## Теоретическая часть

* Понятие о функциях обратного вызова
* Наследование от шаблонного класса
* Понятие об исключениях в **C++**–объект исключения, ключевые слова **try**, **catch**, **throw**
* Собственный класс для идентификации исключения
* Иерархия стандартных классов для обработки исключений
* Понятие о «раскрутке» стека, обработка исключений в методах, функциях

## Практическая часть

**Задача 1.**Разработайте иерархию классов для работы с массивами чисел.

Базовый шаблонный абстрактный класс **Container** содержит чистые виртуальные методы **sort()**, **foreach()**.

Метод **sort()** будет использоваться для сортировки данных в контейнере, метод **foreach()** – для поэлементной обработки контейнера.

Производные классы работают с данными типа **double**. При работе методов поэлементной обработки контролировать условия выполнимости операции. Если операция не выполнима – выбрасывать исключение, обрабатываемое в методе.

Разработайте собственный класс исключения, унаследованный от класса **exception**. Передавать в исключение: сообщении об ошибке, значение элемента массива, при обработке исключения сам элемент массива не менять.

В производном классе **Bubble** сортировка реализуется методом пузырька, поэлементная обработка – извлечение квадратного корня.

В производном классе **Choice** сортировка реализуется методом выбора, поэлементная обработка – вычисление натурального логарифма **log().**

В производном классе **QSort** сортировка реализуется методом быстрой сортировки, вызовом функции **qsort(),** поэлементная обработка – вычисление десятичного логарифма **log10().**

**Задача 2.** Разработать консольное приложение, с использованием класса **Person**, двусвязного списка (класс **DblList<Person>**)объектов класса **Person**. Примените функциональные **try**-блоки в конструкторах класса **Person**.

Класс **Person**имеет поля:

* Табельный номер, целое число, уникальное в пределах приложения
* Фамилия и инициалы – динамическое поле, **char \***
* Оклад, вещественное число, **double**
* Год поступления на работу, **int**
* Количество рабочих дней в месяце, **int**
* Количество фактически отработанных дней в месяце, **int**

Разработать конструкторы и деструктор, аксессоры, метод вычисления начислений – произведение оклада на отношение количества фактически отработанных дней к количеству рабочих дней в месяце.

* Создать двусвязный список из объектов класса **Person**–не менее 12 элементов, показать работу конструкторов, методов, генерацию и обработку исключений.
* Разработать функцию, вычисляющую сумму начислений по двусвязному списку персон
* Добавить данные списка в бинарное дерево поиска, использовать табельный номер для записи в бинарное дерево поиска, вывести данные из бинарного дерева поиска в порядке возрастания табельных номеров

## Дополнительно

Запись занятияможно скачать [**по этой ссылке**](https://cloud.mail.ru/public/d3cK/Q3Put5TyX).Материалы занятия в этом же архиве.