MATURITNÍ PRÁCE - QUALITÄTSSTATISTIK

Martina Bošková

Obsah

[**Úvod** 2](#_Toc193867717)

[**Specifikace programu** 2](#_Toc193867718)

[Skript Decide 2](#_Toc193867719)

[Skript Robot 2](#_Toc193867720)

[Skript Text 3](#_Toc193867721)

[**Použité nástroje** 3](#_Toc193867722)

[Funkce 3](#_Toc193867723)

[**Konečný vzhled** 4](#_Toc193867724)

[**Testování, ladění a vylepšování** 4](#_Toc193867725)

[**Závěr** 4](#_Toc193867726)

# **Úvod**

Při výběru své maturitní práce jsem ze začátku velmi váhala. Přála jsem si vybrat něco nejen pro mé schopnosti dostupného, ale i smysluplného. Základním kritériem se mi tedy stala myšlenka mé budoucnosti. Jelikož bych ráda navázala na studium na Mikulášském gymnáziu studiem na fakultě ekonomické, má mysl směřovala jako první k vytvoření programu spojeného s účetnictvím. S touto základní úvahou jsem začala vnímat problémy, s kterými jsem se setkala, při pracování ve firmě ADP. Jako brigádnice jsem často žádána o vyřešení problémů spojených s manuální kontrolou. Co takhle ulehčit si práci, a ještě splnit nutnost pro splnění maturity? Nějak takto vznikl nápad mého tématu, protože právě jedním z opakujících se manuálních úkolů je statistika kvality.

Co dělá ADP? „*ADP je přední světový poskytovatel cloudových řešení pro správu lidského kapitálu.*“[[1]](#footnote-1) Tým, kterého jsem součástí, zpracovává účetnictví pro firmu z Německa. Každý měsíc se musí dělat výpis statistiky kvality. Ten musí být ve formátu daném ADP a zaznamenává důvody zásahu do výplatnic zaměstnancům v minulých měsících. Můj vytvořený program má tedy za úkol automatizované rozpoznání těchto důvodů, podle určitých kritérií. Vytvoří nebo doplní konečnou zprávu podle uživatelského vstupu.

# **Specifikace programu**

Hlavní skript jsem rozdělila na tři části. Program totiž využívá dvě hlavní větve. Základem se stalo dodání dat od uživatele. Firma ADP totiž používá pro zpracování těchto dat dvě možnosti. Jednou z nich je Excel, který obsahuje jen mzdové údaje lidí potřebných ke zpracování. Druhá dodává osobní data v textovém souboru obsahující i informace dalších lidí. K textovému souboru je připojen i Excel ve formátu konečné zprávy s informacemi o lidech, u nichž se má důvod zásahu určit. Do něj se výsledná data jen doplňují.

## Skript Decide

Úkolem tohoto skriptu je požádání uživatele o vstup Excelu. Jelikož jsem při psaní programu počítala, že cílová skupina budou zaměstnanci pověřeni sepsáním statistiky kvality pro ADP, již předpokládám formát používaný danou firmou. Díky němu jednoduše rozpoznám, kterou větví se má můj program dále vydat. Tento vstup tedy vyhodnotím, spustím další z mých skriptů a předám vstup následujícímu skriptu.

## Skript Robot

Tato možnost tedy předpokládá, že vložený Excel obsahuje jen lidi s daty potřebnými k zpracování. Skript pomocí cyklu projede celý soubor a rozdělí zaměstnance do tříd. Rovnou jim přidá vlastnosti, jako například Personal Nummer neboli číslo specifické pro každého zaměstnance klientské firmy nebo Abrechnungskreis, což je oblast firmy se stejným fakturováním (může se skládat i z více poboček). Při tomto kroku podle daných kritérií určí i Fall neboli důvod zásahu.

Dalším úkolem při této možnosti je vytvoření konečné zprávy. Ta musí splňovat jasné formátování dat a vysvětlivku na konci Excelu. Také pomocí uživatele určuji měsíc, na který se kvalita dělá. Podle toho se mění i rozsah časových údajů v souboru.

## Skript Text

Prvním úkolem je, podobně jako u *Skriptu Robot*, vytvoření tříd ze zaměstnanců zadaných ke kontrole. Dále je uživatel požádán o textový soubor. Ten je prohledáván pomocí cyklu, za účelem nalezení lidí z Excelu. U těch je poté podle daných pravidel určen Fall. Oproti první možnosti se už konečná zpráva nemusí vytvářet. Pro jistotu se udělá kopie prvního listu a do ní se doplní zjištěné údaje.

# **Použité nástroje**

Z různých možností zpracování jsem se rozhodla pro programovací jazyk Python. A proč tedy? Přemýšlela jsem o naprogramování mé práce v excelovském makru, ale to přinášelo několik nevýhod. Sama si v něm nejsem jistá a bylo mi od ostatních doporučeno programování v Pythonu pro jeho jednoduchost. Navíc, při dotazu na firmu mi bylo řečeno, že excelovská makra nejsou kvůli zabezpečení využívána.

Pro mne důležitá knihovna byla Openpyxl. Tu jsem předtím nikdy nevyužívala, ale byla mi velmi nápomocná. Její využití vyniklo při vytváření závěrečné zprávy ve *Skriptu Robot*, ale využila jsem ji i při kopírování a doplňování Excelu ve *Skriptu Text*. Dále jsem použila například modul sys, nebo os.

## Funkce

Ráda bych zmínila vlastní funkce, které jsem vytvořila, jako například funkci *not\_valid*, *people\_classes* nebo *final\_report*. Funkce not\_valid je opatření proti nesprávnému vstupu od uživatele. Používám ji víckrát, abych ověřila koncovku programu a možnost jeho otevření. Funkce people\_classes se vyskytuje ve *Skriptu Robot*. Prochází určený Excel a ukládá měsíce, ve kterých nastala změna. Zároveň také podle firmou daných pravidel určuje důvod změny. Poslední funkce *final\_report* využívá pomocný Excel, ve kterém je předpřipravená vysvětlivka, která musí být na konci konečné zprávy. Kopíruje tuto vysvětlivku i s jejím formátem a vkládá ji do vytvářeného závěrečného Excelu. Také doplňuje vyzjištěná data a formátuje je.

Mnou často využívanou vestavěnou funkcí je *with open*. Funkce *open* slouží k otevření souboru. Já nepoužívám tuto funkci samotnou, kvůli přehlednosti a bezpečnosti kódu. *With open* snižuje riziko chyb, jelikož i když nastane chyba v kódu a kód skončí, tato funkce soubor automaticky uzavře. Jak tedy přesně funguje?„*Důvodem je, že with příkaz volá 2 vestavěné metody za scénou – \_\_enter()\_\_a \_\_exit()\_\_.*“[[2]](#footnote-2)

# **Konečný vzhled**

Svůj dokončený program jsem nechala ve formátu komprimované složky. V ní se nachází tři, již předem popsané, Python skripty a také pomocný Excel a textový soubor, které ve skriptech používám. Pro zajištění správného virtuálního prostředí používám textový soubor requirements.txt.

# **Testování, ladění a vylepšování**

V této době ještě nemám potvrzenou ani vyvrácenou možnost testování svého programu ve firmě ADP. Můj program musí být ověřen a schválen kvůli bezpečnosti. Přepsala jsem pro testování pozměněná, a tím pádem anonymizovaná, data a průběžně zkoušela program na nich, ale jen při větším množství dat se dá ověřit, jestli program funguje na sto procent tak, jak má.

Program jsem psala v editoru Visual Studio Code a pro čistější vzhled kódu jsem používala rozšíření Flake8. Snažila jsem se dodržovat pojmenovávání datových typů a pro transparentnost popisovala pomocí poznámek. Využívala jsem pro postupné zálohování úložiště Github, proto ani nebyl při odhalování chyb problém, vrátit se k minulé verzi.

Možnost vylepšení samozřejmě vidím. Daly by se ještě naprogramovat další Fally, ale upřímně si myslím, že Fall 30, který můj program dokáže určit, je jasně nejčastější. Nevidím tedy jako problém nechat si toto vylepšení až do budoucna. Zvažovala jsem využití regulérního výrazu (RegEx), ale pro náročnost to ponechávám, jako možnost budoucího vylepšení.

# **Závěr**

Jsem spokojená s výběrem tématu maturitní práce. Věřím, že v budoucnu bude využita, což byl jeden z mých základních požadavků. Nejsem zkušený programátor, a proto jsem velmi hrdá na to, že mou práci firma testuje a snad i implementuje. Při vytváření tohoto programu jsem se naučila pracovat s novými funkcemi i knihovnami a rozvíjela jsem své znalosti s Pythonem. Ale nejen to, také jsem například prvně využila úložiště Github. Celkově jsem se zlepšila v řešení problémů, vyhledávání informací a programování.

1. <https://adp.jobs.cz/mzdovapozice> [↑](#footnote-ref-1)
2. <https://www.freecodecamp.org/news/with-open-in-python-with-statement-syntax-example/> [↑](#footnote-ref-2)