**Задание 1.**

Создать класс «Одномерный массив», в котором описать следующие элементы:

- закрытое поле – массив целых чисел,
- свойство для определения длины массива,
- индексатор для доступа к элементам поля-массива,
- конструктор с параметрами,
- метод ToString(),
- статический метод с переменным числом параметров для вычисления общей суммы отрицательных элементов в нескольких массивах,
- операции умножения массива на целое число и числа на массив,
- унарная операция - (знаки элементов меняются на противоположные)
- операция неявного преобразования строки в объект этого класса (Например, строка вида «1 2 -4 3 -2» преобразуется в объект класса Одномерный массив с полем-массивом (1 2 -4 3 -2) ). При попытке неявного преобразования строки неправильного формата должно генерироваться исключение.

Разработать программу, выполняющую следующие действия:

- Ввод и вывод трех массивов A, B и C;
- Вычисление общей суммы отрицательных элементов в массивах 5*A и C;
- Вычисление общей суммы отрицательных элементов в массивах 2*B, -A и C*4;
- Если сумма отрицательных элементов в массиве –A больше суммы отрицательных элементов в массиве A, заменить все отрицательные повторяющиеся элементы этого массива на значение этой суммы.

**Задание 2 (было в EPAM).**

Разработать *базовый класс*, определяющий покупку товара:

**Поля:**

- название товара;
- цена в рублях;
- кол-во единиц товара.

**Конструкторы:**

- по умолчанию;
- с параметрами.

**Методы:**

- установки/считывания полей;
- GetCost( ) – вычисляет стоимость покупки;
- ToString( ) – переводит объект в строку с разделителями «;»;
- Equals( ) – сравнивает две продажи (считаются равными, если совпадают название и цена).

Разработать *первый производный класс*, для покупки товара с фиксированной скидкой от цены и переопределить нужные методы (GetCost( ) и ToString( ).

Разработать *второй производный класс* (от базового) со скидкой к цене, если количество единиц товара не меньше некоторой константы подкласса. Переопределить нужные методы.

Разработать консольное приложение, выполняющее следующее:

1 Создать массив из шести объектов (2 – базового класса, по 2 – каждого производного класса).

2 Вывести массив на консоль.

3 Вывести покупку с максимальной стоимостью.

4 Определить, являются ли все покупки равными.

Задачи 2–4 реализовать в одном общем цикле.

**Задание 3.**

Если в квадратной матрице **A** сумма элементов столбцов, состоящих из положительных элементов, больше чем такая же сумма в квадратной матрице **B**, заменить все нулевые элементы матрицы **B** на значение суммы элементов диагоналей этой матрицы. В противном случае определить сумму элементов диагоналей матрицы **A**. При создании объектов класса матрицы-аргументы конструктора создавать с использованием синтаксиса инициализаторов. С клавиатуры не вводить.

Для решения задачи создать класс **Matrix**, содержащий

- закрытое поле-массив для хранения данных,
- конструктор без параметров для создания единичной матрицы 3×3,
- конструктор с параметрами (параметр – матрица целых чисел),
- метод **ToString()**, возвращающий строковое представление матрицы,
- индексатор для доступа к элементам поля-массива,
- метод **GetLenth** – аналог одноименного метода из Array,
- закрытый (private) метод, возвращающий true, если столбец состоит из положительных элементов (параметр – номер столбца),
- метод, возвращающий сумму элементов столбцов, состоящих из положительных элементов,
- свойство, возвращающее сумму элементов диагоналей матрицы.