

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

Тема: Классы Object Pascal, C++

*Цель: Сформировать практические навыки: реализации абстрактного типа данных 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 по критерию С2, используя средства модульного тестирования Visual Studio.
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  $\geq$  0). Р-ичные числа неизменяемые.

**Операции**

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

<b>Конструктор Число</b>	
Вход:	Вещественное число (a), основание системы счисления (b), точность представления числа (c)
Предусловия:	Основание системы счисления (b) должно принадлежать интервалу [2..16], точность представления числа c $\geq$ 0.

Процесс:	<p>Инициализирует поля объекта <code>this</code> <math>p</math>-ичное число: система счисления (b), точность представления (c). В поле (n) числа заносится (a).</p> <p>Например:</p> <p><math>TPNumber(a,3,3)</math> = число a в системе счисления 3 с тремя разрядами после троичной точки.</p> <p><math>TPNumber(a,3,2)</math> = число a в системе счисления 3 с двумя разрядами после троичной точки.</p>
Постусловия:	Объект инициализирован начальными значениями.
Выход:	Нет.
<b>Конструктор Строка</b>	
Вход:	Строковые представления: $p$ -ичного числа (a), основания системы счисления (b), точности представления числа (c)
Предусловия:	Основание системы счисления (b) должно принадлежать интервалу $[2..16]$ , точность представления числа $c \geq 0$ .
Процесс:	<p>Инициализирует поля объекта <code>this</code> <math>p</math>-ичное число: основание системы счисления (b), точностью представления (c). В поле (n) числа <code>this</code> заносится результат преобразования строки (a) в числовое представление. <math>b</math>-ичное число (a) и основание системы счисления (b) представлены в формате строки.</p> <p>Например:</p> <p><math>TPNumber("20","3","6") = 20</math> в системе счисления 3, точность 6 знаков после запятой.</p>

	TPNumber ("0","3","8") = 0 в системе счисления 3, точность 8 знаков после запятой.
Постусловия:	Объект инициализирован начальными значениями.
Выход:	Нет.
<b>Копировать:</b>	
Вход:	Нет.
Предусловия:	Нет.
Процесс:	Создаёт копию самого числа this (тип TPNumber).
Выход:	р-ичное число.
Постусловия:	Нет.
<b>Сложить</b>	
Вход:	Р-ичное число d с основанием и точностью такими же, как у самого числа this.
Предусловия:	Нет.
Процесс:	Создаёт и возвращает р-ичное число (тип TPNumber), полученное сложением полей (n) самого числа this и числа d.
Выход:	р-ичное число.
Постусловия:	Нет
<b>Умножить</b>	
Вход:	Р-ичное число d с основанием и точностью такими же, как у самого числа this.
Предусловия:	Нет.
Процесс:	Создаёт и возвращает р-ичное число (тип TPNumber), полученное умножением полей (n)

	самого числа <code>this</code> и числа <code>d</code> .
Выход:	$p$ -ичное число (тип <code>TPNumber</code> ).
Постусловия:	Нет.
<b>Вычитать</b>	
Вход:	$p$ -ичное число <code>d</code> с основанием и точностью такими же, как у самого числа <code>this</code> .
Предусловия:	Нет.
Процесс:	Создаёт и возвращает $p$ -ичное число (тип <code>TPNumber</code> ), полученное вычитанием полей ( <code>n</code> ) самого числа <code>this</code> и числа <code>d</code> .
Выход:	$p$ -ичное число (тип <code>TPNumber</code> ).
Постусловия:	Нет.
<b>Делить</b>	
Вход:	$p$ -ичное число <code>d</code> с основанием и точностью такими же, как у самого числа.
Предусловия:	Поле ( <code>n</code> ) числа ( <code>d</code> ) не равно 0.
Процесс:	Создаёт и возвращает $p$ -ичное число (тип <code>TPNumber</code> ), полученное делением полей ( <code>n</code> ) самого числа <code>this</code> на поле ( <code>n</code> ) числа <code>d</code> .
Выход:	$p$ -ичное число (тип <code>TPNumber</code> ).
Постусловия:	Нет.
<b>Обратить</b>	
Вход:	Нет.
Предусловия:	Поле ( <code>n</code> ) самого числа не равно 0.
Процесс:	Создаёт $p$ -ичное число, в поле ( <code>n</code> ) которого заносится значение, полученное как $1/(n)$ самого

	числа this.
Выход:	Р-ичное число (тип TPNumber).
Постусловия:	Нет.
<b>Квадрат</b>	
Вход:	Нет.
Предусловия:	Нет.
Процесс:	Создаёт р-ичное число, в поле (n) которого заносится значение, полученное как квадрат поля (n) самого числа this.
Выход:	Р-ичное число (тип TPNumber).
Постусловия:	Нет.
<b>ВзятьРЧисло</b>	
Вход:	Нет.
Предусловия:	Нет.
Процесс:	Возвращает значение поля (n) самого числа this.
Выход:	Вещественное значение.
Постусловия:	Нет.
<b>ВзятьРСтрока</b>	
Вход:	Нет.
Предусловия:	Нет.
Процесс:	Возвращает р-ичное число (q) в формате строки, изображающей значение поля (n) самого числа this в системе счисления (b) с точностью (c).
Выход:	Строка.
Постусловия:	Нет.

<b><i>ВзятьОснованиеЧисло</i></b>	
Вход:	Нет.
Предусловия:	Нет.
Процесс:	Возвращает значение поля (b) самого числа this
Выход:	Целочисленное значение
Постусловия:	Нет.
<b><i>ВзятьОснованиеСтрока</i></b>	
Вход:	Нет.
Предусловия:	Нет.
Процесс:	Возвращает значение поля (b) самого числа this в формате строки, изображающей (b) в десятичной системе счисления.
Выход:	Строка.
Постусловия:	Нет.
<b><i>ВзятьТочностьЧисло</i></b>	
Вход:	Нет.
Предусловия:	Нет.
Процесс:	Возвращает значение поля (c) самого числа this.
Выход:	Целое значение.
Постусловия:	Нет.
<b><i>ВзятьТочностьСтрока</i></b>	
Вход:	Нет.
Предусловия:	Нет.
Процесс:	Возвращает значение поля (c) самого числа this в формате строки, изображающей (c) в десятичной системе счисления.

Выход:	Строка.
Постусловия:	Нет.
<b>УстановитьОснованиеЧисло</b>	
Вход:	Целое число (newb).
Предусловия:	$2 \leq \text{newb} \leq 16$ .
Процесс:	Устанавливает в поле (b) самого числа this значение (newb).
Выход:	Нет.
Постусловия:	Нет.
<b>УстановитьОснованиеСтрока</b>	
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 this значением, полученным в результате преобразования строки (bs).
Выход:	Строка.
Постусловия:	Нет.
<b>УстановитьТочностьЧисло</b>	
Вход:	Целое число (newc).
Предусловия:	$\text{newc} \geq 0$ .
Процесс:	Устанавливает в поле (c) самого числа значение (newc).
Выход:	Нет.
Постусловия:	Нет.

<b>Установить Точность Строка</b>	
Вход:	Строка (newc).
Предусловия:	Строка (newc) изображает десятичное целое $\geq 0$ .
Процесс:	Устанавливает в поле (с) самого числа this значение, полученное преобразованием строки (newc).
Выход:	Нет.
Постусловия:	Нет.

**end TPNumber**

#### Рекомендации к выполнению

1. Тип данных реализовать, используя класс C++ в составе проекта Консольное приложение.
2. Для возбуждения исключений опишите класс исключительных ситуаций, как наследника библиотечного класса.
3. Для тестирования используйте проект модульного теста.
4. Тип данных реализовать в отдельном классе в составе 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?