Задание №1 (8 баллов). Множества.

1. Создать абстрактный класс Set для работы с множествами. Элементы множеств – натуральные числа. Класс должен содержать следующие методы:

* Абстрактные:
  1. включение элемента в множество;
  2. исключение элемента из множества;
  3. проверка наличия элемента в множестве.
* Реализованные:
  1. два перегруженных метода заполнения: один – для заполнения из строки элементов (с параметром типа string), второй – для заполнения из массива элементов (с параметром типа int[]);

*Примечание 1. Этот массив элементов может быть не отсортированным и содержать повторяющиеся элементы.*

* 1. метод вывода (печати) всех элементов множества на экране.

Реализованные методы **не могут** учитывать внутреннюю организацию множества, вместо этого должны использоваться имеющиеся абстрактные методы.

1. Создать класс SimpleSet, наследующий от класса Set – реализация работы с множествами на основе логического массива (m[i]=true ⇔ элемент i принадлежит множеству). Описать конструктор с 1 параметром – максимальным числом, представимым в множестве.
2. В классе SimpleSet описать операторный метод «+», оба параметра типа SimpleSet, выполняющий объединение множеств. Аналогично – операторный метод «\*», выполняющий пересечение множеств.
3. Создать класс BitSet, наследующий от класса Set – реализация работы с множествами на основе битового массива (32 элемента «упаковываются» в 1 ячейку типа int). Описать конструктор и операторные методы, перечисленные в п. 3.
4. Создать класс MultiSet, наследующий от класса Set – реализация работы с мультимножествами на основе целочисленного массива (m[i] = количеству вхождений элемента i в мультимножество, в том числе если m[i] = 0, то элемент в мультимножестве отсутствует). Описать конструктор с 1 параметром – максимальным числом, представимым в множестве.

*Примечание 2. При проверке наличия элемента в множестве и печати всех элементов количество вхождений не важно (важно только есть ли хотя бы одно вхождение).*

1. Создать собственное исключение – выход за пределы множества. Генерировать это исключение, когда в множество пытаются добавить элемент, превышающий максимально допустимое значение.
2. В основной программе предусмотреть диалог с пользователем.
   1. Пользователь выбирает вариант представления (логический или битовый массив либо мультимножество).
   2. Исходное множество вводится либо с клавиатуры в виде одной строки элементов (используется метод заполнения с параметром типа string), либо из файла, в котором каждый элемент располагается в отдельной строке (используется метод заполнения с параметром типа int[]).
   3. Далее пользователь может добавлять/исключать элементы и проверять наличие элемента в множестве до тех пор, пока не выберет пункт «Выход». При добавлении элемента обрабатывать возможное исключение.

**Для выполнения указанных действий программный код не должен дублироваться (для разных вариантов представления)!**

*Примечание: можно реализовать как графический интерфейс, так и режим командной строки.*

1. Для тестирования операторных методов (пункт 3) предусмотреть отдельный тест. С клавиатуры вводятся два множества А и В (в виде строк элементов). Для каждого из двух вариантов представления (логический или битовый массив) создаются множества С = А ∪ В и D = А ∩ В и выводятся на экран.

Оценивание.

Пункты 1, 2, 5, 6, 7 – 5 баллов

пункт 4 – +2 балла

пункты 3, 8 – +1 балл

Оценка может быть снижена за ошибки в программе, неэффективную реализацию, неэффективное использование памяти, плохо структурированный код, дублирование кода, невыполнение указанных в задании требований.

Положительная оценка выставляется только в случае, если на защите работы студент правильно отвечает на вопросы по тексту программы, способен пояснить назначение отдельных фрагментов программы, внести небольшие изменения в программу.