

ZÁRÓDOLGOZAT

Sipos Szilvia

2023



Magyar Agrár- és Élettudományi Egyetem

Károly Róbert Campus

Műszaki intézet

**Programtervező informatikus felsőoktatási szakképzés
szak**

**Szakmai gyakorlat bemutatása, központban a Python Excel-
automatizáló programmal**

Belső konzulens: Dr. Novák Tamás

Egyetemi docens

**Belső konzulens
intézete/tanszéke:** Műszaki intézet

Készítette: Sipos Szilvia

Gyöngyös

2023

Tartalomjegyzék

1	Bevezetés	2
1.1	Köszönetnyilvánítás	2
2	Szakirodalmi áttekintés	3
2.1	Rendszergazdai feladatok	3
2.2	SQL lekérdezések	4
2.3	Python	5
2.3.1	Könyvtárak	7
2.3.2	OpenPyXL	8
2.3.3	PyCharm	9
3	A vállalat bemutatása	11
4	Szakmai gyakorlaton végzett feladatok	14
4.1	Rendszergazdai feladatok	14
4.2	SQL lekérdezések	16
4.3	Python programozási feladat	17
5	Összegzés.....	26
	Irodalomjegyzék	27
	Ábrajegyzék	30
	Mellékletek.....	31
	Nyilatkozatok.....	34

1 Bevezetés

Közel 3 éve dolgozom az U-Partner National Kft.-nél, eredetileg kereskedelmi feladatkörben, idén július óta azonban munkaköröm informatikai feladatokkal bővült. Így lehetőségem nyílt, hogy szakmai gyakorlatomat a nyár folyamán munkahelyemen tudjam teljesíteni. Ezirányú feladataim sokrétűek voltak, a régóta fennálló problémák, hiányosságok javítása mellett, az aktuális munkafolyamatok informatikai támogatása volt a feladatom. Az informatika több területén tevékenykedtem, de főként rendszergazdai, adatbázis-kezelői és programozási feladataim voltak.

A dolgozatomban ezt a három területet mutatom be. Először bemutatom ezek elméleti hátterét, utána a vállalatot, végül a konkrét feladataimat részletezem.

Kereskedelmi munkám során sokat dolgozom Excel táblákkal, ezért mindig is érdekelt ezek működésének, hatékonyságának programozással történő gyorsítása, automatizálása. Szakmai gyakorlatom során írtam egy ilyen programot Python programozási nyelven, a dolgozatomban erre fókuszálok, és részletesen bemutatom a programot és működését.

1.1 Köszönetnyilvánítás

Szeretnék köszönetet mondani konzulensemnek, Dr. Novák Tamás egyetemi docensnek, akinek végtelen türelme és észrevételei, javaslatai, instrukciói nélkül szakdolgozatom nem készülhetett volna el.

Köszönöm Végh Ildikó tanulmányi osztályvezetőnek és Dr. Zörög Zoltán egyetemi docens és főigazgató-helyettesnek a sok technikai segítségnyújtást és az egyetemi hallgatóknak szervezett online konzultációkat, melyek nagy segítséget jelentettek a dolgozat elkészültében.

Köszönettel tartozom egyetemi oktatóimnak, akik lelkiismeretes munkája révén szert tettem a szükséges elméleti és gyakorlati ismeretekre.

Köszönöm az U-Partner National Kft. ügyvezetőinek, D. Tóth Krisztiánnak és Kaffka Gábornak, amiért lehetőséget biztosítottak munkahelyemen informatikai munkakör ellátására és ezáltal szakmai gyakorlatom teljesítésére is.

Végül szeretnék köszönetet mondani családomnak és barátaimnak, akik mindvégig türelmesek és segítőkészek voltak velem szemben.

2 Szakirodalmi áttekintés

2.1 Rendszergazdai feladatok

A rendszergazda, azon emberek egyike, akik a színfalak mögött dolgoznak a számítógépes infrastruktúra konfigurálásán, üzemeltetésén, karbantartásán és hibaelhárításán. Támogatja a modern élet nagy részét. (Haber et al., 2011).

A rendszergazda telepít, támogat, üzemelteti a szervert és más számítógépes rendszereket. Szolgáltatás leállás, valamint egyéb problémák esetére előzetes stratégiával rendelkezik. További teendői lehetnek különböző szkript írások, kisebb programozási munkák, projekt menedzsment is. (http21, 2023)

A rendszergazdát hívják vírustámadások, nem működő levelezőrendszerek, elfelejtett jelszavak, lefagyott rendszerek, elromlott eszközök kapcsán, mint egy irodai nyomtató vagy egy céges laptop, őt keresik biztonsági kérdések és még sok egyéb más miatt is. (http20, 2023)

A rendszergazda legfontosabb feladatai közé tartozik:

- rendszerek üzemeltetése és frissítése,
- rendszer erőforrások kezelése és optimalizálása,
- hibaelhárítás,
- spamszűrés,
- adatmentés,
- fájlrendszerek kiépítése,
- jelszavak kezelése,
- technikai segítségnyújtás. (http20, 2023)

A rendszergazda munkája kritikus, hiszen egy munkahelyen létfontosságú, hogy a dolgozók mindent gyorsan és problémamentesen tudjanak használni. Nagyon ügyelni kell arra, hogy mindig megfelelő legyen az internetkapcsolat és a szerverek, valamint az iroda működéséhez elengedhetetlen gépek mindig elérhetőek legyenek. (http20, 2023)

2.2 SQL lekérdezések

Az SQL (Structured Query Language, magyarul Strukturált Lekérdező Nyelv) egy relációsadatbázis-kezelő lekérdezési nyelv. Ez a szabványos nyelv a relációs adatbázis-kezelő rendszerekkel (Relational Database Management System, RDBMS) történő interakciókhoz, mint például a MySQL, az IBM DB2, az Oracle, a Microsoft SQL Server, és a Microsoft Access. (http23, 2023)

Bár nem tekintjük programozási nyelvnek, mégis hasonlóan működik, itt is vannak utasítások, amelyek segítségével műveleteket végezhetünk. Adatbázisokat hozhatunk létre, adatokat adhatunk hozzá, törölhetünk belőle, módosíthatjuk az egyes adatokat, lekérdezhetjük azokat, vagy rendezhetjük is. (http22, 2023)

A lekérdezési funkciók mellett olyan elemekkel is rendelkezik, amelyek más adatkezelési funkciók végrehajtására is alkalmasak. Legújabb szabványos változatai pedig már egészen kiterjedt adatbázis-kezelési műveletek megvalósítására is használhatók. (Holovács, 2006)

Az SQL-lekérdezések SQL-ben írt utasítások, amelyek lehetővé teszik az adatbázissal való kommunikációt különféle műveletek végrehajtása érdekében, mint például:

- Adatok lekérése: SQL lekérdezések segítségével lekérhetjük az adatokat egy adatbázistáblából, megadva a lekérni kívánt oszlopokat és a szűrési feltételeket.
- Adatok beszúrása: az INSERT utasítás segítségével új rekordokat adhatunk hozzá egy táblához.
- Adatok frissítése: az UPDATE utasítással az adatbázistábla meglévő rekordjait lehet módosítani.
- Adatok törlése: A DELETE utasítás segítségével rekordokat távolíthatunk el egy adatbázistáblából. Itt figyelni kell arra, hogy a WHERE kulcsszóval feltételt adjunk meg, különben kitörölhetjük az egész táblát. (http24, 2023)

Az SQL-lekérdezések egy meghatározott szintaxist és struktúrát követnek, amelyek jellemzően kulcsszavakból állnak, például SELECT, INSERT, UPDATE, DELETE, FROM, WHERE stb. az adatbázison végrehajtani kívánt műveletek meghatározásához. Az SQL-lekérdezések kulcsfontosságúak az adatlekéréshez, az adatkezeléshez és az adatbázis-kezeléshez különböző alkalmazásokban és rendszerekben. (http23, 2023)

2.3 Python

A Python egy objektum orientált, dinamikus, bővíthető, magas-szintű programozási nyelv, aminél nem fordító, hanem interpreter (értelmező) kell a kód futtatásához, vagyis a forráskódot közvetlenül, soronként, egy interpreter hajtja végre, anélkül, hogy gépi kódba kellene fordítani. (docs.python.org, 2023)

Egy holland programozó, Guido van Rossum 1989 végén kezdte el fejleszteni a Python-t, majd 1991-ben hozta nyilvánosságra. A nyelvet kifejezetten úgy tervezték, hogy az „olvashatóság” és a programozói munka megkönnyítése kerüljön előtérbe a futási sebességgel szemben. (http1, 2023)

A Python nyilvánosságra hozatala óta hatalmas népszerűsége tett szert, és széles körben használják különféle területeken, beleértve többek között a szoftverfejlesztést, webfejlesztést, az adatelemzést, vagy a gépi tanulást (machine learning). (http2, 2023)



1. ábra: Python logó. Forrás: <https://www.python.org/>

A Python hordozható (portábilis), azaz elérhető többféle operációs rendszerre is, mint például Linux, Windows, MacOS stb. A Python ingyenes, mégis korlátozás nélkül használható kereskedelmi projektekben is. (Fermigier, 1998)

Rövid scriptekhez és több tízezer soros komplex projektekhez egyaránt használható. Szintaxisa nagyon egyszerű, nagyon tömör, mindemellett jól olvasható programok írhatók vele. Miután a kód hossza lerövidül, ezért a többi programozási nyelvhez képest egy Python

program rövidebb fejlesztési időt és lényegesen egyszerűbb karbantartást igényel. (Fermigier, 1998)

A Python a kódblokkok meghatározásához behúzást használ, ami a kódsor elején lévő szóközökre vonatkozik. A behúzás nem csak az esztétikát, az olvashatóságot szolgálja, ez a nyelv szerves része. A szóközök számát mindenki maga dönti el, a legáltalánosabb négy szóköz használata, de legalább egynek kell lennie. Viszont konzisztensen ugyanannyi szóközt kell használni egyazon kódblokkban, különben a Python hibát fog adni. (http3, 2023)

A Python, mint minden más programozási nyelv, kezeli a különféle adattípusokat és adatstruktúrákat, mint számok (egész/int, lebegőpontos/float...), szöveg (string), logikai értékek (boolean), listák (list), szótárak (dictionary) stb. (Zumstein, 2021). A változók a nevekhez értékek hozzárendelésével jönnek létre. A változók létrehozása dinamikus, így nem kell kifejezetten deklarálni egy változó adattípusát. Ugyanakkor lehetőség van a felhasználók számára egyéni adattípusok létrehozására is osztályok segítségével. (http3, 2023, Zumstein, 2021)

A Python rendelkezik kommentelési lehetőséggel, ami a kód magyarázatát, a kódon belüli dokumentációt szolgálja. A megjegyzések kettőskeresztrel (#) kezdődnek, a Python az így kezdett sor egészét megjegyzésként jeleníti meg, így az interpreter nem veszi azt figyelembe. (http3, 2023)

A Python tartalmazza a szokásos vezérlőstruktúrákat, például if utasításokat, ciklusokat (for és while) és kapcsolószerű (switch) struktúrákat (szótárak vagy if-elif-else segítségével). (Zumstein, 2021)

A Pythonban függvényeket a „def” kulcsszóval lehet létrehozni. Paramétereket vehetnek fel és értékeket adhatnak vissza. A függvény meghívásához a neve után zárójeleket kell tenni, amennyiben a függvény paramétereket vár, azt a zárójelben kell megadni. (http4, 2023)

A Python egy objektumorientált nyelv, így osztályokat és objektumokat használ a valós világ entitásainak és viselkedésének modellezésére. Az osztály olyan, mint egy objektumkonstruktor, vagyis egyfajta tervrajz az objektumok létrehozásához. Osztály létrehozásához a „class” kulcsszót kell használni. (http5, 2023)

A Python támogatja a file input és output műveleteket, megkönnyítve a külső fájlokkal és adatokkal való munkát. Számos funkcióval rendelkezik a fájlok létrehozásához, olvasásához, frissítéséhez és törléséhez. (http6, 2023)

Ez a programozási nyelv robusztus kivételkezelési mechanizmust is biztosít, számos beépített kivétellel a hibák észlelésére és kezelésére. (http7, 2023)

Különböző haladó koncepciókat is tartalmaz a hatékonyabb és elegánsabb kód írásához, mint például a List Comprehension (tömör, jól olvasható kódot biztosít listák létrehozására), a Lambda-függvények (más néven anonim függvények, lehetővé teszik kicsi, tömör névtelen függvények létrehozását), a Generátorok (hatékony iterációt tesznek lehetővé nagy adathalmazokon), vagy a Dekorátorok (segítségükkel módosítható vagy kiterjeszthető a függvények vagy osztályok viselkedése a kódjuk megváltoztatása nélkül). (http8, 2023)

A Python legújabb fő verziója a Python 3, azonban a Python 2, bár a biztonsági frissítéseken kívül mással már nem frissítik, még mindig meglehetősen népszerű (http2, 2023). A Python legfrissebb verziója dolgozatom írásának idején a Python 3.12.0 (docs.python.org, 2023).

2.3.1 Könyvtárak

A Python programozási nyelv igen kiterjedt és széleskörű standard könyvtárral rendelkezik. A standard könyvtár tartalmaz adattípusokat, amelyeket egyébként a nyelv magjának tekintenek (pl. számok és listák), beépített függvényeket és kivételeket. Azonban vannak olyan feladatok, amelyek programozása nehézkes, vagy lassú, ha csak standard könyvtárra támaszkodunk. A standard könyvtárat kiegészítik az egyéb - mások által - megírt publikus könyvtárak (modulok), amiket a használatukhoz először le kell tölteni, importálni kell. A Python Package Index, rövidítve PyPI a harmadik féltől származó szoftverek hivatalos adatbázisa (repository), ami alapján jelenleg több, mint 490.000 külső Python-csomag áll rendelkezésre. (http9, 2023, pypi.org, 2023)

Néhány fontosabb könyvtár és keretrendszer:

- NumPy: Egy numerikus és tudományos számítástechnikai könyvtár, amely támogatja a többdimenziós tömböket és mátrixokat, valamint számos matematikai függvényt a műveletekhez. (http10, 2023)
- Pandas: Adatkezelési és -elemzési könyvtár, amely rugalmas, magas szintű adatstruktúrákat és különböző elemzési eszközöket biztosít. (http10, 2023)

- Matplotlib: Egy népszerű könyvtár a numerikus adatok ábrázolásához, statikus, animált és interaktív vizualizációk készítéséhez, mint például kördiagramok, hisztogramok, grafikonok. (http10, 2023)
- Scikit-Learn: Gépi tanulási (machine learning) könyvtár, amely eszközöket biztosít például osztályozáshoz, lineáris regresszióhoz, klaszterezéshez. (http10, 2023)
- TensorFlow és PyTorch: Mély tanulási (deep learning) keretrendszerek, amelyeket széles körben használnak neurális hálózatok fejlesztésére és betanítására. (http10, 2023)
- Django: Magasszintű webes keretrendszer, amely leegyszerűsíti a webfejlesztést azáltal, hogy robusztus eszközkészletet biztosít webes alkalmazások létrehozásához. (http11, 2023)
- Flask: Könnyű, webes mikrokeretrendszer, amely népszerű kis és közepes méretű webalkalmazások és API-k készítésére. (http11, 2023)
- SQLAlchemy: Széles körben használt Object-Relational Mapping (ORM) könyvtár a Python számára, amely leegyszerűsíti az adatbázisokkal való interakciót. (http11, 2023)
- OpenPyXL: az egyik legszélesebb körben használt könyvtár az Excel-fájlok Pythonban történő kezelésére. (http12, 2023)
- Requests: Népszerű könyvtár a HTTP-kérésekhez, a webszolgáltatásokkal és API-kkal való interakciók egyszerűsítésére. (http11, 2023)
- BeautifulSoup: HTML és XML dokumentumokból történő adatgyűjtésre (web scraping) és elemzésére szolgáló könyvtár. (http13, 2023)
- Twisted: Eseményvezérelt hálózati motor, amelyet hálózati alkalmazások építésére használnak. (http14, 2023)
- OpenCV: Az Open Source Computer Vision Library egy nyílt forráskódú számítógépes látás- és gépi tanulási szoftverkönyvtár. Felgyorsítja a gépi érzékelés használatát a kereskedelmi termékekben. (http15, 2023)
- SciPy: Nyílt forráskódú könyvtár, melyet magas szintű számításokhoz és szimulációkhoz használnak tudományos és mérnöki területeken. (http16, 2023)
- SymPy: Szimbolikus matematikához szükséges Python könyvtár, amely különböző funkciókat kínál, az alapvető szimbolikus aritmetikától az algebrán, diszkrét matematikán át, egészen a kvantumfizikáig. (http16, 2023)
- Pygame: Játékfejlesztési könyvtár, amely eszközöket biztosít 2D játékok létrehozásához. (http17, 2023)

2.3.2 OpenPyXL

Az OpenPyXL egy ingyenesen elérhető, nyílt forráskódú Python könyvtár Excel-fájlok kezelésére, ami a Python standard könyvtár ezirányú hiányosságai miatt lett létrehozva. Eredetileg a PHPExcel könyvtáron alapult. (openpyxl.readthedocs.io/en/stable/, 2023)

Az OpenPyXL elsősorban az .xlsx fájlformátumra összpontosít, ami a Microsoft Excel 2007 és újabb verzióiban használt Excel-fájlok alapértelmezett formátuma. Nem támogatja a régebbi .xls formátumot. Bár a Microsoft Excel szoftvere a leggyakrabban használt táblázatkezelő, vannak ingyenes alternatívák, mint a LibreOffice Calc és az OpenOffice Calc, amelyek működnek az Excel .xlsx fájlformátumával, ami azt jelenti, hogy az openpyxl modul ezekből az alkalmazásokból is tud dolgozni táblázatokon. (Sweigart, 2019)

Az OpenPyXL segítségével új Excel-táblázatokat lehet létrehozni, de alkalmas meglévő táblázatok olvasására, írására, szerkesztésére is. Ez magában foglalja a munkalapok, cellák, sorok, oszlopok és különféle formázási beállítások hozzáadását és módosítását. Segítségével cella értékeket lehet bekérni, ezeken számításokat-, függvényeket lehet használni, és különböző adattípusokkal is lehet dolgozni, beleértve a számokat, szöveget, dátumokat és képleteket. (openpyxl.readthedocs.io/en/stable/, 2023)

Cellák- és munkalapok formázásra is jól használható az OpenPyXL, beleértve a stílusokat, a betűtípusokat, a szegélyeket, a színeket és az igazítást. Támogatja többek között diagramok és grafikonok használatát, adatérvényesítési szabályok beállítását adatellenőrzésre, cellák egyesítését és felosztását, a munkafüzeten belüli munkalapok kezelését, beleértve azok hozzáadását, törlését és átrendezését. (Sweigart, 2019)

2.3.3 PyCharm

A kód megírásához a PyCharm-ot választottam, ami egy rendkívül népszerű integrált fejlesztői környezet (integrated development environment, IDE) a Python fejlesztésekhez. A PyCharm-ot a különféle fejlesztői eszközök és IDE-k létrehozásáról ismert JetBrains szoftverfejlesztő cég alkotta meg. Kifejezetten arra készült, hogy átfogó és hatékony fejlesztői környezetet biztosítson a Python fejlesztők számára. (http18, 2023)

A PyCharm főbb jellemzői és összetevői - amik miatt annyira népszerű – többek között:

- Professional és Community Edition: Ingyenes és nyílt forráskódú közösségi kiadásban és fizetős professzionális kiadásban is elérhető, ami fejlettebb funkciókkal rendelkezik.
- Kódszerkesztő: A PyCharm hatékony kódszerkesztőt (code editor) kínál olyan funkciókkal, mint a kódkiegészítés, a szintaxis kiemelés és az intelligens kódelemzés, ami nagy segítség a Python-kód hatékonyabb írásában.
- Debugger: Tartalmaz egy beépített hibakeresőt (debugger), amely segít megtalálni és kijavítani a Python-kód hibáit. (http19, 2023)

- Verziófelügyeleti integráció: A PyCharm támogatja a népszerű verziókezelő rendszereket, mint például a Git.
- Projektnavigáció: Eszközöket kínál az egyszerű projektnavigációhoz és -kezeléshez, megkönnyítve a nagy kódbázisokon való munkát.
- Kódrefaktorálási eszközök: A PyCharm egy sor átalakítási eszközt kínál, amelyek segítségével karbantartható, javítható, átláthatóbbá tehető a kódbázis, miközben megőrzi funkcionalitását.
- Intelligens kódelemzés: A PyCharm statikus kódelemzést, kódellenőrzéseket és intelligens javaslatokat tartalmaz, ami nagy segítséget jelent a tiszta és hatékony Python-kód írásában.
- Modulok és bővítmények: Nagyon sok beépülő modult és bővítményt tartalmaz, amelyek hozzáadásával bővíthető a funkcionalitás és a különböző programozási nyelvek és technológiák támogatása.
- Webfejlesztési támogatás: Webfejlesztési funkciókat tartalmaz, így megfelelő választás a Python-alapú webes keretrendszerekkel dolgozó webfejlesztők számára.
- Django-támogatás: A PyCharm erősen támogatja a Django-t, egy népszerű Python-alapú webes keretrendszert, beleértve a sablonokat, felügyeleti parancsokat stb.
- Adatbázis-eszközök: Integrálódik különféle adatbázisokkal, lehetővé téve a fejlesztők számára, hogy adatbázisokkal és SQL-lekérdezésekkel dolgozzanak az IDE-n belül. (http19, 2023)

3 A vállalat bemutatása

Az U-Partner National Kft. többféle termékkel foglalkozó nagykereskedő vállalat, de a fő profilja a munkavédelmi lábbeli és ruházat. A cég az olaszországi U-Group Srl. munkavédelmi lábbeliket és munkaruházatot gyártó cégcsoport képviselője, a U-Power és Lupos márkák hivatalos magyarországi disztribútora, melyek nyugat-európai szintű márkák.

A magas minőségű, felsőkategóriás, egyedi technológiákkal készített, olasz dizájn munkavédelmi lábbelik forgalmazása mellett a cég egyéb munkaruházati cikkeket és kiegészítőket is forgalmaz.

A vállalat 2014-ben alapult, és azóta folyamatosan növekszik. 2015-ben még nagyjából 50 millió, 2019-re 500 millió, míg az elmúlt két évben már 1 milliárd feletti volt az éves bevétele. A cégnek 2 ügyvezetője van, és kevés alkalmazottat foglalkoztatnak. A vállalat két telephellyel rendelkezik, ez egyik Budapesten, a másik Debrecenben található. A budapesti telephelyen 3 raktári/logisztikus kolléga és velem együtt 2 irodai alkalmazott van. Az új, debreceni telephelyen egyelőre 1 raktári és 1 értékesítő kolléga dolgozik.

A cégnek mindkét telephelyén van raktára. Az egyik raktár Budapesten a XX. kerületben található, itt van mellette az iroda is, ahol dolgozom. A budapesti raktár többszintes, modern raktár, ahol több ezer védőlábbeli van készleten. A budapesti irodánkban egy jól felszerelt, több védőlábbeli és védőruházati modellt is felvonultató, igényesen kialakított bemutatóterem is található.

Nemrég nyílt meg a második raktár, néhány hónapja működik a cég debreceni telephelyén, amely egy kisebb, egyszintes raktár. Jelenleg a debreceni raktár még nem üzemel teljes kapacitással, a legkedveltebb, legtöbbet értékesített munkavédelmi lábbelikből van méretsoronként néhány pár itt készleten.

Az olasz gyártó Magyarországra kizárólag nekünk értékesít, az U-Partner National Kft. pedig viszonteladó és végfelhasználó partnereket egyaránt kiszolgál, de kizárólag cégeknek, B2B területen értékesít. Partnereink többsége magyar, de vannak külföldi vevőink, beszállítóink is. Vevőink a legkülönbözőbb profilokkal rendelkeznek, vannak köztük például gyártó-, termelő vállalatok, logisztikai-, élelmiszeripari-, vegyipari-, építőipari cégek.

A vállalatnak széles termékválasztéka van, jelenleg csak U-Power védőlábbelikből több, mint 400 fajta modell van. Nagy árukészlete van Magyarországon elsősorban a védőlábbelikből, így a partnerek igényeit hamar ki tudja szolgálni, de az olasz gyártótól is rendelünk heti rendszerességgel, és akkor az egyedi rendelésekhez, a magyar készleten nem tartott modelleket is berendeljük.



2. ábra: U-Power logó. Forrás: olasz gyártói adatbázis alapján.

A vállalat két fő márkája a U-Power és a Lupos, habár főleg a U-Power márkára fókuszálunk. A U-Power védőlábbelik több termékcsaláddal rendelkeznek, vannak olyanok, melyek egyszerűbb, alap modelleket tartalmaznak, és vannak olyanok is, amik professzionális, csúcstechnológiás modelleket. Rendkívül sokféle munkaterületre alkalmazott és sok fajta védelmi kategóriával ellátott U-Power védőlábbeli van. Több, teljesen egyedi technológiát is alkalmaz a U-Power, mint például a Save & Flex® PLUS talpátszúrás elleni védelem, ami egy kompozit szövetanyag, a klasszikus acél középtalphoz képest könnyebb, rugalmasabb és biztonságosabb is. Vagy a WingTex® légcsatornás textil bélés, amely optimális szellőzőképességét biztosít speciális mikrocellák segítségével.

Több modell is van, melyeket rendszeresen forgalmazunk, de három termékcsalád cipői jelenleg nagyon keresettek, ezek a Red Industry, Red Lion és Red Leve.

A Red Industry termékcsalád modelljeiben egyedi, hogy egy speciális testtartásjavító és álláskönnyítő, WOW2® talpbetéttel vannak ellátva, amely egy egyedülállóan kényelmes anatómiai, fáradtságcsökkentő talpbetét. A legnépszerűbb, általunk legtöbbet forgalmazott Red Industry-s modellek: Alicante, Demon, Lisboa, Nero, Scandy, Madrid, Londra.

A Red Lion termékcsalád modelljeiben speciális, Infinergy® technológiát alkalmaznak, ami egy dinamikus párnázás, egy energia visszaadó rendszer, amely által csökken a fáradtságérzet a munkanap folyamán. Érdekesség, hogy az Infinergy® technológiát eredetileg Adidas

futócipőknél alkalmazták, de a U-Group Srl. (az olasz gyártó) később felvásárolta. A cégünk által legtöbbet eladott Red Lion modellek: Active, Push, Emotion, Point, Carbon.

A Red Leve egy viszonylag új termékcsalád, ami a védőlábbelik piacán egyedülálló módon szuperkönnyű munkavédelmi cipőket vonultat fel, új generációs talppal és ultrakönnyű talpbetéttel, amelyek nagymértékben csökkentik a cipő össztömegét. Az általunk legtöbbet forgalmazott Red Leve modellek: Bruce, Frank, Mike, Adam, Ben.



3. ábra: 3 népszerű U-Power védőlábbeli. Balról jobbra: Scandy, Londra, Active. Forrás: saját szerkesztés olasz gyártói adatbázis alapján.

A munkavédelmi cipőkből még egy másik márkát is forgalmazunk, de ennek nem vagyunk hivatalos magyarországi disztribútora. Ez is olasz márka, a Giasco, ebből a Franklin, Volt és Edison modelleket forgalmazzuk, amelyek elektromos védelemmel ellátott, kifejezetten villanszerelőknek szánt védőlábbelik.

A munkavédelmi termékek mellett kisebb részben egyéb termékeket is értékesítünk. Ide tartoznak többek között higiéniai termékek, tisztítószer, egyszerhasználatos kesztyűk. Van két saját márkás egyszerhasználatos kesztyűnk is, ezek UPN márkanévvel ellátott nitril, púdermentes kesztyűk, az egyik hagyományos, a másik lebomló formában. A Covid időszak alatt nagy mennyiségben értékesített a vállalat maszkokat, egyszerhasználatos kesztyűket, a járvány lecsengésével azonban ezen termékek értékesítése visszaesett.

Én 2020. novembere óta dolgozom a cégnél, egészen idén nyár elejéig, mint kereskedelmi- és online marketing munkatárs. Ezen a vonalon eléggé sokrétűek a feladataim, beszerzési-, értékesítési-, irodai-, és marketing feladatok egyaránt beletartoznak. A vezetőség tudott arról, hogy egy informatikai képzésre járok, amihez szakmai gyakorlatot kell teljesíteni, és ezért felajánlották nyáron, hogy ezt náluk teljesíthessem, és ezért kibővítik a munkakörömet informatikai feladatokkal. Így 2023. júliustól, mint kereskedelmi-, online marketing- és informatikus munkatárs dolgozom az U-Partner National Kft.-nél.

4 Szakmai gyakorlaton végzett feladatok

4.1 Rendszergazdai feladatok

A kiépített hálózatot tekintve a vállalatnál van fizikai és vezeték nélküli hálózat is. A fizikai hálózat UTP CAT5e fali kábelekkel van behúzva. A hálózatot kiszolgáló tűzfal BSD alapú tűzfal szerver, ez biztosítja a kliens gépeknek és szervereknek a hálózatot és az internet elérését.

A logikai hálózatot tekintve tipikus C-osztályú IP cím hálózat van, 24-es alhálózati maszkkal (1-től 254-ig oszthatóak ki a címek). DHCP range van beállítva 50-től 200-ig, ezen vannak a szerverek és a nyomtatók. A tűzfalon ICMP szabályok vannak beállítva, Geo-IP, hogy külföldről ne tudják támadni a belső rendszereket, csak magyarországi IP címek vannak engedélyezve, ClamAV-os víruskereső motor stb.

3 szerver van, a fő szerver a Windows 2016-os szerver, Dell R620-as, Dual Xeon processzorral felszerelt, 256 Gb RAM-mal. Ezen van a vállalat ügyviteli rendszere, illetve a fájl szerver is, hálózati mappák vannak létrehozva, specifikus jogosultsági szintekkel. Van egy R620-as szerver, amin egy Debian 10-es szerver fut, ezen van a mentés (UrBackup), ami egy nyílt forráskódú mentési rendszer. Napi inkrementális mentés van beállítva 5 óránként, teljes file backup 3 naponként. Van egy másodlagos mentés is, ami Synology NAS-ra történik, ez a szervertől eltérő helyen van tárolva, így, ha valamelyik épület leégne például, akkor is helyre lehet állítani a rendszert.

A vezeték nélküli hálózatot MikroTik wAP routerrel oldjuk meg, ezen 2,4 és 5 GHz-es hálózat van engedélyezve. Van egy Hotspot vendég hálózata is, ami tűzfallal van elrejtve a belső hálózattól.

Víruskeresőből Bitdefender Small Office Security van a kliens gépekre és Bitdefender GravityZone van a szerverre telepítve. A kliens gépeken Windows 10-es operációs rendszer fut.

A szakmai gyakorlat alatt rendszergazdai feladataim is voltak. Többek között az új raktári kolléga gépét beüzemeltem, a szükséges alkalmazásokat, szoftvereket telepítettem (pl.

Microsoft Office, Bitdefender vírusirtó, Zebra Designer címkenyomtató program...).

Létrehoztam az új felhasználót, beállítottam a jelszavát, jogosultságait stb.

Távoli eléréshez OpenVPN-t használunk, ezt is beállítottam neki. Hardver eszközöket is beüzemeltem, mint például Zebra tekercses címkenyomtató. Jelenleg 5 lapnyomtató és 2 címkenyomtató van ezzel együtt használatban.

Feladatom volt a felhasználók informatikai támogatása, szoftverfrissítések elvégzése, az esetlegesen felmerült hibák, problémák kezelése is.

4.2 SQL lekérdezések

Szakmai gyakorlatom alatt adatbázis-kezelői feladatokat is végeztem. Miután a cégnél használt ügyviteli rendszer már elég régi volt (Revolution Iroda++), régóta tervbe volt véve a lecserélése. Még nyáron megállapodtak az új ügyviteli rendszerről (Revolution Deep), aminek beüzemelése szeptember végére volt tervezve (dolgozatom írásának idejére ez már megtörtént). Az én feladatomból a régi rendszerből a különféle adatok (vevők, szállítók, cikkek, árak...) lekérdezése és az új rendszernek megfelelő formátumra hozása, illetve az ügyvezetők kérésének megfelelő formázása. Ehhez különféle SQL lekérdezéseket, feltételeket, szűréseket használtam.

Az adatbázisban először végig kellett nézni a táblákat, adattisztítást kellett végezni, ami nagyon sok időt vett igénybe. Például végig néztem a cikktörzs adatokat, hogy vannak-e már nem használt, felesleges cikkek, és ezeket törölni kellett, hogy az új adatbázisba felesleges adatok ne kerüljenek át. Az adatok átvizsgálása során az ilyen cikkeket átneveztem, az eredeti név helyére „ZZZ” betűcsoportot tettem, hogy aztán könnyen törölhetőek legyenek.

```
DELETE FROM cikk_karton WHERE megnevezes_a='ZZZ';
```

A régi rendszerben a cikktörzsadatoknak volt egy Megnevezés (A) és Megnevezés (B) mezője, de az új rendszerben egy megnevezés mező van. Ezért át kellett alakítanom a cikk neveket úgy, hogy ahol volt valamilyen adat a Megnevezés (B) mezőben, az átkerüljön a Megnevezés (A)-ba, egy szóközzel utána fűzve. Az erre szolgáló kód:

```
UPDATE cikk_karton
SET megnevezes_a = CONCAT_WS(' ', megnevezes_a, megnevezes_b),
    megnevezes_b = NULL
WHERE megnevezes_b IS NOT NULL;
```

A cikkek forintos listaárain változtatni kellett. A cikk kartonokban két listaár volt tárolva, a forintos és eurós, és a forintos az eurósból volt számolva. Korábban 415-ös árfolyammal voltak átszámolva, ezt kellett átszámítani 400 forintos árfolyammal.

```
UPDATE cikk_karton
SET listaar = listaar_EUR * 400;
```

A fentiekhez hasonlóan jó pár adattisztító, formázó, átszámoló SQL kódot alkalmaztam az adatbázis különféle tábláira.

4.3 Python programozási feladat

A munkahelyemen évente többször kapunk új árlistákat az olasz gyártótól (3 külön árlista: U-Power védőlábbeli, U-Power védőruházat és Lupos védőlábbeli, mindegyikben többszáz modell van), és ezekből nekem a feladatom a vevői árlisták legenerálása. A különféle vevőkategóriákra különböző sávok kedvezményrendszerét használunk. Jelenleg 11 fajta árszint van, ebből kell eurós és forintos változatot is csinálnom, és ezt a 3 külön gyártói árlistára, így összesen 66 Excel táblát kell csinálnom. Ezt a folyamatot szerettem volna meggyorsítani, automatizálni, amire egy programot írtam Python programnyelven, PyCharm fejlesztői környezetben, OpenPyXL könyvtár használatával. Maga a program kiszámolja a kedvezményes árakat és felülírja vele a listaárakat, törli a vevők számára nem releváns oszlopokat, formázza a megmaradt adatokat (betűtípus, méret, első sor rögzítése, árakhoz pénznem formátum EUR/HUF beállítása) és elmenti ezt a legenerált 66 vevői árlistát a megfelelő fájlneven. A 33 forintos árlistában az árak az ügyvezető által megállapított, rögzített árfolyamon számítnak át. Az eredeti 3 gyártói árlistát nem változtatja meg a program.

Az alábbiakban részletesen bemutatom az elkészített programomat. A kód megírásához segítségképpen használtam a hivatalos Python dokumentációt (docs.python.org, 2023), az OpenPyXL dokumentációt (openpyxl.readthedocs.io/en/stable/, 2023) és a W3Schools Python Tutorial adatbázist (w3schools.com/python/, 2023).

Jelen záródolgozatomban a céginformációk bizalmas kezelésére tekintettel a programban eredetileg használt vevői kategóriakódok elnevezéseit és az alkalmazott kedvezmény mértékeket megváltoztattam. Ez a változtatás a program lényegi működését egyáltalán nem befolyásolja.

Miután a programomhoz Excel táblákat kell kezelni, manipulálni – és az ehhez szükséges eszközök nincsenek meg a Python standard könyvtárában –, ezért szükséges külső könyvtár (modul) használata. Ezt többféleképp is meg lehet tenni, de véleményem szerint erre a feladatra a legalkalmasabb az OpenPyXL modul, így én ezt választottam programom elkészítéséhez.

Először tehát telepíteni kell az OpenPyXL modult, amit a PyCharm fejlesztői környezetben a View / Tool Windows / Python Packages menüvel lehet a legegyszerűbben megtenni. Itt kikerestem az OpenPyXL könyvtárat, és installáltam.

Egy könyvtár telepítése után importálni is kell azt a könyvtárat, hogy a tartalma használható legyen a programban. Ezt az import kifejezéssel lehet megtenni.

Magát az OpenPyXL könyvtárat a következőképpen lehet importálni:

```
import openpyxl
```

Ahhoz, hogy a későbbiek során az Excel táblák oszlopaire azok betűjelével tudjak hivatkozni, importálni kellett még a `get_column_letter` függvényt az `openpyxl.utils` modulból. Ezzel a függvénnyel egy numerikus oszlopindexet át lehet konvertálni a megfelelő betűalapú oszlopcímkévé. Így a későbbiekben az egyes oszlopokra már a betűjelükkel is lehet hivatkozni (például 1 helyett A-val). A kód az alábbiak szerint alakul:

```
from openpyxl.utils import get_column_letter
```

A táblázatban alkalmazott formázásokhoz importálni kell többféle osztályt az `openpyxl.styles` modulból. Ezek az `Alignment` (Igazítás: lehetővé teszi a cellán belüli szövegigazítás szabályozását), a `Border` (Szegély: a cellák szegély tulajdonságainak meghatározására szolgál), `Side` (Oldal: a cellaszegély egyes oldalainak tulajdonságai megadására szolgál, így a cella bal, jobb, felső vagy alsó szegélye is beállítható), `Font` (Betűtípus: a cellán belüli szöveg betűtípus-stílusának szabályozására szolgál, amivel beállítható például a betűméret, a félkövérség, a vagy a betűszín). Az importálás kódja a következő:

```
from openpyxl.styles import Alignment, Border, Side, Font
```

A szükséges modulok, modulelemek importálása után a vevő kategóriák és a hozzájuk tartozó kedvezmények meghatározása következett. Ehhez egy `customer_categories` elnevezésű szótárat (dictionary) hoztam létre. 11 vevői kategória van, az egyes vevői kategóriákat jelen esetben betűk jelölik A-tól K-ig. A hozzájuk tartozó kedvezmény mértékek rendre 10%, 15%, 20%, 25%, 30%, 35%, 40%, 42,5%, 45%, 47,5%, 50%. Miután az egyes vevői árjegyzékekben az árat a gyártói listaárból számolom vissza, ezért az egyes kategóriákhoz a kedvezménnyel

csökkentett szorzót rendelem hozzá, ami 10% esetén 0,9, 15% esetén 0,85 stb. Így a következő szótárat hoztam létre:

```
customer_categories = {  
    'A': 0.9,  
    'B': 0.85,  
    'C': 0.8,  
    'D': 0.75,  
    'E': 0.7,  
    'F': 0.65,  
    'G': 0.6,  
    'H': 0.575,  
    'I': 0.55,  
    'J': 0.525,  
    'K': 0.5,  
}
```

A gyártói árlistákat mind euróban kapjuk meg. Nekünk viszont nem csak eurós, hanem forintos vevői árlistákat is kell csinálni (több forintos partnerünk is van). Emiatt minden gyártói árlistából - a 11 eurós mellett – 11 forintos árlistát is kell generálni. A jelenlegi árjegyzékek fix 400 forintos árfolyamon számítnának át. Így bevezettem egy változót, `eur_to_huf` névvel, amiben ezt a 400 forintos szorzót tárolom:

```
eur_to_huf = 400
```

A gyártótól az árlisták, amiket kapunk angol nyelven vannak, a fájl neve és a tartalma is, ezeket – miután többségében magyar partnereink vannak – a vevői árlistákban magyarosítani kell, át kell nevezni. A fájl neve tartalmazza a márkanévet (U-Power vagy Lupos), attól függően, hogy cipőkre vagy ruhákra vonatkozik a védőlábbeli vagy védőruházat kifejezést, a vevői árjegyzék kifejezést, utána az adott árjegyzékhez tartozó kategória kódot, az alkalmazott devizanemet, és végül az évszámot, mindezeket alávonással elválasztva. Így például egy vevői árjegyzék a következő elnevezéssel jöhet létre: `U-Power_védőlábbeli_vevői_árjegyzék_E_EUR_2023.xlsx`.

Az eredeti Excel-fájlok elnevezései, amiket a gyártótól kapunk: `upower_shoes.xlsx`, `upower_clothes.xlsx` és `lupos_shoes.xlsx`. Ezeket a fenti módon át kell nevezni. Emiatt létrehoztam egy `product_names` elnevezésű szótárat, ami megfelelteti az eredeti fájlok neveit, a vevői árjegyzékek elnevezésének első részével.

```
product_names = {
    "upower_shoes.xlsx": "U-Power_védőlábbeli_vevői_árjegyzék",
    "upower_clothes.xlsx": "U-Power_védőruházat_vevői_árjegyzék",
    "lupos_shoes.xlsx": "Lupos_védőlábbeli_vevői_árjegyzék",
}
```

Ahhoz, hogy az eredeti fájlokat később ciklusokban iterálni lehessen, és különféle műveleteket lehessen ezekkel a fájlokkal végrehajtani, létrehoztam egy `original_files` elnevezésű listát, ami az eredeti fájlok elnevezéseit tartalmazza.

```
original_files = ["upower_shoes.xlsx", "upower_clothes.xlsx",
                  "lupos_shoes.xlsx"]
```

A következő lépésként csináltam egy `for` ciklust, ami kezdésnek az `original_files` listán ismétlődik, és a listában szereplő minden `original_file` esetén megpróbálja lekérni a megfelelő `product_name`-et a `product_names` szótárból a `get()` módszerrel. A kódrészlet ehhez:

```
for original_file in original_files:
    product_name = product_names.get(original_file)
```

A fenti kód egy nagy, többszintű `for` ciklus kezdete, gyakorlatilag a programom többi része mind ez alá kerül be.

Ezután be kell tölteni az aktuális eredeti fájlt, amihez az `OpenPyXL` könyvtárat kell használni. Ezt a következő kóddal lehet megtenni:

```
workbook = openpyxl.load_workbook(original_file)
```

Ez a kód elengedhetetlen lépés a Pythonban az `OpenPyXL` használatával végzett Excel-fájlokkal való munka során, mert hozzáférést biztosít az Excel-fájl adataihoz és szerkezetéhez, lehetővé téve a betöltött munkafüzeten a különféle feladatok és műveletek végrehajtását.

A következőkben egy újabb ciklus következik, ami minden ügyfélkategóriához külön Excel-munkafüzeteket hoz létre, egyet euróban (EUR) és egyet forintban (HUF), meghatározott fájlnevek használatával. A ciklus a `customer_categories` szótár elemei (kulcs-érték párok) fölött iterál, amely az ügyfélkategóriákat (`category`, pl. 'A', 'B', 'C') a hozzájuk tartozó kedvezmény szorzókkal (`discount`, pl. 0,9, 0,85, 0,8 stb.) társítja. Két fájlnev jön létre minden ügyfélkategóriához: `eur_filename` (az eurós fájlnev), `huf_filename` (a forintos fájlnev). A

szövegek formázott kiírására f-string-et használok. A megfelelő product_name és category mögé hozzáfűzi eurós fájl esetén az _EUR_2023 szöveget, forintos fájl esetén a _HUF_2023 szöveget. Például, ha a ciklus éppen az 'A' kategóriát dolgozza fel, és a product_name "U-Power_védőlábbeli_vevői_árjegyzék", akkor az U-Power_védőlábbeli_vevői_árjegyzék_A_EUR_2023.xlsx és az U-Power_védőlábbeli_vevői_árjegyzék_A_HUF_2023.xlsx fájlneveket generálja le az eurós, illetve a forintos munkafüzetekhez. A kódrészlet ehhez:

```
for category, discount in customer_categories.items():
    eur_filename = f"{product_name}_{category}_EUR_2023.xlsx"
    huf_filename = f"{product_name}_{category}_HUF_2023.xlsx"
```

A következő részben új munkafüzeteket hozok létre az eredeti munkalap másolatával. Ez a kód létrehoz egy új eurós munkafüzetet, másolja az adatokat az eredeti munkafüzetből (némi szövegtisztítással), és feltölti az eurós munkafüzetet a feldolgozott adatokkal. Először az openpyxl könyvtár segítségével egy eurós (EUR) Excel-munkafüzetet hoz létre, majd egy munkalap objektumot (eur_worksheet) az újonnan létesített eur_workbook aktív munkalapjának elérésével. Ezután egy ciklus indul, amely az eredeti munkafüzet sorai fölött iterál. Az iter_rows() metódus segítségével kéri le a sorokat az aktív munkalapról. A Values_only=True paraméter azt jelzi, hogy itt csak a cellák értékei érdekesek, nem pedig azok formázása vagy képletei. Ezután adatsorokat fűz hozzá az eurós munkalaphoz úgy, hogy egy List Comprehension segítségével feldolgozza a sor minden celláját, és új listát hoz létre. Az egyes szöveg formátumú cellákból az esetleges felesleges szóközöket törli a strip() metódussal (a gyártói árlistákban sokszor több felesleges szóköz is szerepel a termékek nevei után), majd az eredményül kapott megtisztított adatokat tartalmazó listát az eur_worksheet.append() segítségével hozzáfűzi az eurós munkalaphoz. Ezt az egész kódblokkot természetesen a forintos verzióra is meg kell csinálni. Az ezekre vonatkozó kódrészlet:

```
eur_workbook = openpyxl.Workbook()
eur_worksheet = eur_workbook.active
for row in workbook.active.iter_rows(values_only=True):
    eur_worksheet.append([cell.strip() if isinstance(cell, str) else cell
    for cell in row])

huf_workbook = openpyxl.Workbook()
huf_worksheet = huf_workbook.active
for row in workbook.active.iter_rows(values_only=True):
    huf_worksheet.append([cell.strip() if isinstance(cell, str) else cell
    for cell in row])
```

A gyártói árlistákban vannak olyan oszlopok, amik a vevői árlistákban nem szükségesek, törlendőek. Az utolsó két (G és H) oszlopban vannak a vonalkódok és a VTSZ számok, ezeket törlöm az eurós és a forintos árlistákból is (a 7. oszloptól kezdve 2 oszlopot):

```
eur_worksheet.delete_cols(7, 2)
huf_worksheet.delete_cols(7, 2)
```

A következő részben a megmaradt oszlopok fejléceit nevezem át az egyes vevői árlistákban. Ehhez először létrehozok egy headers elnevezésű listát, ami tartalmazza a fejléc neveket. Ezután egy for ciklusban összepárosítom a fejléc neveket a megfelelő oszlopokkal, majd az adott oszlopban lévő első cellát átnevezem a fejlécre. Ezután egy másik for ciklussal végig megyek az egyes fejléceken, és formázom őket. Mindegyik fejléc középre zárt, félkövér és mindegyik oldalán normál szegéllyel határolt lesz. Ezt a kódrészletet szintén két változatban kell megcsinálni; külön az eurós és külön a forintos árlistákhoz:

```
# Fejlécek átnevezése
headers = ["Katalógus", "Kollekció", "Kód", "Megnevezés", "Méret", "Nettó ár"]
for col, header in zip(eur_worksheet.iter_cols(max_col=6, min_col=1), headers):
    col[0].value = header
    # Fejlécek formázása
    for cell in col:
        if cell.row == 1:
            cell.alignment = Alignment(horizontal="center")
            cell.font = Font(bold=True)
            cell.border = Border(left=Side(style="thin"),
                                right=Side(style="thin"), top=Side(style="thin"),
                                bottom=Side(style="thin"))

for col, header in zip(huf_worksheet.iter_cols(max_col=6, min_col=1), headers):
    col[0].value = header
    # Fejlécek formázása
    for cell in col:
        if cell.row == 1:
            cell.alignment = Alignment(horizontal="center")
            cell.font = Font(bold=True)
            cell.border = Border(left=Side(style="thin"),
                                right=Side(style="thin"), top=Side(style="thin"),
                                bottom=Side(style="thin"))
```

Ezek után az eurós és forintos árlisták munkalapjai formázásának beállítása következik. Egy for ciklus iterál az eurós és forintos munkalapok felett: átnevezi a munkalapot Árlista_2023-ra, rögzíti az első sorát, szint ad hozzá a munkalap megnevezés háttéréhez. Ezután egy újabb for

ciklus az egyes sorok felett iterál a második sortól kezdve, és az adott sor minden cellájára beállítja a középre zárást és normál szegélyeket. A kódrészlet a következőképpen alakul:

```
for sheet in [eur_worksheet, huf_worksheet]:
    sheet.title = "Árlista_2023"
    sheet.freeze_panes = "A2"
    sheet.sheet_properties.tabColor = "0000FF"
    for row in sheet.iter_rows(min_row=2):
        for cell in row:
            cell.alignment = Alignment(horizontal="center")
            cell.border = Border(left=Side(style="thin"),
right=Side(style="thin"), top=Side(style="thin"),
                                bottom=Side(style="thin"))
```

A következő részben az árakat kalkulálom ki. Az eurós árakat 2 tizedesre, míg a forintos árakat egész számra kell megadni. Először egy for ciklust alkalmazok az eurós árak meghatározására. Ez a ciklus az EUR-munkalap sorai fölött iterál, a második sortól kezdve (a fejlécek kivételével), és az első 6 oszlopra korlátozódik (ahol az adatok vannak), tehát az A–F oszlopok adatait tartalmazó sorokat dolgozza fel. Utána egy beágyazott for ciklus az aktuális soron belüli minden cella fölött ismétlődik, majd egy if feltétellel kizárólag a 6. (F) oszlopra alkalmazza a számításokat, ahol az árak találhatóak. A kalkuláció során felveszi az aktuális cella értékét, lebegőpontos (float) típusú számmá alakítja, megszorozza a kedvezménnyel, és az eredményt 2 tizedesjegyre kerekíti. Az adott cella értékét ezután olyan számformátumra formázza, amely két tizedesjegyet jelenít meg, és egy szóköz után hozzáfűz egy "EUR"-t, jelezve, hogy az érték euróban van megadva. A forintos ciklusnál majdnem ugyanez történik, annyi különbséggel, hogy itt az értékeket még megszorozza az eur_to_huf 400 forintos árfolyammal, az eredményt egész számokra kerekíti, egész számként is jeleníti meg, majd a számhoz egy "HUF"-ot fűz hozzá. A kód ehhez a részhez a következőképpen alakul:

```
for row in eur_worksheet.iter_rows(min_row=2, max_col=6,
max_row=eur_worksheet.max_row):
    for cell in row:
        if cell.column == 6:
            cell.value = round(float(cell.value) * discount, 2)
            cell.number_format = '#,##0.00 "EUR"'

for row in huf_worksheet.iter_rows(min_row=2, max_col=6,
max_row=huf_worksheet.max_row):
    for cell in row:
        if cell.column == 6:
            cell.value = round(float(cell.value) * discount * eur_to_huf,
0)
            cell.number_format = '#,##0 "HUF"'
```

Ezután az oszlopok szélességeit állítom be. Az oszlopokra automatikus méretezést állítok be, oly módon, hogy az oszlopszélességet automatikusan igazítom a tartalomhoz (az adott oszlop legszélesebb eleméhez), hogy minden kiférjen, és megfelelően jelenjen meg az adott oszlopban. Ehhez több egymásba ágyazott for ciklusra van szükség. A külső ciklus az eurós és a forintos munkalapok felett iterál. A belső for ciklus az aktuális munkalap oszlopai felett ismétlődik. Ebben először nullára állítja a maximális hosszúságot (`max_length`), lekéri az aktuális oszlop betűalapú oszlopcímekjét a `column_letter` változóba. Utána még egy belső for ciklus következik, ami az oszlop minden celláján ismétlődik. A `try-except`-tel lekezelem az esetleges hibákat (kivételeket), és ellenőrzi a cella szöveggé konvertált értékének hosszát (`len(str(cell.value))`), és összehasonlítja azt az aktuális `max_length` értékkel. Ha a cella tartalmának hossza nagyobb, mint az aktuális maximális hossz, akkor a `max_length` értéket az új maximális hosszra frissíti. Ezután kiszámítja az oszlop új beállított szélességét (`adjusted_width`) úgy, hogy kettőt ad hozzá a maximális hosszhoz (hogy legyen egy kis extra hely). Végül beállítja a munkalap aktuális oszlopjának szélességét az `adjusted_width`-re, biztosítva, hogy az oszlop tartalma teljes mértékben látható legyen. A kódrészlet a következő:

```
for sheet in [eur_worksheet, huf_worksheet]:
    for column in sheet.columns:
        max_length = 0
        column_letter = get_column_letter(column[0].column)
        for cell in column:
            try:
                if len(str(cell.value)) > max_length:
                    max_length = len(str(cell.value))
            except:
                pass
        adjusted_width = max_length + 2
        sheet.column_dimensions[column_letter].width = adjusted_width
```

Ezután a 6. (F) oszlopnak, amiben a számított eurós vagy forintos árak vannak, explicit állítom be az oszlopszélességet egy konkrét értékre (a forintos árak esetében kicsit szélesebbre).

```
eur_worksheet.column_dimensions['F'].width = 12
huf_worksheet.column_dimensions['F'].width = 13
```

A táblázatok jobb áttekinthetősége érdekében az eurós és forintos munkalapok minden második sorához egy kék háttérszínt adtam. Ehhez több egymásba ágyazott for ciklust alkalmaztam. A külső for ciklus az eurós és a forintos munkalapok felett iterál. A belső ciklus az egyes munkalapokon belüli sorokon ismétlődik. A második sortól kezdődik és a munkalap

utolsó soráig folytatódik. Az enumerate függvény a sorszám (i) nyomon követésére szolgál, és 1-től kezdődik. Ezután egy feltétel ellenőrzi, hogy az i sorszám páratlan -e (ha 2-vel osztjuk, a maradék 1), azaz minden második sort azonosít a munkalapon. Végül ezekre a beazonosított sorokra egy for ciklussal beállítottam egy világoskék háttérszínt. A kódblokk erre a részre:

```
for sheet in [eur_worksheet, huf_worksheet]:
    for i, row in enumerate(sheet.iter_rows(min_row=2, max_col=6,
max_row=sheet.max_row), start=1):
        if i % 2 == 1:
            for cell in row:
                cell.fill =
openpyxl.styles.PatternFill(start_color="DDEBF7", end_color="DDEBF7",
                             fill_type="solid")
```

Végül már csak el kell menteni az előzőekben létrehozott és módosított munkafüzetet minden tartalmával, formázásával és módosításaival együtt a megadott fájlba. Ha eurós árlistáról van szó, akkor az eur_filename paraméterben megadott fájlnévvel, ha forintsról, akkor a huf_filename paraméterben megadott fájlnévvel.

```
eur_workbook.save(eur_filename)
huf_workbook.save(huf_filename)
```

A program lefutása után végül legenerálja a szükséges, összesen 66 db vevői árjegyzéket.

5 Összegzés

Szakmai gyakorlatomat 2023 nyarán, munkahelyemen, az U-Partner National Kft.-nél végeztem, mely egy többféle termékkel foglalkozó, B2B területen tevékenykedő nagykereskedő cég. Fő profilja a munkavédelmi lábbeli és ruházat, a vállalat az olasz U-Power és Lupos márkák hivatalos magyarországi disztribútora.

Informatikai feladataim sokrétűek voltak, de főként rendszergazdai, adatbázis-kezelői és programozási feladatok adták a nagyrészt. A dolgozatomban ezt a három területet mutattam be, fókuszálva a programozási részterületre. Először bemutattam ezek elméleti hátterét, utána a vállalatot, végül a szakmai gyakorlaton végzett feladataimat részleteztem.

Rendszergazdai feladataim voltak például kliens számítógépek és egyéb eszközök telepítése, konfigurálása, felügyelete, a felhasználók informatikai támogatása, az esetlegesen felmerült hibák, problémák kezelése, a kliens számítógépeken használt alkalmazások telepítése, konfigurálása, szoftverfrissítések elvégzése, hálózati összeköttetés karbantartása, felügyelete, jogosultságok kezelése stb.

A cégnél használt ügyviteli rendszer lecserélése kapcsán adatbázis-kezelői feladatokat is elláttam. Az én feladatom volt a régi rendszerből a különféle adatok (vevők, szállítók, cikkek, árak...) kinyerése és az új rendszernek megfelelő formátumra hozása, illetve az ügyvezetők kérésének megfelelő formázása. Ehhez különféle SQL lekérdezéseket, feltételeket, szűréseket használtam.

Programozói tevékenységemben egy Excel-automatizáló programot csináltam vevői árlisták generálására, Python programozási nyelven, Pycharm fejlesztői környezetben, Openpyxl könyvtár használatával. A program 11 különféle vevői kedvezménykategóriára eurós és forintos vevői árlistákat generál, mindezt három gyártói árlistából, így összesen 66 Excel táblát hoz létre. Dolgozatomban részletesen bemutattam a programot és működését.

Nagyon örültem, hogy a szakmai gyakorlatot a saját munkahelyemen tudtam teljesíteni, felhasználhattam és kamatoztathattam a programtervező informatikus képzés során megszerzett tudásomat (operációs rendszerek, hálózatok, adatbázisrendszerek, programozás...). Bízom benne, hogy a jövőben is számítanak majd rám a munkahelyemen a korábbiakhoz hasonló és új informatikai feladatok elvégzése terén.

Irodalomjegyzék

Python 3. dokumentáció. Letöltés dátuma: 2023.10.27. Forrás:

<https://docs.python.org/3/>

Python Package Index adatbázis. Letöltés dátuma: 2023.10.30. Forrás: <https://pypi.org/>

OpenPyXL dokumentáció. Letöltés dátuma: 2023.10.27. Forrás:

<https://openpyxl.readthedocs.io/en/stable/>

W3Schools Python Tutorial adatbázis: Letöltés dátuma: 2023.10.28. Forrás:

<https://www.w3schools.com/python/>

Fermigier, S. (1998): A Python nyelv - bemutatja Stéfane Fermigier. Programmez! magazin. In: Gérard Swinnen: Tanuljunk meg programozni Python nyelven. Online oktatási segédlet. 6 p.

Letöltés dátuma: 2023.10.30. Forrás: <https://mek.oszk.hu/08400/08435/08435.pdf>

Haber, E.M. - Kandogan, E. - Maglio, P.P. (2011): Collaboration in System Administration. Communications of the ACM, 54(1), 46 p.

Holovács J. (2006): Adatbázis-kezelés (Elmélet, SQL, VFP, Oracle) Jegyzet. EKF, Számítástudomány és Információtechnológia Tanszék. Letöltés dátuma: 2023.11.01. Forrás:

<https://aries.ektf.hu/~holovacs/AB/>

Sweigart, A. (2019): Automate the Boring Stuff with Python: Practical Programming for Total Beginners. Letöltés dátuma: 2023.10.29. Forrás:

<https://automatetheboringstuff.com/chapter12/>

Zumstein, F. (2021): Python for Excel - A Modern Environment for Automation and Data Analysis. Sebastopol, California, United States of America. O'Reilly Media, Inc., 43-45 p., 52-63 p., 65-66 p.

http1: History of Python. Letöltés dátuma: 2023.10.28. Forrás:

<https://www.geeksforgeeks.org/history-of-python/>

http2: Python Introduction. Letöltés dátuma: 2023.10.28. Forrás:

https://www.w3schools.com/python/python_intro.asp

http3: Python Syntax. Letöltés dátuma: 2023.10.29. Forrás:

https://www.w3schools.com/python/python_syntax.asp

http4: Python Functions. Letöltés dátuma: 2023.10.28. Forrás:

https://www.w3schools.com/python/python_functions.asp

http5: Python Classes and Objects. Letöltés dátuma: 2023.10.30. Forrás:

https://www.w3schools.com/python/python_classes.asp

http6: Python File Open. Letöltés dátuma: 2023.10.31. Forrás:

https://www.w3schools.com/python/python_file_handling.asp

http7: Python Exception Handling. Letöltés dátuma: 2023.10.30. Forrás:

<https://www.geeksforgeeks.org/python-exception-handling/>

http8: Karimov, E.: Python Backend Dev Roadmap 3 -Advanced Python: List comprehensions, lambda functions, decorators, and generators. Letöltés dátuma: 2023.10.30. Forrás:

<https://www.linkedin.com/pulse/python-backend-dev-roadmap-3-advanced-list-lambda-elshad-karimov>

http9: A Python programozási nyelv - Standard könyvtárak. Letöltés dátuma: 2023.10.29.

Forrás: <http://nyelvek.inf.elte.hu/leirasok/Python/index.php?chapter=14>

http10: Libraries in Python. Letöltés dátuma: 2023.10.29. Forrás:

<https://www.geeksforgeeks.org/libraries-in-python/>

http11: Best Python Libraries for Modern Developers. Letöltés dátuma: 2023.10.30. Forrás:

<https://www.bairesdev.com/blog/best-python-libraries/>

http12: Hande, L.: Best Python Packages for Excel. Letöltés dátuma: 2023.10.28. Forrás:

<https://learnpython.com/blog/python-package-for-excel/>

http13: What is Web scraping? Letöltés dátuma: 2023.10.31. Forrás:

<https://realpython.com/beautiful-soup-web-scraper-python/#what-is-web-scraping>

http14: Twisted - An event-driven networking engine. Letöltés dátuma: 2023.10.31. Forrás:

<https://twisted.org/>

http15: About. Letöltés dátuma: 2023.10.31. Forrás: <https://opencv.org/about/>

http16: 40 Most Popular Python Scientific Libraries. Letöltés dátuma: 2023.10.31. Forrás: <https://www.stxnnext.com/blog/most-popular-python-scientific-libraries/>

http17: Introduction to Pygame in Python: A Game Development Library. Letöltés dátuma: 2023.10.31. Forrás: <https://intellipaat.com/blog/what-is-pygame/>

http18: Getting started. Letöltés dátuma: 2023.10.28. Forrás: <https://www.jetbrains.com/help/pycharm>

http19: PyCharm Features. Letöltés dátuma: 2023.10.28. Forrás: <https://www.jetbrains.com/pycharm/features/>

http20: Rendszergazda szolgáltatás. Letöltés dátuma: 2023.11.01. Forrás: <https://x-com.hu/rendszergazda-szolgalatas/>

http21: El tudod magyarázni, hogy mi a rendszergazda szerepe? Letöltés dátuma: 2023.11.01. Forrás: <https://www.ndistudio.hu/mi-rendszergazda-szerepe/>

http22: Az adatbázis-kezelés alapjai (SQL alapok). Letöltés dátuma: 2023.11.02. Forrás: <https://webiskola.hu/sql-ismeretek/az-adatbazis-kezeles-alapjai/>

http23: Introduction to SQL. Letöltés dátuma: 2023.11.01. Forrás: https://www.w3schools.com/sql/sql_intro.asp

http24: Carnes, B.: Basic SQL Commands - The List of Database Queries and Statements You Should Know. Letöltés dátuma: 2023.11.01. Forrás: <https://www.freecodecamp.org/news/basic-sql-commands/>

Ábrajegyzék

1. ábra: Python logó. Forrás: https://www.python.org/	5
2. ábra: U-Power logó. Forrás: olasz gyártói adatbázis alapján.	12
3. ábra: 3 népszerű U-Power védőlábbeli. Balról jobbra: Scandy, Londra, Active. Forrás: saját szerkesztés olasz gyártói adatbázis alapján.	13

Mellékletek

Python Excel-automatizáló program kódja teljes egészében

```
import openpyxl
from openpyxl.utils import get_column_letter
from openpyxl.styles import Alignment, Border, Side, Font

# A vevői kategóriák és a hozzájuk tartozó kedvezmények meghatározása.
customer_categories = {
    'A': 0.9,
    'B': 0.85,
    'C': 0.8,
    'D': 0.75,
    'E': 0.7,
    'F': 0.65,
    'G': 0.6,
    'H': 0.575,
    'I': 0.55,
    'J': 0.525,
    'K': 0.5,
}

# Az árfolyam meghatározása.
eur_to_huf = 400

# Szótár létrehozása, mely megfelelteti az eredeti fájlok neveit, a vevői
# árjegyzékek elnevezésének első részével.
product_names = {
    "upower_shoes.xlsx": "U-Power_védőlábbeli_vevői_árjegyzék",
    "upower_clothes.xlsx": "U-Power_védőruházat_vevői_árjegyzék",
    "lupos_shoes.xlsx": "Lupos_védőlábbeli_vevői_árjegyzék",
}

# Az eredeti fájlok feletti iteráció.
original_files = ["upower_shoes.xlsx", "upower_clothes.xlsx",
                  "lupos_shoes.xlsx"]

for original_file in original_files:
    product_name = product_names.get(original_file)

    # Az eredeti Excel fájl betöltése.
    workbook = openpyxl.load_workbook(original_file)

    # Külön EUR és HUF munkafüzetek létrehozása minden ügyfélkategóriához.
    for category, discount in customer_categories.items():
        eur_filename = f"{product_name}_{category}_EUR_2023.xlsx"
        huf_filename = f"{product_name}_{category}_HUF_2023.xlsx"

        # Új munkafüzetek létrehozása az eredeti munkalap másolatával.
        eur_workbook = openpyxl.Workbook()
        eur_worksheet = eur_workbook.active
        for row in workbook.active.iter_rows(values_only=True):
            eur_worksheet.append([cell.strip() if isinstance(cell, str)
                                else cell for cell in row])

        huf_workbook = openpyxl.Workbook()
        huf_worksheet = huf_workbook.active
        for row in workbook.active.iter_rows(values_only=True):
```

```

        huf_worksheet.append([cell.strip() if isinstance(cell, str)
else cell for cell in row])

# A szükségtelen oszlopok eltávolítása: G és H oszlopok törlése.
eur_worksheet.delete_cols(7, 2)
huf_worksheet.delete_cols(7, 2)

# Fejlécek átnevezése
headers = ["Katalógus", "Kollekció", "Kód", "Megnevezés",
"Méretek", "Nettó ár"]
for col, header in zip(eur_worksheet.iter_cols(max_col=6,
min_col=1), headers):
    col[0].value = header
    # Fejlécek formázása
    for cell in col:
        if cell.row == 1:
            cell.alignment = Alignment(horizontal="center")
            cell.font = Font(bold=True)
            cell.border = Border(left=Side(style="thin"),
right=Side(style="thin"), top=Side(style="thin"),
bottom=Side(style="thin"))

    for col, header in zip(huf_worksheet.iter_cols(max_col=6,
min_col=1), headers):
        col[0].value = header
        # Fejlécek formázása
        for cell in col:
            if cell.row == 1:
                cell.alignment = Alignment(horizontal="center")
                cell.font = Font(bold=True)
                cell.border = Border(left=Side(style="thin"),
right=Side(style="thin"), top=Side(style="thin"),
bottom=Side(style="thin"))

# A formázás beállítása.
for sheet in [eur_worksheet, huf_worksheet]:
    sheet.title = "Árlista_2023"
    sheet.freeze_panes = "A2"
    sheet.sheet_properties.tabColor = "0000FF"
    for row in sheet.iter_rows(min_row=2):
        for cell in row:
            cell.alignment = Alignment(horizontal="center")
            cell.border = Border(left=Side(style="thin"),
right=Side(style="thin"), top=Side(style="thin"),
bottom=Side(style="thin"))

# Az árak kiszámítása és a cellaformátum beállítása.
for row in eur_worksheet.iter_rows(min_row=2, max_col=6,
max_row=eur_worksheet.max_row):
    for cell in row:
        if cell.column == 6:
            cell.value = round(float(cell.value) * discount, 2)
            cell.number_format = '#,##0.00 "EUR"'

    for row in huf_worksheet.iter_rows(min_row=2, max_col=6,
max_row=huf_worksheet.max_row):
        for cell in row:
            if cell.column == 6:
                cell.value = round(float(cell.value) * discount *
eur_to_huf, 0)
                cell.number_format = '#,##0 "HUF"'

```

```

# Az oszlopszélesség igazítása a tartalomhoz.
for sheet in [eur_worksheet, huf_worksheet]:
    for column in sheet.columns:
        max_length = 0
        column_letter = get_column_letter(column[0].column)
        for cell in column:
            try:
                if len(str(cell.value)) > max_length:
                    max_length = len(str(cell.value))
            except:
                pass
        adjusted_width = max_length + 2
        sheet.column_dimensions[column_letter].width =
adjusted_width

# Az F oszlop oszlopszélességének beállítása.
eur_worksheet.column_dimensions['F'].width = 12
huf_worksheet.column_dimensions['F'].width = 13

# Minden második sorhoz kék háttérszín beállítása.
for sheet in [eur_worksheet, huf_worksheet]:
    for i, row in enumerate(sheet.iter_rows(min_row=2, max_col=6,
max_row=sheet.max_row), start=1):
        if i % 2 == 1:
            for cell in row:
                cell.fill =
openpyxl.styles.PatternFill(start_color="DDEBF7", end_color="DDEBF7",
fill_type="solid")

# A fájlok elmentése.
eur_workbook.save(eur_filename)
huf_workbook.save(huf_filename)

```

NYILATKOZAT

a záródolgozat nyilvános hozzáféréséről és eredetiségéről

A hallgató neve: Sipos Szilvia
A Hallgató Neptun kódja: FNGGGO
A dolgozat címe: Szakmai gyakorlat bemutatása, központban a Python Excel-automatizáló programmal
A megjelenés éve: 2023
A konzulens intézetének neve: Műszaki intézet
A konzulens tanszékének a neve: Alkalmazott Informatikai Tanszék, Gyöngyös

Kijelentem, hogy az általam benyújtott záródolgozat egyéni, eredeti jellegű, saját szellemi alkotásom. Azon részeket, melyeket más szerzők munkájából vettem át, egyértelműen megjelöltem, és az irodalomjegyzékben szerepeltettem.

Ha a fenti nyilatkozattal valótlan állítottam, tudomásul veszem, hogy a záróvizsga-bizottság a záróvizsgából kizár és a záróvizsgát csak új dolgozat készítése után tehetek.

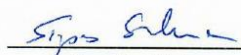
A leadott dolgozat, mely PDF dokumentum, szerkesztését nem, megtekintését és nyomtatását engedélyezem.

Tudomásul veszem, hogy az általam készített dolgozatra, mint szellemi alkotás felhasználására, hasznosítására a Magyar Agrár- és Élettudományi Egyetem mindenkori szellemitulajdon-kezelési szabályzatában megfogalmazottak érvényesek.

Tudomásul veszem, hogy dolgozatom elektronikus változata feltöltésre kerül a Magyar Agrár- és Élettudományi Egyetem könyvtári repozitori rendszerébe. Tudomásul veszem, hogy a megvédett és

- nem titkosított dolgozat a védést követően
- titkosításra engedélyezett dolgozat a benyújtásától számított 5 év eltelte után nyilvánosan elérhető és kereshető lesz az Egyetem könyvtári repozitori rendszerében.

Kelt: 2023 év 11. hó 10. nap


Hallgató aláírása

Konzulensi nyilatkozat

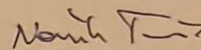
NYILATKOZAT

Sipos Szilvia (név) (hallgató Neptun azonosítója: FNGGGG) konzulenseként nyilatkozom arról, hogy a záródolgozatot/szakdolgozatot/diplomadolgozatot/portfóliót¹ áttekintettem, a hallgatót az irodalmi források korrekt kezelésének követelményeiről, jogi és etikai szabályairól tájékoztattam.

A záródolgozatot/szakdolgozatot/diplomadolgozatot/portfóliót a záróvizsgán történő védésre javaslom / nem javaslom².

A dolgozat állam- vagy szolgálati titkot tartalmaz: igen nem^{*3}

Kelt: 2023. év 11. hó 11. nap



belső konzulens

¹ A megfelelő dolgozattípus meghagyása mellett a többi típus törlendő.

² A megfelelő aláhúzendő.

³ A megfelelő aláhúzendő.