МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное

образовательное учреждение высшего образования

**«Челябинский государственный университет»**

**(ФГБОУ ВО «ЧелГУ»)**

**ОТЧЕТ**

по производственной практике: Сбор и представление данных по готовому шаблону в ГБ Миасс №2

Колледж ЧелГУ

Киселев Артемий Кириллович

Группа: СПМИС-301

|  |  |
| --- | --- |
| Руководитель практики от профильной  организации:  Сидоров Иван Сергеевич  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись )  «\_\_\_\_» \_\_\_\_\_\_\_\_\_2023 г. | Руководитель практики от образовательной организации:  Кирилл Андреевич Лихачёв \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись ) |
|  | Оценка за практику по  результатам защиты отчёта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)  «\_\_\_\_»\_\_\_\_\_\_\_\_\_2023 г. |

Челябинск, 2023 г.

## ЛИЧНАЯ КАРТОЧКА ИНСТРУКТАЖА\*

**ПО ОЗНАКОМЛЕНИЮ С ТРЕБОВАНИЯМИ ОХРАНЫ ТРУДА,**

**ТЕХНИКИ БЕЗОПАСНОСТИ, ПОЖАРНОЙ БЕЗОПАСНОСТИ, ПРАВИЛАМИ ВНУТРЕННЕГО РАСПОРЯДКА**

Обучающегося ФГБОУ ВО «ЧелГУ» Киселева Артемия Кирилловича при прохождении производственной практики: эксплуатация и модификация информационных систем в ООО «ИТС»

|  |  |  |
| --- | --- | --- |
| **Вид инструктажа** | **Инструктаж проведён\*\*** | **Ознакомлен** |
| **по требованиям охраны труда** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Ф.И.О., должность, подпись**    **\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **дата** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **подпись обучающегося**    **\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_**  **дата** |
| **по технике безопасности** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Ф.И.О., должность, подпись**    **\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **дата** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **подпись обучающегося**    **\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_**  **дата** |
| **по пожарной безопасности** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Ф.И.О., должность, подпись**    **\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **дата** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **подпись обучающегося**    **\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_**  **дата** |
| **по правилам внутреннего трудового распорядка** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Ф.И.О., должность, подпись**    **\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **дата** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **подпись обучающегося**    **\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_**  **дата** |

\* в соответствии с ПОЛОЖЕНИЕМ об организации практик обучающихся по основным профессиональным образовательным программам среднего профессионального образования, основным профессиональным образовательным программам высшего образования - программам бакалавриата, программам специалитета, программам магистратуры в ФГБОУ ВО «ЧелГУ»

\*\* **инструктаж** **проводит** специалист по охране труда или работник, на которого приказом работодателя (или уполномоченного им лица) возложены эти обязанности

Руководитель практики от профильной организации \_\_\_\_\_\_\_\_\_\_\_\_\_\_С.И.Сидоров.

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное

образовательное учреждение высшего образования

**«Челябинский государственный университет»**

**(ФГБОУ ВО «ЧелГУ»)**

*Колледж ЧелГУ*

*направление подготовки 09.02.04 «Информационные системы (по отраслям)»*

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ**

**на производственную практику: сбор входных данных и отображение их по готовому шаблону**

**студенту Киселеву Артемию Кирилловичу, группа СПМИС-301**

Место прохождения практики ГБ Города Миасс №2

Сроки прохождения практики с 25.05.2023 по 21.06.2023.

Перечень заданий и вопросов, подлежащих исследованию (в соответствии с программой практики):

1. Установка необходимых программ для разработки.
2. Ознакомление с языком программирования Python.
3. Написание алгоритма сбора данных.
4. Написание алгоритма представления данных по готовому шаблону.
5. Сборка скрипта в готовое приложение и написание инструкции по использованию.

Руководитель практики от ЧелГУ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. К. Лихачев

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ К.А. Киселев

СОГЛАСОВАНО:

Руководитель практики от профильной организации \_\_\_\_\_\_\_\_\_ С.И. Сидоров

|  |  |  |  |
| --- | --- | --- | --- |
| Дата | Наименование работ | Оценка | Подпись руководителя  практики |
| 25.05.23 | Ознакомление с техническим заданием |  |  |
| 26.05.23 | Сбор информации о предстоящей работе |  |  |
| 30.05.23 | Установка Pycharm Community |  |  |
| 03.06.23 | Установка Python 3.10.6 на Windows 10 |  |  |
| 06.06.23 | Ознакомление с Python |  |  |
| 07.06.23 | Выполнение простого скрипта на Python |  |  |
| 09.06.23 | Установка Python библиотеки для работы с Excel |  |  |
| 14.06.23 | Написание скрипта для сбора данных |  |  |
| 14.06.23 | Написание скрипта для вывода данных по готовому шаблону |  |  |
| 14.06.23 | Проверка работоспособности скрипта |  |  |
| 15.06.23 | Сборка готового скрипта в приложение |  |  |
| 15.06.23 | Написание инструкции по использованию приложения |  |  |
| 17.06.23 | Написание отчета и дневника практики. |  |  |
| 20.06.23 | Проверка отчетов. |  |  |
| 21.06.23 | Сдача отчетов. |  |  |

Содержание и объём выполненных работ подтверждаю:

руководитель практики от профильной организации

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ С.И. Сидоров /

**ОТЧЕТ**

**по результатам производственной практики в ООО «ИТС»**

**студента группы СПМИС-202**

**ФГБОУ ВО «Челябинский государственный университет»**

**Колледж ЧелГУ**

**Катаева Данилы Андреевича**

**СОДЕРЖАНИЕ**

Введение

1. Ознакомление с техническим заданием и оценка выполнения задания.

2. Теоретическая часть. Знакомство с Python. Установка PyCharm.

3. Написание скрипта для выполнения технического задания.

4. Сборка готового приложения и написание инструкции к использованию

Заключение

Список литературы

# Введение

Актуальностью производственной практики является изучение языка программирования Python, установка PyCharm – программы для написания кода на языке программирования Python и дальнейшее написание скрипта для сбора вводных данных, и помещения их в готовый шаблон, для ускорения работы с данными по ГБ Миасс №2.

Целью производственной практики является изучение Python, установка PyCharm. Создание программы для сбора и представления данных по готовому шаблону на языке программирования Python.

Задачи производственной практики:

* Изучение технического задания, Excel таблиц, полученных от руководителя практики.
* Изучение и установка языка программирования Python.
* Установка и настройка PyCharm для написания скрипта на языке программирования Python.
* Установка библиотеки Python для работы с Excel таблицами, полученными от руководителя практики.
* Написание скрипта на языке программирования Python, который будет соответствовать требованиям технического задания в полном объеме.
* Сборка готового скрипта в отдельное приложение
* Написание инструкции по использованию приложения

Теоретической базой данного отчета являются электронные и печатные ресурсы 1, 2, 3.

Объектом исследования является язык программирования Python, предназначенный для написания различных программ.

Предметом исследования является создание скрипта для работы с Excel таблицей на языке программирования Python.

Отчет состоит из 4 глав, введения, содержания, списка литературы, заключения и электронных и печатных приложений.

# Ознакомление с техническим заданием и оценка выполнения задания

В качестве технического задания, от руководителя практики, было получено несколько Excel таблиц (Приложение 3). Первая таблица (Рисунок 2) содержит в себе данные про работников ГБ Миасс №2.

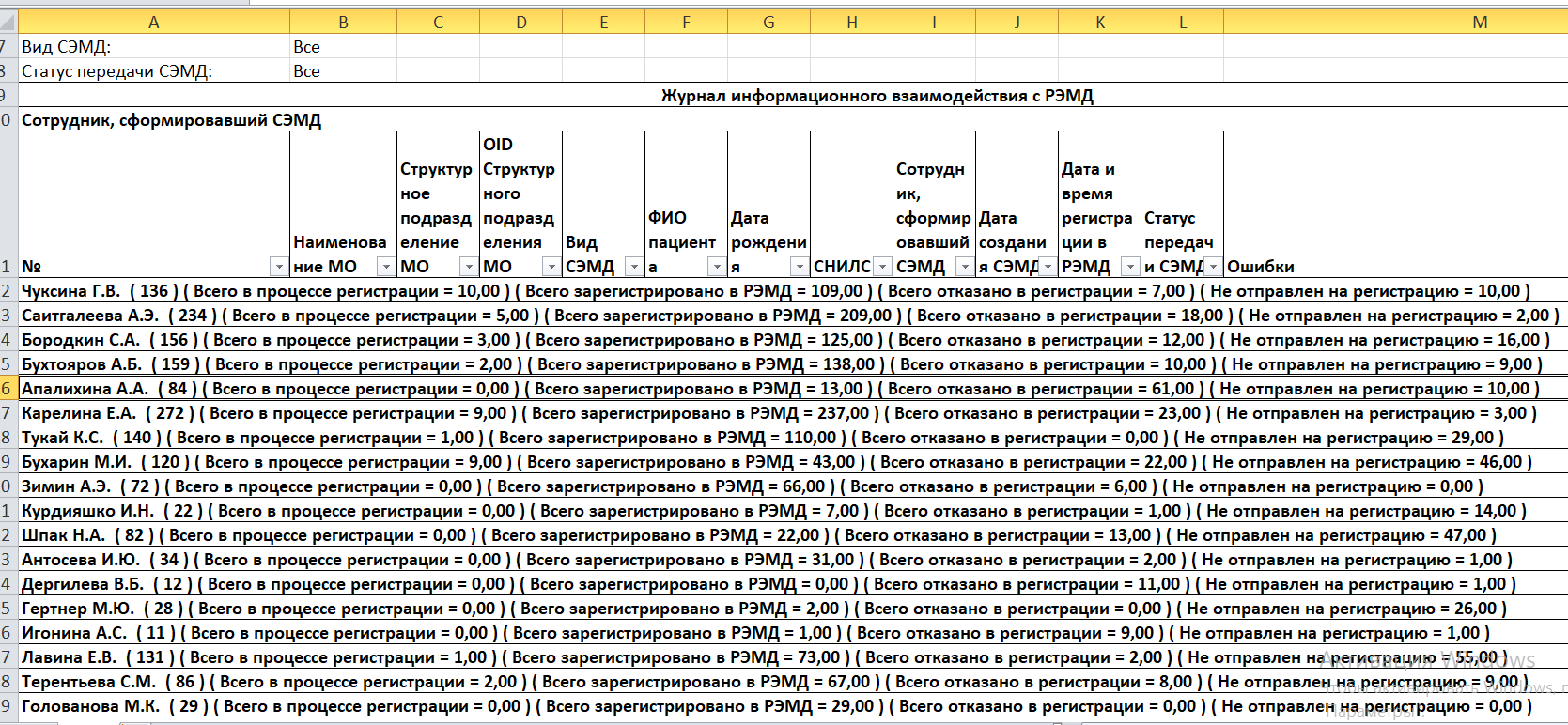


Рисунок 1 - данные по работникам городской больницы

В этой таблице у нас есть следующие данные:

- Имя сотрудника

- Общее кол-во очков сотрудника

- Количество заявок в процессе регистрации

- Количество зарегистрированных заявок

- Количество отказанных заявок на регистрацию

- Количество отправленных заявок на регистрацию

Эти данные, согласно техническому заданию, нужно вставлять во вторую таблицу (Рисунок 2). При этом если сотрудник уже есть в таблице – нужно добавить его данные только в новый блок отчета. Но если сотрудника в таблице нет – его нужно добавить в конец, заполнить существующие пропуски и вывести его данные в новый блок отчета. Также не забыть посчитать его данные в итоговой ячейке, где суммируются все очки сотрудников.

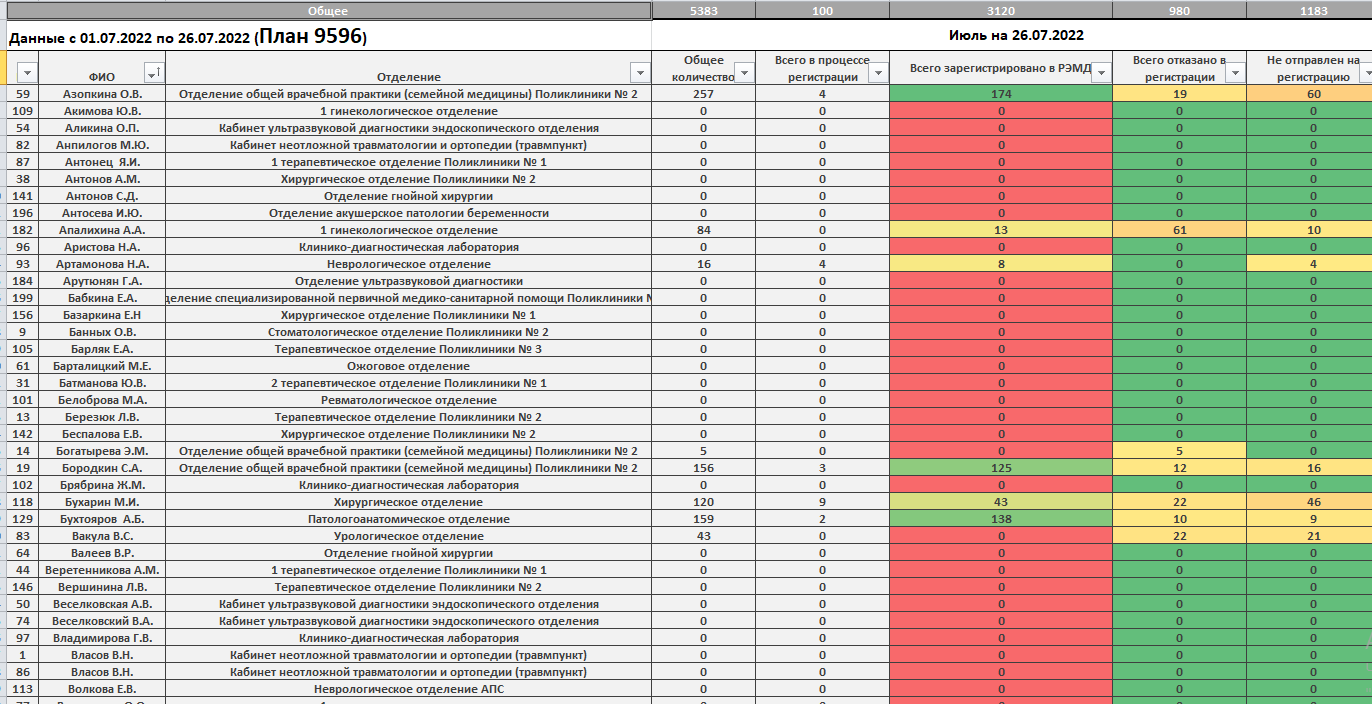


Рисунок 2 - таблица с существующими данными сотрудников

В этой таблице мы видим следующие столбцы:

- ID – порядковый номер сотрудника, выдается при добавлении нового сотрудника в таблицу.

- ФИО сотрудника – используется как ключ к существующим данным, и проверке, есть ли данные по этому сотруднику.

- Отделение – место работы сотрудника, заполняется вручную, так как этих данных нет в таблице входных данных.

- Общее количество – общее количество очков за данный период по этому сотруднику.

- Всего в процессе регистрации – берется из таблицы входных данных по этому сотруднику.

- Всего зарегистрировано - берется из таблицы входных данных по этому сотруднику. Также используется цветовая гамма, чтобы выделять эффективных сотрудников.

- Всего отказано - берется из таблицы входных данных по этому сотруднику. Также используется цветовая гамма, чтобы выделять эффективных сотрудников.

- Не отправлен на регистрацию - берется из таблицы входных данных по этому сотруднику. Также используется цветовая гамма, чтобы выделять эффективных сотрудников.

Также у каждого отчета есть заголовки (Июль на 26.07.2022, Итог Июль на 02.08.2022 и т.д.)

И над каждым отчетом есть поля, которые показывают сумму по всем столбцам снизу (Рисунок 3)

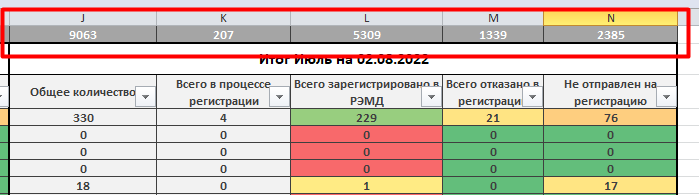


Рисунок 3 - сумма всех ячеек в столбце

Для выполнения этого технического задания я выбрал язык программирования Python - это мощный, универсальный и интерпретируемый язык программирования, который обладает простым и читаемым синтаксисом, позволяющим разработчикам писать код быстро и эффективно. Он поддерживает различные парадигмы программирования, включая процедурное, объектно-ориентированное и функциональное программирование, и предоставляет богатый выбор пользовательских библиотек, которые делают его идеальным выбором для разработки разнообразных приложений, от веб-сайтов до научных вычислений. В нашем случае, он идеально подойдет для выполнения технического задания. На нем будет легко разработать скрипт и собрать его в приложение.

# Теоретическая часть. Знакомство с Python. Установка PyCharm

# PyCharm - это интегрированная среда разработки (IDE) для языка программирования Python, разработанная компанией JetBrains. Она предоставляет разработчикам широкий набор инструментов и функций, упрощающих процесс создания, отладки и тестирования приложений на Python.

# Рекомендуемые требования для работы с PyCharm:

# Операционная система: Windows, macOS или Linux

# Процессор: Intel Core i5 или выше

# Оперативная память: рекомендуется минимум 8 ГБ

# Свободное место на жестком диске: минимум 4 ГБ для установки и дополнительное пространство для проектов

# Разрешение экрана: рекомендуется 1280x800 или выше

# Версия Python: PyCharm поддерживает разные версии Python, включая Python 2.7, 3.5 и выше. Мы будем использовать Python 3.10.6. На момент написания отчета, это LTSR (Latest Stable Release).

# Мой рабочий ноутбук полностью соответствуют рекомендуемым требованиям. Для установки перейдем на сайт <https://www.jetbrains.com/pycharm/download/#section=windows> и скачаем Community версию программы для Windows. Community – бесплатная версия программы, которая подойдет для написания небольших скриптов и изучения языка программирования Python. Скачиваем, запускаем установщик, следуем инструкциям и после установки перезагружаем ноутбук. Программа установлена. Чтобы это проверить, попробуем найти программу через поисковик Windows (Рисунок 4).

# 

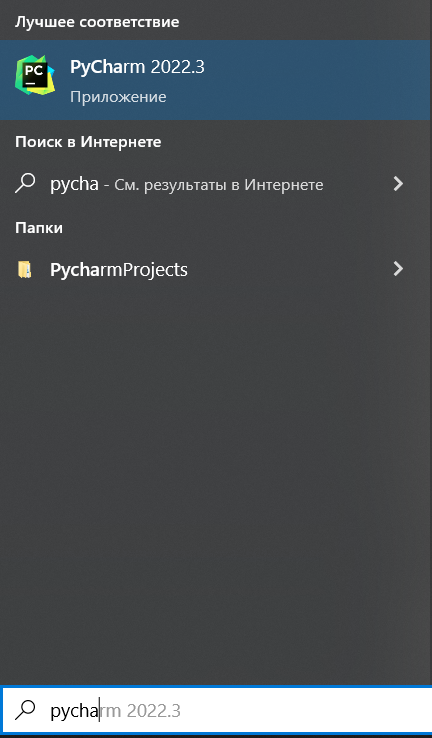


Рисунок 4 - поиск приложения в Windows

При запуске программы будет виден список проектов, которые были открыты (после установки он будет пуст, но так как я уже работал с этой программой, у меня есть другие проекты, не относящиеся к практике). Нас интересует кнопка New Project (Рисунок 5)

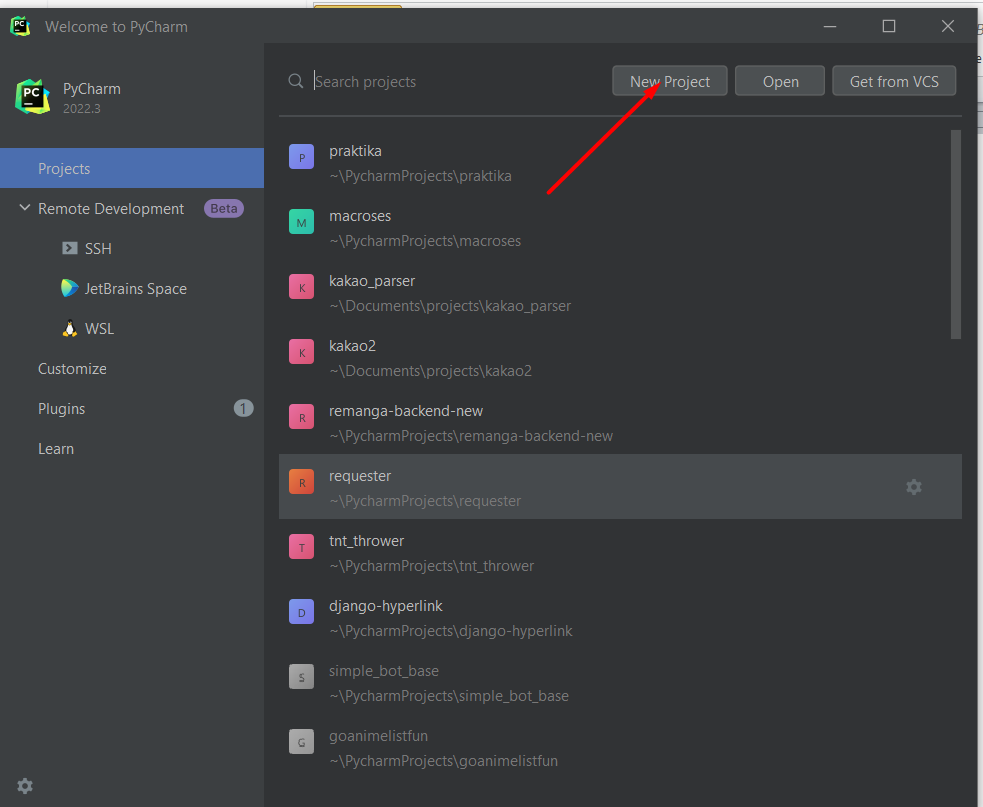


Рисунок 5 - окно выбора проектов PyCharm

Для создания проекта нам потребуется загрузить Python версии 3.10.6. Для этого, переходим на официальный сайт <https://www.python.org/downloads/release/python-3106/>, листаем вниз и выбираем 64 разрядную версию для Windows (Рисунок 6).

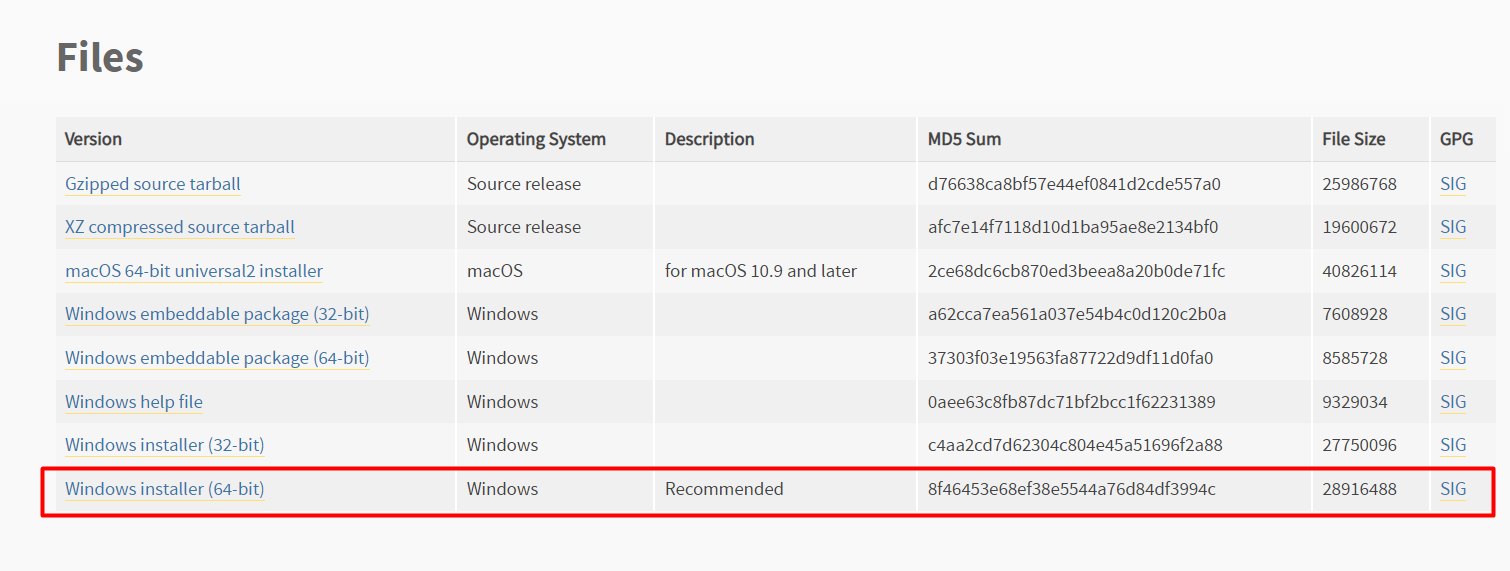


Рисунок 6 - файлы Python 3.10.6

После загрузки и установки, в окне создания проекта у нас появится путь к установленному Python в строке «Base interpreter». Также в качестве названия проекта я указал «praktika» (Рисунок 7).

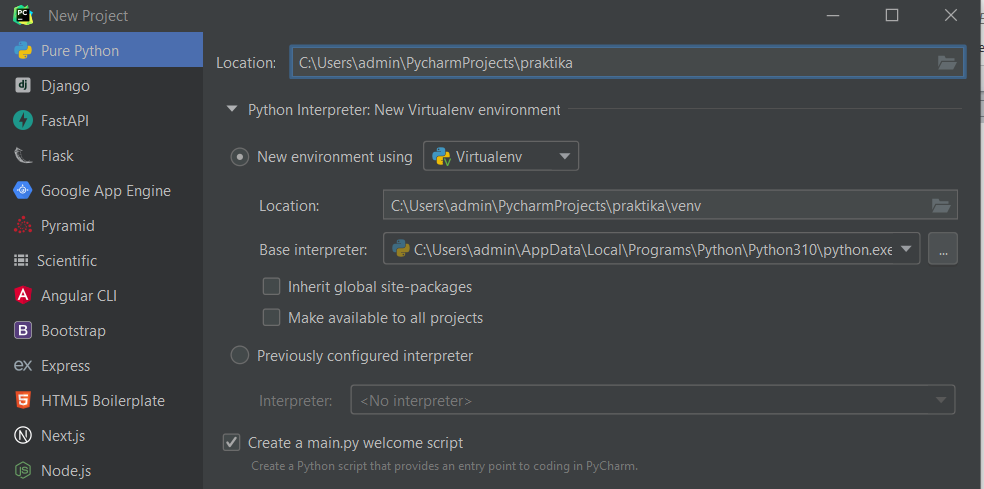


Рисунок 7 - создание нового проекта PyCharm

После нажатия кнопки создания проекта у нас откроется окно с файлами проекта. Также PyCharm создаст виртуальное окружение venv. Оно служит для того, чтобы все установленные модули хранились только в рамках этого проекта, но не глобально.

Теперь установим необходимые библиотеки. Для работы с Excel таблицей нам понадобится библиотека openpyxl. Для сборки Python скрипта в exe приложение нам понадобится библиотека pyinstaller.

Для установки библиотек в Python используется модуль pip. Устанавливаем обе библиотеки командами:

- pip install openpyxl

- pip install pyinstaller

После установки выполняем команду pip freeze > requirements.txt. Это создаст файл с нужными библиотеками, чтобы в следующий раз можно было установить все нужные библиотеки для проекта командой pip install –r requirements.txt.

Теперь приступим к изучению языка программирования Python. Python - это язык программирования с простым и читаемым синтаксисом, что делает его идеальным для начинающих разработчиков. Он предлагает множество встроенных функций и библиотек, что упрощает написание кода и повышает производительность разработки. Синтаксис Python является выразительным и позволяет писать чистый и элегантный код, что способствует легкому пониманию и сопровождению программы.

Для начала напишем простейший скрипт для сложения двух вводимых чисел (Рисунок 8)

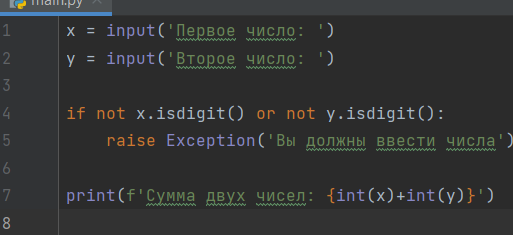


Рисунок 8 - простой скрипт на Python

Сначала мы получаем 2 числа, потом используем функцию строки isdigit для проверки что строка является числом. Если введенные данные не числа – прерываем программу ошибкой. Иначе складываем 2 числа и выводим результат в консоль.

Для проверки запускаем скрипт командой python main.py в консоли. При первом запуске я введу 2 числа. При втором запуске я введу число и слово, чтобы полностью проверить работоспособность скрипта (Рисунок 9)

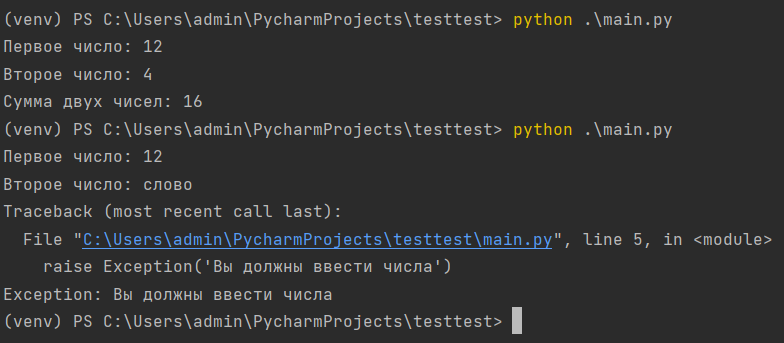


Рисунок 9 - Проверка работоспособности скрипта

Как можно видеть – всё работает правильно. Переходим к написанию скрипта для выполнения технического задания.

# Написание скрипта для выполнения технического задания

# Для начала напишем скрипт для получения всех данных с входной таблицы. Нам понадобится извлекать из входной таблицы имя сотрудника, сумму его очков, заявки в процессе, в процессе регистрации, отклоненные заявки и проигнорированные заявки. Для этого напишем следующий скрипт (Рисунок 10):

# 

Рисунок 10 - Получение данных с входной таблицы

# Для начала открываем Excel файл с названием wb.xlsx, затем в нем получаем лист с названием «Лист». Затем берем все строки с 11 и до предпоследней. Первые 11 строк общая информация, последняя строка – общая статистика, которая нам не интересна. Входные данные – строки вида «Чуксина Г.В. ( 136 ) ( Всего в процессе регистрации = 10,00 ) ( Всего зарегистрировано в РЭМД = 109,00 ) ( Всего отказано в регистрации = 7,00 ) ( Не отправлен на регистрацию = 10,00 )»

# Чтобы разделить я использовал разделение по символу «(», а затем прошёлся по каждой полученной строке и убрал символ «)», а также удалил лишнее пробелы и знаки препинания вначале и в конце слова.

Затем я обрабатываю каждую строку, приводя числа в тип int (целочисленное число), а также записываю это всё в один список formatted\_rows, в котором хранятся все строки с входного файла. Если посмотреть данные, которые хранятся в списке, мы увидим много списков с данными по сотрудникам (Рисунок 11)

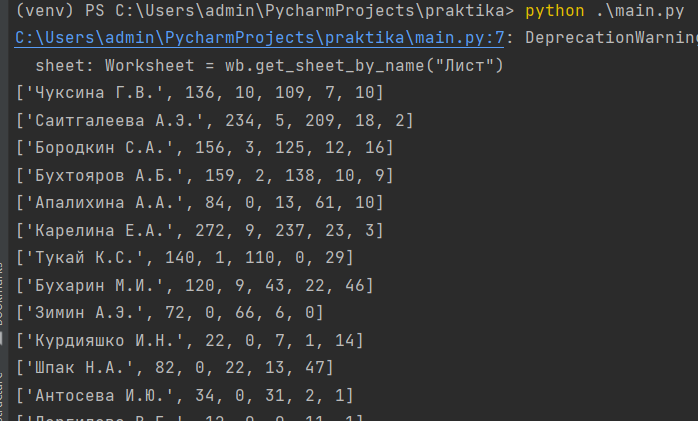


Рисунок 11 - полученные данные из таблицы

Теперь нам нужно получить имена из таблицы, в которой уже есть данные по сотрудникам. Для этого я открою эту таблицу, пройдусь по второму столбцу и помещу их в словарь, где ключ будет ФИО сотрудника, а значение будет номер строки, чтобы быстро их искать. Это сделано для оптимизации, так как ключи словаря хранятся в виде хеш таблицы (каждое фио представлено как уникальный хеш, и для каждого фио существует только 1 уникальный хеш), поиск в ней будет в разы быстрее. Если мы будем проходиться по каждому имени, сложность такого подхода будет O(n), тогда как если искать значение по хешу в хеш таблице, при уникальности хеша, сложность будет O(1), то есть потребуется всего 1 операция. O(n) – способ представления сложности алгоритма. Он указывает наихудший вариант, за который отработает тот или иной алгоритм. Например если у нас будет 10000 имен, при обычном поиске наихудший вариант – искомое имя стоит в самом конце, и нам придется пройти все 10000 имен, чтобы найти нужное. При поиске в хеш таблице по хешу имени, так как существует только один уникальный хеш, операция будет выполнена за 1 операцию. Это и представляют значения О-большое (O(1) и O(n)).

Для того чтобы создать словарь, я написал скрипт (Рисунок 12)

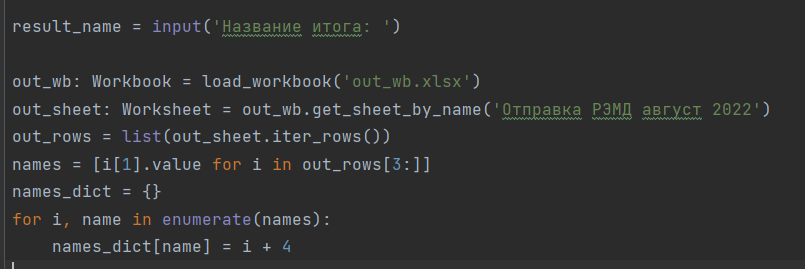


Рисунок 12 - создание словаря с именами и индексами строки

Для каждого имени значение будет I + 4 (индекс в списке имен + 4, так как строки имен начинаются с 4ой строки в таблице)

Далее нужно найти первую свободную ячейку справа, чтобы вставить туда новый отчет, который будет занимать 5 столбцов. Для этого я просто прохожусь по первой строке и ищу первую ячейку, в которую ничего не записано (Рисунок 13 и 14)

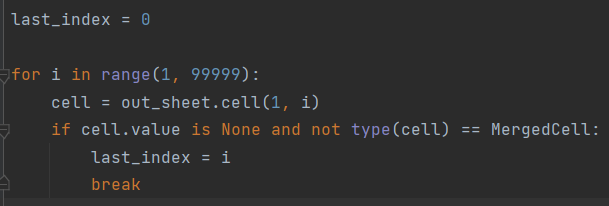


Рисунок 13 - алгоритм вычисления первой свободной ячейки

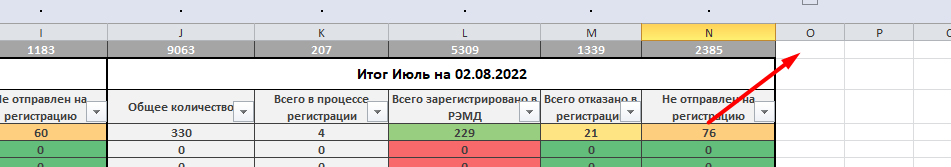


Рисунок 14 - первая свободная ячейка в таблице

Далее напишем скрипт для заполнения таблицы данными, полученными с первой таблицы с входными данными (Рисунок 15)

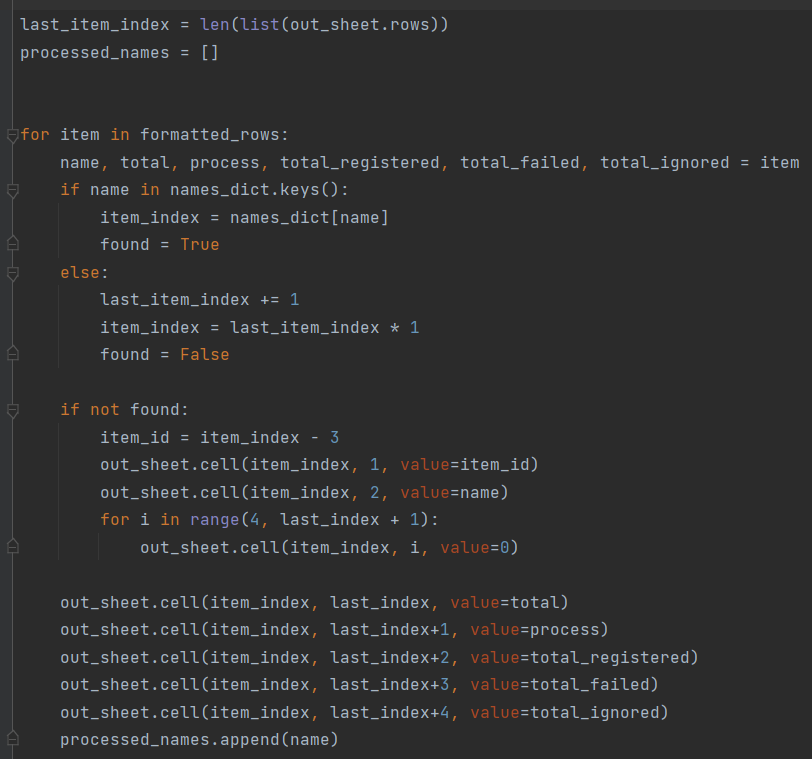


Рисунок 15 - алгоритм заполнения полученными данными

Здесь мы проходимся по списку из входных данных, распаковываем их в отдельные переменные. Затем проверяем, есть ли уже такое ФИО в таблице. Если есть – берем номер строки, если нету – добавляем к кол-ву строк 1 и выделяем её под этого сотрудника. Затем, если сотрудник не был в таблице, заполняем данные за прошлые отчеты нулями. Ну и в любом случае в конце вставляем данные, полученные из входных данных. И так делаем для каждого сотрудника.

Затем нам нужно заполнить нулями всех сотрудников, которые есть в таблице с данными, но нет в таблице с входными данными (Рисунок 16)

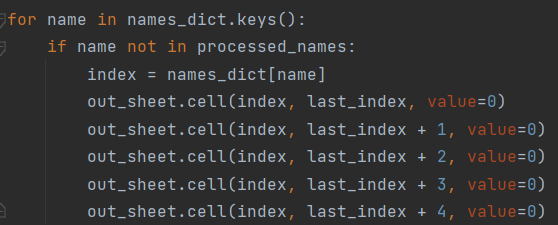


Рисунок 16 - заполнение пропущенных сотрудников

Довольно простой скрипт: берем все имена из словаря имен, проверяем, было ли обработано это имя. Если не было – заполняем данные за текущий отчет нулями.

Ну и последнее что осталось сделать – заполнить верхнюю строчку с общим итогом, названия столбцов и название отчета. Для этого спрашиваем название отчета у пользователя и заполняем оставшиеся ячейки. Результат сохраняем в файл out\_wb2.xlsx. Таким подходом мы создадим новый файл, не испортив исходный, если в алгоритме произойдет какая-то ошибка. (Рисунок 17)

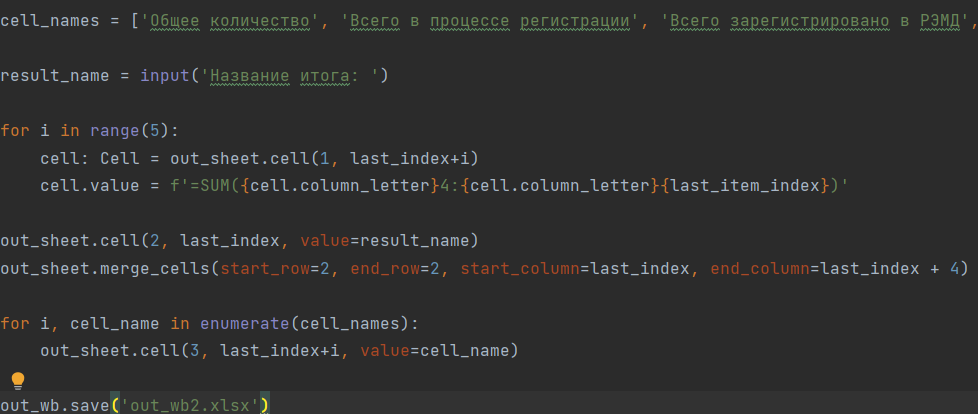


Рисунок 17 - завершение алгоритма и сохранение файла

Запускаем скрипт командой python main.py и проверяем результат (Рисунок 18, 19, 20)

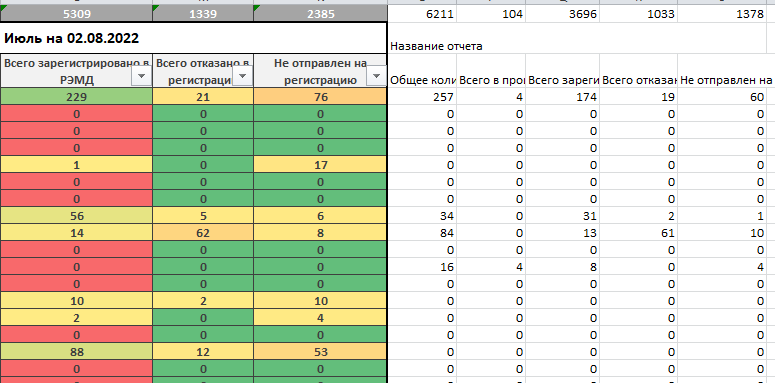


Рисунок 18 - новые полученные данные для существующих сотрудников



Рисунок 19 - новые сотрудники

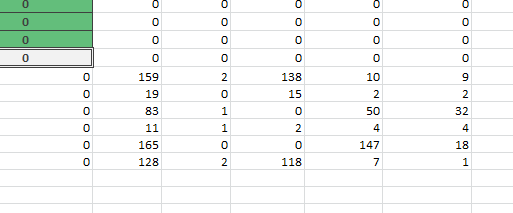


Рисунок 20 - данные новых сотрудников

Как можно видеть – все данные подставились верно. Всё что осталось пользователю – продлить вручную таблицу, использовав форматирование по образцу (Рисунок 21 и 22)

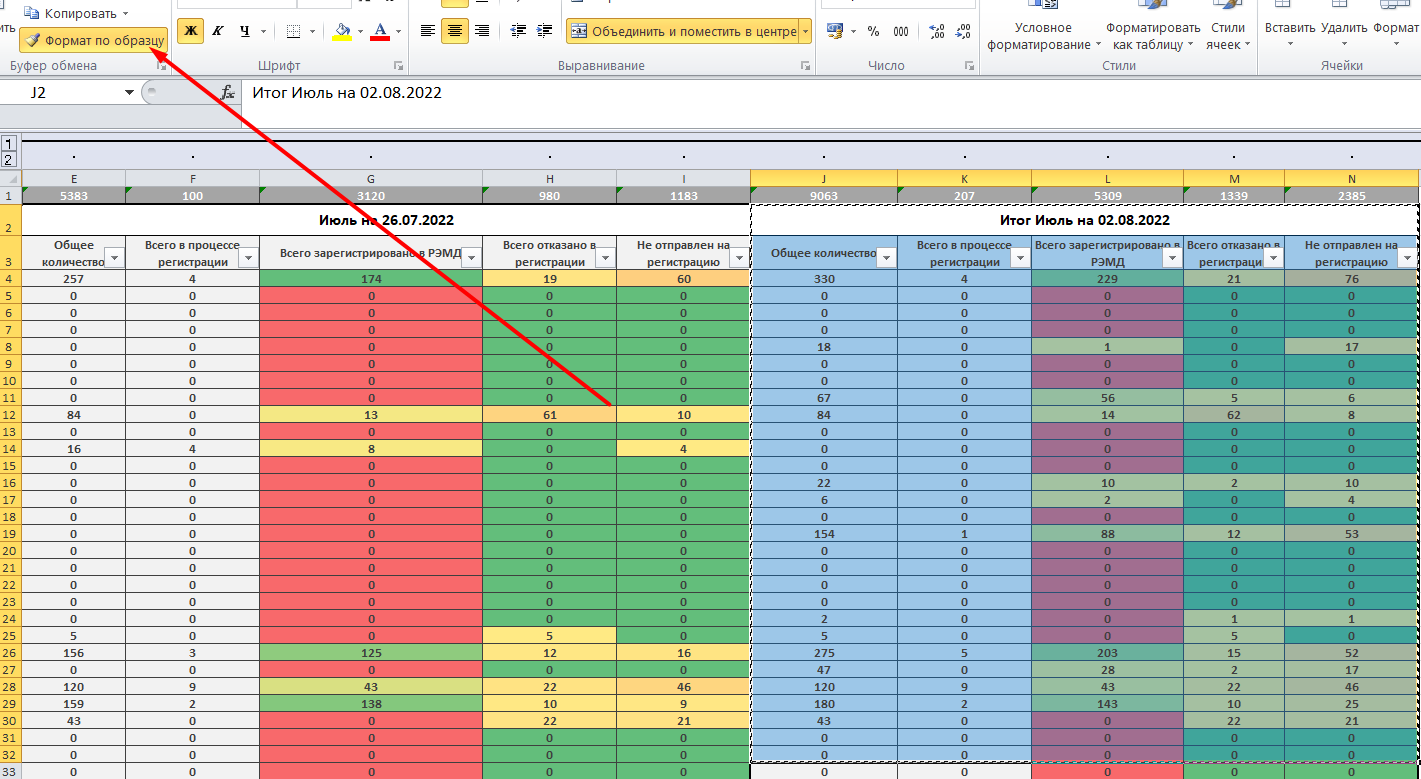


Рисунок 21 - выделение существующих данных и использование форматирования

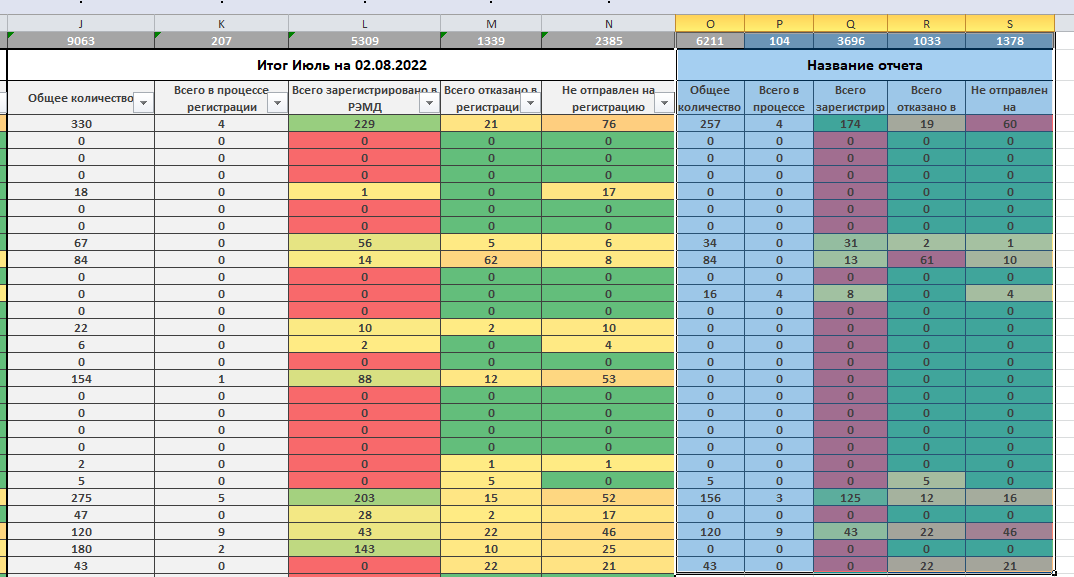


Рисунок 22 - результат использования форматирования

Скрипт полностью выполняет необходимое техническое задание. Теперь нам нужно собрать скрипт в полноценное exe приложение, которое будет использоваться пользователями.

# Сборка готового приложения и написание инструкции к использованию

Для сборки Python скрипта в exe файл я использовал библиотеку pyinstaller. PyInstaller анализирует структуру приложения на Python, определяет зависимости и автоматически включает все необходимые модули, библиотеки и файлы данных в итоговый исполняемый файл. Это позволяет упаковывать приложение в один файл, что облегчает его распространение и запуск на других компьютерах. Для упаковки в один файл потребуется аргумент --onefile. Запускаем упаковку командой pyinstaller --onefile main.py и ждем завершения работы. После завершения работы у нас создалось 2 папки: build и dist (Рисунок 23). Папка build нас не интересует, в ней хранятся файлы, которые pyinstaller использует при сборке приложения. В папке dist находится наш собранный exe файл.

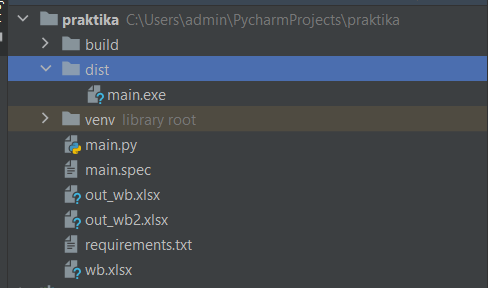


Рисунок 23 - структура проекта после сборки приложения

Перейдем в папку dist в проводнике, положим наши 2 таблицы wb.xlsx и out\_wb.xlsx в эту папку и запустим собранное приложение (Рисунок 24). В качестве названия используем «Отчет Проверка собранного приложения»

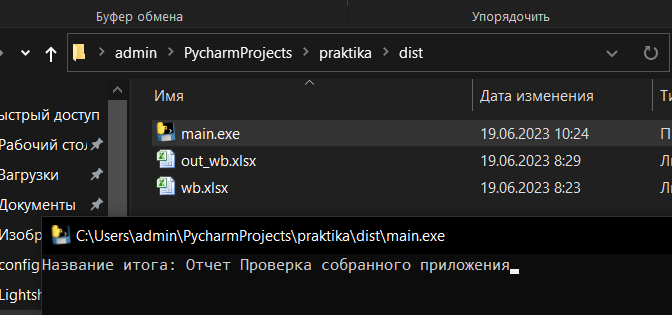


Рисунок 24 - запуск собранного приложения

После ввода названия консоль закрывается, и появляется файл out\_wb2.xlsx. Открываем и смотрим результат (Рисунок 25).

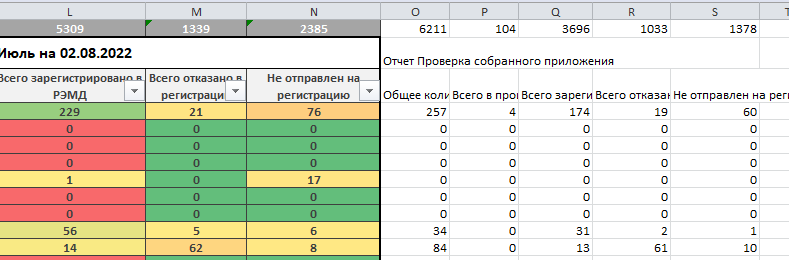


Рисунок 25 - результат работы приложения

Как и ожидалось – всё работает успешно, пользователю лишь останется отформатировать сгенерированные данные.

Инструкция к эксплуатации: