. Для чтения и записи р - ичного числа в вещественном и строковом формате используйте свойства (property).
5. Тип данных реализовать в отдельном модуле UPNumber в режиме консольного приложения.

Содержание отчета

1. Задание.
2. Текст программы.

3. Тестовые наборы данных для тестирования типа данных.

Контрольные вопросы

1. Что такое инкапсуляция?
2. Как синтаксически представлено поле в описании класса?
3. Как синтаксически представлен метод в описании класса?
4. Как синтаксически представлено простое свойство в описании класса?
5. Особенности описания методов класса?
6. Особенности описания и назначение конструктора класса?
7. Видимость идентификаторов в описании класса?
8. Особенности вызова методов применительно к объектам класса?