

ŠCV, ERŠ

Trg mladosti 3, 3320 Velenje

MLADI RAZISKOVALCI ZA RAZVOJ ŠALEŠKE DOLINE

RAZISKOVALNA NALOGA

**POENOSTAVLJANJE RAČUNALNIŠKIH OPRAVIL V POSLOVNEM
OKOLJU Z UPORABO SKRIPTNEGA JEZIKA AUTOHOTKEY**

Tematsko področje:

Računalništvo ali telekomunikacije

Avtor:

Anže Maj Blagus

Mentor:

Gregor Hrastnik, univ. dipl. inž. rač. in inf.

Velenje, 2020/21

A. Blagus, Poenostavljanje računalniških opravil v poslovnem okolju z uporabo skriptnega jezika AutoHotKey
Raz. nal., ERS, 2020/2021

Raziskovalna naloga je bila opravljena na Elektro in računalniški šoli Velenje, 2021.

Mentor: Gregor Hrastnik, univ. dipl. inž. rač. in inf..

Datum predstavitve:

KLJUČNA DOKUMENTACIJSKA INFORMACIJA

ŠD Elektro in računalniška šola, šolsko leto 2020/2021

KG avtomatizacija/ahk/autohotkey/rpa

AV BLAGUS, Anže

SA HRASTNIK, Gregor

KZ

ZA

LI 2020/2021

IN POENOSTAVLJANJE RAČUNALNIŠKIH OPRAVIL V POSLOVNEM OKOLJU Z
UPORABO SKRIPTNEGA JEZIKA AUTOHOTKEY

TD Raziskovalna naloga

OP

IJ SL

JI sl

AI Velik del delovnega časa ljudje na delovnih mestih v pisarnah opravljajo ponavljajoča opravila. Z uporabo jezika Autohotkey in drugih podobnih orodij je možno nekatera od teh opravil avtomatizirati. Po večkratnih pogovorih z ljudmi na pisarniških delovnih mestih sem opazil nepoznavanje področja avtomatizacije računalniških opravil, zato sem se odločil to področje bolj raziskati in izdelati primere programov za avtomatizacijo. Sprva me je zanimalo, če je avtomatizacija v podjetjih res nepoznana, nato pa sem se odločil izdelati še programa za avtomatizacijo različnih opravil v gradbenem podjetju. Ugotavljal sem tudi, koliko časa ti programi prihranijo. Po anketiranju v podjetjih sem ugotovil, da približno polovica anketiranih ni bila seznanjena z možnostmi avtomatizacije, kar ustvari zanimivo možnost za izboljšavo v podjetjih. Z izdelavo dveh programov za avtomatizacijo sem v konkretnem podjetju prihranil več dni dela na posamezni projekt in to nadomestil z le nekaj minutami dela.

A. Blagus, Poenostavljanje računalniških opravil v poslovnem okolju z uporabo skriptnega jezika AutoHotKey
Raz. nal., ERS, 2020/2021

KEY WORDS DOCUMENTATION

ND Elektro in računalniška šola, šolsko leto 2020/2021

CX automation/ahk/autohotkey/rpa

AU BLAGUS, Anže

AA HRASTNIK, Gregor

PP

PB

PY 2020/2021

TI BUSINESS AUTOMATION USING AUTOHOTKEY

DT RESEARCH WORK

NO

LA SL

AL sl/en

AB Office workers do a lot of repetitive work during their average work day. Using the Autohotkey programming language and other similar tools, some of these tasks can be automated. After several conversations with people in office workplaces, I noticed how unaware they were of the possibilities of automation, so I decided to explore this field and make some examples of automation programs. At first, I was interested in how much most companies knew about automation. After that, I created two programs for a construction company that automated some of their office work. I also found out how much time these programs save. After surveying in companies, I found that about half of the respondents were not familiar with the possibilities of automation, which creates an interesting opportunity for improvement in companies. By creating the automation programs, I saved several days of work per project in a company and replaced it with only a few minutes of work.

KAZALO VSEBINE

1	UVOD	1
1.1	HIPOTEZE RAZISKOVALNE NALOGE	1
2	PREGLED OBJAV	2
2.1	AVTOMATIZACIJA RAČUNALNIŠKIH OPRAVIL	2
2.2	AUTOHOTKEY	2
2.2.1	PRIMERI KODE	3
2.2.2	RAZLIČNE FUNKCIONALNOSTI AHK	3
2.3	ALTERNATIVE	7
2.3.1	AutoIt	7
2.3.2	Makri v Microsoft Office programih	7
2.3.3	Selenium	8
2.3.4	Različni snemalniki makrov	8
3	MATERIALI IN METODE DELA	8
3.1	ANKETIRANJE PODJETIJ	8
3.2	AVTOMATIZACIJA DODAJANJA SKLICEV V PREDRAČUNIH	9
3.3	PROGRAM ZA AVTOMATSKO ODKLEPANJE .XLSX DATOTEK	10
4	REZULTATI IN RAZPRAVA	12
4.1	ANKETA	12
4.2	UPORABA V PODJETJU	14
4.3	HIPOTEZE	14
5	ZAKLJUČEK	15
6	POVZETEK	15
7	ZAHVALA	16
8	VIRI IN LITERATURA	17
9	PRILOGE	18

KAZALO SLIK

Slika 1: Novo okno beležnice	6
Slika 2: Izbira izgleda pisave	6
Slika 3: Primer vmesnika	7
Slika 4: Vsebina .xlsx datoteke (videna s preimenovanjem v .zip datoteko).....	12

KAZALO KODE

Koda 1: Sprememba funkcije tipke F12	3
Koda 2: Iskanje vsebine odložišča v iskalniku Google.....	3
Koda 3: Primer hotstringa	4
Koda 4: Primer hotkeya	4
Koda 5: Primer uporabe COM.....	5
Koda 6: Primer simulacije uporabniškega vnosa za spremembo velikosti pisave na urejevalniku »Notepad«	5
Koda 7: Primer enostavnega grafičnega vmesnika	7
Koda 8: Odklepanje Excel datotek.....	11

KAZALO GRAFOV

Graf 1: Koliko zaposlenih imate v podjetju?	12
Graf 2: Ali ste se vi osebno že srečali z avtomatizacijo opravil na računalniku?	13
Graf 3: Ali vam je znano, če v vašem podjetju uporabljate avtomatizacijo opravil na računalniku (npr. masovno pošiljanje e-sporočil, avtomatsko izpolnjevanje obrazcev)?	13
Graf 4: Za katera od naštetih orodij za avtomatizacijo ste že slišali?	14

KAZALO PRILOG

Priloga 1: Navodila za uporabo programa za ustvarjanje sklicev	18
Priloga 2: Navodila za uporabo programa za odklepanje .xlsx datotek	23
Priloga 3: Anketni vprašalnik.....	25
Priloga 4: Koda programa za odklepanje .xlsx datotek	29
Priloga 5: Koda programa za ustvarjanje sklicev.....	31

1 UVOD

Idejo za izdelavo raziskovalne naloge sem dobil ob pogovoru z mamo in njenimi sodelavci, v katerem sem ugotovil, koliko je ponavljajočih opravil na računalniku vsak dan. Ker sem pred tem že imel izkušnje z avtomatizacijo z jezikom AutoHotkey sem se odločil izdelati avtomatizacijo za olajšanje opravil v podjetju, kjer delajo. Poleg tega me je zanimalo tudi, če je to problem, ki se pojavlja v vseh podjetjih ali samo v omenjenem.

Zadal sem si cilj, da ugotovim, kako poznana in kako razširjena je avtomatizacija in katera orodja so najpogosteje uporabljena. Potem sem se odločil izdelati programe, ki bodo v podjetju prihranili čas.

1.1 HIPOTEZE RAZISKOVALNE NALOGE

Zastavil sem si naslednji hipotezi:

1. Večina podjetji ni seznanjena z avtomatizacijo preprostih opravil na računalniku z uporabo programske opreme.
2. Z AutoHotKey je mogoče skrajšati čas izpolnjevanja predračuna v gradbenem podjetju za polovico.

2 PREGLED OBJAV

V tem delu je najprej opisana avtomatizacija računalniških opravil na splošno, potem pa skriptni jezik AutoHotKey, njegova pravila zapisa in primeri uporabe.

2.1 AVTOMATIZACIJA RAČUNALNIŠKIH OPRAVIL

RPA (angl. Robotic Process Automation) je izraz, ki opisuje izdelavo ali konfiguriranje programske opreme (ali »robot«), tako da simulira dejanja človeka. S tem dosežemo avtomatsko izvajanje določenega procesa (po navadi s poslovnim namenom). Pogosto se uporablja za avtomatizacijo uporabniškega vmesnika, zbiranje podatkov iz njega, izvajanje dejanj glede na sprožilnike (e-pošta, klik uporabnika, dogodek na zaslonu). Prednost RPA pred človeškim delom je njegova zanesljivost, relativno nizka cena, glede na delo, ki ga lahko opravlja ter dejstvo, da računalnik deluje brez premorov, ki so pri človeškem delu nujni. [1]

Poznamo več tipov RPA:

- **Delna avtomatizacija**

Pri delni avtomatizaciji govorimo o programski opremi, ki opravi le del človekovega dela. Pri njej je še vedno potreben uporabnik, ki program nadzoruje ali mu celo pomaga (z vnašanjem podatkov, preverjanjem/potrjevanjem vnesenih podatkov, izvajanje stvari, ki so za računalnik nemogoče/prezahtevne za avtomatizacijo).

- **Popolna avtomatizacija (s strukturiranimi podatki)**

Pri tem tipu RPA govorimo o programski opremi, ki opravi vse delo in za njeno delovanje ni potreben človek. Podatki, na osnovi katerih deluje, so strukturirani in računalniku enostavno razumljivi (po navadi jih ne ustvari človek, temveč drug računalniški sistem).

- **Popolna avtomatizacija (z nestrukturiranimi podatki)**

Ta tip RPA je najzahtevnejši in najmanj razvit. Uporablja različne tipe umetne inteligence (nevronske mreže, strojno učenje) za razumevanje podanih podatkov in izvajanje dejanj, glede na njih.

2.2 AUTOHOTKEY

Autohotkey ali AHK je odprtokodni skriptni jezik za Windows operacijske sisteme.

Skriptni jezik ni točno definiran pojem, je pa v splošnem programski jezik, ki se interpretira – sproti izvaja vrstico po vrstico. Na splošno so skriptni jeziki namenjeni dodajanju funkcionalnosti nekemu

A. Blagus, Poenostavljanje računalniških opravil v poslovnem okolju z uporabo skriptnega jezika AutoHotKey
Raz. nal., ERS, 2020/2021

drugemu programu (VBA v Excelu, Lua v nekaterih računalniških igrah in podobno).

AutoHotkey se uporablja za vse od spreminjanja funkcij tipk do izdelave grafičnih vmesnikov, spletnih pajkov (angl. web spiders), programov za avtomatizacijo določenih programov in nešteto drugih stvari. [2]

2.2.1 PRIMERI KODE

Preprost primer kode v nadaljevanju spremeni funkcijo tipke F12 na način, da jo programi zaznajo, kot bi bil pritisnjen levi gumb na računalniški miški.

```
f12::LButton
```

Koda 1: Sprememba funkcije tipke F12

Še en primer kode, ki ob pritisku kombinacije tipk ctrl+alt+g odpre spletni brskalnik Google in v njem poišče tekst, ki je trenutno v odložišču.

```
^!g::
    if(!clipboard)
    {
        MsgBox, Odložišče je prazno!
        return
    }
    Run, https://www.google.com/search?q=%clipboard%
return
```

Koda 2: Iskanje vsebine odložišča v iskalniku Google

V prvi vrstici vidimo znaka »^« ter »!«, kjer »^« pomeni tipko Control ter »!« pomeni tipko Alt. Za tem preverimo, če je odložišče (angl. clipboard) prazno in če je, prikažemo sporočilo ter prenehamo izvajanje trenutnega hotkeya – rutina v kodi, ki se izvede ob pritisku na tipko. Potem odpremo Google z iskalnim nizom enakem spremenljivki »clipboard«, ki vsebuje vsebino odložišča.

2.2.2 RAZLIČNE FUNKCIONALNOSTI AHK

V nadaljevanju bo na kratko opisanih nekaj funkcionalnosti AHK, ki so najbolj zanimive za namene te naloge.

1. Spreminjanje funkcij tipk (hotkeys, hotstrings)

Funkcionalnost jezika AHK po imenu »hotkeys« omogoča enostavno pisanje rutin, ki se izvedejo ob pritisku na tipko ali kombinacijo tipk. AHK omogoča tudi »hotstring«, ki so podobni hotkeyem, le da se izvedejo ob zapisu določenega teksta.

Naslednji primer kode prikazuje hotstring, ki ob zapisu zaporedja črk »np« to zamenja z »ni problema«. Zvezdica (»*«) pomeni, da se tekst zamenja takoj ob končanju zapisa (ni potrebno pritisniti enter tipke, da se zamenja). Hotstringi so zanimivi v uporabi pri pisanju ponavljajočih tekstov. Lahko vsebujejo spremenljivke ali celo izvedejo kodo ob njihovem vnosu (kar odpre še veliko možnosti).

```
:*:np::ni problema
```

Koda 3: Primer hotstringa

Naslednja koda prikazuje primer hotkeya, ki spremeni trenutno aktivno okno tako, da je vedno »na vrhu« (nobeno okno se ne more prikazati čez njega). Ob ponovnem kliku to izklopi.

```
^+e::  
Winset, Alwaysontop, Toggle, A  
return
```

Koda 4: Primer hotkeya

2. Uporaba COM (Component Object Model)

V operacijskem sistemu Windows obstaja programski vmesnik z imenom COM. Ta omogoča komuniciranje med različnimi procesi, ki se izvajajo na računalniku neodvisno od programskega jezika, v katerem so napisani. Z uporabo COM vmesnika lahko na primer upravljamo programe kot so Microsoft Excel, Word, Powerpoint in podobne. Omogoča nam, da v njih izvedemo vse akcije, ki so mogoče z uporabo vizualnega uporabniškega vmesnika.

Prav tako nam omogoča komunikacijo s spletnim brskalnikom Internet Explorer, kar odpre možnost avtomatskega upravljanja spletnih strani ter vnašanja in pridobivanja informacij iz njih.

A. Blagus, Poenostavljanje računalniških opravil v poslovnem okolju z uporabo skriptnega jezika AutoHotKey
Raz. nal., ERS, 2020/2021

```
wb := ComObjCreate("InternetExplorer.Application") ; ustvarimo novo COM
povezavo do aplikacije Internet Explorer
wb.Visible := True ; prikažemo brskalnikovo okno (ta vrstica je opcijska,
ne vpliva na delovanje programa)

wb.Navigate("http://mladiraziskovalci.scv.si/") ; brskalnik preusmerimo na
spletni naslov

While wb.readyState!=4 || wb.document.readyState!="complete" || wb.busy ;
počakamo, da je stran naložena
    Sleep 50

prviParagraf := wb.document.getElementsByTagName("p")[0] ; preko COM
vmesnika dobimo prvi paragraf na strani
msgbox, % prviParagraf.innerText ; v pojavnem oknu prikažemo tekstovno
vsebino prvega paragrafa
```

Koda 5: Primer uporabe COM

3. Simulacija uporabniškega vnosa (Pritiskanje tipk ter premikanje miške)

Pri avtomatizaciji vseh programov ni mogoča uporaba COM vmesnika, zato moramo včasih preprosto simulirati uporabniški vnos, da dosežemo željen vnos podatkov v določenem programu.

```
+n:: ; ob pritisku kombinacije tipk shift+n
run, notepad.exe ; zaženemo urejevalnik Notepad

while(!WinActive("ahk_exe notepad.exe")) ; počakamo, da se odpre
    sleep, 50

send, {alt}o ; kombinacija tipk alt+o odpre meni "Format"

send, {down}{Enter} ; izberemo 2. opcijo na meniju

while(!WinActive("ahk_class #32770")) ; počakamo, da se odpre pojavno okno
za spremembo izgleda pisave
    sleep, 50

send, {tab 2} ; premaknemo se na izbiro velikosti pisave

send, {up 17} ; ne glede na trenutno lokacijo na seznamu pojdimo na vrh

send, {down 8} ; izberemo 8. opcijo velikosti pisave

send, {enter} ; potrdimo izbiro
return ; konec izvajanja "hotkey"-a
```

Koda 6: Primer simulacije uporabniškega vnosa za spremembo velikosti pisave na urejevalniku

A. Blagus, Poenostavljanje računalniških opravil v poslovnem okolju z uporabo skriptnega jezika AutoHotKey
Raz. nal., ERS, 2020/2021

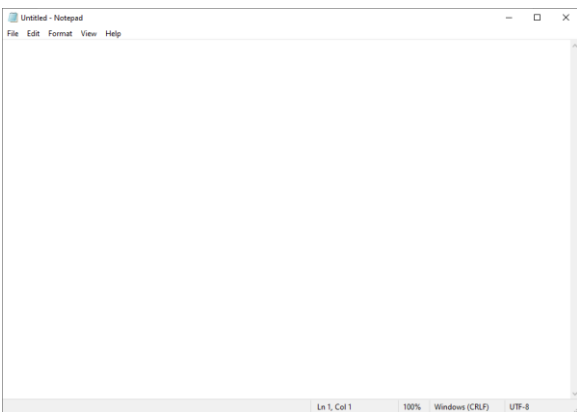
»Notepad«

Cilj zgoraj prikazane kode je, da se ob pritisku tipk Shift in n odpre novo okno beležnice ter spremeni velikost pisave na 20.

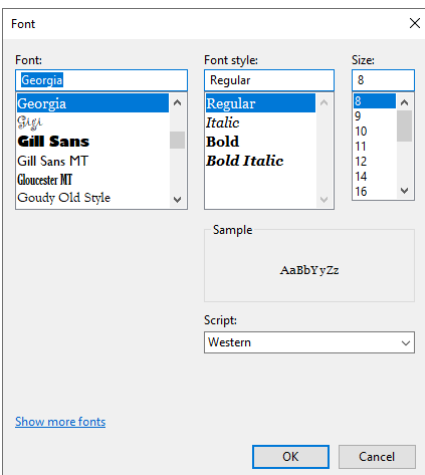
Kot pričakovano se ob pritisku kombinacije tipk odpre okno (Slika 1

Slika 1: Novo okno beležnice

), nato se odpre meni za izbiro izgleda pisave (Slika 2) ter s pritiskanjem različnih kombinacij tipk spremeni pisavo na velikost 20.



Slika 1: Novo okno beležnice



Slika 2: Izbira izgleda pisave

4. Izdelava grafičnih vmesnikov

AHK omogoča tudi izdelavo preprostih grafičnih vmesnikov. Te lahko nato povežemo z drugo

A. Blagus, Poenostavljanje računalniških opravil v poslovnem okolju z uporabo skriptnega jezika AutoHotKey
Raz. nal., ERS, 2020/2021

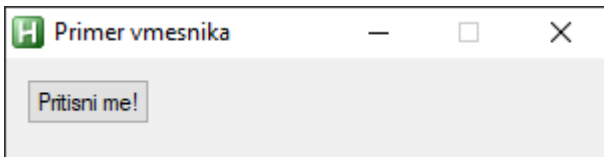
funkcionalnostjo in dobimo možnost enostavnega ustvarjanja zelo uporabnih programov. Prednost AHK pred drugimi načini ustvarjanja grafičnih vmesnikov je njegova preprostost. V Koda 7 vidimo primer, kjer ustvarimo novo okno ter v njem gumb. Ko je gumb pritisnjen se izvede del rutine v kodi imenovani »GumbPritisnjen«. Ta primer nam pokaže, kako enostavno je ustvariti grafične vmesnike ter s kako malo kode se to lahko naredi.

```
Gui, Add, Button, x10 y10 gGumbPritisnjen, Pritisni me!
Gui, Show, w300 h50
return

GumbPritisnjen:
MsgBox, Gumb je bil pritisnjen!
return
```

Koda 7: Primer enostavnega grafičnega vmesnika

Ob izvajanju se pokaže okno, prikazano na Slika 3.



Slika 3: Primer vmesnika

2.3 ALTERNATIVE

V tem poglavju je opisanih nekaj alternativ v Autohotkey. Vsaka izmed njih je uporabljena na enem od področji, v katerih se uporablja AHK.

2.3.1 AutoIt

AutoIt je brezplačna programska oprema, namenjena avtomatizaciji grafičnega vmesnika v operacijskem sistemu Windows. V primerjavi z AHK ni odprtokodni program, omogoča pa lažjo povezavo s COM vmesnikom. [3]

2.3.2 Makri v Microsoft Office programih

Makri v programih Microsoft Office so namenjeni avtomatizaciji ponavljajočih opravil. Lahko jih »posnamemo«, kjer se z našimi kliki in pisanjem v času snemanja napiše preprosta koda, ki potem enak postopek izvede le z enim klikom. Kodo lahko tudi urejamo in ustvarimo naprednejše programe. Slabost makrov pri splošni avtomatizaciji je, da delujejo le v določenih programih. Upravljanje

uporabniškega vmesnika zunaj teh oz. opravljanje drugih opravil je zelo zapleteno ali celo nemogoče.

2.3.3 Selenium

Selenium je brezplačen odprtokodni skupek programov (angl. program suite) za testiranje in avtomatiziranje spletnih aplikacij. Mogoče ga je uporabiti z različnimi programskimi jeziki (Java, C#, Python) in omogoča lažje upravljanje spletnih vmesnikov, kot bi ga ti jeziki drugače.

Slabost tega je, da je potrebno poglobljeno znanje zgoraj omenjenih jezikov, prednost pa da omogoča zelo enostavno upravljanje spletnih aplikacij. Ne omogoča avtomatizacije uporabniškega vmesnika.

[4]

2.3.4 Različni snemalniki makrov

Poleg makrov v programih Microsoft Office obstajajo tudi programi, namenjeni le snemanju in izvajanju makrov. Pri teh programih »posnameš« svoje premike miške in pritiske tipk na tipkovnici, nato pa jih program lahko ponovno predvaja. To omogoča zelo preprosto in hitro avtomatizacijo, ne omogoča pa avtomatizacije kompleksnejših opravil, ki zahtevajo določeno logiko. [5] [6]

3 MATERIALI IN METODE DELA

V tej raziskovalni nalogi sem se problema avtomatizacije lotil na dva načina. Najprej me je zanimalo, kako poznana je avtomatizacija opravil na računalniku, nato pa sem izdelal programe za avtomatizacijo in ugotavljal, če so uporabni.

Sprva sem izvedel anketo, kjer so anketiranci (podjetja oz. predstavniki podjetij) povedali, če poznajo avtomatizacijo in kje bi jo lahko uvedli.

V drugem delu sem naredil 2 programa, ki sta namenjena avtomatizaciji preprostih opravil. Sprva sem se dobil z direktorjem gradbenega podjetja in izvedel, kaj želi pohitriti/avtomatizirati, nato pa sem prek več sestankov programe dodelal.

3.1 ANKETIRANJE PODJETIJ

V sklopu te raziskovalne naloge sem se odločil ugotavljati tudi kako poznana je avtomatizacija opravil na računalniku v poslovnem okolju. To sem naredil na platformi Google Forms. Ankete sem podjetjem poslal po e-pošti, v kateri sem napisal, naj jo reši ena oseba v podjetju, in sicer tista, ki ima o področju avtomatizacije računalniških opravil največ znanja.

Najprej me je zanimalo število zaposlenih v podjetju. Prek tega vprašanja bom ugotovil ali je

poznavanje avtomatizacije v podjetju odvisno od njegove velikosti.

Zatem anketa sprašuje po nalogi tistega, ki jo rešuje v podjetju. To vprašanje mi pove, kdo v podjetju je v bistvu odgovoren za avtomatizacijo.

Tretje vprašanje sprašuje, če se je anketiranec že osebno srečal z avtomatizacijo, četrto pa, če mu je znano, da je prav ta uvedena v njihovem podjetju. To odkrije primere, kjer je anketiranec seznanjen z avtomatizacijo, pa ta še ni uvedena v podjetju.

V naslednjih vprašanjih sprašujem po tipu avtomatizacije, s katero so se srečali (če so se sploh) in kje v njihovem podjetju bi avtomatizacijo lahko uvedli.

Na koncu še vprašam, katere od naštetih orodij za avtomatizacijo anketiranec pozna (to še dodatno pokaže njegovo seznanjenost z avtomatizacijo).

Celoten anketni vprašalnik je v **Error! Reference source not found..**

3.2 AVTOMATIZACIJA DODAJANJA SKLICEV V PREDRAČUNIH

Prvi program za avtomatizacijo je namenjen olajšanju izdelave predračunov v gradbenem podjetju. Delo olajša z ustvarjanjem sklicev v Excel datoteki. Uporabnik bi brez tega programa šel čez vse postavke v datoteki in jim dopisal ceno, ta program pa naj bi na vseh enakih postavkah napisal ceno, ki jo je uporabnik določil za to postavko.

Ta program le delno avtomatizira postopek, saj je potrebna konfiguracija pred zagonom.

Sprva sem se tega izziva lotil tako, da bi v eni datoteki (seznam postavk) bile zapisane vse postavke in njihove cene, nato bi program pregledal vse postavke v izbranem predračunu in jih primerjal s temi iz seznama. Nato bi jim glede na to določil cene. Kot ideja zveni dobro, toda po testiranju in razmisleku sem prišel do ugotovitve, da je to v resnici neizvedljivo. Problem je, da postavke v gradbeništvu niso standardizirane, zato jih vsaka projektantska hiša poimenuje drugače. S tem vedenjem bi bilo potrebno ustvariti zelo velik seznam postavk, iz katerih bi potem program jemal cene.

A. Blagus, Poenostavljanje računalniških opravil v poslovnem okolju z uporabo skriptnega jezika AutoHotKey
Raz. nal., ERS, 2020/2021

Naslednja ideja za skrajšanje te naloge je bila naslednja: ko uporabnik odpre predračun pregleda vse postavke in ob prvem pojavu posamezne postavke določi ceno. Nato program gre čez vse postavke in vsem enakim doda sklic na prvo določeno ceno. S tem bi se delo zelo skrajšalo, saj se v takšnih datotekah postavke veliko ponavljajo.

Program je bil v podjetju testiran in po nekaj popravkih (nekompatibilnost z nekaterimi tipi Excel datotek in podobni mali problemi) ga zdaj uporabljajo pri vseh večjih projektih.

Celotna koda programa je v Priloga 5, navodila za uporabo, ki so bila predana podjetju pa v Priloga 1.

3.3 PROGRAM ZA AVTOMATSKO ODKLEPANJE .XLSX DATOTEK

Drugi program, ki sem ga izdelal, je namenjen odklepanju datotek za program Microsoft Excel. Uporablja se, da se pred oddajo predračuna pregledajo vse formule v datoteki (oz. več datotekah). To ni možno, če je posamezen list v datoteki zaklenjen z geslom, zato ga je potrebno odkleniti.

Ker zaklepanje listov v Excelu ni preveč varno, lahko liste odklenemo na preprost način.

Ker je .xlsx datoteka v resnici le .zip datoteka, jo lahko razpakiramo in spreminjamo njeno vsebino.

Postopek odklepanja je naslednji:

1. Spremenimo končnico datoteke na .zip.
2. Razpakiramo njeno vsebino.
3. V razpakiranih datotekah poiščemo mapo »worksheets.«
4. V njej vsaki datoteki v tej mapi, ki jo želimo odkleniti spremenimo določeno vsebino (odstranimo XML značka (angl. XML node) z imenom »SheetProtection«).
5. List je odklenjen.

Pri izdelavi programa za opravljanje tega opravila sem prav tako uporabil Autohotkey. Za razpakiranje datotek uporabljam orodje 7zip prek ukazne vrstice, potem pa v datoteki vsakega lista poiščem zaporedje znakov »<sheetProtection« in za njim prvi pojav znakov »/>« ter odstranim vse znake med njima. S tem odklenem list. Na koncu še vse zapakiram nazaj in preimenujem v novo datoteko.

Celotna koda programa je v Priloga 4, navodila za uporabo, ki so bila predana podjetju pa v Priloga 2.

A. Blagus, Poenostavljanje računalniških opravil v poslovnem okolju z uporabo skriptnega jezika AutoHotKey
Raz. nal., ERS, 2020/2021

```

zip = %dir%\%brez_konc%.zip ; path do zip datoteke

outZip = out_%brez_konc%.zip ; path do izhodne zip datoteke
outFile = ODKLENJENO_%brez_konc%.xlsx ; path do izhodne datoteke
outDir=%dir%\out_%brez_konc% ; path do mape za razpakiranje

FileMove, %path%, %zip%, 1 ; preimenujemo datoteko v .zip

runwait, "%7zPath%" x "%zip%" -o"%outDir%" ; razpakiramo

sheetsDir:=outdir "\xl\worksheets" ; pot do mape z vsemi listi

while(!fileExist(sheetsDir)) ; za vsak slucaj pocakamo, da obstaja mapa
    sleep, 10

loop, Files, %sheetsDir%\*.xml ; za vsak list
{
    if(substr(a_loopfilename,1,5)!="sheet")
        continue

    FileRead, XMLdata, %A_LoopFileFullPath% ; preberemo vsebino

    ; naslednje vrstice:
    ; - locirajo XML znacko "sheetProtection"
    ; - locirajo njen konec
    ; - jo odstranijo
    loc := instr(XMLdata,"<sheetProtection")
    if(!loc)
        continue
    rest := substr(XMLdata,loc)
    loc2 := instr(rest,"/>")

    before:=substr(XMLdata,1,loc-1)
    after:=substr(rest,loc2+2)

    FileDelete, %A_LoopFileFullPath% ; zbrisemo staro datoteko
    FileAppend, %before%%after%, %A_LoopFileFullPath% ; napisemo odklenjeno
    datoteko
}

; nazaj zapakiramo in izbrisemo preostale datoteke

runwait, "%7zPath%" a "%outZip%" ".\out_%brez_konc%\*"
FileRemoveDir, %outDir%, 1
FileMove, %zip%, %path%
FileMove, %outZip%, %outFile%

```

Koda 8: Odklepanje Excel datotek

A. Blagus, Poenostavljanje računalniških opravil v poslovnem okolju z uporabo skriptnega jezika AutoHotKey
Raz. nal., ERS, 2020/2021

Name	Type	Compressed size	Password ...	Size	Ratio	Date modified
_rels	File folder					07-Jan-21 12:58 PM
docProps	File folder					07-Jan-21 12:58 PM
xl	File folder					07-Jan-21 12:58 PM
[Content_Types].xml	XML File	1 KB	No	3 KB	85%	30-Dec-99 1:00 AM

Slika 4: Vsebina .xlsx datoteke (videna s preimenovanjem v .zip datoteko)

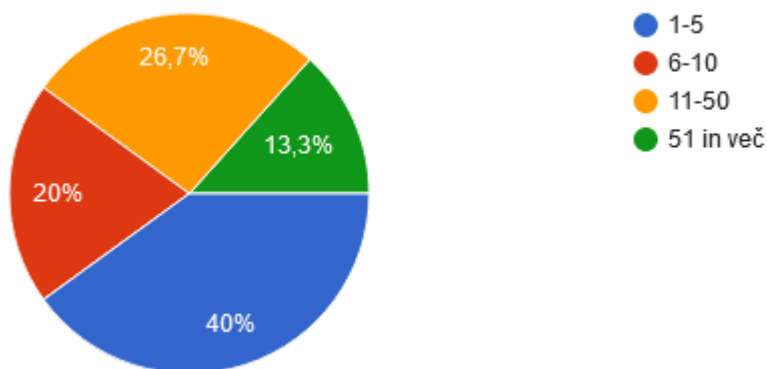
4 REZULTATI IN RAZPRAVA

4.1 ANKETA

Na anketni vprašalnik je odgovorilo 15 podjetij.

1. Koliko zaposlenih imate v podjetju?

Kot vidimo iz Graf 1 je na anketo odgovorilo največ manjših podjetji (1-5 zaposlenih), toda bilo je tudi nekaj večjih. To je dobro, saj bomo videli uporabo avtomatizacije v podjetjih različnih velikosti.



Graf 1: Koliko zaposlenih imate v podjetju?

2. Kaj je vaša naloga v podjetju (marketing, računovodstvo, izdajanje ponudb, IT administrator ...)?

Odgovore na to vprašanje lahko razdelimo na tri večje skupine: vodstvo, izdajanje ponudb in računovodstvo ter ostalo.

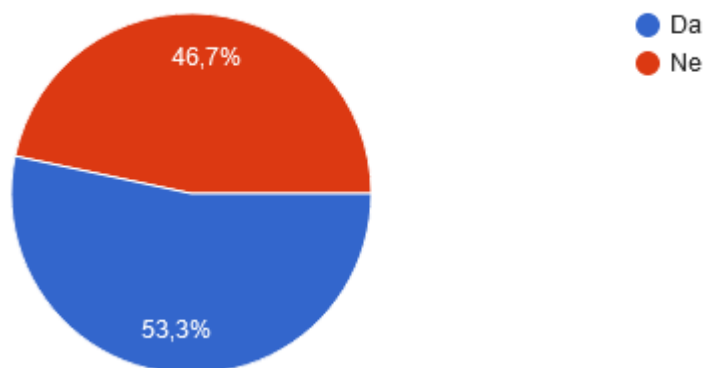
Največ odgovorov je od oseb, ki se ukvarjajo z izdajanjem ponudb in računovodstvom (7). To je razumljivo, saj ti dve deli vključujeta veliko ponavljajočih opravil, zato so osebe, ki jih opravljajo najbolj seznanjene z avtomatizacijo.

Nekaj odgovorov je tudi iz vodstva (5). To je pričakovano, saj v manjših podjetjih ti najbolj poznajo potek dela v podjetju in kje bi se ga dalo poenostaviti ali avtomatizirati.

Bila sta še dva odgovora (administrator in »vse delo«)

3. Ali ste se vi osebno že srečali z avtomatizacijo opravil na računalniku?

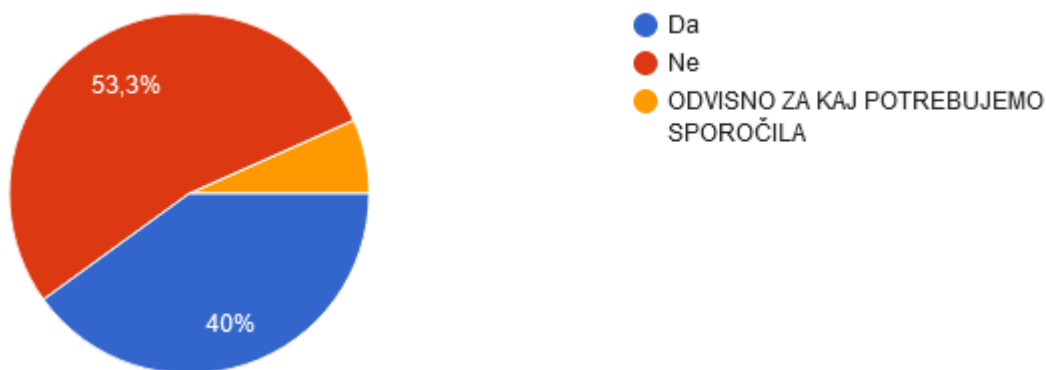
Tukaj vidimo, da se je približno polovica anketirancev že srečala z avtomatizacijo. Večina teh je iz večjih podjetji, toda pri tako malem številu anketirancev to ni merodajno.



Graf 2: Ali ste se vi osebno že srečali z avtomatizacijo opravil na računalniku?

4. Ali vam je znano, če v vašem podjetju uporabljate avtomatizacijo opravil na računalniku (npr. masovno pošiljanje e-sporočil, avtomatsko izpolnjevanje obrazcev)?

Tu vidimo, da čeprav so se nekateri posamezniki že srečali z avtomatizacijo, te ne uporabljajo v podjetju. To pokaže, da bi si v nekaterih podjetjih lahko pomagali s programi za avtomatizacijo.



Graf 3: Ali vam je znano, če v vašem podjetju uporabljate avtomatizacijo opravil na računalniku (npr. masovno pošiljanje e-sporočil, avtomatsko izpolnjevanje obrazcev)?

5. Na katerem področju v vašem podjetju bi lahko uvedli avtomatizacijo opravil na računalniku (poleg obstoječe, če je že na mestu)?

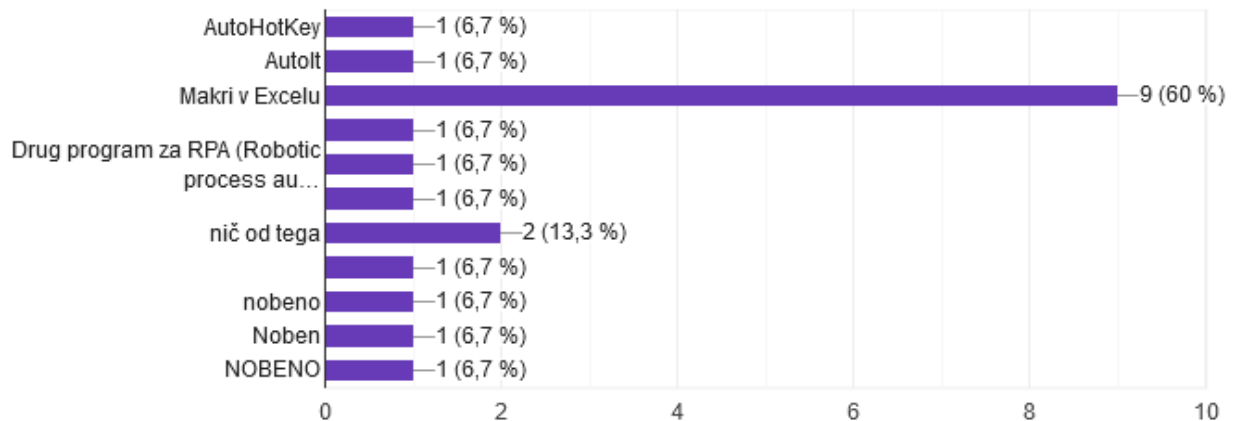
Iz odgovorov na to vprašanje je razvidno, da je zelo težko posplošiti, za kaj v podjetju je potrebna avtomatizacija. V vsakem podjetju je potrebna drugače, programe je potrebno prilagoditi za posamezno opravilo.

6. Za katera od naštetih orodij za avtomatizacijo ste že slišali?

Iz odgovorov na to vprašanje vidimo, da so daleč najbolj poznan način za avtomatizacijo opravil na

računalniku makri v Excelu. To ni presenetljivo, saj so prav ti najstarejši in najbolj razširjeni od naštetih orodji.

Prav tako vidimo, da približno tretjina anketirancev ne pozna nobenega od naštetih orodji.



Graf 4: Za katera od naštetih orodij za avtomatizacijo ste že slišali?

Na splošno iz rezultatov ankete vidimo, da je na področju avtomatizacije možna izboljšava (saj jo le približno polovica anketirancev pozna). Poleg makrov v Excelu so drugi tipi avtomatizacije nepoznani, iz česar lahko sklepamo, da popolna avtomatizacija skoraj vsem anketirancem ni poznana.

4.2 UPORABA V PODJETJU

Po nekaj tednih uporabe programov za avtomatizacijo, izdelanih v sklopu te raziskovalne naloge je bilo v podjetju prihranjeno več kot 90 % časa pri opravilih, ki so jih ti programi avtomatizirali. To pomeni, da so lahko zaposleni v tem času delali na drugih opravilih, kar pomeni, da lahko podjetje z uporabo teh programov izda več ponudb. Z uporabo programa za odklepanje datotek so si zmanjšali število napak in olajšali delo s temi datotekami.

4.3 HIPOTEZE

1. Z AutoHotKey je mogoče skrajšati čas izpolnjevanja predračuna v gradbenem podjetju za polovico.

Po pogovoru z direktorjem podjetja, v katerem so nekaj časa uporabljali programe sem dobil nekaj podatkov o tem, koliko časa so jim programi iz te raziskovalne naloge prihranili. Prvi program (za dodajanje sklicev) je pri samo enem projektu prihranil po njegovi oceni tri dni dela. Projekt je obsegal pet excel datotek s čez dvajset listov v vsaki. Program jih je vse obdelal v le nekaj minutah. S tem lahko potrdimo uspešnost in uporabnost tega programa, prav tako pa to hipotezo.

2. Večina podjetji ni seznanjena z avtomatizacijo preprostih opravil na računalniku z uporabo programske opreme.

Kot vidimo iz ankete približno polovica anketirancev ni seznanjena z avtomatizacijo preprostih opravil na računalniku in se s tem še ni srečala. Iz rezultata ankete vidimo, da lahko to hipotezo potrdimo.

5 ZAKLJUČEK

Raziskovalno nalogo sem začel delati, ko sem opazil, koliko dela na računalniku se opravlja ročno, čeprav je glavna naloga računalnika, da opravi delo namesto nas. Skozi izdelavo programov in opravljanje ankete, sem potrdil svoje hipoteze, kar pokaže, da je na tem področju potencial. Sprva sem si pri prvem programu zadal neizvedljiv cilj, kar je pokazalo, da nas računalniki ne morejo popolnoma zamenjati, toda z malo spremembami ideje, sem še vedno prišel do rešitve.

Že pred izdelavo raziskovalne naloge sem se ukvarjal z izdelavo podobnih orodji, ta naloga pa mi je pokazala nekatere nove probleme, ki jih je potrebno odpraviti. Tudi v prihodnosti ne bom prenehal s poenostavljanjem svojega dela in dela ostalih, saj vidim v temu velik problem današnje družbe.

V prihodnje bi rad bolj raziskal popolno avtomatizacijo z nestrukturiranimi podatki, saj je po mojem mnenju to najbolj uporaben tip avtomatizacije oziroma RPA, obenem pa v tem vidim prihodnost RPA. Pri programu za avtomatsko dodajanje sklicev v Excelu bi lahko ta tip RPA uporabil in naredil program, ki se lahko uporabi pri več tipih datotek (drugačen izgled in zaporedje postavk).

6 POVZETEK

Velik del delovnega časa ljudje na delovnih mestih v pisarnah opravljajo ponavljajoča opravila. Z uporabo jezika Autohotkey in drugih podobnih orodij je možno nekatera od teh opravil avtomatizirati. Po večkratnih pogovorih z ljudmi na pisarniških delovnih mestih sem opazil nepoznavanje področja avtomatizacije računalniških opravil, zato sem se odločil to področje bolj raziskati in izdelati primere programov za avtomatizacijo. Sprva me je zanimalo, če je avtomatizacija v podjetjih res nepoznana, nato pa sem se odločil izdelati še programa za avtomatizacijo različnih opravil v gradbenem podjetju. Ugotavljal sem tudi, koliko časa ti programi prihranijo. Po anketiranju v podjetjih sem ugotovil, da približno polovica anketiranih ni bila seznanjena z možnostmi avtomatizacije, kar ustvari zanimivo možnost za izboljšavo v podjetjih. Z izdelavo dveh programov za avtomatizacijo sem v konkretnem podjetju prihranil več dni dela na posamezni projekt in to nadomestil z le nekaj minutami dela.

7 ZAHVALA

Za pomoč pri izdelavi raziskovalne naloge se zahvaljujem:

- Mentorju g. Gregorju Hrastniku za nasvete in pomoč pri izdelavi raziskovalne naloge.
- Gospe dr. Nataši Meh Peer, prof.
- Staršema za ideje in podporo.
- Podjetju NIVIG in zaposlenim za pomoč pri testiranju programov.

8 VIRI IN LITERATURA

- [1] „What is Robotic Process Automation?“, [Elektronski]. Available: <https://www.uipath.com/rpa/robotic-process-automation>. [Poskus dostopa 27 3 2021].
- [2] „AutoHotkey“, [Elektronski]. Available: <https://www.autohotkey.com/>. [Poskus dostopa 23 3 2021].
- [3] „Home - AutoIt“, [Elektronski]. Available: <https://www.autoitscript.com/site/>. [Poskus dostopa 27 3 2021].
- [4] „SeleniumHQ“, [Elektronski]. Available: <https://www.selenium.dev/>. [Poskus dostopa 23 3 2021].
- [5] „ 20 BEST Selenium Alternatives in 2021“, [Elektronski]. Available: <https://www.guru99.com/selenium-alternatives.html>. [Poskus dostopa 23 3 2021].
- [6] T. Nguyen, „12 Best Automation Tools for Desktop Apps in 2021“, [Elektronski]. Available: <https://www.logigear.com/blog/test-automation/12-best-automation-tools-for-desktop-apps-in-2021/#section9>. [Poskus dostopa 23 3 2021].

A. Blagus, Poenostavljanje računalniških opravil v poslovnem okolju z uporabo skriptnega jezika AutoHotKey
Raz. nal., ERS, 2020/2021

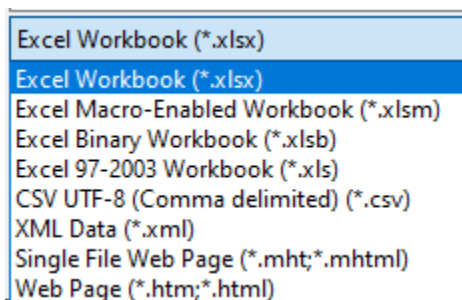
9 PRILOGE

Priloga 1: Navodila za uporabo programa za ustvarjanje sklicev

Navodila za program Sklicar

Postopek obdelave:

- Odpremo datoteko, na kateri želimo narediti sklice. Datoteka, ki jo obdelujemo mora biti shranjena v .xlsx formatu. Če ni, jo shranimo v tem formatu z naslednjim postopkom:
 - Pritisnemo gumb »Datoteka« (zgoraj levo), nato
 - »Shrani kot«
 - Izberemo lokacijo nove datoteke
 - Izberemo opcijo .xlsx (prva v seznamu):



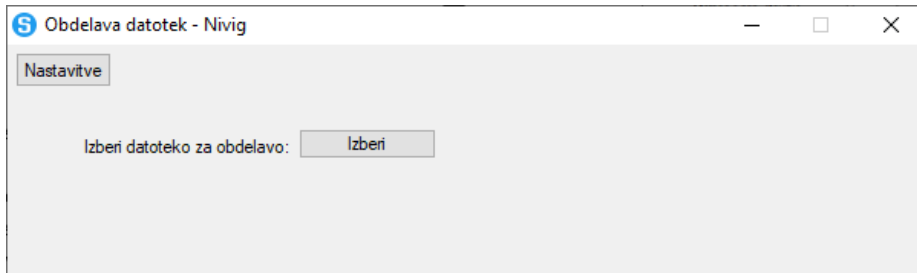
- Datoteka, ki jo želimo obdelati mora imeti odklenjene celice. Če jih nima, za odklepanje uporabimo program Odklepanje.
- Ker se popisi v Excel razlikujejo – niso poenoteni, program mogoče ne bo deloval v vseh oblikah popisov. V primeru, da pride do problemov, me kontaktirajte na e-pošto, navedeno na koncu dokumenta.
- Primer izgleda postavk, na katerih program deluje:

Zap. št	POSTAVKA	Enota	Količina	Cena na enoto	Cena skupaj
E	ZAKLJUČNA DELA				
1	Čiščenje in izpiranje kanalizacije	ura	10.00	62.00 €	620.00 €
2	Tlačni preiskus tesnosti cevovoda skladno z SIST EN 1610	m	389.59	1.50 €	584.39 €
3	Tlačni preiskus tesnosti jaškov skladno z SIST EN 1610.	kom	10	10.00 €	100.00 €
4	Pregled kanala s kamero po končanih delih	m	389.59	1.35 €	525.95 €

Postopek uporabe

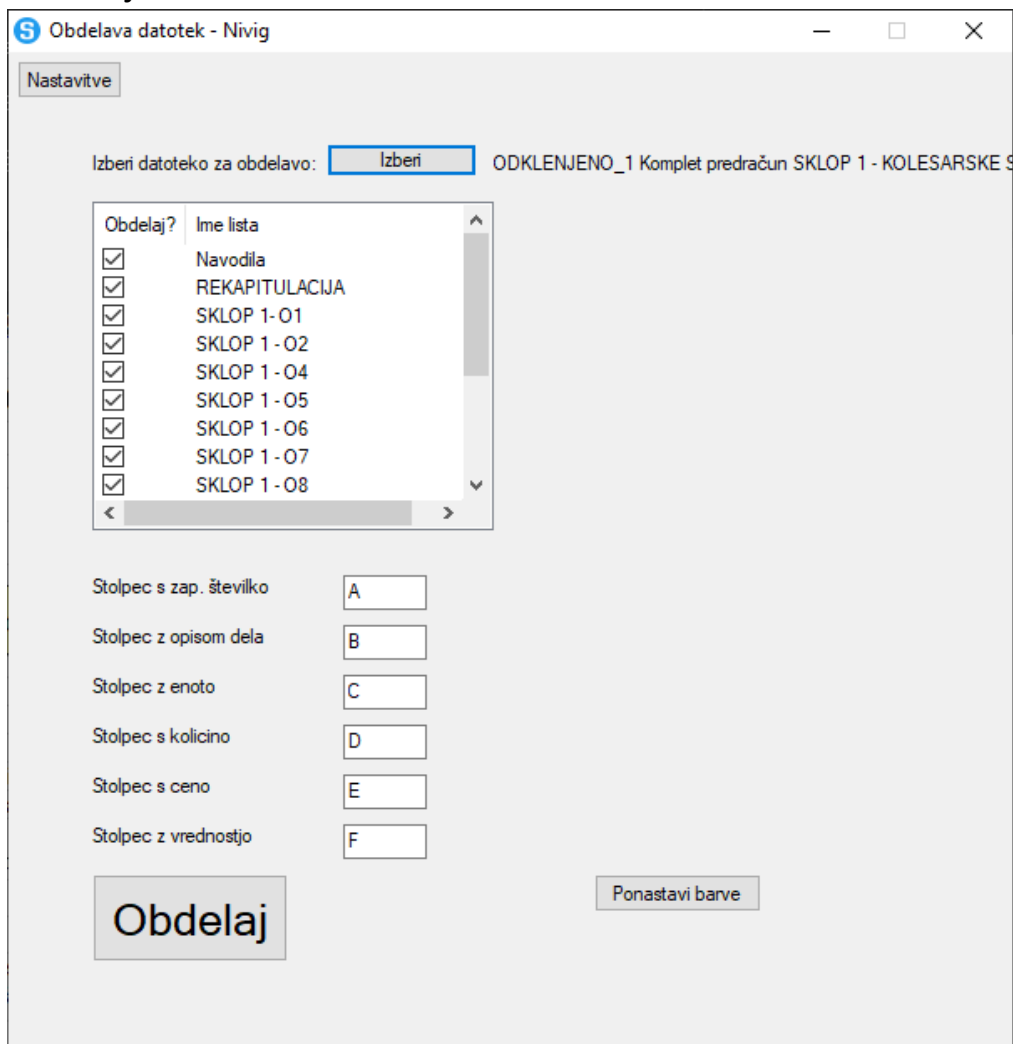
1. Odpremo program (Dvojni klik na »Sklicar«)
2. Izberemo datoteko s pritiskom na gumb »Izberi«

A. Blagus, Poenostavljanje računalniških opravil v poslovnem okolju z uporabo skriptnega jezika AutoHotKey
Raz. nal., ERS, 2020/2021



3. Počakamo, da se excel datoteka odpre

4. Prikažejo se ostale nastavitve



5. Odstranimo liste, katerih ne želimo obdelati (z odstranitvijo kljukice pred njihovim imenom)

A. Blagus, Poenostavljanje računalniških opravil v poslovnem okolju z uporabo skriptnega jezika AutoHotKey
Raz. nal., ERS, 2020/2021

6. Pravilno nastavimo stolpce!

	A	B	C	D	E	F	G
1	Projekt: NAVEZAVA ŠOLSKEGA CENTRA NA KOLESARSKO OMREŽJE SLOVENJ GRADCA						
2	V OBMOČJU MESTNE OBČINE SLOVENJ GRADEC						
3	Projekt:	SKLOP 1					
4	Odsek:	1					
5	Načrt:	PZI					
6	Faza:		PREDRAČUN				
7							
8	Postavka	Normativ	Opis postavke	Enota	Količina	Cena za enoto	Cena skupaj

V tem primeru so pravilne nastavitve to:

Stolpec s zap. številko	<input type="text" value="A"/>
Stolpec z opisom dela	<input type="text" value="C"/>
Stolpec z enoto	<input type="text" value="D"/>
Stolpec s količino	<input type="text" value="E"/>
Stolpec s ceno	<input type="text" value="F"/>
Stolpec z vrednostjo	<input type="text" value="G"/>

Opomba: Če se zgodi, da kateri stolpec manjka, na primer, da ni stolpca »Enota«, napišemo tja vrednost stolpca, ki se pojavi v istih vrsticah kot manjkajoči stolpec. Če bi v tem excelu manjkal stolpec »Enota«, bi namesto C tja napisali D, saj se količina pojavi v vseh vrsticah, kjer se pojavi enota.

7. Pritisnemo gumb »Obdelaj«
8. Programu pustimo, da deluje. Med delovanjem programa v excelu ne smemo delati karkoli drugega (to lahko prepreči njegovo delovanje in povzroči težave)

Druge možnosti:

- Nastavitve:

V njih lahko izklopimo in vklopimo pobarvanje prvih pojavitev postavke z rdečo barvo.



- Gumb »Ponastavi barve«:
Ta gumb nam omogoča ponastavitev barv prvih pojavitev postavk (rdeče obarvane, če je ta možnost omogočena v nastavitvah). Ob kliku na gumb bodo barve odstranjene *le iz izbranih listov* (na vrhu vmesnika).

Pogoste napake in kako jih odpravimo:

- Program je prikazal napako:
Če je zapisan problem, kot na primer »Excel je najverjetneje zaklenjen« sledimo tem navodilom in ga odklenemo
- Program je prikazal neprepoznavno napako (črke, številke in opis napake v angleščini):

Sporočite meni na kontaktno e-pošto, napisano na koncu tega dokumenta. Če je možno, fotografirajte napako (z uporabo telefona ali tipke »print screen«) in mi jo posredujete.

- Program ne zazna postavk/jih zazna napačno/drug problem pri obdelavi:

Napako sporočite meni na e-pošto, navedeno na koncu dokumenta. V pomoč mi bo Excel datoteka, ki povzroča napako in opis problema.

E-pošta za kontakt: anzeblagus@gmail.com

Prosim vas, da mi sporočite tudi pripombe na navodila, če vam karkoli ni jasno.

Bom popravil tudi to.

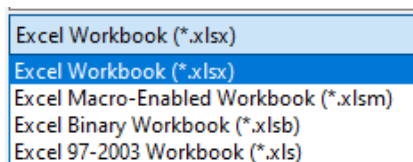
A. Blagus, Poenostavljanje računalniških opravil v poslovnem okolju z uporabo skriptnega jezika AutoHotKey
Raz. nal., ERS, 2020/2021

Priloga 2: Navodila za uporabo programa za odklepanje .xlsx datotek

Navodila za program Odklepanje

Postopek odklepanja:

- Datoteka, ki jo želimo odkleniti mora biti zaprta (ne sme biti odprta z Excelom)
- Biti mora v .xlsx formatu. Če ni, jo pred odklepanjem shranimo v ta format z naslednjim postopkom:
 - Z miško se postavimo na gumb »Datoteka« (zgoraj levo)
 - Potem pritisnemo »Shrani kot«
 - Izberemo lokacijo nove datoteke
 - Izberemo opcijo .xlsx (prva v seznamu):



Ko smo datoteko shranili, lahko nadaljujemo po naslednjem postopku:

1. Odpremo program (Dvojni klik na »Odklepanje«)
2. Izberemo zaklenjeno datoteko, ki jo želimo odkleniti
3. Počakamo nekaj sekund (lahko do 30), da program odklene datoteko
4. Ustvarjena bo nova, istoimenska .xlsx datoteka, kateri bo na začetku dodana beseda »ODKLENJENO_«

Pri postopku je tudi možnost napak. Le-te odpravimo na spodaj navedene načine:

- Če se pri izbiri datoteke nič ne zgodi:

Najprej počakamo cca eno minuto, potem pa ponovno poskusimo, če tudi v drugo ne gre, kontaktirajte na spodnji mail.

- V primeru, da program prikaže napako:

Prepričamo se, da je datoteka res v .xlsx formatu, da ni dvakrat odprta in poskusimo še enkrat. Če še vedno ne deluje me kontaktirajte na e-pošto anzeblagus@gmail.com

Priloga 3: Anketni vprašalnik

Anketa o poznavanju avtomatizacije v poslovnem okolju

<https://docs.google.com/forms/u/0/d/1Uz0DIDp0ZTth1gI2kWza0onff5d...>

Anketa o poznavanju avtomatizacije v poslovnem okolju

Sem Anže Maj Blagus, dijak Elektro in računalniške šole Velenje. V letošnjem šolskem letu delam raziskovalno nalogo na temo avtomatizacije opravil na računalniku. Kot del naloge bom ugotavljal, kako se podjetja spopadajo s ponavljajočimi opravili na posameznih delovnih mestih. Vesel bom, če si boste vzeli nekaj minut za izpolnjevanje kratke ankete.

Anketa je namenjena osebi, ki je najbolj seznanjena s stanjem avtomatizacije v vašem podjetju (računovodja, direktor, IT administrator...).

Avtomatizacija opravil na računalniku vključuje programe za avtomatsko pošiljanje e-pošte, programe za avtomatsko izpolnjevanje spletnih anket, programe za pridobivanje podatkov iz spletnih strani (zbiranje e-poštnih naslovov iz socialnih omrežij in podobno), programe za pomoč pri določenih ponavljajočih opravilih in podobno.

Nekaj primerov avtomatizacije preprostih opravil (klik na namizje zapre vsa okna, nekaj dodatkov za starejše verzije Windowsa, izklop monitorja s pritiskom kombinacije tipk in veliko drugih): <https://www.dcmembers.com/akrommel/downloads/>

Nekaj bolj naprednih primerov pa najdete tukaj: <https://www.autohotkey.com/docs/scripts>

* Zahtevano

1. Koliko zaposlenih imate v podjetju? *

Označite samo en oval.

- 1-5
- 6-10
- 11-50
- 51 in več
- Drugo: _____

2. Kaj je vaša naloga v podjetju (marketing, računovodstvo, izdajanje ponudb, IT administrator...)?

3. Ali ste se vi osebno že srečali z avtomatizacijo opravil na računalniku?

Označite samo en oval.

- Da
- Ne
- Drugo: _____

4. Ali vam je znano, če v vašem podjetju uporabljate avtomatizacijo opravil na računalniku (npr. masovno pošiljanje e-sporočil, avtomatsko izpolnjevanje obrazcev)? *

Označite samo en oval.

- Da *Prekočite na vprašanje 5*
- Ne *Prekočite na vprašanje 7*
- Drugo: _____

Prekočite na vprašanje 8

Sklop 2

5. Prosim na kratko opišite, s kakšno avtomatizacijo opravil na računalniku ste se srečali. *

6. Na katerem področju v vašem podjetju bi lahko uvedli avtomatizacijo opravil na računalniku (poleg obstoječe, če je že na mestu)? *

Preskočite na vprašanje 8

Skop
2

Avtomatizacija opravil na računalniku vključuje programe za avtomatsko pošiljanje e-pošte, programe za avtomatsko izpolnjevanje spletnih anket, programe za pridobivanje podatkov iz spletnih strani (zbiranje e-poštnih naslovov iz socialnih omrežij in podobno), programe za pomoč pri določenih ponavljajočih opravilih in podobno.

Primeri programov najdete na:
<https://www.domembers.com/skrommel/downloads/>
 in
<https://www.autohotkey.com/docs/scripts>

7. Na katerem področju v vašem podjetju bi lahko uvedli avtomatizacijo opravil na računalniku? *

Preskočite na vprašanje 6

Skop 3

A. Blagus, Poenostavljanje računalniških opravil v poslovnem okolju z uporabo skriptnega jezika AutoHotKey
Raz. nal., ERS, 2020/2021

Anketa o poznavanju avtomatizacije v poslovnem okolju

<https://docs.google.com/forms/w/0/d/1Uz0DIDp0ZTth1gI2kWza0onf65d...>

8. Za katera od naštetih orodij za avtomatizacijo ste že slišali? *

Izberite vse primerno odgovore.

- AutoHotKey
- AutoIt
- Makri v Excelu
- Selenium za avtomatizacijo brskalnika
- Drug program za RPA (Robotic process automation)
- Drug program za ustvarjanje makrov

Drugo: _____

9. Če želite, lahko spodaj vpišete e-pošto, katero lahko kontaktiram po končanju raziskovalne.

Google ni mtl ustvaril niti odobril te vsebine.

Google Obrazci

A. Blagus, Poenostavljanje računalniških opravil v poslovnem okolju z uporabo skriptnega jezika AutoHotKey
Raz. nal., ERS, 2020/2021

Priloga 4: Koda programa za odklepanje .xlsx datotek

```
#SingleInstance force
SetBatchLines, -1
SetWorkingDir, %a_scriptdir%
onexit, exitme
7zPath:="C:\Program Files\7-Zip\7z.exe"
args=%0%

filesToDo:=[]

if(args==0)
{
    FileSelectFile, files, M3,,Izberi datoteke za odklepanje, *.xlsx
    if(files=="")
        ExitApp
    Loop, parse, files, `n
    {
        if (A_Index = 1)
            path := A_LoopField
        else
        {
            filesToDo.push(path "\" A_LoopField)
        }
    }
} else {
    loop, %args%
    {
        thisFile:=%a_index%
        SplitPath, thisFile, name, dir, ext, noext
        if(ext!="xlsx")
        {
            msgbox, Datoteke morajo biti tipa .xlsx, da ta program pravilno
            deluje! Pretvorite in poskusite ponovno
            exitapp
        }
        filesToDo.push(thisFile)
    }
}

for k,v in filesToDo
{
    doFile(v)
}

Msgbox, ,ODKLEPANJE,Odklepanje koncano!,10

;~ path:="D:\Projekti\!Other\nivig unlock\asd\test.xlsx"

return
```

A. Blagus, Poenostavljanje računalniških opravil v poslovnem okolju z uporabo skriptnega jezika AutoHotKey
Raz. nal., ERS, 2020/2021

```
doFile(path)
{
    global 7zPath

    SplitPath, path, n, dir, e, brez_konc

    SetWorkingDir, %dir%

    zip = %dir%\%brez_konc%.zip ; path do zip datoteke

    outZip = out_%brez_konc%.zip ; path do izhodne zip datoteke
    outFile = ODKLENJENO_%brez_konc%.xlsx ; path do izhodne datoteke
    outDir=%dir%\out_%brez_konc% ; path do mape za razpakiranje

    FileMove, %path%, %zip%, 1 ; preimenujemo datoteko v .zip

    runwait, "%7zPath%" x "%zip%" -o"%outDir%",,Hide ; razpakiramo

    sheetsDir:=outdir "\xl\worksheets" ; pot do mape z vsemi listi

    while(!fileExist(sheetsDir)) ; za vsak slucaj pocakamo, da obstaja mapa
        sleep, 10

    loop, Files, %sheetsDir%\*.xml ; za vsak list
    {
        if(substr(a_loopfilename,1,5)!="sheet")
            continue

        FileRead, XMLdata, %A_LoopFileFullPath% ; preberemo vsebino

        ; naslednje vrstice:
        ; - locirajo XML znacko "sheetProtection"
        ; - locirajo njen konec
        ; - jo odstranijo
        loc := instr(XMLdata,"<sheetProtection")
        if(!loc)
            continue
        rest := substr(XMLdata,loc)
        loc2 := instr(rest,"/>")

        before:=substr(XMLdata,1,loc-1)
        after:=substr(rest,loc2+2)

        FileDelete, %A_LoopFileFullPath% ; zbrisemo staro datoteko
        FileAppend, %before%%after%, %A_LoopFileFullPath% ; napisemo
        odklenjeno datoteko
    }

    ; nazaj zapakiramo in izbrisemo preostale datoteke

    runwait, "%7zPath%" a "%outZip%" ".\out_%brez_konc%*",,Hide
    FileRemoveDir, %outDir%, 1
    FileMove, %zip%, %path%
    FileMove, %outZip%, %outFile%
```

A. Blagus, Poenostavljanje računalniških opravil v poslovnem okolju z uporabo skriptnega jezika AutoHotKey
Raz. nal., ERS, 2020/2021

```

}

exitme:
try
{
    xmlDoc.quit ()
}
FileRemoveDir, %outDir%, 1
FileMove, %zip%, %path%
exitapp
return

```

Priloga 5: Koda programa za ustarjanje sklicev

```

#SingleInstance force
SetBatchLines, -1
SetWorkingDir, %a_scriptdir%

args=%0%
autoLoad:=""
if (args!=0)
{
    autoLoad = %1%
}

file:=a_scriptdir "\main.xls"
list:=["1. faza-I. Pripravljalna dela", "1. faza-II_cestno omrežje ", "1. faza-
MK I"]

hideList:=["PriceColT", "PriceCol", "DescColT", "DescCol", "SheetList", "StartBtn",
"ResetClrBtn", "NumColT", "NumCol", "makeDictBtn"]

defaultSettings:={"pobarvaj":
{"value": "false", "type": "checkbox", "text": "Pobarjav prve pojavitve"}}

settingsFile:= A_ScriptDir "\nastavitve.json"

settings.Load ()
settings.Save ()

currentExcel:=""
; ++++++ USTVARJANJE GUI
+++++
{
Gui, Add, text, x50 y60, Izberi datoteko za obdelavo:
Gui, Add, Button, gBrowse x140 y55 w90 h20, Izberi
Gui, Add, text, x100 y60 w500 vChosenFileName

Gui, Add, ListView, r10 x50 yp30 AltSubmit Checked vSheetList Hdr,
Obdelaj?|Ime lista

; Stolpci

gui, Add, Text, x50 yp225 vNumColT, Stolpec s zap. tevilko

```

A. Blagus, Poenostavljanje računalniških opravil v poslovnem okolju z uporabo skriptnega jezika AutoHotKey
Raz. nal., ERS, 2020/2021

```

gui, Add, Edit, xp150 yp w50 vNumCol,A

gui, Add, Text, x50 yp30 vDescColT,Stolpec z opisom dela
gui, Add, Edit, xp150 yp w50 vDescCol,B

gui, Add, Text, x50 yp30 vUnitColT,Stolpec z enoto
gui, Add, Edit, xp150 yp w50 vUnitCol,C

gui, Add, Text, x50 yp30 vAmountColT,Stolpec s kolicino
gui, Add, Edit, xp150 yp w50 vAmountCol,D

gui, Add, Text, x50 yp30 vPriceColT,Stolpec s ceno
gui, Add, Edit, xp150 yp w50 vPriceCol,E

gui, Add, Text, x50 yp30 vValueColT,Stolpec z vrednostjo
gui, Add, Edit, xp150 yp w50 vValueCol,F

gui,font,s20
gui, Add, Button, x50 yp30 w50 vStartBtn gStart,Obdelaj
gui,font

gui, Add, Button, xp300 yp w100 vResetClrBtn gResetClrs,Ponastavi barve

for k,v in hideList
    guicontrol, hide, %v%

gui, add,Button, x5 y5 gsettingsBtn, Nastavitve

Gui, Show,h150 w600 Center, Obdelava datotek - Nivig

; ----- USTVARJANJE GUI -----
}

gui,submit,nohide

global globObj:=[]
sestevkArr := []

if(autoLoad!="")
{
    LoadFile(autoLoad)
}

;~ mainXL:=OpenExcel(file)

;~ msgbox, here
return

ResetClrs:
gui,submit,nohide
resetColors(mainXL, LV_GetChecked(), PriceCol, 3, 2)
Msgbox, Barve ponastavljene!
return

```

A. Blagus, Poenostavljanje računalniških opravil v poslovnem okolju z uporabo skriptnega jezika AutoHotKey
Raz. nal., ERS, 2020/2021

```

; ++++++ KLIK GUMBA ZACNI ++++++
; ++++++
{
Start:
Gui, submit, nohide
;~ msgbox, % o2s(LV_GetChecked())
doExcel(mainXL,LV_GetChecked())

Msgbox, Excel obdelan!

; reset everything
globObj:=[]
sestevArr := []

coloredArr := {}
return
; ----- KLIK GUMBA ZACNI -----
}

; ++++++ KLIK GUMBA IZBERI ++++++
; ++++++
{
Browse:
FileSelectFile, SelectedFile, 3, , Izberi datoteko za obdelavo, Excel (*.xls;
*.xlsx; *.xlsm)
if(!SelectedFile)
return

LoadFile(SelectedFile)
Return

LoadFile(SelectedFile)
{
global

SplitPath,SelectedFile,Name,dir,ext,noext

; if not excel file
if(substr(ext,1,3)!="xls")
{
Msgbox, Podana ni bila Excel datoteka!
return
}
fileLoadGUI := new LoadGUI("Odpiranje","Odpiranje izbrane
datoteke...",true)

GuiControl, ,ChosenFileName,%Name%
mainXL:=OpenExcel(SelectedFile)
currentExcel:=SelectedFile
fileLoadGUI.Hide()
fileLoadGUI:=""
LV_Delete()

For sheet in mainXL.Worksheets

```

A. Blagus, Poenostavljanje računalniških opravil v poslovnem okolju z uporabo skriptnega jezika AutoHotKey
Raz. nal., ERS, 2020/2021

```

    {
        LV_Add(Checked, "", sheet.name)
        LV_Modify(Row%a_index%, "Check")
    }

    for k,v in hideList
        guicontrol, show, %v%

    Gui, Show, h600 w600 Center
}

; ----- KLIK GUMBA IZBERI -----
}

; ++++++ KLIK GUMBA NASTAVITVE
++++++
{
settingsBtn:
setGUI:=new SettingsGUI(globalSettings)

return
; ----- KLIK GUMBA NASTAVITVE -----
}

;~ f10::
;~ msgbox, doing
;~ SerDes(sestevekArr, "sestevekArr.txt")
;~ msgbox, done
;~ return

processPostavka(post)
{
    global mainXL, sestevekArr, globalSettings
    val := mainXL.ActiveSheet.Range(post.location).value
    ;~ msgbox, % post.text
    if(referenceFrom:=hasPostavka(post))
    ;~ if(false)
    {
        ;~ msgbox, % post.location " to " referenceFrom
        if(val == "") ; ne povezi
        {
            mainXL.ActiveSheet.Range(post.location).value := "=" referenceFrom
            ;~ msgbox, % "last " sestevekArr[post.text]
            sestevekArr[post.text] += post.amount
        }
    } else {
        sestevekArr[post.text] := post.amount
        if(globalSettings.pobarvaj.value)
        {
            ;~ msgbox, % post.location
            ;~
            coloredArr.Push(mainXL.ActiveSheet.Range(post.location).Interior.ColorIndex)
            try {
                mainXL.ActiveSheet.Range(post.location).Interior.ColorIndex := 3
            } catch e {

```


A. Blagus, Poenostavljanje računalniških opravil v poslovnem okolju z uporabo skriptnega jezika AutoHotKey
Raz. nal., ERS, 2020/2021

```

        msgbox, Problem pri obdelavi. Najverjetneje je excel datoteka
zaklenjena!
        exitapp
    }
}
;~ msgbox, % "this is first " post.location

globObj[post.text]:=post.location
}

}

hasPostavka(post)
{
    return globObj[post.text]
}

resetColors(xl, list, column, fromI, toI)
{
    global NumCol, DescCol, UnitCol, AmountCol, PriceCol, ValueCol
    gui, submit, nohide
    for k,v in list
    {
        xl.Worksheets(v).Activate
        emptyRows:=0

        while(emptyRows<20)
        {
            val1 := rng(NumCol a_index)
            val2 := rng(DescCol a_index)
            val3 := rng(UnitCol a_index)
            val4 := rng(AmountCol a_index)
            val5 := rng(PriceCol a_index)
            val6 := rng(ValueCol a_index)
            if(val1="" && val2="" && val3="" && val4="" && val5="" &&
val6="")
            {
                ;~ msgbox, % o2s(row) " is empty"
                emptyRows++
                continue
            }
            emptyRows:=0
            cell := xl.ActiveSheet.Range(column a_index)
            if(cell.Interior.ColorIndex == fromI)
            {
                cell.Style := "Normal"
            }
        }
    }
}

doExcel(xl, list)
{

```

A. Blagus, Poenostavljanje računalniških opravil v poslovnem okolju z uporabo skriptnega jezika AutoHotKey
Raz. nal., ERS, 2020/2021

```

global
NumCol, DescCol, UnitCol, AmountCol, PriceCol, ValueCol, sestevkaArr, currentExcel
gui, submit, nohide
for k, v in list
{
    xl.Worksheets(v).Activate
    emptyRows:=0

    thisPostavka:=new postavka()
    i:=findFirstLine(xl.ActiveSheet)
    ;~ msgbox, %i%
    while(emptyRows<20)
    {
        i++

        row:=new row(rng(NumCol i), rng(DescCol i), rng(UnitCol i), rng(AmountCol
i), rng(PriceCol i), rng(ValueCol i))
        if(row.isEmpty())
        {
            ;~ msgbox, % o2s(row) " is empty"
            emptyRows++
            continue
        }
        emptyRows:=0

        ;~ if(row.onlyDesc())
            ;~ continue

        thisPostavka.hadNum:=(row.hasNum() ? true : thisPostavka.hadNum)
        thisPostavka.hadOther:=(row.hasOther() ? true : thisPostavka.hadOther)

        if(!thisPostavka.hadNum)
            thisPostavka.text:=prevPostavkaText "`n"

        thisPostavka.text:=(row.desc) ? (" " row.desc) : ("")
        thisPostavka.amount := ((row.amount) ? (row.amount) :
(thisPostavka.amount))
        ;~ msgbox, % o2s(row) "`n" o2s(thisPostavka)
        if(thisPostavka.hadOther && thisPostavka.location=="")
        {
            thisPostavka.location := "" xl.ActiveSheet.name "!" PriceCol i
        }

        if(thisPostavka.shouldBeProcessed())
        {
            ;~ msgbox, % "postavka na " thisPostavka.location "`n"
            thisPostavka.text
            processPostavka(thisPostavka)
            if(thisPostavka.hadNum) ; samo za glavne
                prevPostavkaText:=thisPostavka.text
            thisPostavka:=new postavka()
        }
    }
}
}

```

A. Blagus, Poenostavljanje računalniških opravil v poslovnem okolju z uporabo skriptnega jezika AutoHotKey
Raz. nal., ERS, 2020/2021

```

SplitPath, currentExcel, name, dir, ext, noext
outTxtName = %noext%_%a_now%_sestevek.txt
for k,v in sestevekArr
{
    if(v!="")
        FileAppend, `n%k%`n%v%`n, %outTxtName%
}

}

findFirstLine(xl)
{
    global NumCol,DescCol,UnitCol,AmountCol,PriceCol,ValueCol
    gui, submit, nohide
    fullLineFound:=false
    i:=1
    while(!fullLineFound && i < 200)
    {
        row:=new row(rng(NumCol i,xl),rng(DescCol i,xl),rng(UnitCol
i,xl),rng(AmountCol i,xl),rng(PriceCol i,xl),rng(ValueCol i,xl))
        if(row.isFull())
        {
            break
        }
        i++
    }

    return i + 1
}

rng(what,xl:="DEF")
{
    global mainXL
    if(xl=="DEF")
        xl:=mainXL.ActiveSheet
    return xl.Range(what).Value
}

OpenExcel(openpath,hidden:=false) {
    SplitPath, openpath, name
    if(!WinExist(name)) {
        XL := ComObjCreate("Excel.Application")
        ;~ msgbox, %openpath%
        XL.Workbooks.Open(openpath)

        XL.Visible := !hidden
        return XL
    } else {
        ;~ XL := ComObjActive("Excel.Application")
        XL := ComObjGet(openpath)
        return XL
    }
}

; + CLASSI +
{

```

A. Blagus, Poenostavljanje računalniških opravil v poslovnem okolju z uporabo skriptnega jezika AutoHotKey
Raz. nal., ERS, 2020/2021

```

; GUI za nalaganje
Class LoadGUI
{
    __New(Title, Text)
    {
        this.title:=Title
        this.text:=Text
        this.Show()
    }
    __Delete()
    {
        gui loading:destroy
    }
    Show()
    {
        gui, loading:-MaximizeBox -MinimizeBox +AlwaysOnTop
        gui loading:Add, Text,,% this.text
        gui loading:Show,Autosize Center,% this.title
    }
    Hide()
    {
        gui loading:Hide
    }
}

class postavka
{
    __New()
    {
        this.text:=""
        this.location:=""
        this.hadNum:=false
        this.hadOther:=false
        this.amount:=""
    }

    shouldBeProcessed()
    {
        if(this.text!="" && this.hadOther)
        ;~ if(this.text!="" && this.hadNum && this.hadOther)
            return true
        return false
    }
}

class row
{
    __New(num,desc,unit,amount,price,value)
    {
        this.num:=num
        this.desc:=desc
        this.unit:=unit
        this.amount:=amount
        this.price:=price
        this.value:=value
    }
}

```

A. Blagus, Poenostavljanje računalniških opravil v poslovnem okolju z uporabo skriptnega jezika AutoHotKey
Raz. nal., ERS, 2020/2021

```

    }

    isEmpty()
    {
        ;~ if (this.desc != "" && this.unit == "" && this.amount == "")
        ;~ return true
        for k,v in this
            if(v!="")
                return false

        return true
    }
    isFull()
    {
        for k,v in this
            if(v=="")
                return false

        return true
    }
    onlyDesc()
    {
        if(this.desc != "" && this.num=="" && this.unit=="" && this.amount=="")
        && this.price=="")
            return true
        return false
    }
    hasNum()
    {
        if(this.num!="")
            return true
        return false
    }
    hasOther()
    {
        if(this.unit!="" && this.amount!="")
            return true
        return false
    }
}

; GUI za nalaganje
Class SettingsGUI
{
    __New(settings)
    {
        this.settingsObj:=settings
        this.Show()
    }
    Show()
    {
        global
        gui settings:destroy
        gui settings:Font,s20
    }
}

```

A. Blagus, Poenostavljanje računalniških opravil v poslovnem okolju z uporabo skriptnega jezika AutoHotKey
Raz. nal., ERS, 2020/2021

```

gui settings:Add, Text, x50 y30,Nastavitve
gui settings:Font,s0
gui settings:Add, Text,yp20 x-500
gui settings:Font,s11

for name,v in this.settingsObj
{
    text:=v.text
    value:=v.value
    type:=v.type
    switch (type)
    {
        case "edit":
            gui settings:Add, Text,x50 yp+30,%text%

            gui settings:Add, Edit,x50 v%name% yp xp+300,%value%
            break
        case "file":

            gui settings:Add, Text,x50 yp+30,%text%
            gui settings:Add, Button,yp xp+300 gSelectFileSettings

v%name%,Izberi
            gui settings:font, s7
            gui settings:Add, Text,xp+100 yp v%name%Text w400,%value%
            gui settings:font, s11

            break

        case "checkbox":

            gui settings:Add, Text,x50 yp+30,%text%

            gui settings:Add, Checkbox,x50 v%name% g%name%Change yp xp+300
            GuiControl, settings:,%name%, %value%
            break
    }
}

}

gui settings:Show,Autosize Center,% this.title

}
Hide()
{
    gui settings:Hide
}
}

class settings {
    Save()
    {
        static
        global globalSettings,settingsFile
        saveSettings:=Jxon_Dump(globalSettings)
        FileDelete, %settingsFile%
    }
}

```

A. Blagus, Poenostavljanje računalniških opravil v poslovnem okolju z uporabo skriptnega jezika AutoHotKey
Raz. nal., ERS, 2020/2021

```

        FileAppend, %saveSettings%, %settingsFile%
    }
    Load()
    {
        static
        global globalSettings, settingsFile, defaultSettings
        if (FileExist(settingsFile))
        {
            fileread, settingsRead, %settingsFile%
            globalSettings := Jxon_Load(settingsRead)
        } else {
            globalSettings := defaultSettings
        }
    }
}

; - CLASSI -
}

guiclose:
exitapp
return

pobarvajChange:
GuiControlGet, val, ,pobarvaj
globalSettings.pobarvaj.value := val
settings.Save()
return

; Dobi vse izbrane vrstice iz ListView
LV_GetChecked() {
    checkedRowList := []
    while rowNumber := LV_GetNext(rowNumber, "C")
        checkedRowList.Push(rowNumber)
    return checkedRowList
}

; + OBJECT TO STRING +
{
o2s(obj) {

    Linear := isLinear(obj)

    For e, v in obj {
        if (Linear == False) {
            if (isObject(v))
                r .= e ":" o2s(v) ", "
            else {
                r .= e ":"
                if v is number
                    r .= v ", "
            }
        }
    }
}
}

```

A. Blagus, Poenostavljanje računalniških opravil v poslovnem okolju z uporabo skriptnega jezika AutoHotKey
Raz. nal., ERS, 2020/2021

```

        else
            r .= "" v "", "
    }
} else {
    if (isObject(v))
        r .= o2s(v) ", "
    else {
        if v is number
            r .= v ", "
        else
            r .= "" v "", "
    }
}
}
return Linear ? "[" trim(r, ", ") "]"
              : "{" trim(r, ", ") }"
}

isLinear(obj) {
    n := obj.count(), i := 1
    loop % (n / 2) + 1
        if (!obj[i++] || !obj[n--])
            return 0
    return 1
}
; - OBJECT TO STRING -
}

; JXON LIB
{
Jxon_Load(ByRef src, args*)
{
    static q := Chr(34)

    key := "", is_key := false
    stack := [ tree := [] ]
    is_arr := { (tree): 1 }
    next := q . "{[01234567890-tfn"
    pos := 0
    while ( (ch := SubStr(src, ++pos, 1)) != "" )
    {
        if InStr(" `t`n`r", ch)
            continue
        if !InStr(next, ch, true)
        {
            ln := ObjLength(StrSplit(SubStr(src, 1, pos), "`n"))
            col := pos - InStr(src, "`n",,, -(StrLen(src)-pos+1))

            msg := Format("{}: line {} col {} (char {})"
, (next == "")      ? ["Extra data", ch := SubStr(src, pos)][1]
: (next == "'")    ? "Unterminated string starting at"
: (next == "\")    ? "Invalid \escape"
: (next == ":")    ? "Expecting ':' delimiter"
: (next == q)      ? "Expecting object key enclosed in double
quotes"

```


A. Blagus, Poenostavljanje računalniških opravil v poslovnem okolju z uporabo skriptnega jezika AutoHotKey
Raz. nal., ERS, 2020/2021

```

        : (next == q . "}") ? "Expecting object key enclosed in double
quotes or object closing '}'"
        : (next == ",}") ? "Expecting ',' delimiter or object closing
'}'"
        : (next == ",]") ? "Expecting ',' delimiter or array closing
']'"
        : [ "Expecting JSON value(string, number, [true, false, null],
object or array)"
          , ch := SubStr(src, pos, (SubStr(src, pos)~="[\]\},\s|$$")-1)
] [1]
        , ln, col, pos)

    throw Exception(msg, -1, ch)
}

is_array := is_arr[obj := stack[1]]

if i := InStr("{[" , ch)
{
    val := (proto := args[i]) ? new proto : {}
    is_array? ObjPush(obj, val) : obj[key] := val
    ObjInsertAt(stack, 1, val)

    is_arr[val] := !(is_key := ch == "{")
    next := q . (is_key ? "}" : "{[0123456789-tfn")
}

else if InStr("}]"] , ch)
{
    ObjRemoveAt(stack, 1)
    next := stack[1]==tree ? "" : is_arr[stack[1]] ? ",]" : ",}"
}

else if InStr(",:." , ch)
{
    is_key := (!is_array && ch == ",")
    next := is_key ? q : q . "{[0123456789-tfn"
}

else ; string | number | true | false | null
{
    if (ch == q) ; string
    {
        i := pos
        while i := InStr(src, q,, i+1)
        {
            val := StrReplace(SubStr(src, pos+1, i-pos-1), "\\\"",
"\u005C")

            static end := A_AhkVersion<"2" ? 0 : -1
            if (SubStr(val, end) != "\"")
                break
        }
        if !i ? (pos--, next := "") : 0
            continue

        pos := i ; update pos
    }
}

```

A. Blagus, Poenostavljanje računalniških opravil v poslovnem okolju z uporabo skriptnega jezika AutoHotKey
Raz. nal., ERS, 2020/2021

```

        val := StrReplace(val, "\/", "/")
        val := StrReplace(val, "\" . q, q)
        val := StrReplace(val, "\b", "`b")
        val := StrReplace(val, "\f", "`f")
        val := StrReplace(val, "\n", "`n")
        val := StrReplace(val, "\r", "`r")
        val := StrReplace(val, "\t", "`t")

i := 0
while i := InStr(val, "\",, i+1)
{
    if (SubStr(val, i+1, 1) != "u") ? (pos -=
StrLen(SubStr(val, i)), next := "\") : 0
        continue 2

    ; \uXXXX - JSON unicode escape sequence
    xxxx := Abs("0x" . SubStr(val, i+2, 4))
    if (A_IsUnicode || xxxx < 0x100)
        val := SubStr(val, 1, i-1) . Chr(xxxx) . SubStr(val,
i+6)
}

if is_key
{
    key := val, next := ":"
    continue
}
}

else ; number | true | false | null
{
    val := SubStr(src, pos, i := RegExMatch(src, "[\]\},\s]|$",,,
pos)-pos)

; For numerical values, numerify integers and keep floats as is.
; I'm not yet sure if I should numerify floats in v2.0-a ...
    static number := "number", integer := "integer"
    if val is %number%
    {
        if val is %integer%
            val += 0
    }

; in v1.1, true,false,A_PtrSize,A_IsUnicode,A_Index,A_EventInfo,
; SOMETIMES return strings due to certain optimizations. Since it
; is just 'SOMETIMES', numerify to be consistent w/ v2.0-a
    else if (val == "true" || val == "false")
        val := %value% + 0

; AHK_H has built-in null, can't do 'val := %value%' where value
== "null"
; as it would raise an exception in AHK_H(overriding built-in var)
    else if (val == "null")
        val := ""

; any other values are invalid, continue to trigger error
    else if (pos--, next := "#")
        continue

    pos += i-1

```

A. Blagus, Poenostavljanje računalniških opravil v poslovnem okolju z uporabo skriptnega jezika AutoHotKey
Raz. nal., ERS, 2020/2021

```

    }

    is_array? ObjPush(obj, val) : obj[key] := val
    next := obj==tree ? "" : is_array ? ",]" : ",}"
  }
}

return tree[1]
}

Jxon_Dump(obj, indent:="", lvl:=1)
{
  static q := Chr(34)

  if IsObject(obj)
  {
    static Type := Func("Type")
    if Type ? (Type.Call(obj) != "Object") : (ObjGetCapacity(obj) == "")
      throw Exception("Object type not supported.", -1, Format("<Object
at 0x{:p}>", &obj))

    is_array := 0
    for k in obj
      is_array := k == A_Index
    until !is_array

    static integer := "integer"
    if indent is %integer%
    {
      if (indent < 0)
        throw Exception("Indent parameter must be a postive integer.",
-1, indent)
      spaces := indent, indent := ""
      Loop % spaces
        indent .= " "
    }
    indt := ""
    Loop, % indent ? lvl : 0
      indt .= indent

    lvl += 1, out := "" ; Make #Warn happy
    for k, v in obj
    {
      if IsObject(k) || (k == "")
        throw Exception("Invalid object key.", -1, k ? Format("<Object
at 0x{:p}>", &obj) : "<blank>")

      if !is_array
        out .= ( ObjGetCapacity([k], 1) ? Jxon_Dump(k) : q . k . q )
; // key
        . ( indent ? ": " : ":" ) ; token + padding
      out .= Jxon_Dump(v, indent, lvl) ; value
        . ( indent ? ",\n" . indt : "," ) ; token + indent
    }

    if (out != "")
    {

```

A. Blagus, Poenostavljanje računalniških opravil v poslovnem okolju z uporabo skriptnega jezika AutoHotKey
Raz. nal., ERS, 2020/2021

```

        out := Trim(out, ",`n" . indent)
        if (indent != "")
            out := "`n" . indt . out . "`n" . SubStr(indt,
StrLen(indent)+1)
    }

    return is_array ? "[" . out . "]" : "{" . out . "}"
}

; Number
else if (ObjGetCapacity([obj], 1) == "")
    return obj

; String (null -> not supported by AHK)
if (obj != "")
{
    obj := StrReplace(obj, "\", "\\")
    , obj := StrReplace(obj, "/", "\/")
    , obj := StrReplace(obj, q, "\" . q)
    , obj := StrReplace(obj, "`b", "\b")
    , obj := StrReplace(obj, "`f", "\f")
    , obj := StrReplace(obj, "`n", "\n")
    , obj := StrReplace(obj, "`r", "\r")
    , obj := StrReplace(obj, "`t", "\t")

    static needle := (A_AhkVersion<"2" ? "O" : "") . "[^\x20-\x7e]"
    while RegExMatch(obj, needle, m)
        obj := StrReplace(obj, m[0], Format("\u{:04X}", Ord(m[0])))
}

return q . obj . q
}
}

```