

Automatska izrada PDF izvješća koristeći programske jezike

Štrkalj, Josip

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:235:491989>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-01-03**

Repository / Repozitorij:

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Josip Štrkalj

Zagreb, 2023. godina

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Mentor:

Dr. sc. Tomislav Stipančić, dipl. ing.

Student:

Josip Štrkalj

Zagreb, 2023. godina

Izjavljujem da sam ovaj rad izradio samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.

Zahvaljujem se svom mentoru dr. sc. Tomislavu Stipančiću na zadavanju teme završnog rada te pruženoj pomoći, uputama i savjetima.

Zahvaljujem se svojoj obitelji na pomoći i potpori tokom studiranja te kolegama i prijateljima bez kojih provedeno vrijeme na fakultetu ne bi bilo isto.

Josip Štrkalj



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

Središnje povjerenstvo za završne i diplomske ispite

Povjerenstvo za završne i diplomske ispite studija strojarstva za smjerove:
 proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo
 materijala i mehatronika i robotika



| | |
|-------------------------------------|--------|
| Sveučilište u Zagrebu | |
| Fakultet strojarstva i brodogradnje | |
| Datum | Prilog |
| Klasa: 602 – 04 / 23 – 6 / 1 | |
| Ur.broj: 15 - 1703 - 23 - | |

ZAVRŠNI ZADATAK

Student: **Josip Štrkalj** JMBAG: **0035220748**

Naslov rada na hrvatskom jeziku: **Automatska izrada PDF izvješća koristeći programske jezike**

Naslov rada na engleskom jeziku: **Automatic creation of PDF reports using programming languages**

Opis zadatka:

Programski jezik Python nudi velik izvor metoda i biblioteka koje omogućuju brzo i automatsko kreiranje i dijeljenje sadržaja među aplikacijama.

U radu je potrebno izraditi PDF izvješće za projekt čija je tema obrada podataka prateći sljedeće korake:

- identificirati i proučiti prikladne Python računalne biblioteke za rješavanje ovog zadatka,
- izraditi kostur PDF dokumenta uključujući osnovni izgled dokumenta, smještaj teksta, te izgled zaglavlja i podnožja dokumenta,
- definirati stilove te prijelome teksta i stranice,
- koristeći prikladnu programsku biblioteku omogućiti integraciju grafikona i slika koji se automatski generiraju iz podataka,
- koristeći prikladnu programsku biblioteku dodati tablice s podacima,
- dati kritički osvrt na korištenu metodologiju.

U radu je potrebno navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:

30. 11. 2022.

Datum predaje rada:

1. rok: 20. 2. 2023.
 2. rok (izvanredni): 10. 7. 2023.
 3. rok: 18. 9. 2023.

Predviđeni datumi obrane:

1. rok: 27. 2. – 3. 3. 2023.
 2. rok (izvanredni): 14. 7. 2023.
 3. rok: 25. 9. – 29. 9. 2023.

Zadatak zadao:

Doc. dr. sc. Tomislav Stipančić

Predsjednik Povjerenstva:

Prof. dr. sc. Branko Bauer

SADRŽAJ

| | |
|---|-----|
| SADRŽAJ | I |
| POPIS SLIKA | II |
| POPIS TABLICA | III |
| SAŽETAK | IV |
| SUMMARY | V |
| 1. UVOD..... | 1 |
| 1.1. Vrste programskih jezika..... | 1 |
| 1.2. Python..... | 5 |
| 1.3. Python biblioteke..... | 5 |
| 1.4. Poznate Python biblioteke | 6 |
| 1.4.1. Numpy biblioteka | 6 |
| 1.4.2. Scikit-learn biblioteka | 7 |
| 1.4.3. Tensorflow biblioteka..... | 8 |
| 1.4.4. Django biblioteka..... | 8 |
| 1.5. Python biblioteke korištene u završnom radu..... | 9 |
| 2. KREIRANJE PDF DOKUMENTA..... | 10 |
| 2.1. Zaglavlje i podnožje PDF-a | 10 |
| 2.2. Tekstualni dio PDF-a..... | 12 |
| 2.3. Prvi dio sintakse PDF dokumenta..... | 15 |
| 3. KREIRANJE TABLICA I GRAFA U PDF DOKUMENTU..... | 17 |
| 3.1. Generiranje tablice u PDF dokumentu..... | 17 |
| 3.1.1. Zaglavlje tablice..... | 19 |
| 3.1.2. Sadržaj tablice..... | 20 |
| 3.2. Generiranje grafa u PDF dokumentu..... | 23 |
| 3.2.1. Generiranje grafa u PDF dokumentu – kod..... | 24 |
| 4. ZAVRŠNI KOD | 26 |
| 5. ZAKLJUČAK..... | 28 |
| LITERATURA..... | 29 |
| PRILOG..... | 31 |

POPIS SLIKA

| | | |
|-----------|------------------------------------|----|
| Slika 1. | Programski jezik Python..... | 1 |
| Slika 2. | Programski jezik JavaScript..... | 2 |
| Slika 3. | Programski jezik C..... | 2 |
| Slika 4. | Programski jezik Java..... | 3 |
| Slika 5. | Programski jezik C++..... | 3 |
| Slika 6. | Programski jezik Ruby..... | 3 |
| Slika 7. | Programski jezik Swift..... | 4 |
| Slika 8. | Programski jezik Go | 4 |
| Slika 9. | Python biblioteke..... | 6 |
| Slika 10. | Primjer Numpy sintakse..... | 7 |
| Slika 11. | Primjer Scikit-learn sintakse..... | 8 |
| Slika 12. | Stvaranje PDF-a..... | 10 |
| Slika 13. | Veličina ćelije | 10 |
| Slika 14. | Zaglavlje i podnožje PDF-a..... | 11 |
| Slika 15. | Naslov PDF-a..... | 12 |
| Slika 16. | Datum i autor..... | 12 |
| Slika 17. | Okomiti razmak..... | 13 |
| Slika 18. | Lorem biblioteka..... | 13 |
| Slika 19. | Lorem Ipsum..... | 14 |
| Slika 20. | Prvi dio sintakse..... | 15 |
| Slika 21. | Prvi dio sintakse – PDF..... | 16 |
| Slika 22. | Pandas biblioteka..... | 17 |
| Slika 23. | Generiranje tablice..... | 18 |
| Slika 24. | Oblikovanje tablice..... | 19 |
| Slika 25. | Način oblikovanja tablice..... | 20 |
| Slika 26. | Matplotlib.pyplot biblioteka..... | 23 |
| Slika 27. | Seaborn biblioteka..... | 24 |
| Slika 28. | Generiranje grafa..... | 24 |
| Slika 29. | Završni kod..... | 26 |
| Slika 30. | PDF dokument..... | 27 |

POPIS TABLICA

Tablica 1. Početni podatci.....18

Tablica 2. Zaglavlje tablice.....20

Tablica 3. Tablica s podacima.....22

SAŽETAK

U današnjem modernom dobu, uloga programskih jezika postaje sve važnija u oblikovanju našeg tehnološkog okruženja. Programski jezici su precizni alati koje programeri koriste za komunikaciju s računalom te izvršavanje različitih zadataka.

U ovom završnom radu, dublje će se razumjeti njihovo značenje danas na način da su programski jezici temelj i podloga softvera, aplikacija i društvenih mreža koji su postali neizostavni dio naše svakodnevnice. Svaki se programski jezik ističe svojim specifičnim karakteristikama i svrhom. Primjerice, Python koji je poznat po svojoj jednostavnosti i čitljivosti, koristi se u web razvoju, analizi podataka i umjetnoj inteligenciji dok se Java koristi za pokretanje Android aplikacija, a C++ pruža detaljnu uslugu čineći ga glavnom izborom za razvoj operativnih sustava te igara. Vrhunac ovog istraživanja je jedan praktični primjer automatske izrade PDF dokumenta u Pythonu. Iskoristili smo njegove mogućnosti da stvorimo PDF dokument te pokazali kako se mogu automatizirati zadatci te pojednostavniti složeni procesi.

Zaključno, može se reći da su programski jezici mostovi koji povezuju logičnost računala sa ljudskom kreativnošću. Programski jezici nisu samo sredstvo za postizanje cilja već se u njima krije jezik inovacije i budućnosti te nam omogućavaju da svoje ideje pretvorimo u stvarnost u današnjem digitalnom dobu.

Ključne riječi: moderno doba, programski jezik, alat, programer, računalno, softver, Python, Java, C++, PDF dokument, automatizirati

SUMMARY

In today's modern era, the role of programming languages is becoming increasingly important in shaping our technological environment. These languages are precise tools that programmers use to communicate with computers and execute various tasks.

In this final work, we will delve deeper into their significance today and their role as the foundation of software, applications, and social networks that have become an integral part of our daily lives. Each programming language stands out with its specific characteristics and purposes. For example, Python, known for its simplicity and readability is used in web development, data analysis and artificial intelligence. Java is used to run Android applications, while C++ provides detailed control, making it the primary choice for developing operating systems and games. The culmination of this research is a practical example of automatic PDF document generation in Python. We harnessed its capabilities to create a PDF document and demonstrated how tasks can be automated, simplifying complex processes.

In conclusion, programming languages can be seen as bridges that connect the logic of computers with human creativity. Programming languages are not just tools to achieve a goal but a language of innovation and the future, enabling us to turn our ideas into reality in today's digital age.

Key words: modern era, programming languages, tool, programmer, computer, software, Python, Java, C++, PDF document, automate

1. UVOD

Programski jezici u dinamičnom svijetu suvremenog inženjeringa postaju sve neophodniji. Kako se s vremenom rad prenosi s čovjeka na strojeve, tako su i programski jezici zaživjeli kao podloga automatizaciji. U procesu automatizacije koriste se znanja iz strojarstva, elektronike te računarstva odnosno samim tim i programiranja.

Programski je jezik skup pravila i simbola kojim se prikazuje proces računanja poznat kao algoritam ili program.[1] Ti simboli, koji su posloženi po pravilima određenog programskog jezika, imaju jedinstvene karakteristike i primjenu prilagođavajući se zahtjevima programiranja u različitim područjima. Svrha svakog programskog jezika je da izvršava zadatke, od najjednostavnijih matematičkih operacija do suvremene kompleksnosti umjetne inteligencije.

1.1. Vrste programskih jezika

Razvoj programskih jezika prisutan je od samog početka računalstva. Danas postoji pregršt programskih jezika, svaki sa svojim prednostima, ali i ograničenjima. Određeni se jezici fokusiraju na brzinu izvedbe, dok se drugi baziraju na jednostavnosti sintakse.

Osnovna podjela programskih jezika je na niže, odnosno strojne jezike koji su manje prikladni ljudskom radu zbog binarnog zapisa te na više, odnosno jezike orijentirane ljudima.

Neki od najutjecajnijih programskih jezika današnjice su:

- **Python:** Python je programski jezik za opću namjenu poznat po svojoj jednostavnosti. Ističe se u raznim znanostima te ga njegova raznolikost i mnoštvo biblioteka čine uvijek popularnim izborom za programere.



Slika 1. Programski jezik Python [2]

- **JavaScript:** JavaScript univerzalni je programski jezik poznat po svojoj ulozi u web razvoju. Pruža korisnicima dinamičnost te interaktivni sadržaj u web preglednicima, što ga danas čini jako bitnim za izradu modernih web aplikacija. [3]



Slika 2. Programski jezik JavaScript [3]

- **SQL:** („Structured Query Language“) specifičan je programski jezik za upravljanje bazama podataka. Omogućuje standardizirani način upravljanja, stvaranja, ažuriranja te brisanja podataka u bazama podataka. Koristan je za zadatke stvaranja tablica, baza podataka te uspostavljanja odnosa između više tablica. Prisutan je u gotovo svim aplikacijama koje imaju podlogu na podacima i njihovoj analizi. [4]
- **C:** C je programski jezik razvijen još 1970-ih. Vrlo je poznat po svojoj jednostavnosti što ga čini naširoko korištenim za razvoj aplikacija te sistemsko programiranje. Zbog svoje pristupačnosti, C je postao podloga za razvoj mnogih drugih programskih jezika. [5]



Slika 3. Programski jezik C [5]

- **Java:** Java je univerzalni jezik visoke razine stvoren sredinom 1990-ih. Koristi se za izradu web aplikacija, softvera koji se koriste u poduzećima te Android mobilnih aplikacija. Objektno je orijentiran jezik što ga čini odličnim izborom za velike sustave. [6]



Slika 4. Programski jezik Java [6]

- **C++** : C++ jezik je visoke razine koji se kao što samo ime nalaže, temelji na već spomenutom programskom jeziku C-u. Poznat je po svojoj fleksibilnosti što ga čini popularnim za sistemsko programiranje, posebice za razvoj igara. C++ programerima omogućuje direktno upravljanje memorijom, čime se optimiziraju performanse u zahtjevnim aplikacijama kao što su operativni sustavi. [7]



Slika 5. Programski jezik C++ [7]

- **Ruby**: Objektno orijentiran programski jezik Ruby jezik je visoke razine poznat po pojednostavljenoj sintaksi. Njegova kombinacija čitljivosti i vještine web razvoja čini ga pravim alatom za brzo pretvaranje ideja u web aplikacije. [8]



Slika 6. Programski jezik Ruby [8]

- **Swift:** Swift je Appleov moderni programski jezik predstavljen 2014. godine. Osmišljen je za kreaciju iOS softvera kao i poznatih macOS, watchOS te tvOS uređaja i aplikacija. Njegova sintaksa jako je izražajna i učinkovita. Poznat je po stvaranju stabilnih aplikacija te automatskom upravljanju memorijom. [9]



Slika 7. Programski jezik Swift [9]

- **Go:** Jezik Go, poznat kao i Golang, programski jezik koji je razvio Google. Popularan je po svojoj jednostavnosti i učinkovitosti. Posjeduje ugrađenu podršku za paralelno programiranje te izradu aplikacija visokih performansi. Koristi se u računalstvu u „Cloud computingu“ poznatom kao i „Računarstvo u oblaku“ koji opisuje prostor za pohranu podataka putem interneta. [10]



Slika 8. Programski jezik Go [10]

Svaki od navedenih programskih jezika služi za rad u specifičnim domenama, omogućavajući programerima da izaberu pravi alat za razvoj vlastitih aplikacija i projekata.

U okviru ovog završnog rada fokusirati ću se upravo na Python programski jezik jer se najviše nametnuo zbog svoje jednostavnosti, čitljivosti te računalnih biblioteka te zbog najlakše automatske izrade PDF izvješća.

1.2. Python

Python se kao programski jezik isprofilirao još početkom 1990-ih. Osmislio ga je Nizozemski programer Guido van Rossum. Python je dobio ime po britanskoj komičarskoj grupi Monty Python čiji je obožavatelj bio i sam Guido van Rossum. Nakon rada od skoro dvije godine na programu, prva verzija Pythona objavljena je u veljači 1991. godine. Guidov cilj bio je napraviti programski jezik koji će biti jednostavan i pristupačan za pisanje i čitanje uz naglasak na jednostavnost samog koda. [11]

Kroz naredna dva desetljeća, Python postaje sve popularniji sa novom verzijom Python 2.0 koja omogućava korisnicima široku namjenu u razvoju softvera. Međutim, 2008. godine predstavljen je Python 3.0, što je dovodi do značajnih promjena u samom dizajnu jezika te od 2020. godine postaje preporučena verzija za kreiranje novih projekata.

Python glasi za jedan od najboljih programskih jezika što je i vidljivo po prisutnosti i usvojenosti u industriji. Posebnim ga čine njegove biblioteke (1.3) te sintaksa prilagođena korisniku. Ovakva jednostavnost koda smanjuje greške i čini kod pogodnijim za korištenje od strane ljudi. Koristi se u brojnim domenama, od web razvoja, strojnog učenja do umjetne inteligencije. [2]

Zaključno, podrška zajednice programera postavila je Python visoko na listi popularnosti i kvalitete. Njegove buduće svrhe još više pojačavaju njegov potencijal, primjenjujući ga u raznim znanostima poput umjetne inteligencije, Internetu stvari („Internet of Things“), kvantnom računalstvu, itd.

1.3. Python biblioteke

Python jezik ističe se svojim širokim spektrom biblioteka („libraries“) koje su zaslužne za pružanje podrške programerima. Biblioteke su neprocjenjivi alati koji omogućuju repetitivno iskorištavanje koda, odnosno već napisanih modula i definiranih funkcionalnosti te na taj način smanjuju potrebno vrijeme za razvoj softvera. Detaljno su i svrhovito osmišljene kako bi zadovoljile potrebe specifičnih domena. Korisne su za sve programere, od onih koji se bave najjednostavnijim matematičkim problemima sve do onih koji razvijaju napredne znanstvene

softvera poput strojnog učenja i dr. Značaj ovih biblioteka je da ubrzavaju proces programiranja i razvoj softvera pružanjem jednostavno dostupnih rješenja. [2]

Pythonove biblioteke igraju ključnu ulogu u programerskom svijetu razvoja softvera. Sadrže već optimizirane algoritme koji omogućuju programerima da potreban programski kod ne pišu od nule. Na ovaj način olakšava se programerima u smislu da se potencijalno više mogu usredotočiti na strateške i kreativne aspekte svojih radova, umjesto cijelog postavljanja i programiranja logike razmišljanja.

U samoj biti, Python biblioteke predstavljaju kolektivni napor i znanje svih programera raspodijeljeno u pojedinačne datoteke. Njihovim kombinacijama i upravljanjem, stručnjaci su u mogućnosti stvarati moderne aplikacije.



Slika 9. Python biblioteke [12]

1.4. Poznate Python biblioteke

1.4.1. NumPy biblioteka

- Domena: Numeričko računalstvo
- Opis: Biblioteka Numpy igra ključnu ulogu u pružanju usluge za numerička izračunavanja i proračune s nizovima i matricama. Ona pruža osnovu za manipulaciju podacima uz širok raspon alata za matematičke operacije, statistiku i dr. NumPy je jedna od ključnih biblioteka Python jezika, lagano se integrira unutar Pythona sa bibliotekama Pandas te Matplotlib i posjeduje sljedeće značajke:

- Višedimenzionalni niz objekata – ovakva struktura podataka omogućava pohranu i rukovanje nizovima raznih dimenzija. Nizovi mogu biti svih veličina te oblika, najčešće u obliku vektora ili matrica.
- Razne numeričke operacije – Numpy osigurava sigurno i učinkovito korištenje numeričkim proračunima. Univerzalno poznate funkcije omogućavaju operacije nad nizovima poboljšavajući samu čitljivost koda.
- Matematičke i statističke funkcije – ova biblioteka nudi razne funkcije od statističkih do funkcija napredne algebre. Također sadrži razne logaritamske i eksponencijalne funkcije za rad nad nizovima. [13]

```
import numpy as np

# Creating a NumPy array
my_array = np.array ([1, 2, 3, 4, 5])

# Performing an operation on the array
squared_array = np.square (my_array)

print("Original Array: ", my_array)
print("Squared Array: ", squared_array)
```

Slika 10. Primjer Numpy sintakse

1.4.2. Scikit-learn biblioteka

- Domena: Strojno učenje
- Opis: Scikit-learn biblioteka pruža alate za zadatke strojnog učenja. Primarno je izgrađena na drugim već spomenutim bibliotekama Numpy te Matplotlib. Odlična je za programere početnike, jednostavna je za korištenje i lagano se integrira s drugim Python bibliotekama. [14]

```
import numpy as np
from sklearn.linear_model import LinearRegression

# Generate example data
X = np.random.rand(100, 1) * 10
Y = 2 * X + 1 + np.random.randn(100, 1) * 2

# Train a linear regression model
model = LinearRegression().fit(X, Y)

# Make a prediction
new_X = np.array([[5]])
predicted_Y = model.predict(new_X)
print(f"Predicted Y for new_X: {predicted_Y[0][0]}")
```

Slika 11. Primjer Scikit-learn sintakse

1.4.3. TensorFlow biblioteka

- Domena: Neuronske mreže i umjetna inteligencija
- Opis: TensorFlow biblioteka razvijena je od strane Google Brain tima za njihovu privatnu upotrebu. Koristi se u velikom broju programskih jezika poput C++, JavaScript, Java te Python. Primjenjiva je u različitim sektorima razvoja softvera. Pruža dobru platformu za strojno učenje i pomaže pri učitavanju podataka za obuku modela. [15]

1.4.4. Django biblioteka

- Domena: Web razvoj
- Opis: Django biblioteka pruža uslugu pojednostavljenja procesa izgradnje složenih web aplikacija. Omogućava alate za rukovanje bazama podataka nalik SQL-u. [16]

1.5. Python biblioteke korištene u završnom radu

Za izradu sintakse automatske izrade PDF izvješća, potrebne su sljedeće biblioteke:

- Pandas
- Matplotlib.pyplot
- Seaborn
- Lorem

Svaka od njih je pojedinačno objašnjena tokom pisanja i upotrebe iste u programskom kodu.

2. KREIRANJE PDF DOKUMENTA

Kreiranje PDF dokumenta započinje stvaranjem praznog lista PDF dokumenta.

```
pdf = FPDF()  
pdf.add_page()  
pdf.output(f'./example.pdf', 'F')
```

Slika 12. Stvaranje PDF-a

- Ovim kodom stvorena je komponenta nazvana 'pdf' koja će se koristiti za korištenje funkcionalnosti 'FPDF' klase.
- Zatim je dodana prazna stranica koja će služiti kao podloga PDF-a na koju će se spremati tekst, grafovi te tablice.
- Na kraju se 'output()' metodom omogućava spremanje dokumenta, u ovom slučaju kao „example.pdf“.

U nastavku pisanja koda, prije 'output()' metode će se pisati ostatak sintakse kako bi se uspješno sačuvao dokument prije prikazivanja.

Dodana stranica ima format A4 u portretu te margine od 1 cm.

2.1. Zaglavlje i podnožje PDF-a

Sljedeće linije koda služe za stvaranje zaglavlja i podnožja svake stranice PDF dokumenta.

Prvo je navedena veličina ćelije „ch“ koja će biti konstantna kroz sljedeći kod:

```
# cell height  
ch = 8
```

Slika 13. Veličina ćelije

Nakon navedene varijable 'ch', zapisuje se kod pomoću funkcije 'class'. Ova funkcija omogućava definirati podatke, odnosno objekte koji će se nalaziti unutar zaglavlja i podnožja ove stranice PDF-a:

- Redak 'class PDF(FPDF):' definira novu klasu pod nazivom PDF iz klase FPDF. U nastavku koda pozivaju se funkcije koje kreiraju objekte ove klase.

```
class PDF(FPDF):
    def __init__(self):
        super().__init__()
    def header(self):
        self.set_font('Arial', '', 12)
        self.cell(0, 8, 'Zaglavljje', 0, 1, 'C')
    def footer(self):
        self.set_y(-15)
        self.set_font('Arial', '', 12)
        self.cell(0, 8, f'Stranica {self.page_no()}', 0, 0, 'C')
```

Slika 14. Zaglavljje i podnožje PDF-a

- Kodom 'def __init__(self):' definira se inicijalizacija svake naredbe unutar klase.
- Metodom 'def header (self)' definira se zaglavljje PDF-a.
 - Pomoću komande 'self.set_font ('Arial', ' ', 12)' postavlja se font „Arial“, normalnog dizajna te veličine slova 12.
 - Metodom 'self.cell (0, 8, 'Zaglavljje', 0, 1, 'C')' dodaje se tekstualna ćelija na PDF stranicu. Tekst „Zaglavljje“ se dodaje na sredinu vrha stranice s prijelomom retka iza teksta.
- Metodom 'def footer (self):' definira se podnožje PDF-a.
 - Komandom (self.set_y(-15)) postavlja se kursor na dno stranice.
 - Zatim se kao i u zaglavljju dodaje font „Arial“, normalnog dizajna te veličina slova 12.
 - Naredbom 'cell (0, 8, f'Stranica {self.page_no ()}, 0, 0, 'C')' dodaje se tekstualna ćelija u kojoj će se daljnjim kodom centrirati tekst i dodati broj stranice PDF-a.

2.2. Tekstualni dio PDF-a

Nakon napisanog koda za zaglavlje i podnožje PDF-a što čini svaki PDF dokument standardnim, slijedi kod za tekstualni dio PDF-a.

- Kada se napravila prazna stranica PDF-a, linijom koda 'pdf.set_font('Arial', 'B', 24)' uspostavlja se font za sljedeći tekst koji se dodaje u PDF dokument. Koristi se font „Arial“, podebljanog stila ('B') i veličinom fonta 24.

```
pdf = PDF()
pdf.add_page()

pdf.set_font('Arial', 'B', 24)
pdf.cell(w=0, h=20, txt="Automatska izrada PDF izvjesca", ln=1)
```

Slika 15. Naslov PDF-a

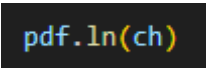
- Retkom 'pdf.cell (w=0, h=20, txt="Automatska izrada PDF izvjesca", ln=1)' uvrštavamo ćeliju s tekстом „Automatska izrada PDF izvjesca“.
 - Parametrom „w“ se određuje širina ćelije (0 označava automatsku širinu ćelije).
 - Parametrom „h“ određuje se visina ćelije (20).
 - Kratica „txt“ je oznaka za sadržaj teksta („Automatska izrada PDF izvjesca“).
 - Komanda 'ln=1' označava da se prijelom retka dodaje odmah iza ćelije.
- U sljedećoj liniji koda koristi se komanda 'pdf.set_font ('Arial', '', 16)' čime se postavlja font idućeg teksta „Arial“, regularnog oblika te veličinom fonta 16.

```
pdf.set_font('Arial', '', 16)
pdf.cell(w=30, h=ch, txt="Datum: ", ln=0)
pdf.cell(w=30, h=ch, txt="25/08/2023", ln=1)
pdf.cell(w=30, h=ch, txt="Autor: ", ln=0)
pdf.cell(w=30, h=ch, txt="Josip Strkalj", ln=1)
```

Slika 16. Datum i autor

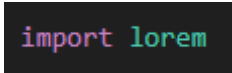
- Nadalje se kodom 'pdf.cell (w=30, h=ch, txt="Datum: ", ln=0)' dodaje nova tekstualna ćelija.
 - Parametrom „w“ označava se širina ćelija od 30 jedinica.
 - Parametrom se „ln=0“ ili „ln=1“ označava da (ne)postoji prijelom retka nakon ćelije.
 - Parametrom „h=ch“ označavamo visinu ćelije od 8 jedinica [3].
 - U svakom od 4 iduća retka, označava se različiti tekst. Prvo se navodi tekst „Datum“, zatim datum pisanja završnog rada, tekst „Autor“ te ime autora završnog rada.

- Zatim se kodom 'pdf.ln (ch)' dodaje prijelom retka koji služi za stvaranje okomitog razmaka između 2 odjeljka.

A screenshot of a code editor showing the code `pdf.ln(ch)` in a dark-themed font.

Slika 17. Okomiti razmak

Nakon postavljanja uvodnog dijela teksta, piše se kod za generiranje teksta koji će biti slučajan i nov svaki put kada se sintaksa spremi te pokrene novi PDF dokument.

A screenshot of a code editor showing the code `import lorem` in a dark-themed font.

Slika 18. Lorem biblioteka

Biblioteka Lorem će omogućiti generiranje tzv. „Lažnog“ teksta poznatog kao „Lorem Ipsum“ tekst. Ova biblioteka često se koristi u prilikama kada se želi ispuniti neki prostor određenim tekstom koji nalikuje na pravi, ali nema određeno značenje. Neke od značajki ove biblioteke su:

- Jednostavno i brzo kreiranje teksta koje ubrzava proces pisanja teksta te se tekst sam generira.
- Funkcije Lorem biblioteke prilagođavaju duljinu teksta specifičnim zahtjevima unutar određene sintakse.
- Sadrži razne oblike tekstova, od tehničkih do latinskih kakav će biti prikazan unutar programske sintakse.

- Primjenjuje se u izradbi raznih maketa sa lažnim tekstovima, u razvoju softvera pri demonstraciji slučajnog teksta unutar aplikacija te u obrazovnim svrhama pri podučavanju manipulacija teksta. [17]

Potom se zapisuje kod kojim se dodaje „Lorem Ipsum“ tekst unutar PDF dokumenta:

```
pdf.ln(ch)

pdf.multi_cell(w=0, h=5, txt=lorem.paragraph())

pdf.ln(ch)

pdf.multi_cell(w=0, h=5, txt=lorem.paragraph())

pdf.ln(ch)
```

Slika 19. Lorem Ipsum

- Komandom 'pdf.ln (ch)' stvara se prijelom retka koji služi kao vertikalni razmak između 2 odjeljka.
- Linijom koda 'pdf.multi_cell (w=0, h=5, txt=lorem.paragraph())' dodaje se ćelija s više redaka:
 - Varijablom 'w=0' postavlja se navedena širina „w=0“.
 - Varijablom 'h=5' postavlja se visina ćelije koja iznosi „h=5“.
 - Komandom 'txt=lorem.paragraph ()' generira se slučajni tekst „Lorem Ipsum“ kako bi se popunila ćelija.
- Komandom 'pdf.ln (ch)' stvara se prijelom retka koji služi kao vertikalni razmak između 2 odjeljka.
- Zatim se ponavlja jednaka linija koda pri stvaranju „Lorem Ipsum“ teksta.
- Komandom 'pdf.ln (ch)' stvara se prijelom retka koji služi kao vertikalni razmak između 2 odjeljka.

Ukratko, linije koda pod naslovom 2.2. stvaraju PDF dokument koristeći se PDF klasom iz 'fpdf' klase, postavljaju fontove, dodavaju ćelije sa određenim sadržajima i dizajnom te generiraju više dijelova teksta sa razmacima i prijelomima redaka. Dokumentu su se također dodali naslov, datum i podatci o autoru.

2.3. Prvi dio sintakse PDF dokumenta

```
import lorem

from fpdf import FPDF

class PDF(FPDF):
    def __init__(self):
        super().__init__()
    def header(self):
        self.set_font('Arial', '', 12)
        self.cell(0, 8, 'Zaglavljje', 0, 1, 'C')
    def footer(self):
        self.set_y(-15)
        self.set_font('Arial', '', 12)
        self.cell(0, 8, f'Stranica {self.page_no()}', 0, 0, 'C')

# cell height
ch = 8

# Generate PDF

pdf = PDF()
pdf.add_page()
pdf.set_font('Arial', 'B', 24)
pdf.cell(w=0, h=20, txt="Automatska izrada PDF izvjesca", ln=1)

pdf.set_font('Arial', '', 16)
pdf.cell(w=30, h=ch, txt="Datum: ", ln=0)
pdf.cell(w=30, h=ch, txt="25/08/2023", ln=1)
pdf.cell(w=30, h=ch, txt="Autor: ", ln=0)
pdf.cell(w=30, h=ch, txt="Josip Strkalj", ln=1)

pdf.ln(ch)

pdf.multi_cell(w=0, h=5, txt=lorem.paragraph())

pdf.ln(ch)

pdf.multi_cell(w=0, h=5, txt=lorem.paragraph())

pdf.ln(ch)
```

Slika 20. Prvi dio sintakse

Dosadašnja sintaksa prikazuje sljedeći isprogramirani PDF dokument:

Zaglavlje

Automatska izrada PDF izvjesca

Datum: 25/08/2023

Autor: Josip Strkalj

Numquam dolore neque non porro neque sit. Est non tempora adipisci porro consectetur. Ut ut sed sit magnam quisquam quiquia. Quaerat velit adipisci quisquam eius porro ipsum sit. Quiquia labore adipisci consectetur modi. Adipisci ut ut dolorem non dolorem quiquia quaerat. Eius amet magnam neque voluptatem numquam. Aliquam eius voluptatem ipsum magnam amet dolore.

Numquam labore labore sit dolorem numquam velit. Dolorem numquam dolor magnam. Sed magnam tempora eius eius. Dolore quisquam sit dolorem. Sed numquam numquam adipisci velit. Ipsum tempora non modi aliquam velit ipsum. Etincidunt dolor porro voluptatem aliquam labore velit amet. Ipsum magnam quisquam amet dolore.

Stranica 1

Slika 21. Prvi dio sintakse - PDF

3. KREIRANJE TABLICE I GRAFA U PDF DOKUMENTU

3.1. Generiranje tablice u PDF dokumentu

U drugom dijelu izrade sintakse automatskog PDF izvješća, generirati će se kod za izradu tablice i grafa u PDF dokumentu.

Prvotno se uvodi nova biblioteka „pandas“.

```
import pandas as pd
```

Slika 22. Pandas biblioteka

Pandas je biblioteka zaslužna za manipulaciju te analizu podataka unutar Python jezika. Sastoji se od funkcija za jednostavno upravljanje analizom strukturiranih podataka što ju čini vrlo popularnom u programerskom svijetu. Neke od glavnih značajki ove biblioteke su:

- Sastoji se od dviju struktura podataka:
 - „DataFrame“ koja zapisuje podatke nalik tablici, odnosno omogućava pohranu podataka u stupce i redove. Ovakav oblik podatkovnog zapisa poslužiti će u ovom završnom radu.
 - „Series“, tj. niz je jednodimenzionalni niz koji sadrži različite vrste podataka. Dakle „DataFrame“ je skup različitih nizova.
- Jednostavna manipulacija podacima koja omogućava filtraciju i uspoređivanje podataka. Također pomaže pri spajanju različitih vrsti podataka te njihovom raspoređivanju unutar analize.
- Omogućava uvrštavanje podataka iz drugih podatkovnih baza poput CSV-a, Excel-a te SQL baze podataka.
- Integracija sa NumPy bazom podataka. NumPy biblioteka stoji za Numerical Python te se koristi za velike nizove te matrice podataka. Njihova integracija omogućava uspješan rad nad nizovima matematičkih funkcija.
- Pandas se koristi u raznim domenama poput poslovnih i financijskih analiza, zdravstvu, energetici te obrazovanju. [18]

Stoga se shodno tome koristi „Pandas“ kod generiranja tablice u ovom završnom radu.

Nakon uvrštavanja ove biblioteke, zapisujemo 'df = pd.DataFrame ({'Apscisa': ['Kategorija 1', 'Kategorija 2', 'Kategorija 3', 'Kategorija 4'], 'Ordinata': [400, 600, 300, 500] })'. [6]

```
# Load or prepare pandas DataFrame
df=pd.DataFrame({'Apscisa':['Kategorija 1','Kategorija 2','Kategorija 3','Kategorija 4'],'Ordinata':[400,600,300,500]})
```

Slika 23. Generiranje tablice

- Komentarom „# Load or prepare pandas DataFrame“ postavlja se uputa o korištenju funkcije DataFrame kao dio biblioteke Pandas.
- U navedenom retku programskog koda, pomoću datoteke Pandas kreiran je DataFrame pod nazivom 'df' koji sadrži dva stupca: 'Apscisa' i 'Ordinata'. Izraz DataFrame označava podatkovnu strukturu koja organizira podatke u dvodimenzionalnu tablicu stupaca i redaka.
- Zapisom ' 'Apscisa' : ['Kategorija 1', 'Kategorija 2', 'Kategorija 3', 'Kategorija 4] ' definira se stupac 'Apscisa' te pridodaje mu niz od četiri tekstualne vrijednosti, „Kategorija 1“, „Kategorija 2“, „Kategorija 3“ te „Kategorija 4“.
- Zapisom ' 'Ordinata' : [400, 600, 300, 500] ' definira se stupac 'Ordinata' te pridodaje niz od četiri brojčane vrijednosti, 400, 600, 300 te 500.

Ovim se postupkom generira DataFrame sa dva stupca; 'Apscisa' te 'Ordinata' sa pripadajućim vrijednostima.

Tablica 1. Početni podatci

| Apscisa | Ordinata |
|--------------|----------|
| Kategorija 1 | 400 |
| Kategorija 2 | 600 |
| Kategorija 3 | 300 |
| Kategorija 4 | 500 |

3.1.1. Zaglavlje tablice

U sljedećim linijama koda, oblikovati će se navedena tablica:

```
# Table Header
pdf.set_font('Arial', 'B', 16)
pdf.cell(40, ch, 'Apscisa', 1, 0, 'C')
pdf.cell(40, ch, 'Ordinata', 1, 1, 'C')
```

Slika 24. Oblikovanje tablice

- Komentaram „# Table Header“ zapisuje se komentar kao uputa u snalaženju tokom pisanja programskog koda.
- Komandom 'pdf.set_font ('Arial', 'B', 16)' postavlja se font koji će biti dodan u ovaj PDF dokument. Koristiti će se font „Arial“, podebljanog dizajna 'B' te veličine slova 16.
- Kodom 'pdf.cell (40, ch, 'Apscisa', 1, 0, 'C')' dodajemo novu ćeliju unutar PDF dokumenta sa sljedećim parametrima:
 - Varijabla '40' označava širinu ćelije od 40 jedinica.
 - Varijabla 'ch' označava visinu ćelije od 8 jedinica. [3]
 - Tekst „Apscisa“ se zapisuje kao sadržaj ćelije.
 - Varijabla '1' označava stil obruba, odnosno da je ćelija obrubljena.
 - Varijabla '0' označava da ne postoji prijelom retka nakon ćelije, odnosno da je nastavak daljnjeg teksta u istom retku.
 - Oznakom 'C' centrirana se navedeni tekst u samo središte ćelije.
- Kodom 'pdf.cell (40, ch, 'Ordinata', 1, 1, 'C')' dodajemo novu ćeliju unutar PDF dokumenta sa sljedećim parametrima:
 - Varijabla '40' označava širinu ćelije od 40 jedinica.
 - Varijabla 'ch' označava visinu ćelije od 8 jedinica.

- Tekst „Ordinata“ se zapisuje kao sadržaj ćelije.
- Varijabla 'l' označava stil obruba, odnosno da je ćelija obrubljena.
- Varijabla 'l' za razliku od prošlog retka koda, označava da postoji prijelom retka nakon ćelije, odnosno da je nastavak daljnjeg teksta u sljedećem retku.
- Oznakom 'C' centrira se navedeni tekst u samo središte ćelije.

Tablica 2. Zaglavlje tablice

| Apscisa | Ordinata |
|---------|----------|
|---------|----------|

3.1.2. Sadržaj tablice

U daljnjem tijeku koda prikazuje se način kojim se popunjava tablica:

```
# Table contents
pdf.set_font('Arial', '', 16)
for i in range(0, len(df)):
    pdf.cell(40, ch, df['Apscisa'].iloc[i], 1, 0, 'C')
    pdf.cell(40, ch, df['Ordinata'].iloc[i].astype(str), 1, 1, 'C')
```

Slika 25. Način oblikovanja tablice

- Komentarom „#Table contents“ zapisuje se komentar kao uputa u snalaženju tokom pisanja programskoga koda.
- Komandom 'pdf.set_font('Arial', '', 16)' postavlja se font koji će biti dodan u ovu tablicu PDF dokumenta. Koristiti će se font „Arial“, normalnog dizajna veličine slova 16.
- Komandom 'for i in range(0, len(df)):' stvaramo petlju koja radi iteraciju brojeva od 0 do duljine varijable 'df'.

- Komandom `'pdf.cell (40, ch, df ['Apscisa'].iloc [i], 1, 0, 'C')` dodaju se nove ćelije u PDF dokument s određenim argumentima:
 - Varijabla `'40'` označava širinu ćelija od 40 jedinica.
 - Varijabla `'ch'` označava visinu ćelija od 8 jedinica. [3]
 - Zapisom `'df ['Apscisa'].iloc [i]` se generiraju tekstualne vrijednosti u stupcu `'Apscisa'` pozivajući se na varijablu `'df'` iz `DataFramea`.
 - Varijabla `'1'` označava stil obruba, odnosno da su ćelije obrubljene.
 - Varijabla `'0'` označava da ne postoji prijelom retka nakon ćelije, odnosno da je nastavak teksta u istom retku.
 - Oznakom `'C'` poravnava se navedeni tekst u samo središte ćelije.

U narednom kodu koji je sličan prethodnom, radi se sa stupcem `'Ordinata'`:

- Komandom `'pdf.cell (40, ch, df ['Ordinata'].iloc [i].astype(str), 1, 1, 'C')` dodaju se nove ćelije u PDF dokument s određenim argumentima:
 - Varijabla `'40'` označava širinu ćelija od 40 jedinica.
 - Varijabla `'ch'` označava visinu ćelija od 8 jedinica. [3]
 - Zapisom `'df ['Ordinata'].iloc [i].astype (str)'` pristupa se vrijednostima u stupcu `'Ordinata'` unutar `DataFramea` varijable `'df'` po indeksu `'i'` te se pomoću zapisa `'astype (str)'` osigurava da se navedene vrijednosti tretiraju kao niz podataka.
 - Varijabla `'1'` označava stil obrubljenih ćelija.
 - Varijabla `'1'` označava da postoji prijelom retka nakon ćelije te da se sljedeća ćelija generira u novom retku
 - Oznakom `'C'` centrirana se navedeni tekst u središte ćelije.

Navedenim se kodom omogućilo popunjavanje tablice unutar nekog PDF dokumenta. Postavljen je font, njegova veličina te pozicija. Pokazan je način na koji se uvrštavaju tekstualne te brojčane vrijednosti kao podatci koji su vizualno pregledniji korisniku:

Tablica 3. Tablica s podacima

| Apscisa | Ordinata |
|----------------|-----------------|
| Kategorija 1 | 400 |
| Kategorija 2 | 600 |
| Kategorija 3 | 300 |
| Kategorija 4 | 500 |

3.2. Generiranje grafa u PDF dokumentu

Kako bi se generirao graf unutar nekog PDF dokumenta, potrebno je uvesti dvije nove biblioteke:

- „Matplotlib.pyplot“ biblioteka jedna je od glavnih komponenti „Matplotlib“ biblioteke kao bitnog alata za generiranje različitih vizualizacija podataka u Pythonu, najčešće u obliku grafova i dijagrama. Jedne od najbitnijih značajki ove biblioteke su:
 - Brzo i jednostavno korištenje kojim se stvara sučelje za stvaranje vizualizacija različitih oblika.
 - Automatsko upravljanje osim značajka koje pruži ova biblioteka, korisnik ne bi morao eksplicitno kreirati ove objekte već biblioteka Matplotlib.pyplot generira umjesto njega.
 - Jednostavnim pozivom funkcija mogu se stvarati različite vrste dijagrama.
 - Ova se biblioteka laganom integrira sa vrijednostima nizova biblioteka NumPy te Pandas što će biti primjenjivo i u ovom završnom radu.
 - Mogućnost dodavanja različitih oznaka i bilješki zbog približavanja konteksta korisniku.
 - Programirane vizualizacije laganom je spremite kao slikovne datoteke (npr. JPEG ili PNG). [19]

Biblioteka „Matplotlib.pyplot“ zapisuje se kao varijabla „plt“:

```
import matplotlib.pyplot as plt
```

Slika 26. Matplotlib.pyplot biblioteka

- „Seaborn“ biblioteka zaslužna je za vizualizaciju podataka unutar Python programskog jezika. Jedne od značajki ove biblioteke su:
 - Mogućnost korištenja kompliciranijih funkcija od Matplotliba.
 - Fokus na dizajn te ugodnu vizualizaciju boja.
 - Jednostavnim kodom se stiže do kompliciranih vizualizacija.

- Nudi laganu integraciju sa Pandas bibliotekom što olakšava proces stvaranja vizualizacije unutar Python programa. [20]

Biblioteka Seaborn zapisuje se kao varijabla „sns“:

```
import seaborn as sns
```

Slika 27. Seaborn biblioteka

3.2.1. Generiranje grafa u PDF dokumentu – kod

Nakon uvrštavanja ovih dvaju biblioteka koje će pomoći pri stvaranju i oblikovanju grafa u Python sintaksu, slijedi kod kojim će se to omogućiti:

```
# Save figures to use in the PDF
fig, ax = plt.subplots(1,1, figsize = (6, 4))
sns.barplot(data = df, x = 'Apscisa', y = 'Ordinata')
plt.title("Graf")
plt.savefig('./example_Graf.png', transparent=False, facecolor='white', bbox_inches="tight")
plt.close()
```

Slika 28. Generiranje grafa

- Komentarom '# Save figures to use in the PDF' objašnjava se svrha i zadatak sljedećeg bloka sintakse. Govori da je idući kod namijenjen za generiranje oblika dijagrama koji će spremi u obliku slikovne datoteke.
- Retkom 'fig, ax = plt.subplots (1, 1, figsize = (6, 4))' se stvara skup podcrteža pomoću funkcije 'plt.subplots ()', upravo pomoću Matplotlibovog modula „pyplot“.
 - Varijabloma (1, 1) određuje se zasebno polje slike sa jednim redom te jednim stupcem.
 - Varijablom 'figsize = (6, 4)' uspostavljaju se dimenzije slikovne datoteke u inčima. Širina slikovne datoteke će biti 6 inča dok će visina iznositi 4 inča.
 - Finalna varijabla 'fig' referenca je na kreiranu slikovnu datoteku. Ona služi kao spremnik za sve elemente koji se žele prikazati ovim kodom.
 - Varijabla 'ax' odnosi se podcrteže kreirane u ovom crtežu.

- Redak `'sns.barplot (dana = df, x = 'Apscisa', y = 'Ordinata')` koristi se funkcijom `'sns.barplot ()'` iz biblioteke Seaborn za generiranje dijagrama.
 - Varijablom `'data = df'` specificira se DataFrame odnosno, `'df'` koji sadržava podatke spremne za iscrtavanje.
 - Varijablom `'x = 'Apscisa' '` postavlja se redak u DataFrameu koji se koristi za prikazivanje podataka duž x-osi.
 - Varijablom `'y = 'Ordinata' '` postavlja se stupac u DataFrameu koji se koristi za prikazivanje podataka duž y-osi.
- Retkom `'plt.title ('Graf')` se grafu dodaje naziv „Graf“.
- Linijom koda `'plt.savefig (' ./example_Graf.png', transparent = False, facecolor = 'white', bbox_inches = „tight“)` sprema se trenutno isprogramirana slika kao slikovna datoteka.
 - Pomoću `' ./example_Graf.png'` navodi se putanja datoteke te naziv slike kako bi se mogla spremiti.
 - Varijabla `'transparent = False'` govori da pozadina slike ne smije biti prozirna.
 - Varijabla `'facecolor = 'white' '` postavlja bijelu pozadinu na sliku.
 - Varijabla `'bbox_inches = 'tight' '` omogućuje da spremljena slikovna datoteka uključuje sve elemente slike.
- Redak `'plt.close ()'` naređuje da se trenutna slika zatvori.

Navedena sintaksa generira graf koristeći se Seaborn bibliotekom. Ona svojim funkcijama poput `'barplot'` potpomaže da se grafu zadaju razne boje za svaki postojeći stupac. Također se zahvaljujući Seabornu dodaje naslov grafu, postavlja bijela pozadina te zatvara slikovna datoteka i sprema.

4. ZAVRŠNI KOD

Sveukupna sintaksa za automatsku izradu PDF izvješća izgleda sljedeće:

```
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import seaborn as sns
import lorem
from fpdf import FPDF

# cell height
ch = 8

class PDF(FPDF):
    def __init__(self):
        super().__init__()
    def header(self):
        self.set_font('Arial', '', 12)
        self.cell(0, 8, 'Zaglavlje', 0, 1, 'C')
    def footer(self):
        self.set_y(-15)
        self.set_font('Arial', '', 12)
        self.cell(0, 8, f'Stranica {self.page_no()}', 0, 0, 'C')

# Load or prepare pandas DataFrame
df=pd.DataFrame({'Apscisa':['Kategorija 1','Kategorija 2','Kategorija 3', 'Kategorija 4'], 'Ordinata':[400, 600, 300, 500]})

# Save figures to use in the PDF
fig, ax = plt.subplots(1,1, figsize = (6, 4))
sns.barplot(data = df, x = 'Apscisa', y = 'Ordinata')
plt.title("Graf")
plt.savefig('./example_Graf.png', transparent=False, facecolor='white', bbox_inches="tight")
plt.close()

# Generate PDF
pdf = PDF()
pdf.add_page()
pdf.set_font('Arial', 'B', 24)
pdf.cell(w=0, h=20, txt="Automatska izrada PDF izvjesca", ln=1)
pdf.set_font('Arial', '', 16)
pdf.cell(w=30, h=ch, txt="Datum: ", ln=0)
pdf.cell(w=30, h=ch, txt="25/08/2023", ln=1)
pdf.cell(w=30, h=ch, txt="Autor: ", ln=0)
pdf.cell(w=30, h=ch, txt="Josip Strkalj", ln=1)
pdf.ln(ch)
pdf.multi_cell(w=0, h=5, txt=lorem.paragraph())
pdf.image('./example_chart.png', x = 10, y = None, w = 100, h = 0, type = 'PNG', link = '')
pdf.ln(ch)
pdf.multi_cell(w=0, h=5, txt=lorem.paragraph())
pdf.ln(ch)

# Table Header
pdf.set_font('Arial', 'B', 16)
pdf.cell(40, ch, 'Apscisa', 1, 0, 'C')
pdf.cell(40, ch, 'Ordinata', 1, 1, 'C')

# Table contents
pdf.set_font('Arial', '', 16)
for i in range(0, len(df)):
    pdf.cell(40, ch, df['Apscisa'].iloc[i], 1, 0, 'C')
    pdf.cell(40, ch, df['Ordinata'].iloc[i].astype(str), 1, 1, 'C')

pdf.output(f'./example.pdf', 'F')
```

Slika 29. Završni kod

Finalna sintaksa prikazuje sljedeći isprogramirani PDF dokument:

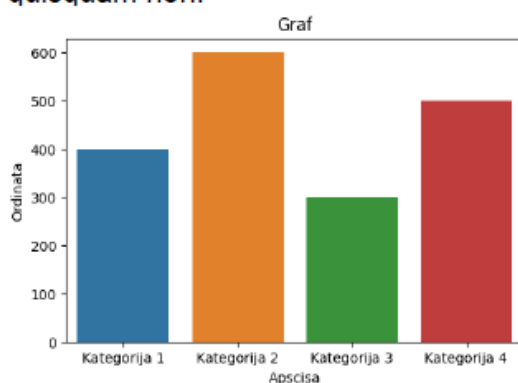
Zaglavlje

Automatska izrada PDF izvjesca

Datum: 25/08/2023

Autor: Josip Strkalj

Neque sit velit ipsum modi dolore neque ipsum. Numquam quiquia ipsum dolorem numquam. Consectetur dolore eius dolore numquam sit amet adipisci. Quiquia neque quisquam modi. Ipsum adipisci sed tempora ut tempora porro. Voluptatem adipisci ipsum ipsum quisquam modi porro ipsum. Numquam etincidunt eius voluptatem amet sit ipsum. Quaerat porro numquam quiquia ipsum adipisci. Numquam etincidunt aliquam dolor quisquam non.



Dolor consectetur tempora quisquam quiquia. Modi tempora porro consectetur sit. Aliquam quisquam tempora amet voluptatem ipsum. Neque dolorem quiquia velit numquam dolore non. Neque quiquia dolor numquam eius sit. Dolor magnam magnam magnam non aliquam consectetur etincidunt. Labore est consectetur est quaerat aliquam labore sed. Numquam dolore eius est sit. Voluptatem sit dolor sit neque labore. Magnam eius quaerat magnam sit neque aliquam velit.

| Apscisa | Ordinata |
|--------------|----------|
| Kategorija 1 | 400 |
| Kategorija 2 | 600 |
| Kategorija 3 | 300 |
| Kategorija 4 | 500 |

Stranica 1

Slika 30. PDF dokument

5. ZAKLJUČAK

Tijekom izrade ovog PDF izvješća korišten je programski jezik Python.

Python je nedvojbeno jedan od najsvestranijih i najpopularnijih programskih jezika na svijetu. Njegova široka prihvaćenost pripisuje se njegovoj čitljivosti i obilju biblioteka koje nudi. No, kao i svi drugi alati, Python ima svoje prednosti i mane. Njegova elegantna sintaksa čini ga odličnim izborom za početnike programere. Može se koristiti u web razvoju, analizi podataka, umjetnoj inteligenciji i dr. Njegova velika aktivna zajednica programera uvijek je prisutna za pronalazak rješenja problema putem internetskih foruma. Python je dostupan na više operacijskih sustava, od Windowsa, macOS-a do Linuxa što znači nesmetan rad na različitim platformama bez većih izmjena.

U usporedbi s drugim jezicima poput C++, Python je sporiji u izvršavanju zadataka. To je možebitno veliki nedostatak za razvoj aplikacija koje zahtijevaju velike napore ili obradu podataka u stvarnom vremenu. Također, Python nije najbolji izbor kao jezik za razvoj mobilnih igara i aplikacija što je danas iznimno popularno. Pruža nižu razinu performansi i kontrole od jezika poput Java, Swifta ili C++.

Zaključno, Pythonova metodologija ima određene vrline i mane koji ga ujedno čine posebnim i vrijednim alatom za širok raspon aplikacija, pogotovo onih koji daju prioritet brzom razvoju i jednostavnosti korištenja. Međutim, pri odabiru Pythona među drugim programskim jezicima, treba razmotriti njegova ograničenja izvedbe, potrošnju memorije i nekompatibilnost u određenim domenama rada. U konačnici, izbor programskog jezika mora se uskladiti sa zahtjevima i ciljevima projekta. U tom smislu, Python je i više nego zadovoljio potrebne kriterije za pisanje ovog završnog rada.

LITERATURA

- [1] Programski jezik, https://hr.wikipedia.org/wiki/Programski_jezik, pristupljeno: 31.08.2023.
- [2] Python (programski jezik), https://en.wikipedia.org/wiki/Python_%28programming_language%29, pristupljeno: 31.08.2023.
- [3] JavaScript, <https://simple.wikipedia.org/wiki/JavaScript>, pristupljeno: 31.08.2023.
- [4] SQL, <https://en.wikipedia.org/wiki/SQL>, pristupljeno: 31.08.2023.
- [5] C (programski jezik), [https://en.wikipedia.org/wiki/C_\(programming_language\)](https://en.wikipedia.org/wiki/C_(programming_language)), pristupljeno 31.08.2023.
- [6] Java (programski jezik), [https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language)), pristupljeno 31.08.2023.
- [7] C++, <https://en.wikipedia.org/wiki/C%2B%2B>, pristupljeno: 31.08.2023.
- [8] Ruby (programski jezik), [https://en.wikipedia.org/wiki/Ruby_\(programming_language\)](https://en.wikipedia.org/wiki/Ruby_(programming_language)), pristupljeno: 31.08.2023.
- [9] Swift (programski jezik), [https://en.wikipedia.org/wiki/Swift_\(programming_language\)](https://en.wikipedia.org/wiki/Swift_(programming_language)), pristupljeno: 01.09.2023.
- [10] Go (programski jezik), [https://en.wikipedia.org/wiki/Go_\(programming_language\)](https://en.wikipedia.org/wiki/Go_(programming_language)), pristupljeno: 01.09.2023.
- [11] Guido van Rossum, https://en.wikipedia.org/wiki/Guido_van_Rossum, pristupljeno: 01.09.2023.
- [12] Best Python Libraries for Machine Learning and Deep Learning, <https://towardsdatascience.com/best-python-libraries-for-machine-learning-and-deep-learning-b0bd40c7e8c>, pristupljeno: 02.09.2023.
- [13] Numpy, <https://en.wikipedia.org/wiki/NumPy>, pristupljeno: 02.09.2023.
- [14] scikit – learn, <https://en.wikipedia.org/wiki/Scikit-learn>, pristupljeno: 02.09.2023.
- [15] TensorFlow, <https://en.wikipedia.org/wiki/TensorFlow>, pristupljeno: 02.09.2023.
- [16] Django (web okvir), [https://en.wikipedia.org/wiki/Django_\(web_framework\)](https://en.wikipedia.org/wiki/Django_(web_framework)), pristupljeno: 02.09.2023.

- [17] Lorem ipsum, https://en.wikipedia.org/wiki/Lorem_ipsum, pristupljeno: 02.09.2023.
- [18] pandas (softver), [https://en.wikipedia.org/wiki/Pandas_\(software\)](https://en.wikipedia.org/wiki/Pandas_(software)), pristupljeno: 02.09.2023.
- [19] Matplotlib, <https://en.wikipedia.org/wiki/Matplotlib>, pristupljeno: 02.09.2023.
- [20] seaborn, <https://seaborn.pydata.org/>, pristupljeno: 02.09.2023.
- [21] OpenAI, <https://openai.com/blog/chatgpt>, pristupljeno: 31.08.2023.
- [22] How to Create a PDF Report for Your Data Analysis in Python, <https://towardsdatascience.com/how-to-create-a-pdf-report-for-your-data-analysis-in-python-2bea81133b>, pristupljeno 31.08.2023.

PRILOG

Programski kod:

```
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import seaborn as sns
import lorem
from fpdf import FPDF

# cell height
ch = 8

class PDF(FPDF):
    def __init__(self):
        super().__init__()
    def header(self):
        self.set_font('Arial', "", 12)
        self.cell(0, 8, 'Zaglavlje', 0, 1, 'C')
    def footer(self):
        self.set_y(-15)
        self.set_font('Arial', "", 12)
        self.cell(0, 8, f'Stranica {self.page_no()}', 0, 0, 'C')

# Load or prepare pandas DataFrame
df=pd.DataFrame({'Apscisa':['Kategorija    1','Kategorija    2','Kategorija    3','Kategorija    4'],'Ordinata':[400,600,300,500]})

# Save figures to use in the PDF
fig, ax = plt.subplots(1,1, figsize = (6, 4))
sns.barplot(data = df, x = 'Apscisa', y = 'Ordinata')
plt.title("Graf")
plt.savefig('./example_Graf.png', transparent=False, facecolor='white', bbox_inches="tight")
plt.close()

# Generate PDF
pdf = PDF()
```

```
pdf.add_page()
pdf.set_font('Arial', 'B', 24)
pdf.cell(w=0, h=20, txt="Automatska izrada PDF izvjesca", ln=1)
pdf.set_font('Arial', "", 16)
pdf.cell(w=30, h=ch, txt="Datum: ", ln=0)
pdf.cell(w=30, h=ch, txt="25/08/2023", ln=1)
pdf.cell(w=30, h=ch, txt="Autor: ", ln=0)
pdf.cell(w=30, h=ch, txt="Josip Strkalj", ln=1)
pdf.ln(ch)
pdf.multi_cell(w=0, h=5, txt=lorem.paragraph())
pdf.image('./example_chart.png', x = 10, y = None, w = 100, h = 0, type = 'PNG', link = "")
pdf.ln(ch)
pdf.multi_cell(w=0, h=5, txt=lorem.paragraph())
pdf.ln(ch)

# Table Header
pdf.set_font('Arial', 'B', 16)
pdf.cell(40, ch, 'Apscisa', 1, 0, 'C')
pdf.cell(40, ch, 'Ordinata', 1, 1, 'C')

# Table contents
pdf.set_font('Arial', "", 16)
for i in range(0, len(df)):
    pdf.cell(40, ch, df['Apscisa'].iloc[i], 1, 0, 'C')
    pdf.cell(40, ch, df['Ordinata'].iloc[i].astype(str), 1, 1, 'C')

pdf.output(f'./example.pdf', 'F')
```