

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

Тема: Классы Object Pascal, C++

Цель: Сформировать практические навыки реализации абстрактных типов данных в соответствии с заданной спецификацией с помощью классов C++ и их модульного тестирования.

Задание

1. Реализовать абстрактный тип данных «р-ичное число», используя класс, в соответствии с приведенной ниже спецификацией.
2. Протестировать каждую операцию, определенную на типе данных, используя средства модульного тестирования.
3. Если необходимо, предусмотрите возбуждение исключительных ситуаций.

Спецификация типа данных «р-ичное число».

ADT TPNumber**Данные**

Р-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 – (с ≥ 0). Р-ичные числа изменяемые.

Операции

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

КонструкторЧисло	
Вход:	Вещественное число (a). Система счисления (b), точность представления числа (c) – целые числа.
Предусловия:	$2 \leq b \leq 16$, $c \geq 0$. В противном случае

	возбуждается исключение.
Процесс:	<p>Создаёт r-ичное число: система счисления (b), точность представления (c). В поле (n) созданного числа заносится (a).</p> <p>Например:</p> <p>$NCreate(a,3,3)$ = число a в системе счисления 3 с тремя разрядами после троичной точки.</p> <p>$NCreate(a,3,2)$ = число a в системе счисления 3 с двумя разрядами после троичной точки.</p>
Постусловия:	Поля объекта проинициализированы начальными значениями.
КонструкторСтрока	
Вход:	Строковое представление r -ичного числа (a), системы счисления (b), точности представления числа (c)
Предусловия:	$2 \leq b \leq 16, c \geq 0$. В противном случае возбуждается исключение.
Процесс:	<p>Создаёт r-ичное число: система счисления (b), точность представления (c). В поле (n) созданного числа заносится результат преобразования строки (a) в числовое представление.</p> <p>b-ичное число (a) и основание системы счисления (b) представлены в формате</p>

	<p>строки.</p> <p>Например:</p> <p>$S\text{Create}('20', '3', '6') = 20$ в системе счисления 3, точность 6 знаков после запятой.</p> <p>$S\text{Create}('0', '3', '8') = 0$ в системе счисления 3, точность 8 знаков после запятой.</p>
Постусловия:	Поля объекта проинициализированы начальными значениями.
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:	р-ичное число.
Постусловия:	Нет
Умножить	
Вход:	Р-ичное число d с основанием и точностью такими же, как у самого числа.
Предусловия:	Нет.
Процесс:	Создаёт и возвращает р-ичное число (тип TPNumber), полученное умножением полей (n) самого числа и числа d .
Выход:	Р-ичное число (тип TPNumber).
Постусловия:	Нет.
Вычесть	
Вход:	Р-ичное число d с основанием и точностью такими же, как у самого числа.
Предусловия:	Нет.
Процесс:	Создаёт и возвращает р-ичное число (тип TPNumber), полученное вычитанием полей (n) самого числа и числа d .
Выход:	Р-ичное число (тип TPNumber).
Постусловия:	Нет.
Делить	
Вход:	Р-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 TPNumber), полученное делением полей (n) самого числа на поле (n) числа d.
Выход:	P -ичное число (тип TPNumber).
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 TPNumber).
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 TPNumber).
Постусловия:	Нет.

<i>ВзятьРЧисло</i>	
Вход:	Нет.
Предусловия:	Нет.
Процесс:	Возвращает значение поля (n) самого числа.
Выход:	Вещественное значение.
Постусловия:	Нет.
<i>ВзятьРСтрока</i>	
Вход:	Нет.
Предусловия:	Нет.
Процесс:	Возвращает р-ичное число (q) в формате строки, изображающей значение поля (n) самого числа в системе счисления (b) с точностью (c).
Выход:	Строка.
Постусловия:	Нет.
<i>ВзятьОснованиеЧисло</i>	
Вход:	Нет.
Предусловия:	Нет.
Процесс:	Возвращает значение поля (b) самого числа (q).
Выход:	Целочисленное значение
Постусловия:	Нет.
<i>ВзятьОснованиеСтрока</i>	
Вход:	Нет.
Предусловия:	Нет.

Процесс:	Возвращает значение поля (b) самого числа в формате строки, изображающей (b) в десятичной системе счисления.
Выход:	Строка.
Постусловия:	Нет.
<i>Взять Точность Число</i>	
Вход:	Нет.
Предусловия:	Нет.
Процесс:	Возвращает значение поля (c) самого числа .
Выход:	Целое значение.
Постусловия:	Нет.
<i>Взять Точность Строка</i>	
Вход:	Нет.
Предусловия:	Нет.
Процесс:	Возвращает значение поля (c) самого числа в формате строки, изображающей (c) в десятичной системе счисления.
Выход:	Строка.
Постусловия:	Нет.
<i>Установить Основание Число</i>	
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.
УстановитьТочностьЧисло	
Вход:	Целое число (newc).
Предусловия:	$\text{newc} \geq 0$.
Процесс:	Устанавливает в поле (c) самого числа значение (newc).
Выход:	Нет.
Постусловия:	Нет.
УстановитьТочностьСтрока	
Вход:	Строка (newc).
Предусловия:	Строка (newc) изображает десятичное целое ≥ 0 .

Процесс:	Устанавливает в поле (с) 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 и точность представления дробной части храните как поля целочисленного типа.
4. Для возбуждения исключений опишите классы исключительных ситуаций, как наследников библиотечного класса.
5. Для тестирования используйте проект модульного теста.
6. 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.
4. Результаты тестирования.

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?