**Počítačová obrana a útok**
Protokol z předmětu

**Tématická oblast**: Log files

**Přednášející**: Ing. Jan Plucar, Ph.D.
**Cvičící**: Ing. Jan Faltýnek

**Jméno a číslo studenta**: Adam Šárek (SAR0083)
**Datum vypracování**: 21. 4. 2022

Zadání:

Vaším úkolem bude naprogramovat aplikaci, která bude kontrolovat vstupní soubory (Users.xlsx a Data.txt) a na základě obsahu bude generovat i log file. Cvičení je mířeno na "error handling" a vytváření logů. Výsledná aplikace nakonec vytvoří dva soubory - LogFile.txt a ErrorsLogFile.txt. Soubor LogFile.txt bude obsahovat veškeré informace (čas spuštění aplikace, kontrola jednotlivých řádků, všechny chyby atd.) a ErrorsLogFile.txt bude obsahovat pouze záznamy o chybách (chybný formát, neočekávaný vstup, atd.).
Aplikace bude číst soubory po řádku a vyhodnocovat, zda odpovídá řádek předepsaným pravidlům. Pokud je řádek bezchybný, tak se do "LogFile.txt" zapíše na řádek {<time> <int id> <int row>}, v případě, že obsahuje nějakou chybu, tak se zapíše řádek do obou log souborů. V takovém případě bude obsahovat řádek časové razítko a co nejpřesnější popisek chyby. Jelikož vstupní dva soubory jsou propojené (ID ze souboru "Users.xlsxt", odpovídá druhému sloupci v "Data.txt"), je potřeba kontrolovat i tohle. Když se objeví řádek v Data.txt s ID uživatele, který neexistuje v Users.xlsx, je to chyba!
Aplikace by nejprve měla zkontrolovat Users.xlsx soubor (informace bude taky nějak reprezentována v LogFile.txt) a následně procházet druhý soubor.
V souboru Data.txt poslední sloupec reprezentuje MD5 otisk dat (čtvrtý sloupec) na daném řádku, proto je potřeba kontrolovat, zda je tato hash dobře vypočítaná. Pro usnadnění parsování, je časové razítko ohraničeno "<>".

Soubor Users.xlsx:
- ID - int
- Jmeno - řetězec přípustných písmen pro jména (CZE kódování)
- Prijmeni - řetězec přípustných písmen pro jména (CZE kódování)
- Vek - int
- Web - přípustná webová adresa, položka není povinná
- Mail - řetězec přípustných znaků pro mail
- Telefon - 9 čísel

Soubor Data.txt:
- ID - int
- UserID - int
- Time - time formát
- Data
  - přípustné znaky:
    - "A" až "F" (včetně)
    - "a" až "f" (včetně)
    - "0" až "5" (včetně)
    - "+" a "-"
  - délka (20 až 50 znaků)
- MD5hash

Součásti odevzdaného řešení bude stručný popis vaší aplikace a popis formátu logů. Dále budete odevzdávat zdrojový kód aplikace + ErrorsLogFile.txt. a LogFile.txt.

# Vypracování

Před samotným procházením datových souborů a zjišťováním obsažených chyb je potřeba tyto datové soubory nejprve načíst. K tomu je využit programovací jazyk C#, s jehož pomocí jsou data ze souborů *Users.xlsx* a *Data.txt* načtena a poté uchovávána ve struktuře *DataTable*. V případě textového souboru je použitý pouze jeden objekt *DataTable*, jelikož tento soubor představuje jednu tabulku, zatímco v případě .xlsx souboru je možné načíst i více takovýchto tabulek, jelikož tento soubor může potenciálně obsahovat více samostatných listů. Výhodou jednotného načítání obou souborů je to, že je možné načtená data kontrolovat jednotně a jelikož se v obou případech jedná o tabulku dat, tak se toto sjednocení nabízí.

Před procházením načtených dat je potřeba zavést požadavky, které mají jednotlivé části splňovat. Tyto požadavky je možné ručně upravovat přímo v parametrech konstruktoru třídy *Logger*, která byla vytvořena právě pro vytváření záznamu o průchodu a chybách již zmíněných souborů. V parametrech je rovněž možné nastavit cestu k oběma datovým souborům a také k souborům obsahujícím již zmíněné záznamy.

Konstruktor třídy *Logger* se stará o samotný průchod a kontrolu chyb dle poskytnutých požadavků tím, že zavolá metodu *CreateLogFiles*. Tato metoda nejprve kontroluje data ze souboru *Users.xlsx* a až poté data ze souboru *Data.txt*. Nejprve je kontrolováno, zda samotný soubor vůbec existuje. Dále je poté vypsán počet nalezených tabulek a řádků. Data jsou procházena po jednotlivých tabulkách (v případě textového souboru pouze po jedné tabulce) a po jednotlivých řádcích. O procházení a kontrolu jednotlivých řádků se stará metoda *ValidateTableColumns*, která přijímá na vstupu kromě dané struktury tabulky také dané požadavky a číslo počátečního řádku. V případě, že na daném řádku není žádná chyba, tak se zapíše záznam podobný tomu, který je uvedený na obrázku Obr. 1. Obrázek Obr. 2 poté obsahuje příklad, jak může vypadat obsah souboru *ErrorsLogFile.txt*. Je možné si všimnout, že pokud daný řádek obsahuje i více než jen jednu chybu, tak jsou jednotlivé chyby vypsány vždy na nový řádek záznamu.

```
Started at <2022-04-20T02:17:52.230>

Users.xlsx (1 table found)
Users.xlsx / List 1 (32 rows found)
{<2022-04-20T02:17:52.422> <1> <2>}
{<2022-04-20T02:17:52.432> <2> <3>}
{<2022-04-20T02:17:52.436> <3> <4>}
{<2022-04-20T02:17:52.440> <4> <5>}
{<2022-04-20T02:17:52.448> <5> <6>}
{<2022-04-20T02:17:52.452> <6> <7>}
{<2022-04-20T02:17:52.456> <7> <8>}
{<2022-04-20T02:17:52.460> <8> <9>}
```

*Obr. 1 - Část záznamu v souboru LogFile.txt (bez nalezených chyb)*

```
Users.xlsx
{<2022-04-20T02:17:52.542> Error at row <31>: Vek <asd> does not contain a valid integer.}
{<2022-04-20T02:17:52.546> Error at row <32>: Telefon should have length of <9> but the given value <4586768890> has length of <10>.}
{<2022-04-20T02:17:52.581> Error at row <33>: Mail <svo@gmail@> does not contain a valid email address.}

Data.txt
{<2022-04-20T02:17:52.608> Error at row <1>: MD5hash <8D327B7C13C> is not equal to the expected hash <8D327B7C13CC1B48EC395CE57CD453ED>.}
{<2022-04-20T02:17:52.620> Error at row <8>: MD5hash <64175AEE10D4C51A5F368BA536F21E48> is not equal to the expected hash <64175AEE10D4C51A5F368BA536F21E49>.}
{<2022-04-20T02:17:52.643> Error at row <24>: Data should have length between <20> and <50> but the given value <+C32FCfDfcF351A+Eb-1Df3eE4f2F+DDECd3dCDCDCD1111222333331111111111CAAAAAAAAAAA35111>
has length of <81>.}
{<2022-04-20T02:17:52.644> Error at row <24>: MD5hash <50E5DE3163C60B1ED27D2F03B1F2EF26> is not equal to the expected hash <063709FE88190BF4FB149F3A222F2506>.}
{<2022-04-20T02:17:52.657> Error at row <35>: UserID <0> is not found in source data column <ID>.}
```

*Obr. 2 - Část záznamu v souboru ErrorsLogFile.txt (obsahující pouze chyby)*

Pro specifikování jednotlivých požadavků byla vytvořena enumerace *Rule*, která nabízí možnosti *Required*, *Length*, *Int*, *Email*, *TimeFormat*, *Num*, *URL*, *Encoding*, *AllowedCharacters*, *MD5Hash* a *JoinSourceTableColumn*. Dle vybraného požadavku se poté odvíjí kontrola daného sloupce, přičemž je možné si na začátku pro daný sloupec zvolit i více než jen jeden požadavek.

- ***Required*** očekává, že bude hodnota v daném sloupci alespoň o délce 1. Tento požadavek je u všech sloupců kromě webu v *Users.xlsx*, jelikož tento sloupec není povinný.
- ***Length*** kontroluje, zda délka položky je v určitém definovaném rozmezí, které může a nemusí být z jedné strany omezeno.
- ***Int*** kontroluje výsledek po zavolání metody *int.TryParse* nad danou položkou.
- ***Email*** kontroluje, zda po zadání emailu do konstruktoru třídy *MailAddress* se nevyvolá výjimka. Pokud se vyvolá výjimka, tak se jedná o neplatný formát emailové adresy.
- ***TimeFormat*** kontroluje výsledek po zavolání metody *DateTime.TryParse* nad danou položkou.
- ***Num*** kontroluje, zda se jedná o správný zápis čísla. Oproti *Int* tento požadavek splňují i čísla, které jsou mimo velikostní rozsah datového typu *int* a jejich kontrola probíhá díky využití regulárních výrazů.
- ***URL*** kontroluje, zda se jedná o validní URL adresu, k čemuž také používá regulární výraz. Vzhledem k tomu, že web není povinnou položkou v tabulce, tak kontroluje pouze ty položky, které skutečně nějaký text obsahují.
- ***Encoding*** na základě regulárního výrazu kontroluje, zda daná položka obsahuje platné znaky v rámci zadaného jazyka (čeština: cs_CZ). Sada znaků obsahuje velká a malá písmena včetně české diakritiky.
- ***AllowedCharacters*** pomocí regulárního výrazu kontroluje, zda daná položka obsahuje pouze vybrané znaky, které vychází ze zadání protokolu.
- ***MD5Hash*** nejprve pomocí metody *GetMD5Hash* generuje MD5 hash z obsahu sloupce Data v souboru *Data.txt* a poté kontroluje, zda tento řetězec odpovídá položce ve sloupci MD5hash.
- ***JoinSourceTableColumn*** zajišťuje kontrolu toho, zda hodnota UserID v *Data.txt* je obsažena také v rámci souboru *Users.xlsx* v některé z položek sloupce ID. Vybrané propojení těchto 2 sloupců a tabulek bylo nastaveno v úvodních požadavcích.

## Závěr

Cílem tohoto cvičení bylo naučit se zpracovávat chyby, které se mohou v práci s nějakými daty objevit a co možná nejdetailněji je zapisovat, aby bylo možné tyto chyby dohledat a případně opravit. Cílem tedy bylo načíst 2 datové soubory *Users.xlsx* a *Data.txt*, které obsahují nějaké chyby vzhledem k předem definovaným požadavkům. Tyto soubory jsme měli pomocí programu projít a vypsat záznamy o daném průchodu do souborů, které obsahují všechny záznamy (*LogFile.txt*) a souboru, který obsahuje pouze chyby (*ErrorsLogFile.txt*).