

Семинар 2. Примеры задач.

№1.

- Декларируйте делегат-тип **Cast** для представления методов с одним параметром типа **double** и возвращаемым значением типа **int**.
- Создайте два экземпляра типа **Cast**. Первый (1) свяжите с анонимным методом, возвращающим ближайшее чётное целое к переданному в параметре вещественному числу. Второй (2) – с анонимным методом, вычисляющим порядок переданного в параметре положительного числа.
- Протестируйте вызовы при помощи делегатов (1) на одном тестовом вещественном значении; (2) на нескольких тестовых вещественных значениях
- Используя операцию += свяжите оба анонимных метода из задачи 1 с одним многоадресным делегатом. Вызовите методы через него.
- Замените анонимные методы лямбда-выражениями.

№2. Напишите класс **FileUsing** для работы с объёмными файлами. В классе должно быть:
закрытое поле path – путь к файлу;
два конструктора – один пустой (задаёт путь к файлу по умолчанию @ "..\..\Default.txt"),
другой с пользовательским путём к файлу;
методы:

WriteFile (bool @override, int strMin, int strMax) – создает файл (по указанному пути path, записывает в него случайное количество строчек в диапазоне от **strMin** до **strMax**). Параметр **@override** определяет будет ли перезаписываться файл при очередном запуске или нет. Каждая строка составляется случайно из букв латинского алфавита (строчных и заглавных) с помощью метода **StringGenerator()** и имеет длину от 60 до 100 символов.

Создайте делегат **GetInfo(string message)** и закрытое поле **getInfo** – экземпляр типа делегата **GetInfo** (так как оно закрытое, то нужен метод **SetMethod()** для присвоения полю **getInfo** метода из основной программы).

В основной программе создать новый файл (без перезаписи), присвоить метод для вывода в консоль соотношения уже обработанных строк от общего количества строк в файле в процентах от 0 до 100 (выводить только целые значения и первую строку, при которой это целое значение достигается *). Посчитать время считывания файла и вывести его на экран.

* – например, было записано 200 строк:

Ada
afafhjFAaltu
fasHSF
jkgpoiUIO
и т.д.

Вывод будет:

Ada 0
afafhjFAaltu 1
jkgpoiUIO 2

№3. Лямбда-выражения, к примеру, используют для того, чтобы задать определённую логику для статических методов класса **Array**. Например, сортировки по убыванию и сортировки по чётности (см. *Task_3.cs*).