

ZADANIA:

1. Research dotyczący rozpoznawania urządzeń USB.

<ftp://ftp.helion.pl/online/usbppr/usbppr-4.pdf>

Generalnie da się (powyżej przykłady w C++, nie będzie problemu żeby albo wykorzystać klasy do urządzeń USB z .NET albo wykorzystać te dane poprzez plInvoke), przechwycenie zdarzenia podłączenia urządzenia do systemu, enumeracja urządzeń – można potem ładnie wyświetlić dialog userowi, żeby sobie wybrał urządzenie – następnie opisał, zapisał, może podał ścieżki skąd brać zdjęcia (to nie musi być aparat, ale telefon, smartfon etc.).

Potem w obsłudze tego samego zdarzenia, przy rozpoznaniu można łatwo odczytać literę dysku, skorzystać ze ścieżek i skopiować podane dane.

Trudność zadania (w skali 1-10, gdzie 10 to najtrudniejsze): **7**

2. Research dotyczący wydajności pola BLOB w przechowywaniu zdjęć w bazie danych.

Generalnie lepiej trzymać wiele małych plików, wczytywanie powinno zrobić się w tle (BackgroundWorker pattern). Dlaczego?

Pola BLOB nie są koszmarnie wolne (ale i tak jest wolne), problemem jest wyświetlenie, bo i tak musimy obrazek zapisać gdzieś na dysku i go wczytać do kontrolki. Zapisywanie samej bitmapy (siatki pikseli) mija się z celem IMHO.

Widzę to tak, że będzie ukryty (ten z kropką) folder w katalogu użytkownika programu, gdzie przechowywane będą zmniejszone obrazki (przy wczytywaniu zdjęć generacja miniaturki także w tle). Stamtąd również w tle, będą wczytywane miniaturki do naszych widoków. Wadą jest to, że trzeba go backupować (przy zapisywaniu ustawień) oraz zajmuje miejsce (mniejsza wada). Wczytanie 100 małych plików (do 300KB) to nie jest spory narzut, odwrotnie jest przy bazie gdzie operacja wczytaj plik z BLOB, zapisz temp, wczytaj jest zabójcza dla wydajności.

Trudność zadania (w skali 1-10, gdzie 10 to najtrudniejsze): **4**