**Информационные системы управления корпоративным контентом**

**Практическая работа №7**

**ФИО:** Гусев Тимофей Андреевич

**Группа:** ИКБО-08-21

**Дата предоставления в СДО:**

**Задание 1. Создать шаблон документа Заявление на предоставление отпуска сотруднику. Реализовать скрипт для заполнения шаблона.**

На рисунке 1 представлен шаблон заявления

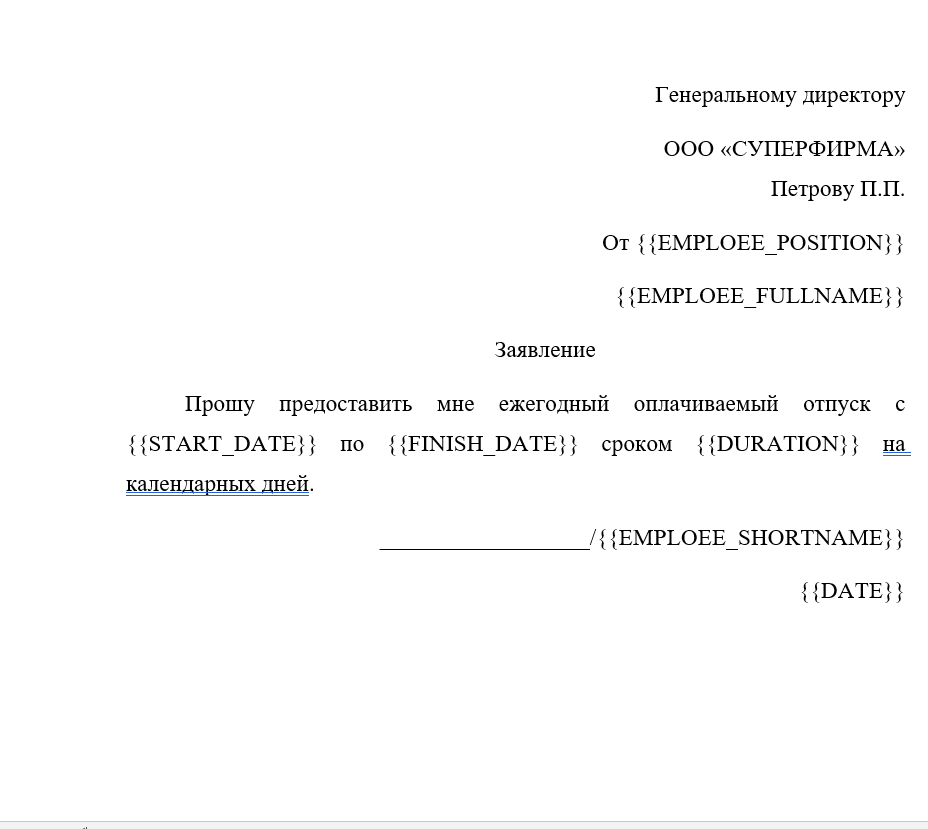


Рисунок 1 – Шаблон заявления

На листинге 1 представлен код программы для задания 1.

Листинг 1 – Код программы подстановки данных в шаблон

from docx import Document  
  
  
def main():  
 template = 'template.docx'  
 result = 'result.docx'  
 data = {  
 "{{EMPLOEE\_POSITION}}": "программиста",  
 "{{EMPLOEE\_FULLNAME}}": "Иванова Ивана Ивановича",  
 "{{EMPLOEE\_SHORTNAME}}": "Иванов И.И.",  
 "{{START\_DATE}}": "01.09.2022",  
 "{{FINISH\_DATE}}": "14.09.2022",  
 "{{DURATION}}": "14",  
 "{{DATE}}": "23.03.2022",  
 }  
 template\_doc = Document(template)  
 for key, value in data.items():  
 for paragraph in template\_doc.paragraphs:  
 replace\_text(paragraph, key, value)  
 template\_doc.save(result)  
  
  
def replace\_text(paragraph, key, value):  
 if key in paragraph.text:  
 paragraph.text = paragraph.text.replace(key, value)  
  
  
main()

На рисунке 2 представлен результат выполнения программы, а именно составленный по шаблону документ.

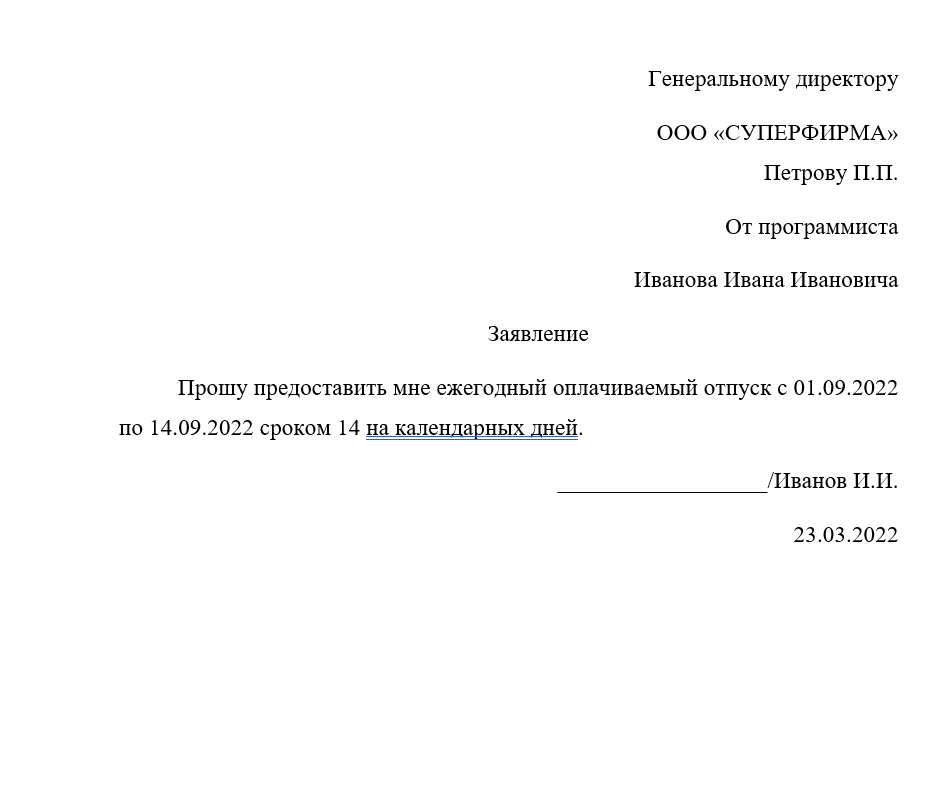


Рисунок 2 – Результат выполнения программы

**Задание 2. Дополнить скрипт для заполнения данных в шаблоне с таблицей.**

На рисунке 3 представлен шаблон документа для задания 2.

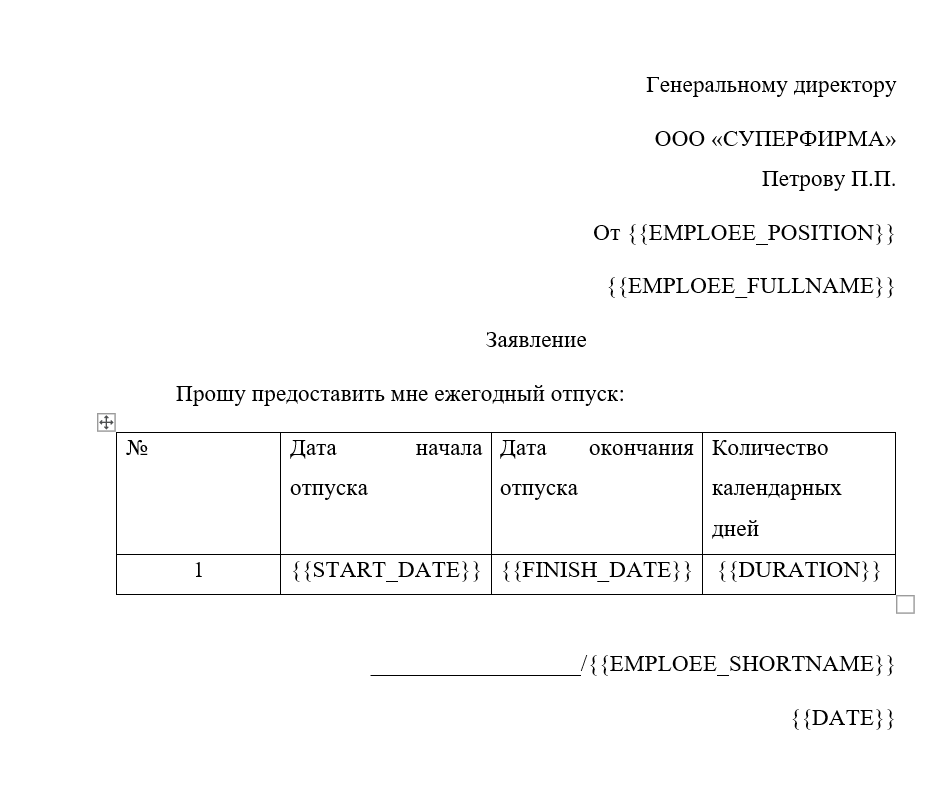


Рисунок 3 – Шаблон заявления с таблицей

На листинге 2 представлен код, при выполнении которого заполняются ячейки в таблице шаблона.

Листинг 2 – Код программы для подстановки данных в таблицу

from docx import Document  
  
  
def main():  
 template = 'template.docx'  
 result = 'result.docx'  
 data = {  
 "{{EMPLOEE\_POSITION}}": "программиста",  
 "{{EMPLOEE\_FULLNAME}}": "Иванова Ивана Ивановича",  
 "{{EMPLOEE\_SHORTNAME}}": "Иванов И.И.",  
 "{{START\_DATE}}": "01.09.2022",  
 "{{FINISH\_DATE}}": "14.09.2022",  
 "{{DURATION}}": "14",  
 "{{DATE}}": "23.03.2022",  
 }

template\_doc = Document(template)  
 for key, value in data.items():  
 for paragraph in template\_doc.paragraphs:  
 replace\_text(paragraph, key, value)  
 template\_doc.save(result)  
 for row in template\_doc.tables[0].rows:  
 for cell in row.cells:  
 replace\_text(cell, key, value)  
 template\_doc.save(result)  
  
  
def replace\_text(paragraph, key, value):  
 if key in paragraph.text:  
 paragraph.text = paragraph.text.replace(key, value)  
  
  
main()

На рисунке 4 представлен результат выполнения кода представленного на листинге 2.

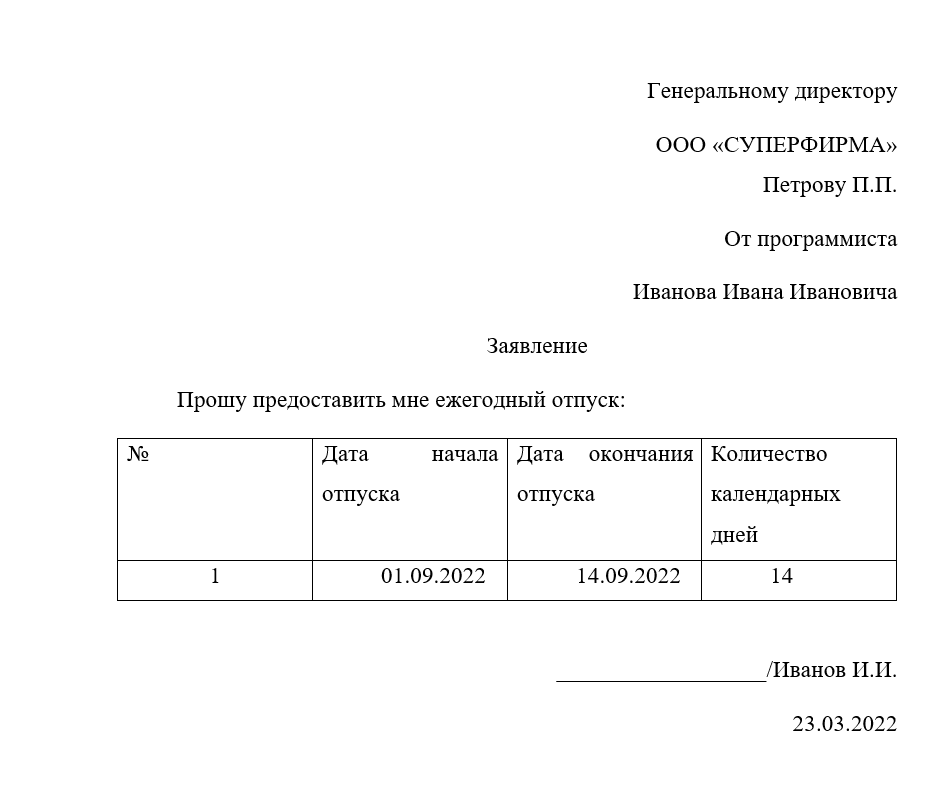


Рисунок 4 – Результат выполнения программы

**Задание 3. Дополнить скрипт для генерации документа с чтением данных из текстового документа. При этом текущая дата там не будет записана, ее нужно будет получать в скрипте средствами соответствующей библиотеки языка.**

На рисунке 5 представлен текстовый документ для генерации документа.

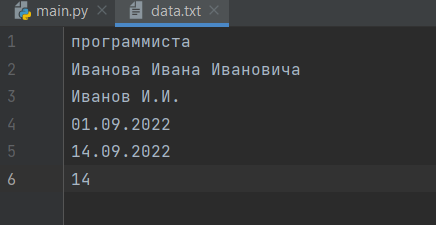


Рисунок 5 – Содержимое текстового документа

Также стоит заметить, что шаблон с прошлого задания не был изменен. На листинге 3 представлен код для решения второго задания.

Листинг 3 – Код программы для составления документа из файла

from docx import Document  
import datetime  
  
  
def main():  
 with open("data.txt", "r", encoding="UTF-8") as f:  
 lines = f.readlines()  
 template = 'template.docx'  
 result = 'result.docx'  
 data = {  
 "{{EMPLOEE\_POSITION}}": lines[0],  
 "{{EMPLOEE\_FULLNAME}}": lines[1],  
 "{{EMPLOEE\_SHORTNAME}}": lines[2],  
 "{{START\_DATE}}": lines[3],  
 "{{FINISH\_DATE}}": lines[4],  
 "{{DURATION}}": lines[5],  
 "{{DATE}}": datetime.datetime.today().date().strftime("%d.%m.%y")  
 }  
 template\_doc = Document(template)  
 for key, value in data.items():  
 for paragraph in template\_doc.paragraphs:  
 replace\_text(paragraph, key, value)  
 template\_doc.save(result)  
 for row in template\_doc.tables[0].rows:  
 for cell in row.cells:  
 replace\_text(cell, key, value)  
 template\_doc.save(result)  
  
  
def replace\_text(paragraph, key, value):  
 if key in paragraph.text:  
 paragraph.text = paragraph.text.replace(key, value)  
  
  
main()

На рисунке 6 представлен документ, полученный в результате выполнения программы.

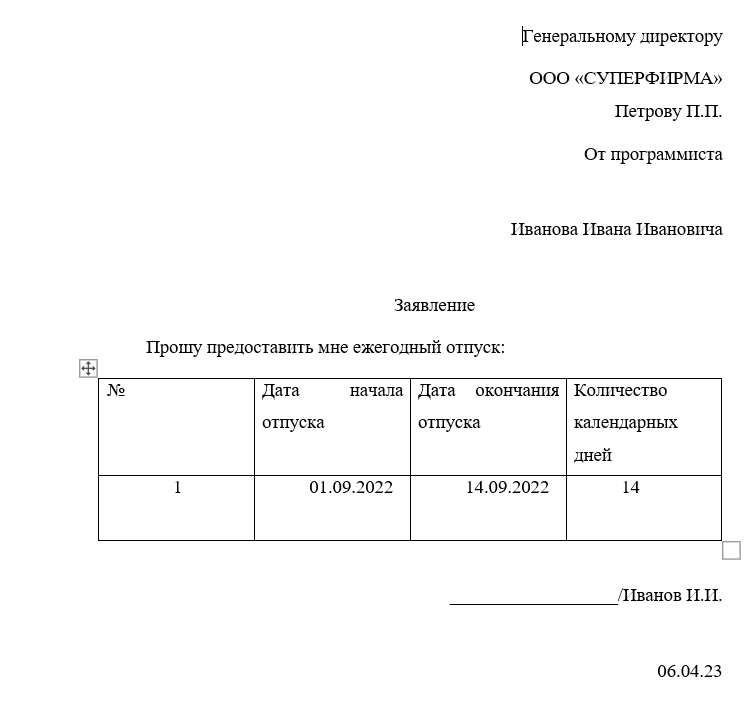


Рисунок 6 – Результат выполнения программы

**Задание 4. Дополнить скрипт для генерации документа с чтением как данных, так и ключей из текстового документа.**

На рисунке 7 представлено содержимое исходного текстового документа с данными.

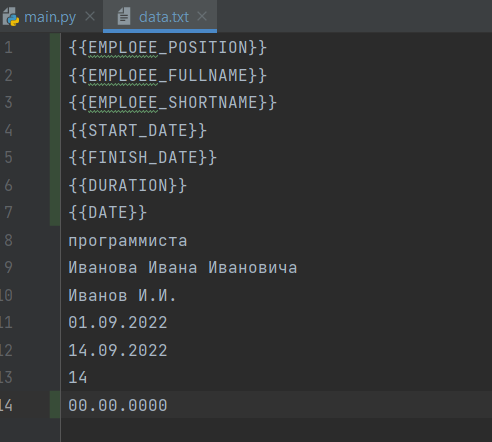


Рисунок 7 – Содержимое текстового файла

Также стоит заметить, что шаблон с прошлого задания не был изменен. На листинге 4 представлена программная реализация задания 4.

Листинг 4 – Код программы для задания 4

from docx import Document  
import datetime  
  
  
def main():  
 data = {}  
 with open("data.txt", "r", encoding="UTF-8") as f:  
 lines = f.readlines()  
 keys = lines[:7]  
 values = lines[7:]  
 for i in range(len(keys)):  
 data[keys[i].strip()] = values[i].strip()  
 template = 'template.docx'  
 result = 'result.docx'  
 template\_doc = Document(template)  
 for key, value in data.items():  
 for paragraph in template\_doc.paragraphs:  
 replace\_text(paragraph, key, value)  
 template\_doc.save(result)  
 for row in template\_doc.tables[0].rows:  
 for cell in row.cells:  
 replace\_text(cell, key, value)  
 template\_doc.save(result)  
  
  
def replace\_text(paragraph, key, value):  
 if key in paragraph.text:  
 paragraph.text = paragraph.text.replace(key, value)  
  
  
main()

На рисунке 8 представлен результат выполнения программы из листинга 4, а именно сгенерированный документ.

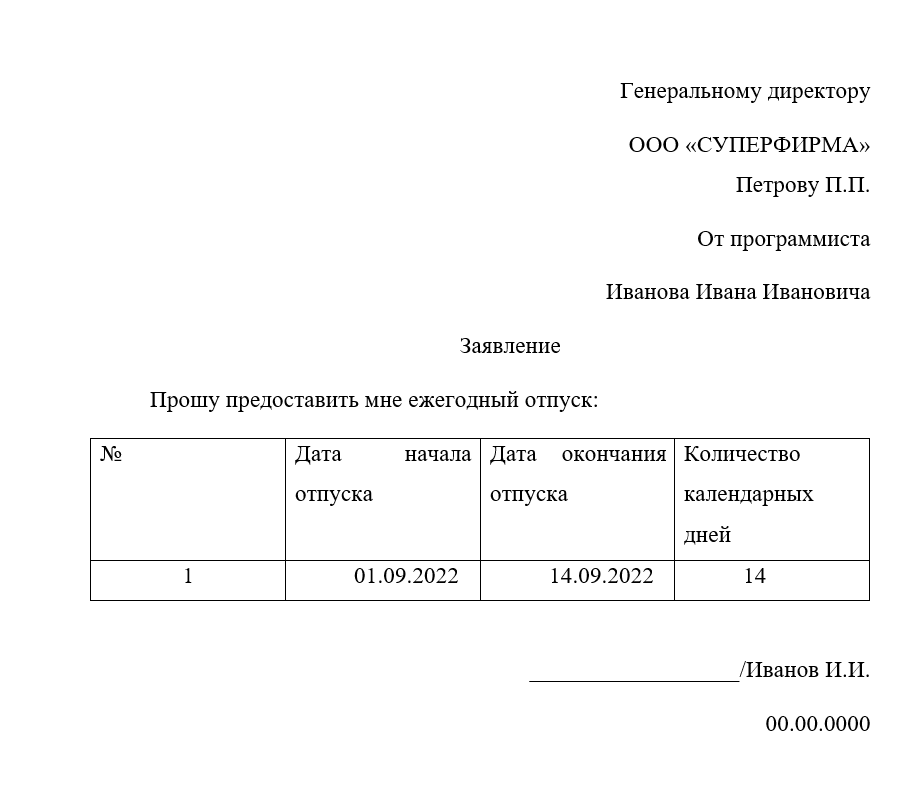


Рисунок 8 – Полученный в ходе выполнения программы документ