

Задание 2 для первого семестра 2 курса

Арифметика. Задача — построить класс, представляющий собой специальное число, и потом проверить некоторые вычислительные алгоритмы с такими числами. Для начала — простейшие арифметические операции, потом что-то более сложное — суммирование рядов, вычисление интегралов, решение уравнений и т.п. Функция печати этого числа в соответствующей наглядной форме.

Задачи этого типа могут оказаться разной сложности. В случае сложных задач надо вовремя остановиться. То есть, построить реализацию, которая может решать некоторые содержательные частные задачи, но не обязательно обладает полной функциональностью. В списке ниже более сложные задачи отмечены звездочкой.

На лекциях разбирался пример

Задание 2.0. Арифметика с оценкой абсолютной погрешности. Число есть значение и модуль его абсолютной погрешности. Это тот самый Number!

Задание 2.1. Комплексная арифметика. Обычное комплексное число — действительная и мнимая части. Реализация операций, элементарных функций. Тест, например, решение квадратного уравнения, вычисление элементарных функций через ряды и т.п.

Задание 2.2. Рациональная арифметика. Число представляется как целочисленная пара — числитель и знаменатель. Арифметические операции, сокращение общего множителя, контроль переполнения знаменателя. Взаимодействие с вещественными числами (приближение).

Длинная целая арифметика. Реализация вычислений с целыми числами произвольной величины. Тут можно предложить несколько способов, основанных на общей идее динамического массива, т.е. массива, которые может менять свою длину при необходимости.

Задание 2.3.* Длинная целая арифметика. Число есть `char` массив (элемент — одна десятичная цифра).

Задание 2.4.* Длинная целая арифметика. Число есть битовый массив, реализуемый через массив целых. Можно ограничиться шестнадцатиричной печатью.

Задание 2.5.* Длинная целая арифметика. Число есть полубайтовый массив, где каждый полубайт представляет десятичную цифру.

Задание 2.6.* Длинная целая арифметика. Число есть массив `int`, где каждый элемент отвечает за несколько десятичных знаков числа (например, 9, т.е. содержит число от 0 до 999999999 или типа того).

Длинная вещественная арифметика. По сути не отличается от целой арифметики, но каждое число хранит номер позиции десятичной точки, и сложение-вычитание подразумевает выравнивание.

Задание 2.7.* Длинная вещественная арифметика. Число есть `char` массив (элемент — одна десятичная цифра).

Задание 2.8.* Длинная вещественная арифметика. Число есть полубайтовый массив, где каждый полубайт представляет десятичную цифру.

Задание 2.9.* Длинная целая арифметика. Число есть массив `int`, где каждый элемент отвечает за несколько десятичных знаков числа (например, 9, т.е. содержит число от 0 до 999999999 или типа того).

Задание 2.10.* Вещественная арифметика с фиксированной точкой. Число — это пара целых чисел (целая и дробная части).

Задание 2.11.* Вещественная арифметика с фиксированной точкой. Число — это целое число, в котором часть разрядов отведена под целую часть, а часть под дробную.

Задание 2.12. Интервальная арифметика. Число — это интервал, которому может принадлежать текущее значение (`min`, `max`). Соответственно результат любой операции — это тоже интервал для максимального и минимального возможных значений.

Задание 2.13. Арифметика с оценкой относительной погрешности. Число есть значение и его относительная погрешность. Арифметические операции и элементарные функции. Тесты — любые вычислительные процедуры и сравнение с результатами в `double` или `float`.

Задание 2.14.* Арифметика с производной. Число есть пара чисел, которые интерпретируются как значение некоторой функции и ее производной. Соответственно результат операций для производной вычисляется по формулам производной суммы, разности, произведения, частного. Выполняя некоторый алгоритм

с такими числами (например, суммируя ряд для синуса), мы должны одновременно получить и значение производной (т.е. значение косинуса). Есть некоторые тонкости при взаимодействии с обычными числами.

Задание 2.15. Алгебра многочленов. Сложение, вычитание, умножение, деление (с остатком). Интересно соединить с комплексными числами. Тест — нахождение корней. Действительные — численным поиском, комплексные — делением с остатком и выделением квадратичных сомножителей.

Внимание!

Некоторые задачи весьма сложны для полной реализации. Поэтому не старайтесь из сразу решать в полном объеме, а уделите существенное внимание именно разработке структуры ваших классов. Иначе может оказаться, что неверные решения, принятые в начале проектирования, не дадут вам возможность реализовать какие-то требования.

При реализации нужно использовать перегрузку операторов, разнообразные конструкторы, в частности, операторы присваивания и сравнения и конструкторы копирования.

Следует предусмотреть возможность проверки и обработки ситуаций, когда операция с объектами не может быть выполнена или не дает корректного результата (например, деление на 0).