**Задание №1**

Разработать Windows Desktop приложение, демонстрирующее обработку изображений по заданному алгоритму. Реализовать возможность выбора алгоритма из списка.

*Требования:*

1) Технология: WinForms.

2) Алгоритмы обработки: контрастирование, градиент (оператор Собеля).

3) Алгоритмы должны выполняться асинхронно относительно UI-потока.

4) Не использовать сторонние библиотеки (OpenCV и т.п.).

5) Для увеличения производительности разрешено использование unsafe кода.

6) Код должен быть написан в соответствии с Code Convention (см. папку CodingConventions.zip).

7) Интерфейс должен быть выполнен в соответствии microsoft windows user experience guidelines. (<https://msdn.microsoft.com/en-us/library/windows/desktop/dn688964(v=vs.85).aspx>).

**Задание №2**

Реализовать интерфейс на основе формы для отображения видео файла. Интерфейс должен содержать элементы управления:

* кнопки «Проиграть» (Play), «Перемотка назад», «Перемотка вперед», «Открыть…»;
* ползунок времени (Slider).

По кнопке «Открыть…» возникает стандартное диалоговое окно выбора видео файла с известными расширениями (например, avi, mp4).

После загрузки видео файла данные у ползунка времени меняются на «00:00:00 – 01:23:45» (используется длина видео файла). При перемещении ползунка времени необходимо позиционировать кадр видео на нужное время.

По нажатию на кнопки «Проиграть» и «Перемотки» текущая позиция на ползунке времени также меняется. Отображение видео необходимо реализовать с применением аппаратного ускорения или GPU. Картинка видео должна подгоняться под размер формы.

Тестовое задание нужно выполнить на языке C# любой версии на основе Windows Forms (WinForms) или Windows Presentation Foundation (WPF).

Параметры оценки: выполнение поставленной задачи, отсутствие багов, структурно чистый и понятный код.

**Задание №3**

Консольное приложение для резервного копирования файлов в архив.

В файле настроек хранятся пути для исходной и целевой папки.

При запуске программы происходит создание папки с временным штампом в целевой папке и копирование в неё всех доступных файлов из исходной. Требуется обрабатывать ситуации с невозможностью доступа к файлам в исходной папке.

Пункты со звездочкой являются дополнительными и не обязательны для выполнения.

* Файл настроек имеет формат JSON.
* Есть возможность указать несколько исходных папок.
* Ведется журналирование процесса копирования. Каждый запуск создает свой файл журнала.

Уровень журналирования можно указать в файле настроек.

Примеры распределения событий:

* Error - Ошибки приложения. Например, те, которые вызвали неожиданное падение.
* Info - Основные события приложения: старт приложения, обработка одной исходной папки или обработанные ошибки.
* Debug - Отладочная информация. Например, скопирован отдельный файл

**Задание №4**

Написать программу на C#, предназначенную для сжатия и расжатия файлов с помощью *System.IO.Compression.GzipStream*.

Параметры программы, имена исходного и результирующего файлов задаются в командной строке следующим образом:

* Для архивации: GZipTest.exe compress [имя исходного файла] [имя архива]имя исходного файла] [имя исходного файла] [имя архива]имя архива]
* Для разархивации: GZipTest.exe decompress [имя исходного файла] [имя архива]имя архива] [имя исходного файла] [имя архива]имя распакованного файла]

В случае успеха программа возвращает 0, при ошибке 1.

Программа должна эффективно распараллеливать и синхронизировать задачи в многопроцессорной среде и уметь обрабатывать файлы, размер которых превышает объем

доступной оперативной памяти.

Код должен корректно обрабатывать все исключения, а при работе с потоками допускается использовать только стандартные классы и библиотеки из .Net (исключая ThreadPool, BackgroundWorker, TPL). Ожидается реализация с использованием Thread-ов.

Код программы должен следовать принципам ООП.