

Задача 3: перегрузка операторов

Bookmark this page

Вариант 2

1. Разработать класс "шестнадцатеричное число" в соответствии со следующим заданием:

Состояние класса -

Шестнадцатеричное число определяется его длиной (количество цифр в числе, например, до 31) и массивом значений шестнадцатеричных цифр (типа char). Число со знаком хранится в прямом коде. При выполнении арифметических операций использовать дополнительный код. Память под массив выделяется статически, во время компиляции, и задается массивом фиксированного предель-ного размера.

Протокол класса -

Определяет возможности создания и инициализации экземпляров класса и правила их исполь-зования (методы класса).

Предусмотреть следующие возможности:

- пустой конструктор для инициализации экземпляров и массивов экземпляров класса по умолчанию;
- создание экземпляров класса с инициализацией шестнадцатеричной константой;
- создание экземпляров класса с инициализацией значением числа как строки символов;
- ввод экземпляров класса из входного потока и вывод их значений в выходной поток

(с помощью перегруженных операторов >> и <<);

- выполнение операций сложения и вычитания шестнадцатеричных чисел (с помощью пере-груженных операторов + и -);
- выполнение операции сдвига числа на заданное количество шестнадцатеричных цифр вле-во с дополнением числа нулями справа (с помощью перегруженного оператора <<=);
- выполнение операции сдвига числа на заданное количество шестнадцатеричных цифр вправо с усечением числа справа (с помощью перегруженного оператора >>=);
- выполнение операции сравнения шестнадцатеричных чисел;
- выполнение операции “проверить четность числа”.

2. Проектирование класса рекомендуется начать с представления состояния класса, учитываю-щего заданные операции, а затем реализации конструкторов и перегруженного оператора вывода. Для отладки и исчерпывающего тестирования других методов разработанного класса реализовать диалоговую программу, которая позволяет вводить параметры, отлаживаемых методов. Для обра-ботки ошибочных ситуаций использовать механизм исключительных ситуаций.

3. Повторить разработку класса при условии, что память под массив необходимой длины вы-деляется динамически, во время выполнения программы (с помощью оператора new; память задает-ся указателем на char в состоянии класса).

Дополнить интерфейс класса следующими возможностями:

- создание экземпляра класса с его инициализацией другим экземпляром класса (копирую-щий конструктор);
- переопределение экземпляра класса (с помощью перегруженного оператора присваива-ния).

4. Написать прикладную программу, использующую разработанный класс.