**Текущее решение:**

Программа перебирает файлы в указанной папке, из каждого файла достает данные план и факт каждого сотрудника, подсчитывает успешность каждого проекта, затем высчитывает среднее арифметические успешности по всем проектам и выводит в порядке убывания успешности всех сотрудников, либо в консоль (при запуске программы в IDE), либо создает отдельный текстовый файл с результатами анализа (при запуске Analysis\_of\_success.exe в папке анализируемыми excel-файлами).

**Перспективы расширения функционала:**

Можно реализовать анализ успешности всех руководителей проектов и анализ каждого проекта в отдельности. Для этого можно использовать данные из словаря, который уже создан и хранит все данные по сотрудникам. Если анализировать успешность проекта смотрим на количество сотрудников и количество затраченных дней каждым сотрудником. Если анализировать успешность руководителей, берем все проекты которыми управлял сотрудник и вычисляем успешность этих проектов.

**Analysis\_of\_success.exe**

Создает текстовый файл с результатами анализа эффективности сотрудников. Файл .exe нужно поместить в паку с excel-файлами.

**Сторонние библиотеки:**

openpyxl — библиотека для записи и чтения файлов excel

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

glob - предоставляет функцию для создания списков файлов по шаблону из найденных в каталоге

<https://docs.python.org/3/library/glob.html>

pyinstaller - объединяет приложение Python и все его зависимости в один пакет

<http://www.pyinstaller.org/>

**Исходный код (main.pyw):**

**import** openpyxl  
**import** glob  
  
all\_data = {}  
count\_employee = 0  
start\_col = 4 *# номер колоки с которой произойдет считывание (счетчик начинается с 0)*  
**for** f **in** glob.glob(**"\*.xlsx"**):  
 book = openpyxl.open(f, read\_only=**True**)  
 sheet = book.active  
 **for** col **in** range(start\_col, sheet.max\_column, 2):  
 name\_employee = sheet[1][col].value  
  
 *# Добавляем сотрудника в словарь если его там нет*  
**if** name\_employee **not in** all\_data:  
 all\_data[name\_employee] = [0, 0, 0, 0, **''**]  
  
 *# Удаляем перенос строки и все лишнее после ФИО*  
index\_new\_line = name\_employee.index(**'\n'**)  
 index\_point = name\_employee.index(**'.'**)  
 index\_end\_name = name\_employee.index(**'.'**, index\_point + 1)  
 **if** index\_new\_line == -1:  
 **if** index\_end\_name == -1:  
 all\_data[name\_employee][start\_col] = name\_employee  
 **else**:  
 all\_data[name\_employee][start\_col] = name\_employee[:index\_end\_name + 1]  
 **else**:  
 all\_data[name\_employee][start\_col] = name\_employee[:index\_new\_line]  
  
 count\_employee += 1  
  
 **for** row **in** range(2, sheet.max\_row + 1):  
 *# Увеличиваем количество проектов на 1 для этого сотрудника*  
all\_data[name\_employee][2] += 1  
  
 plan = sheet[row][col].value  
 fact = sheet[row][col + 1].value  
  
 *# Добавляем в словарь количество смен план*  
**if** type(plan) != int **or** plan < 0:  
 all\_data[name\_employee][0] += 0  
 **else**:  
 all\_data[name\_employee][0] += plan  
  
 *# Добавляем в словарь количество смен факт*  
**if** type(fact) != int **or** fact < 0:  
 all\_data[name\_employee][1] += 0  
 **else**:  
 all\_data[name\_employee][1] += fact  
  
 *# Вычисляем успешность сотрудника по формуле:*  
 *# успешность = (текущая успешность + план/факт) / 2 (2 - так как расчитываем средн. арифм. двух коэф.*  
**if** (type(plan) != int **or** plan == 0) **and** (type(fact) != int **or** fact == 0):  
 *# Сотрудник не участвовал в проекте, убавляем количесвто проектов у него*  
all\_data[name\_employee][2] -= 1  
 **elif** (type(plan) != int **or** plan == 0) **and** fact > 0:  
 all\_data[name\_employee][3] = (all\_data[name\_employee][3] + 1) / 2  
 **elif** plan == fact:  
 all\_data[name\_employee][3] = (all\_data[name\_employee][3] + 1) / 2  
 *# Сотруднику дали работу, но он не приступил к ней, КПД сотрудника в этом проекте 0*  
**elif** type(fact) != int **or** fact == 0:  
 all\_data[name\_employee][3] = (all\_data[name\_employee][3] + 0) / 2  
 **else**:  
 all\_data[name\_employee][3] = (all\_data[name\_employee][3] + plan / fact) / 2  
  
*# Создание списка только с именем сотрудника и его успешностью*  
list = []  
count = 0  
**for** employee **in** all\_data:  
 temp\_list = [0, 0]  
 temp\_list[0] = format(all\_data[employee][3], **'.3f'**)  
 temp\_list[1] = all\_data[employee][4]  
 list.append(temp\_list)  
 count += 1  
  
*# Сортировка списка*  
**for** i **in** range(0, len(list) - 1):  
 **for** j **in** range(i + 1, len(list)):  
 **if** list[j][0] > list[i][0]:  
 list[i][0], list[j][0] = list[j][0], list[i][0]  
 list[i][1], list[j][1] = list[j][1], list[i][1]  
  
*# Вывод в консоль*  
count\_employee = 1  
print(**'№ Успешность Ф.И.О.'**)  
print(**'- - - - - - - - - - - - - - -'**)  
**for** success, employee **in** list:  
 print(count\_employee, **'. '**,success, **' '**,employee)  
 count\_employee += 1  
  
**"""**  
**Убрать документирование для создания текстового файла с результатами анализа**  
  
**# Создание текстового файла и запись в него полученных данных**  
**my\_file = open('Анализ успешности сотрудников.txt', 'w')**  
**count\_employee = 1**  
**my\_file.write('№ Успешность Ф.И.О.\n')**  
**my\_file.write('- - - - - - - - - - - - - - -\n')**  
**for i in list:**  
 **my\_file.write(str(count\_employee))**  
 **my\_file.write('. ')**  
 **my\_file.write(str(i[0]))**  
 **my\_file.write(' ')**  
 **my\_file.write(str(i[1]))**  
 **my\_file.write('\n')**  
 **count\_employee += 1**  
**my\_file.close()**  
  
**"""**