**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  
  
*# Создание сотрудника*  
*#*  
*# name - имя сотруника, count\_project - количество проектов у сотрудника, plan - план, fact - факт*  
**class** Employee:  
 *# Хранит все экземпляры класса:*  
dict\_employee = {}  
  
 **def** \_\_init\_\_(self, name, count\_project = 0, plan = 0, fact = 0, success\_rate = 0):  
 self.name = name  
 self.count\_project = count\_project  
 self.plan = plan  
 self.fact = fact  
 self.success\_rate = success\_rate  
  
*# Удаление переносов строки и пробелов после ФИО*  
**def** remov\_pass (value):  
 index\_new\_line = value.index(**'\n'**)  
 index\_point = value.index(**'.'**)  
 index\_end\_name = value.index(**'.'**, index\_point + 1)  
 **if** index\_new\_line == -1:  
 **if** index\_end\_name == -1:  
 **return** value  
 **else**:  
 **return** value[:index\_end\_name + 1]  
 **else**:  
 **return** value[:index\_new\_line]  
  
*# Вычисляем успешность сотрудника по формуле:*  
*# успешность = (текущая успешность + план/факт) / 2 (2 - так как расчитываем средн. арифм. двух коэф.)*  
*#*  
*# value - Экземпляр класса Employee, plan - план, fact - факт*  
**def** success\_rate (value, plan, fact):  
 **if** (type(plan) != int **or** plan == 0) **and** (type(fact) != int **or** fact == 0):  
 *# Сотрудник не участвовал в проекте, убавляем количество проектов у него*  
value.count\_project -= 1  
 **elif** (type(plan) != int **or** plan == 0) **and** fact > 0:  
 value.success\_rate = (value.success\_rate + 1) / 2  
 **elif** plan == fact:  
 value.success\_rate = (value.success\_rate + 1) / 2  
 *# Сотруднику дали работу, но он не приступил к ней, КПД сотрудника в этом проекте 0*  
**elif** type(fact) != int **or** fact == 0:  
 value.success\_rate = (value.success\_rate + 0) / 2  
 **else**:  
 value.success\_rate = (value.success\_rate + plan / fact) / 2  
  
*# Вывод отсортированного результата успешности сотрудников*  
*#*  
*# dict\_employee - словарь с экземплярами класса*  
**def** get\_sorted\_list(dict\_employee):  
 *# Создание списка только с именем сотрудника и его успешностью*  
list = []  
 count = 0  
 **for** employee **in** dict\_employee:  
 temp\_list = [0, 0]  
 temp\_list[0] = format(dict\_employee[employee].success\_rate, **'.3f'**)  
 temp\_list[1] = dict\_employee[employee].name  
 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  
  
*# номер колоки с которой произойдет считывание (счетчик начинается с 0)*  
start\_col = 4  
  
**for** file **in** glob.glob(**"\*.xlsx"**):  
 book = openpyxl.open(file, read\_only=**True**)  
 sheet = book.active  
  
 *# перебираем колонки через одну, так как 1ая - план сотрудника, 2-ая - факт,*  
 *# выходит что каждый новый сотрудник начниается на нечетной колонке (1, 3, ...)*  
**for** col **in** range(start\_col, sheet.max\_column, 2):  
 name\_employee = remov\_pass(sheet[1][col].value)  
  
 *# Добавляем сотрудника в словарь если его там нет*  
**if** name\_employee **not in** Employee.dict\_employee:  
 Employee.dict\_employee[name\_employee] = Employee(name\_employee)  
 **else**:  
 **continue**  
  
*# Перебор со 2ой и до последней строки файла, первая строка с данными ФИО*  
**for** row **in** range(2, sheet.max\_row + 1):  
 *# Увеличиваем количество проектов на 1 для этого сотрудника*  
Employee.dict\_employee[name\_employee].count\_project += 1  
  
 *# Cокращение:*  
plan = sheet[row][col].value  
 fact = sheet[row][col + 1].value  
  
 *# Добавляем в словарь количество смен план*  
**if** type(plan) == int **and** plan > 0:  
 Employee.dict\_employee[name\_employee].plan += plan  
  
 *# Добавляем в словарь количество смен факт*  
**if** type(fact) == int **and** fact > 0:  
 Employee.dict\_employee[name\_employee].fact += fact  
  
 *# Вычисление и запись успешности сотруднику*  
success\_rate(Employee.dict\_employee[name\_employee], plan, fact)  
  
*# Вывод результата*  
get\_sorted\_list (Employee.dict\_employee)  
  
**"""**  
**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 item in list:**  
 **my\_file.write(str(count\_employee))**  
 **my\_file.write('. ')**  
 **my\_file.write(str(item[0]))**  
 **my\_file.write(' ')**  
 **my\_file.write(str(item[1]))**  
 **my\_file.write('\n')**  
 **count\_employee += 1**  
**my\_file.close()**  
  
**"""**