## Теоретическая часть

* Перегрузка унарных операций
* Особенности перегрузки префиксного и постфиксного инкремента и декремента
* Понятие о классах-контейнерах
* Перегрузка операции [] для доступа к элементам массива в классе-контейнере
* Перегрузка операции () для передачи нескольких параметров объекту класса

## Практическая часть

С использованием шаблона **OopConsoleApplication.zip**, приведенного в папке CWразработайте, пожалуйста, приложение по следующему заданию.

Напишите консольное многофайловое приложение с использованием ООП по следующему заданию.

**Задача 1.** Создайте класс, описывающий усеченный конус **Conoid** (радиус верхнего основания, радиус нижнего основания, высота – тип полей **double**). Реализуйтеконструкторы (только два: по умолчанию и инициирующий – т.е. задающий все поля класса), геттеры и сеттеры (при некорректных значениях выбрасывать в сеттерах исключение **exception**), методы для вычисления [**площади**](https://www-formula.ru/2011-09-21-04-35-14) и [**объема**](https://www-formula.ru/2011-09-21-10-55-40) усеченного конуса.

Перегрузите операции:

* **++** префиксный и постфиксный инкременты – увеличение на 1 высоты и радиусов оснований, перегрузка методом класса
* **--** префиксный и постфиксный декременты – уменьшение на 1 высоты и радиусов оснований, перегрузка дружественной функцией класса, выбрасывать исключение при нулевом или отрицательном результате
* **<<**операция вывода, выводить радиусы и высоту, площадь поверхности и объем в виде строки таблицы

Создайте класс

**ArrayConoid**, хранящий динамический массив из усеченных конусов – объектов класса **Conoid** и размер этого массива. Массив создавать в конструкторах, требуется только два конструктора: по умолчанию, создающий массив из 3 элементов, конструктор с параметром – количеством создаваемых элементов). Также разработайте деструктор для освобождения памяти, занятой массивом усеченных конусов.

Геттеры и сеттеры в классе **ArrayConoid** не разрабатывайте, реализуйте только константный метод **size()**, возвращающий размер массива.

В классе **ArrayConoid**перегрузите операции:

* индексирования **[]**, при выходе индекса за допустимые пределы выбрасывайте исключения
* << вывода массива в виде таблицы (элементы массива выводить перегруженной операцией вывода)

Реализуйте методы класса **ArrayConoid**:

* поиск индекса первого элемента массива с минимальным объемом
* поиск индекса элемента массива с максимальной площадью.

С использованием перегруженной операции класса **ArrayConoid[]** реализуйте функции (передавайте в функции объект класса **ArrayConoid** по ссылке):

* вычисления суммарного объема конусов
* вычисления суммарной площади поверхности конусов
* вывод массива в консоль с выделением цветом конуса/конусов с максимальной площадью.

Создайте объект (экземпляр) класса **ArrayConoid**, продемонстрируйте разработанные методы.

**Задача 2.** В том же консольном приложении, с использованием класса Matrix из приложения, рассмотренного в классной работе, решите задачу. Разработайте функции, в которые по ссылке передавайте объект класса Matrix.

* Заполните объект класса Matrixиз 5 строк и 8 столбцов случайными числами
* Найдите сумму элементов в тех столбцах, которые не содержат отрицательных элементов
* Найдите количество столбцов, содержащих хотя бы один нулевой элемент

# Дополнительно

Материалы занятия и задачники – в архиве.Скринкаст занятия можно скачать[**по этой ссылке**.](https://cloud.mail.ru/public/W2Az/HhUxS4q1U)