

< Previous				Next >
-------------------------------	---	--	---	---------------------------

Задача 3: перегрузка операторов

Bookmark this page

Вариант 29

1. Разработать класс "множество символов" в соответствии со следующим заданием:

Состояние класса -

Множество состоит из элементов типа char (в множестве не может быть двух элементов с оди-наковыми значениями) и определяется мощностью (количеством элементов в множестве) и значени-ями своих элементов, например: множество из 6 элементов - строка abcdef. Память под множество выделяется статически, во время компиляции, и задается массивом фиксированного предельного значения (например, максимальная мощность множества 100 элементов).

Протокол класса -

Определяет возможности создания и инициализации экземпляров класса и правила использо-вания их (методы класса).

Предусмотреть следующие возможности:

- пустой конструктор для инициализации экземпляров и массивов экземпляров класса по умолчанию;
- создание экземпляров класса с инициализацией мощностью множества (начальное множе-ство - символы, начиная с символа пробел);
- создание экземпляров класса с инициализацией значениями элементов множества как стро-ки символов;
- ввод экземпляров класса из входного потока и вывод их значений в выходной поток

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

- объединение двух множеств (с помощью перегруженного оператора сложения +; резуль-тирующее множество должно содержать элементы множества 1 и отличающиеся элемен-ты множества 2);
- добавление нового элемента в множество (с помощью перегруженного оператора +=);
- вычисление пересечения двух множеств (с помощью перегруженного оператора умноже-ния *; результирующее множество должно состоять из элементов, имеющихя и в множе-стве 1, и в множестве 2);
- вычисление разности двух множеств (с помощью перегруженного оператора вычитания -; результирующее множество должно состоять из элементов первого множества без тех, которые есть во втором множестве);
- определение, имеется ли некоторый заданный элемент в множестве.

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 множество выделяется необходи-мой длины динамически, во время выполнения программы (с помощью оператора new; память задается 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.

