Retry operation pattern

<https://alastaircrabtree.com/implementing-a-simple-retry-pattern-in-c/>

**Vystrihovadlo – StatusChangeServiceAction**

const int retryAttempts = 2;

var pauseBetweenFailures = TimeSpan.FromSeconds(20);

// zkusi pustit operaci. Pokud v ramci operace bude vyhozena vyjimka, zkusi to znovu a pak vypise error na konzoli. Pokud tato operace neprojde, party menit nebude.

RetryOnException(retryAttempts, pauseBetweenFailures, () =>

{

**m\_statusChangeService.ChangePublicationStatus(request.PublicationId, request.PublicationStatusId, request.PartIdsInPublication, m\_mediaDataDaoSource);**

});

private static void RetryOnException(int times, TimeSpan delay, Action **operation**)

{

var attempts = 0;

do

{

try

{

attempts++;

operation();

break; // Sucess! Lets exit the loop!

}

catch (Exception exception)

{

if (attempts == times)

//throw;

//System.Windows.MessageBox.Show(CuttingResources.CriticalError);

Console.WriteLine($@" Nemuzes pokracovat protoze vyjimka byla vyhozena znovu . Jsem v Core a nemuzu si vyhazovat dialogy do UI. Pokus c.: {attempts}");

if (m\_log.IsErrorEnabled)

{

m\_log.Error(exception.Message, exception);

m\_log.Error($"Exception with message: {exception.Message}, caught on attempt {attempts} - will retry after delay {delay}", exception);

}

Task.Delay(delay).Wait();

}

} while (true);

}