

# Введение

В данном разделе показан пример более гибкой вариативности выбора фильтра. Если вы полностью разобрались с делегатами, классами и Зей и 4ой лабораторной работой - можете выполнить это дополнительное задание.

## Суть

Если вдруг вы обратили внимание, то большая часть наших фильтров имеет схожую сигнатуру.

```
/// <summary>
/// Инвертирует цвета на изображении.
/// </summary>
Ссылок: 1
public static void Invert(Image image)...

/// <summary>
/// Преобразует изображение в оттенки серого.
/// </summary>
Ссылок: 0
public static void Grayscale(Image image)...

/// <summary>
/// Придает изображению эффект "старой фотографии".
/// </summary>
Ссылок: 0
public static void Sepia(Image image)...
```

*laba4\_signature\_sample.png*

В прошлой лабораторной работе, в качестве одного из аргументов функции **Filter** мы передавали делегат. Так как у части наших фильтров для изображения сигнтура выглядит как **void MethodName(Image img)**, эти фильтры могут быть добавлены в один словарь.

Пример реализации через словарь:

Нам нужно создать подходящий под сигнтуру методов тип делегата:

```
// Это можно объявить прямо в файле Program.cs, над
классом Program
```

```
public delegate void ImageFilterDelegate(Image image);
```

Далее мы можем создать словарь, у которого ключом будет название метода (`string`), а значением будет метод (`ImageFilterDelegate`).

Создание словаря с ссылками на метод:

```
var availableFilters = new Dictionary<string,
ImageFilterDelegate> {
    { "invert", ImageFilters.Invert },
    { "grayscale", ImageFilters.Grayscale },
    // Чтобы добавить новый, просто пишем его имя
    { "sepia", ImageFilters.Sepia }
};
```

Для методов с отличающейся сигнатурой (например:

`AdjustBrightness(Image img, int value)`) можно создать метод с сигнатурой делегата (`void ApplyAdjustBrightness(Image img)`). Этот новый метод будет вызывать метод `AdjustBrightness`, передавать туда `img` из аргумента, а значение яркости нужно спарсить из консоли.

Наше допущение:

Предполагается, если пользователь ввел название метода для яркости, то он введет четвертый аргумент значения. Его нужно будет получить из `args` при вызове `ApplyAdjustBrightness`.

Думайте...