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**

**Киселева Артемия Кирилловича**

**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

Приложения

# 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.

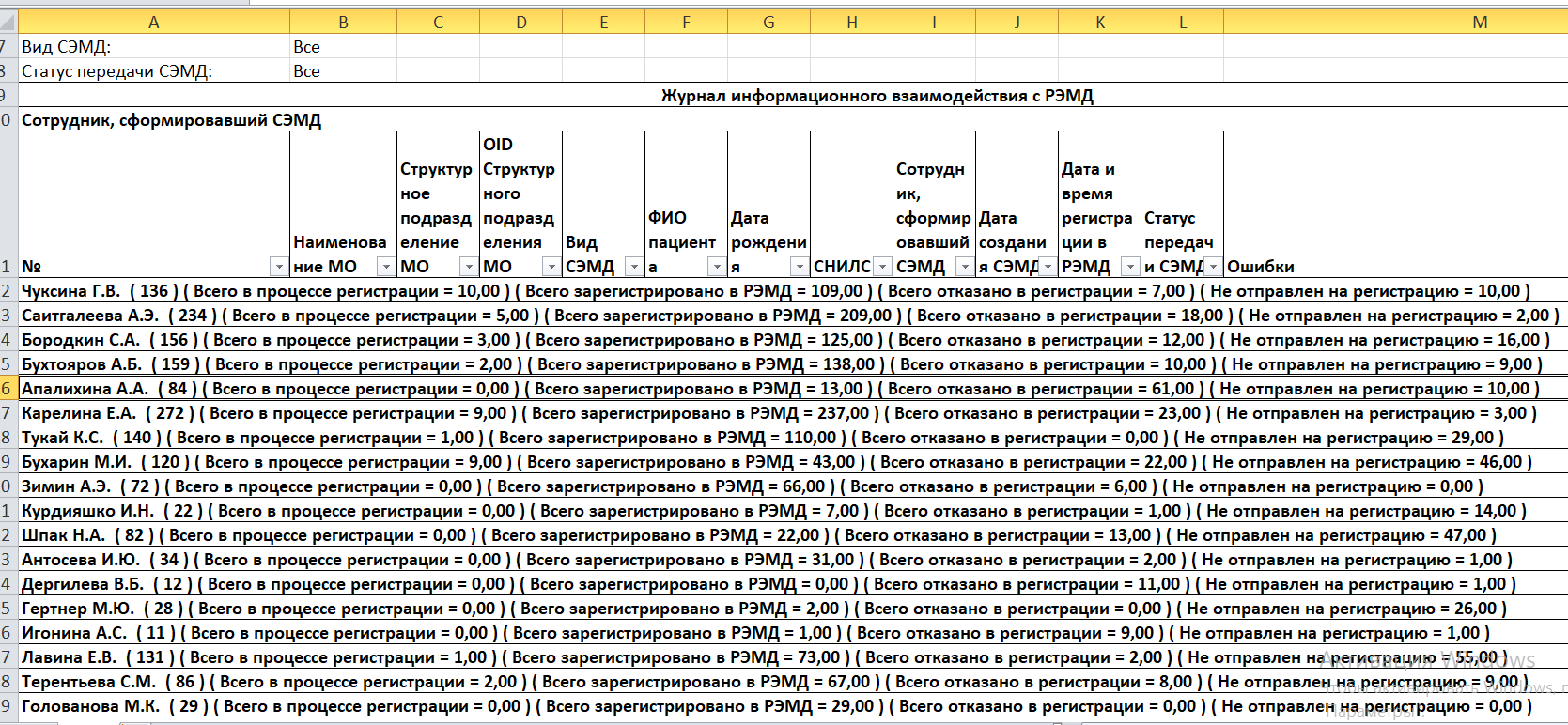


Рисунок 1 - данные по работникам городской больницы

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.

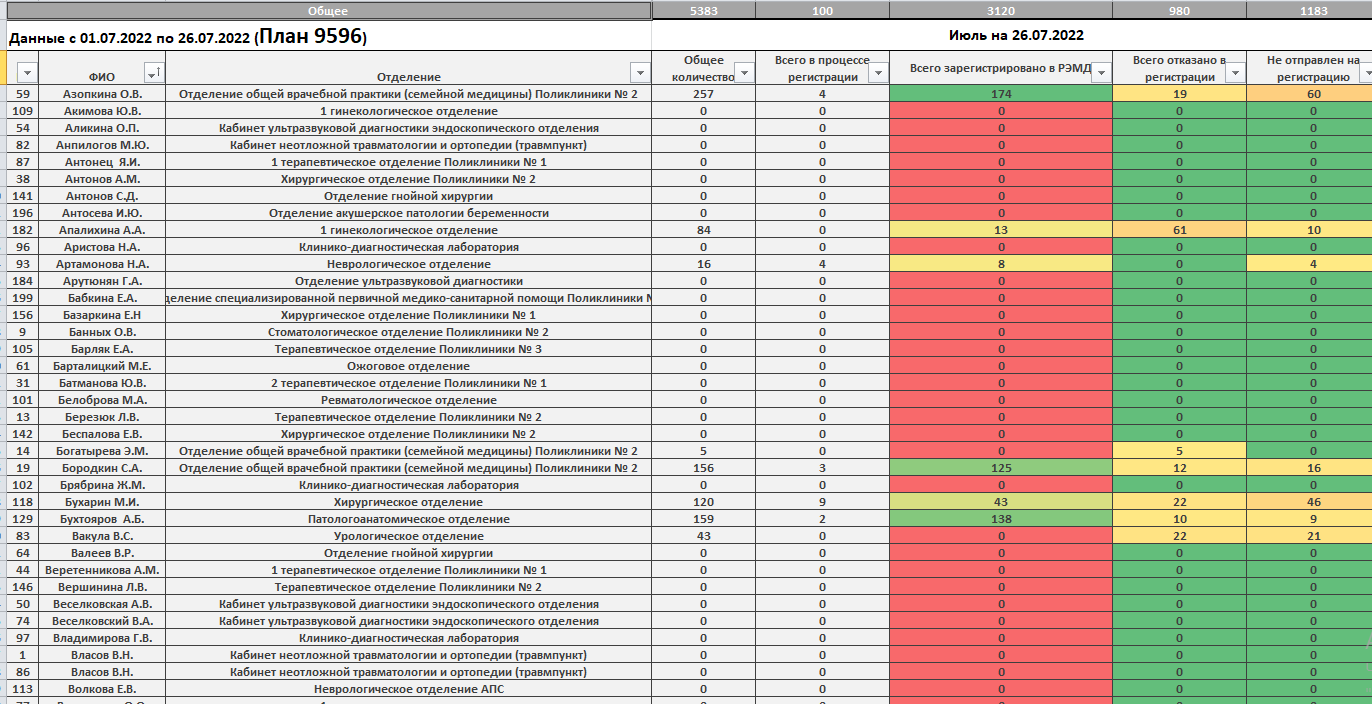


Рисунок 2 - таблица с существующими данными сотрудников

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)

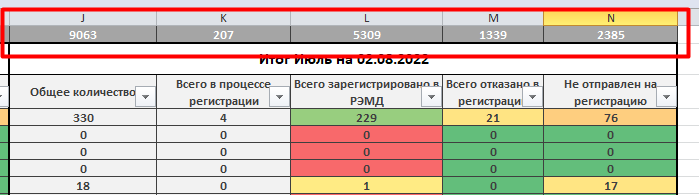


Рисунок 3 - сумма всех ячеек в столбце

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).

# 

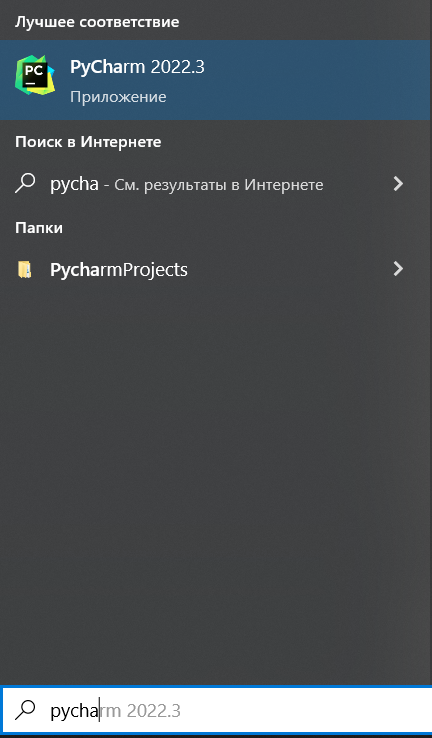


Рисунок 4 - поиск приложения в Windows

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)

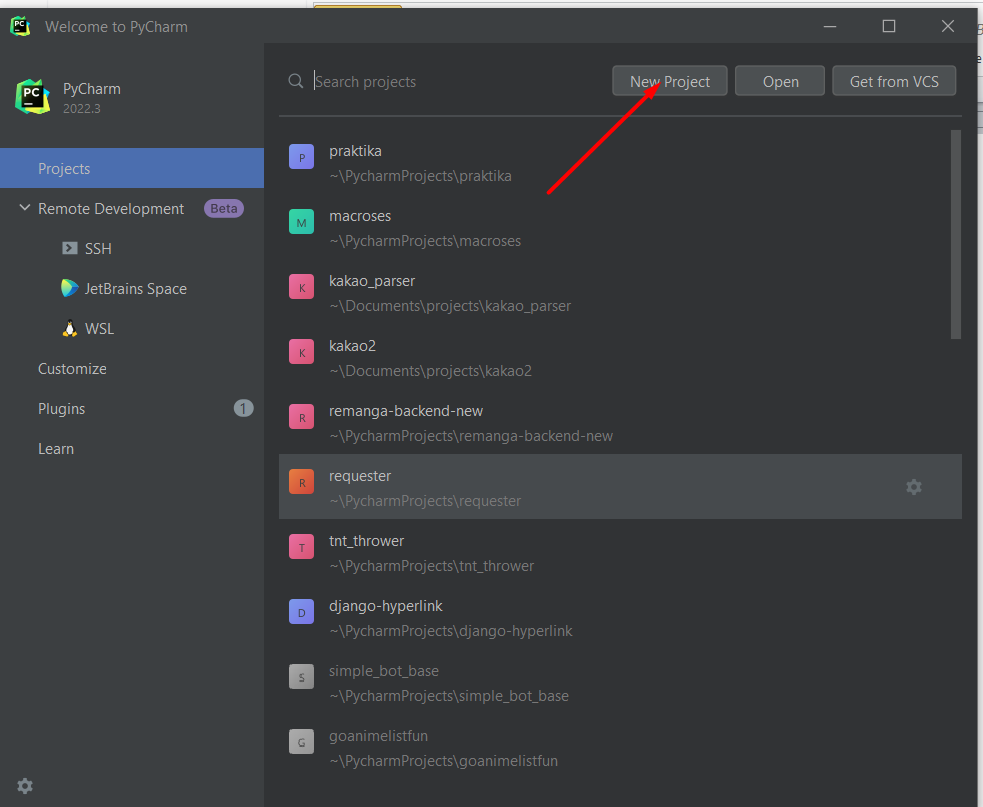


Рисунок 5 - окно выбора проектов PyCharm

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).

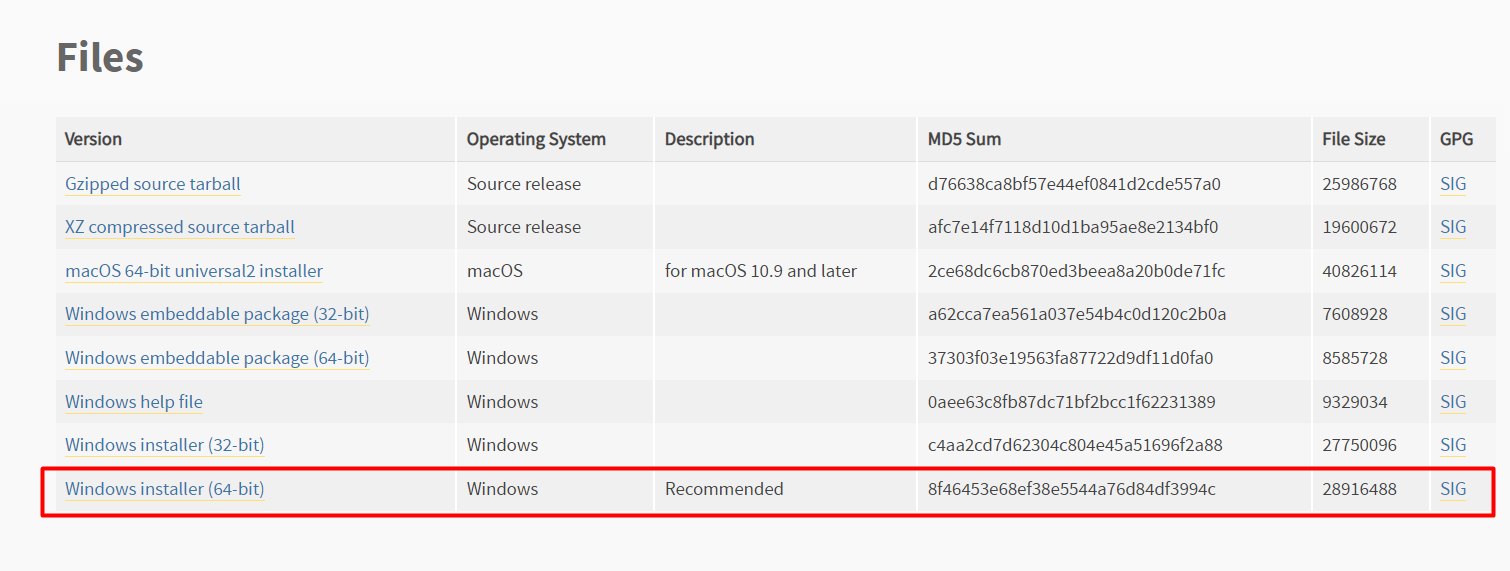


Рисунок 6 - файлы Python 3.10.6

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).

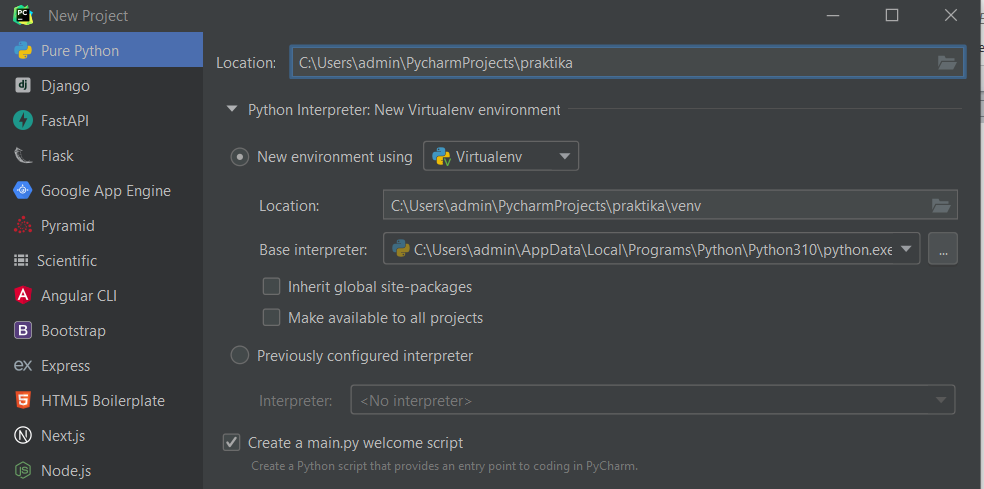


Рисунок 7 - создание нового проекта PyCharm

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)

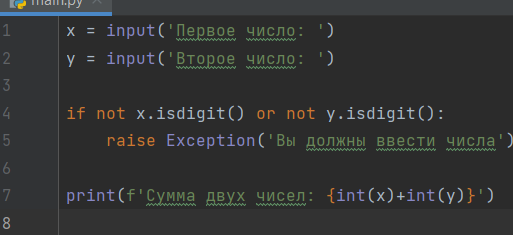


Рисунок 8 - простой скрипт на Python

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)

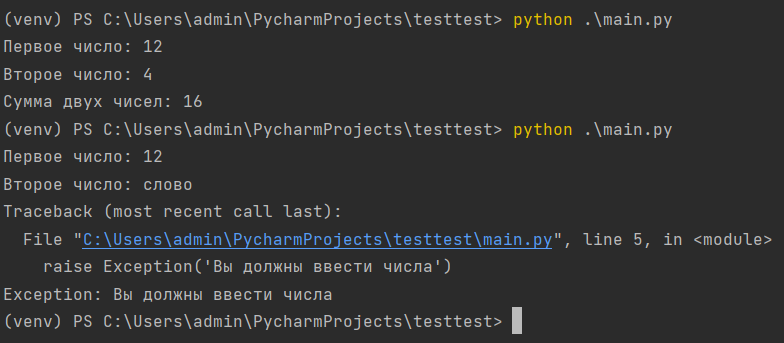


Рисунок 9 - Проверка работоспособности скрипта

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)

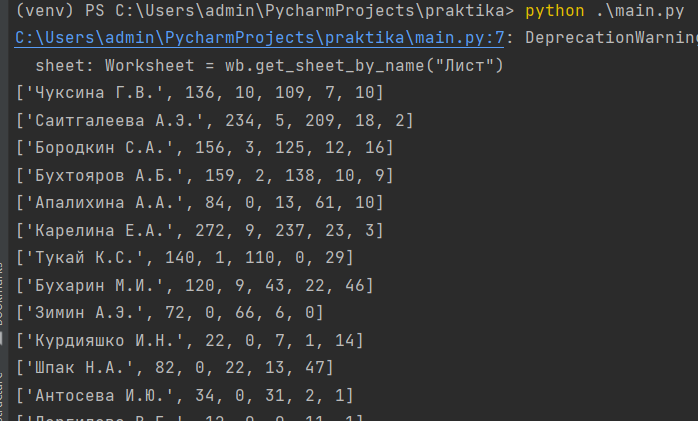


Рисунок 11 - полученные данные из таблицы

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)

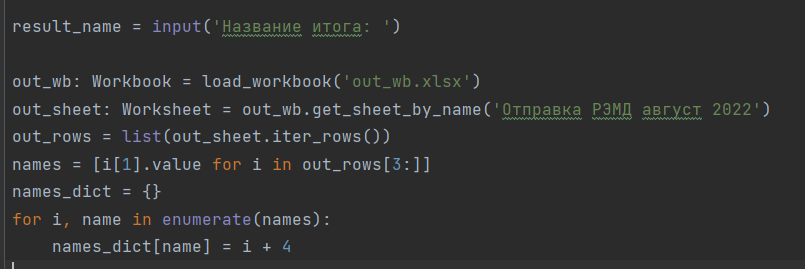


Рисунок 12 - создание словаря с именами и индексами строки

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)

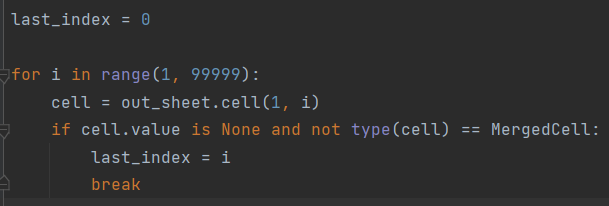


Рисунок 13 - алгоритм вычисления первой свободной ячейки

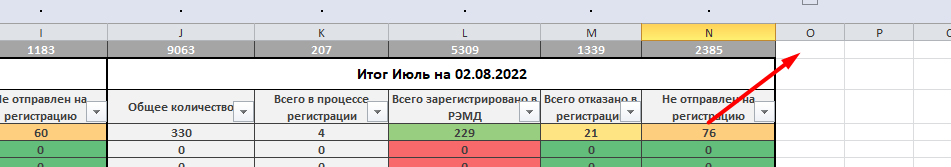


Рисунок 14 - первая свободная ячейка в таблице

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

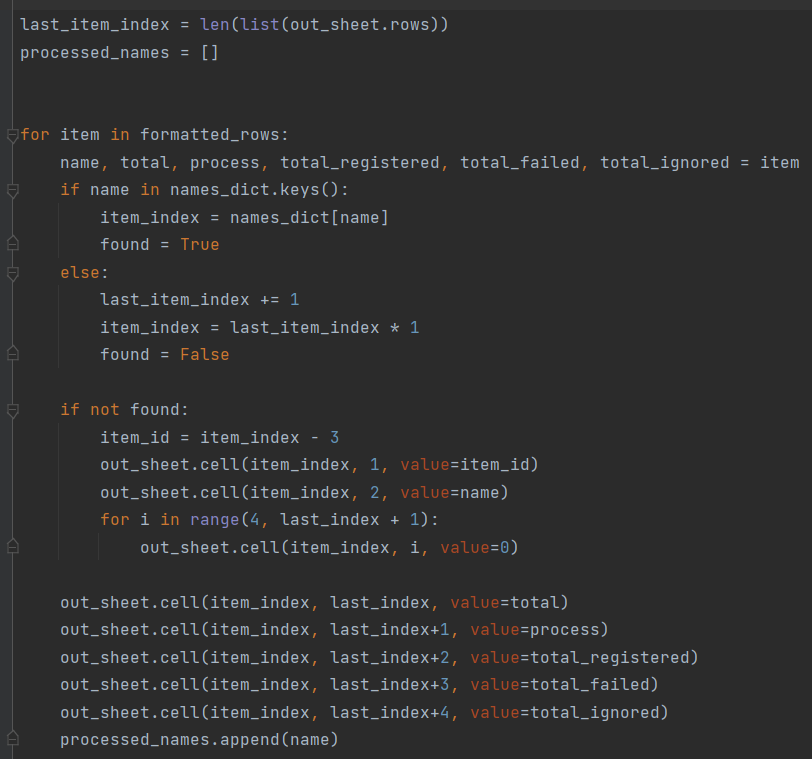


Рисунок 15 - алгоритм заполнения полученными данными

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)

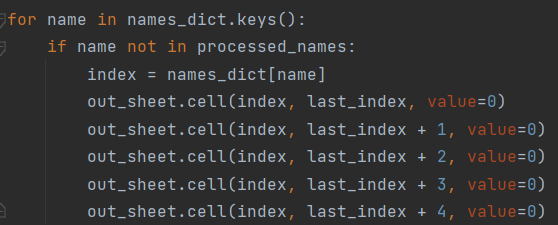


Рисунок 16 - заполнение пропущенных сотрудников

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)

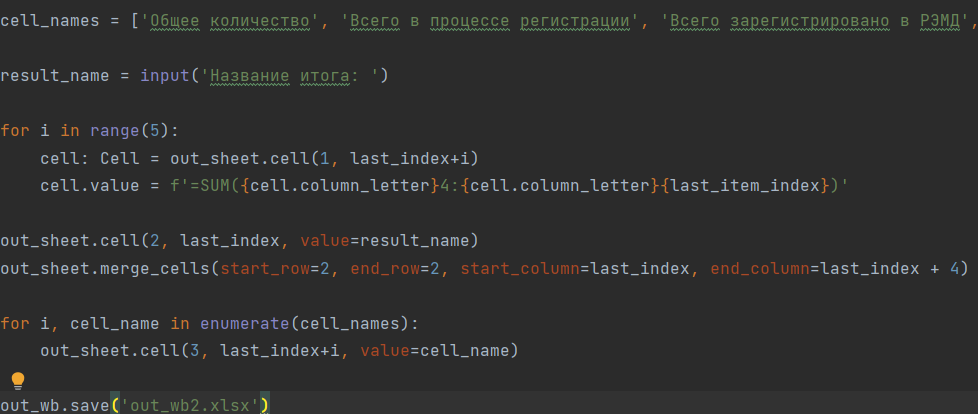


Рисунок 17 - завершение алгоритма и сохранение файла

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

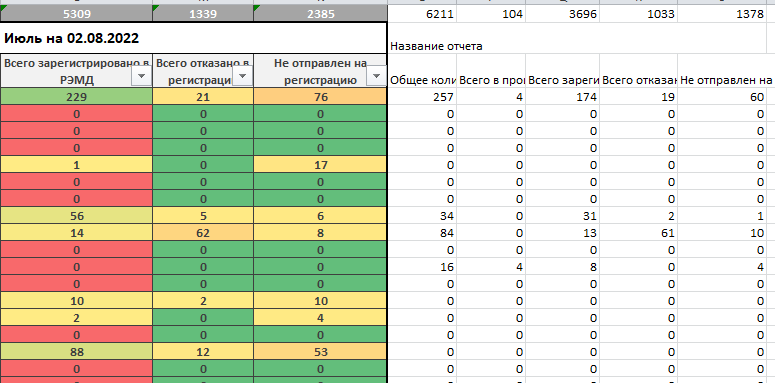


Рисунок 18 - новые полученные данные для существующих сотрудников



Рисунок 19 - новые сотрудники

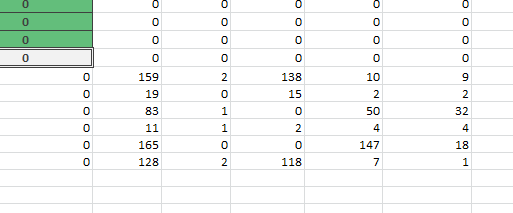


Рисунок 20 - данные новых сотрудников

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

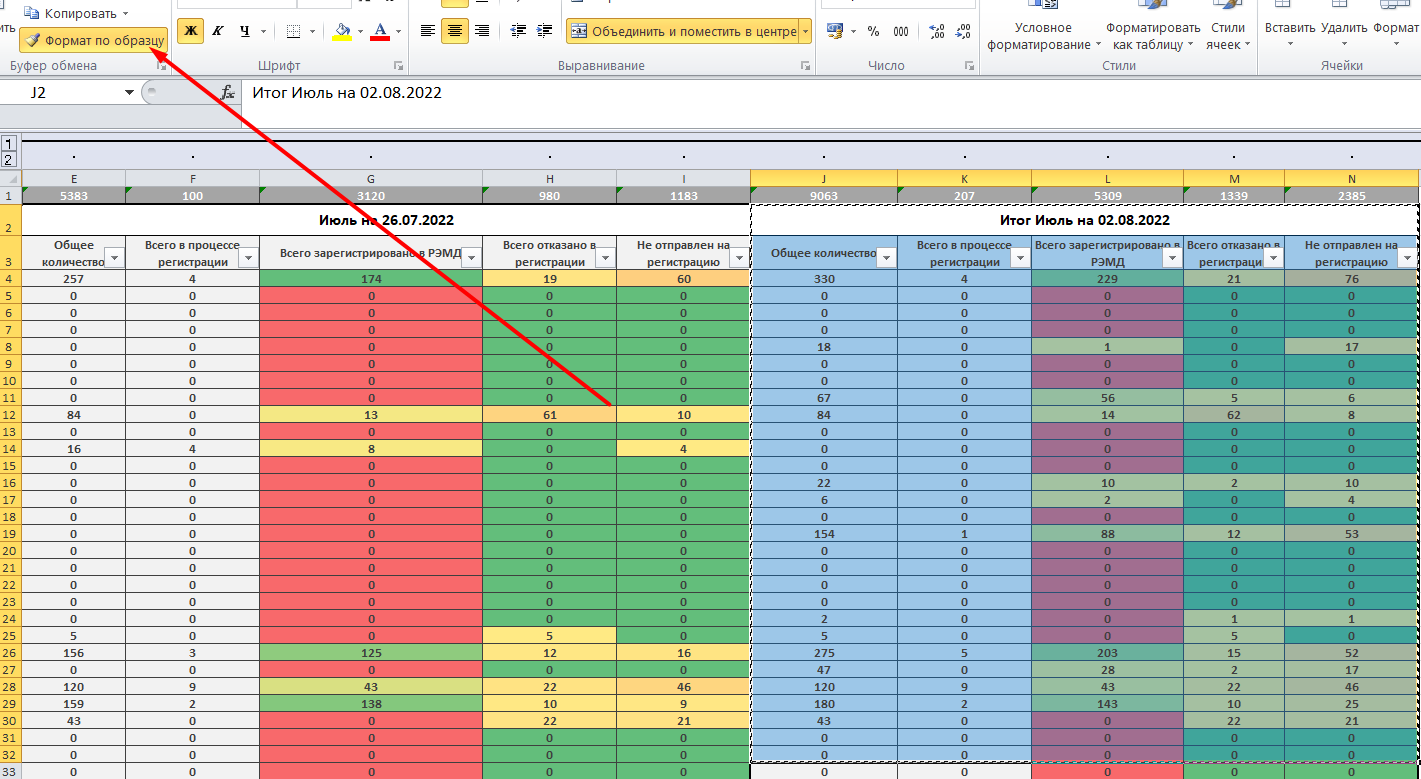


Рисунок 21 - выделение существующих данных и использование форматирования

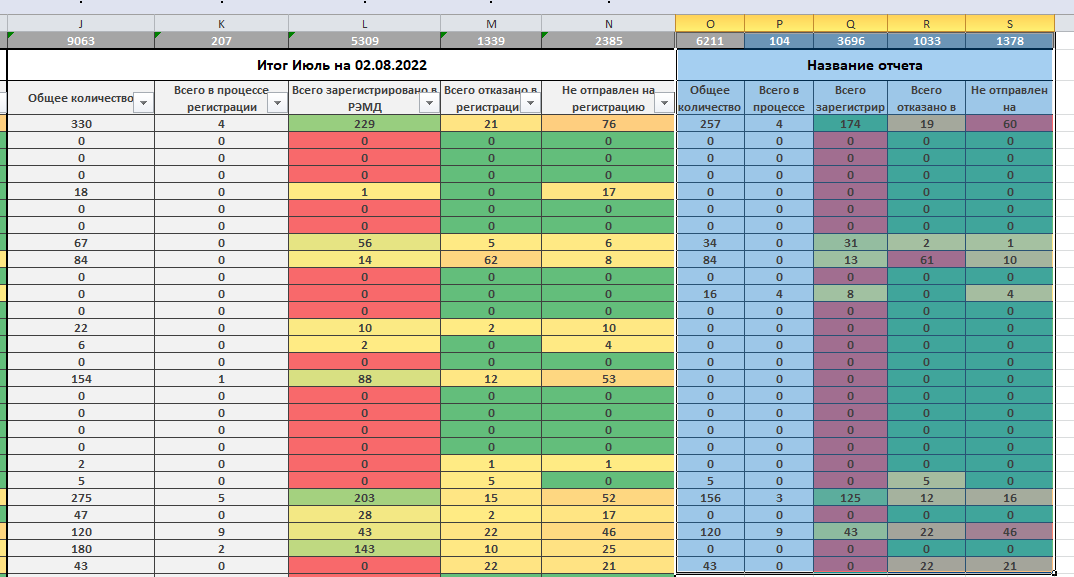


Рисунок 22 - результат использования форматирования

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.

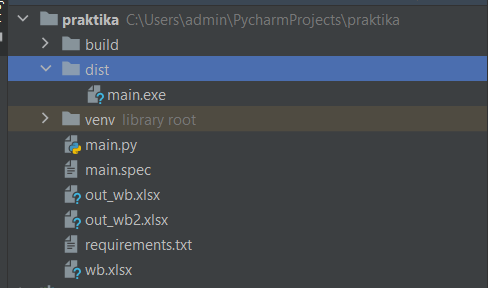


Рисунок 23 - структура проекта после сборки приложения

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»

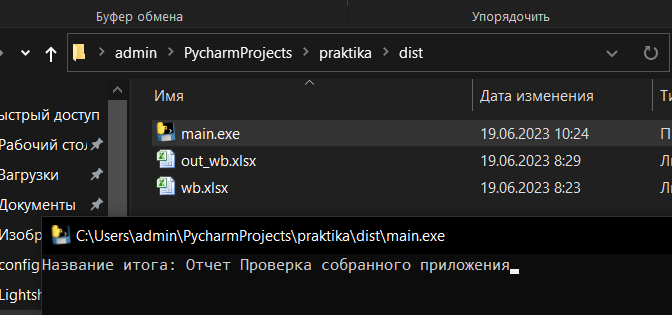


Рисунок 24 - запуск собранного приложения

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

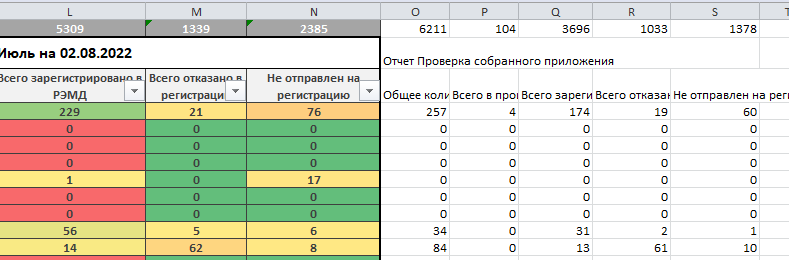


Рисунок 25 - результат работы приложения

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

Инструкция к эксплуатации:

* + - 1. Загрузить exe файл с github репозитория <https://github.com/nelttjen/collage-praktika/blob/main/dist/main.exe>
      2. Поместить в любую удобную папку
      3. Поместить таблицу с новыми данными под именем wb.xlsx в одну папку вместе с exe приложением. Лист с данными должен называться «Лист»
      4. Поместить таблицу с имеющимися данными по сотрудникам под именем out\_wb.xlsx в одну папку вместе с exe приложением. Лист с данными должен называться «Отправка РЭМД август 2022»
      5. Запустить файл main.exe
      6. Вписать нужное название, которое будет использовано как название отчета
      7. После работы скрипта файл с новыми данными появится в папке приложения с именем out\_wb2.xlsx. Данные будут в листе «Отправка РЭМД август 2022»

Техническое задание выполнено в полном необходимом объеме.

# Заключение

В результате производственной практики, были изучены материалы, по языку программирования Python и было создано приложение для быстрого сбора входных данных и представления их по готовому шаблону.

В ходе производственной практики (преддипломной) были выполнены поставленные задачи такие как:

* Инсталляции и настройки рабочего места для написания кода на языке программирования Python;
* Изучение языка программирования Python, его библиотек, синтаксиса и функциональности;
* Написание скрипта для быстрого сбора входных данных и представления их по готовому шаблону в соответствии с техническим заданием;
* Сборка скрипта в отдельное Standalone приложение для Windows, без необходимости дополнительной установки Python, библиотек и среды разработки;
* Написана документация к пользованию программы для использования программы пользователями;

Кроме того, студент написал отчет о производственной практике, в котором описывал свои задачи и поставленные цели, полученный результат и подводил итоги работы за период практики.

Таким образом, студент успешно выполнил поставленные задачи и овладел навыками работы с языком программирования Python, а также средой разработки PyCharm, быстро адаптировался к новой среде и умело применял свои знания для решения поставленных задач.

# 

# 

Студент группы СПМИС-202

Киселев Артемий Кириллович \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**(подпись)**

# Список литературы

1. Основы языка программирования Python за 10 минут <https://habr.com/ru/articles/31180/>
2. Документация библиотеки PyInstaller <https://pyinstaller.org/en/stable/index.html>
3. Использование Python и Excel для обработки и анализа данных. Часть 2: библиотеки для работы с данными <https://habr.com/ru/companies/otus/articles/331998/>
4. Материалы пользователей Американского интернет-форума Reddis <https://www.reddit.com/>
5. Ответы пользователей на ошибки и вопросы на международном кодинг-форуме Stack Overflow   
   <https://stackoverflow.com/questions/tagged/python>
6. Книга «Python, Django и PyCharm для начинающих» Анатолий Постолит  
   <https://www.litres.ru/book/anatoliy-postolit/python-django-i-pycharm-dlya-nachinauschih-67727943/>

**Приложение 1**

Код на языке программирования Python

from openpyxl import load\_workbook, Workbook  
from openpyxl.worksheet.worksheet import Worksheet  
from openpyxl.cell import Cell, MergedCell  
from typing import Union, List  
  
wb: Workbook = load\_workbook('wb.xlsx')  
sheet: Worksheet = wb.get\_sheet\_by\_name("Лист")  
rows = list(sheet.iter\_rows())[11:-1]  
formatted\_rows = []  
for row in rows:  
 row: List[Union[Cell, MergedCell]]  
 for cell in row:  
 if val := cell.value:  
 items = val.split("(")  
 items = [i.replace(")", '').strip() for i in items]  
 name, total, process, total\_registered, total\_failed, total\_ignored = items  
  
 total = int(total)  
 process = int(float(process.split(" = ")[1].replace(",", '.')))  
 total\_registered = int(float(total\_registered.split(" = ")[1].replace(",", '.')))  
 total\_failed = int(float(total\_failed.split(" = ")[1].replace(",", '.')))  
 total\_ignored = int(float(total\_ignored.split(" = ")[1].replace(",", '.')))  
 formatted\_rows.append([name, total, process, total\_registered, total\_failed, total\_ignored])  
  
out\_wb: Workbook = load\_workbook('out\_wb.xlsx')  
out\_sheet: Worksheet = out\_wb.get\_sheet\_by\_name('Отправка РЭМД август 2022')  
out\_rows = list(out\_sheet.iter\_rows())  
names = [i[1].value for i in out\_rows[3:]]  
names\_dict = {}  
for i, name in enumerate(names):  
 names\_dict[name] = i + 4  
  
last\_index = 0  
  
for i in range(1, 99999):  
 cell = out\_sheet.cell(1, i)  
 if cell.value is None and not type(cell) == MergedCell:  
 last\_index = i  
 break  
  
last\_item\_index = len(list(out\_sheet.rows))  
processed\_names = []  
  
  
for item in formatted\_rows:  
 name, total, process, total\_registered, total\_failed, total\_ignored = item  
 if name in names\_dict.keys():  
 item\_index = names\_dict[name]  
 found = True  
 else:  
 last\_item\_index += 1  
 item\_index = last\_item\_index \* 1  
 found = False  
  
 if not found:  
 item\_id = item\_index - 3  
 out\_sheet.cell(item\_index, 1, value=item\_id)  
 out\_sheet.cell(item\_index, 2, value=name)  
 for i in range(4, last\_index + 1):  
 out\_sheet.cell(item\_index, i, value=0)  
  
 out\_sheet.cell(item\_index, last\_index, value=total)  
 out\_sheet.cell(item\_index, last\_index+1, value=process)  
 out\_sheet.cell(item\_index, last\_index+2, value=total\_registered)  
 out\_sheet.cell(item\_index, last\_index+3, value=total\_failed)  
 out\_sheet.cell(item\_index, last\_index+4, value=total\_ignored)  
 processed\_names.append(name)  
  
for name in names\_dict.keys():  
 if name not in processed\_names:  
 index = names\_dict[name]  
 out\_sheet.cell(index, last\_index, value=0)  
 out\_sheet.cell(index, last\_index + 1, value=0)  
 out\_sheet.cell(index, last\_index + 2, value=0)  
 out\_sheet.cell(index, last\_index + 3, value=0)  
 out\_sheet.cell(index, last\_index + 4, value=0)  
  
cell\_names = ['Общее количество', 'Всего в процессе регистрации', 'Всего зарегистрировано в РЭМД', 'Всего отказано в регистрации', 'Не отправлен на регистрацию']  
  
result\_name = input('Название итога: ')  
  
for i in range(5):  
 cell: Cell = out\_sheet.cell(1, last\_index+i)  
 cell.value = f'=SUM({cell.column\_letter}4:{cell.column\_letter}{last\_item\_index})'  
  
out\_sheet.cell(2, last\_index, value=result\_name)  
out\_sheet.merge\_cells(start\_row=2, end\_row=2, start\_column=last\_index, end\_column=last\_index + 4)  
  
for i, cell\_name in enumerate(cell\_names):  
 out\_sheet.cell(3, last\_index+i, value=cell\_name)  
  
out\_wb.save('out\_wb2.xlsx')

**Приложение 2**

Exe файл собранного приложения

<https://github.com/nelttjen/collage-praktika/blob/main/dist/main.exe>

**Приложение 3**

Таблицы с данными от руководителя практики

<https://github.com/nelttjen/collage-praktika/tree/main/source_tables>

АТТЕСТАЦИОННЫЙ ЛИСТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ

Киселева Артемия Кирилловича

Обучающийся на 3 курсе по специальности 09.02.04 Информационные системы (по отраслям) успешно прошел производственную практику по профессиональному модулю в объеме 144 часа.

С «25» мая 2023 г. по «21» июня 2023 г. ГБ Миасс №2.

**Виды и качество выполнения работ**

|  |  |
| --- | --- |
| Виды и объем работ, выполненных обучающимся во время практики | Качество выполнения работ в соответствии с технологией и (или) требованиями организации, в которой проходила практика (соответствует / не соответствует) Нужное подчеркнуть |
| Задание 1. Введение. | соответствует/ не соответствует |
| вводный инструктаж; соответствует/ не соответствует | соответствует/ не соответствует |
| анализ технического задания; | соответствует/ не соответствует |
| сбор информации о предстоящей работе; | соответствует/ не соответствует |
| установка и настройка среды разработки на языке программирования Python; | соответствует/ не соответствует |
| Задание 2. Выполнение технического задания в полном объеме. | соответствует/ не соответствует |
| установка необходимых библиотек для выполнения задания; | соответствует/ не соответствует |
| разработка скрипта сбора входных данных; | соответствует/ не соответствует |
| разработка скрипта вывода собранных данных по готовому шаблону; | соответствует/ не соответствует |
| Задание 3. Сборка приложения и написание инструкции | соответствует/ не соответствует |
| использование библиотеки для сборки итогового приложения; | соответствует/ не соответствует |
| тестирование собранного приложения; | соответствует/ не соответствует |
| написание инструкции к использованию приложения | соответствует/ не соответствует |
| устранение замечаний пользователя; | соответствует/ не соответствует |

|  |  |  |
| --- | --- | --- |
| **Сформированность общих компетенций обучающегося по результатам практики** | | |
| Код | Наименование компетенций | Уровень сформированности компетенций (уровни: низкий, средний, высокий)\*  Нужное подчеркнуть |
| ОК1 | Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес | низкий, средний, высокий |
| ОК2 | Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество. | низкий, средний, высокий |
| ОК3 | Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность. | низкий, средний, высокий |
| ОК4 | Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития. | низкий, средний, высокий |
| ОК5 | Использовать информационно-коммуникационные технологии в профессиональной деятельности. | низкий, средний, высокий |
| ОК6 | Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями. | низкий, средний, высокий |
| ОК7 | Брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий. | низкий, средний, высокий |
| ОК8 | Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации | низкий, средний, высокий |
| ОК9 | Ориентироваться в условиях частой смены технологий в профессиональной деятельности. | низкий, средний, высокий |

\*

√ Высокий уровень компетенций: работает полностью самостоятельно;

√ Средний уровень компетенций: работает самостоятельно, но иногда обращается к наставнику за помощью;

√ Низкий уровень компетенций: нуждается в помощи и (или) контроле.

Дата «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Подпись руководителя практики

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ А. К. Лихачев /

Подпись ответственного лица организации (базы практики)

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

**Характеристика**

За время прохождения практики с «25» мая 2023 г. по «21» июня 2023 г.

Студент Киселев Артемий Кириллович проходил учебно-ознакомительную практику в ГБ Миасс №2, расположенном по адресу: г. Миасс, ул. Ильменская 81, корп. 1.

В течение всего периода практики, Киселев Артемий Кириллович внимательно и ответственно относился к выполняемой работе. Понял и проанализировал техническое задание, прописал план работы для его выполнения. Настроил среду разработки, разработал нужный скрипт, собрал его в приложение и написал к нему инструкцию.

Всю порученную работу выполнял добросовестно и в срок. Стремился приобретать новые знания, чтобы быть еще более полезным на месте практики. Неоднократно оказывал помощь сотрудникам организации. Руководство организации оценивают работу Киселева Артемия Кирилловича на «отлично».

За время прохождения производственной практики студент освоил (не освоил)

следующие профессиональные компетенции

|  |  |  |
| --- | --- | --- |
| **Коды и название профессиональных компетенций (ПК)** | **Освоил** | **Не освоил** |
| ПК 1.1. Анализ и построение графика работы по техническому заданию, полученному от руководителя практики |  |  |
| ПК 1.2. Взаимодействовать со специалистами смежного профиля при разработке методов, средств и технологий применения объектов профессиональной деятельности |  |  |
| ПК 1.3. Разбираться в установке необходимого программного обеспечения для выполнения технического задания в полном объеме |  |  |
| ПК 1.4. Изучать новый материал, необходимый для выполнения технического задания в полном объеме |  |  |
| ПК 1.5. Разрабатывать фрагменты документации по  эксплуатации разработанного продукта |  |  |
| ПК 1.6. Участвовать в оценке качества и технической  эффективности разработанного продукта |  |  |
| ПК 1.7. Производить тестирование конечного продукта на наличие ошибок, которые могут негативно повлиять на опыт пользователя при использовании продукта |  |  |
| ПК 1.8. Консультировать пользователей разработанного продукта и разрабатывать фрагменты методики обучения  пользователей разработанного продукта |  |  |
| ПК 1.9. Выполнять регламенты по обновлению,  техническому сопровождению и восстановлению данных  разработанного продукта  документацией |  |  |
| ПК 1.10. Обеспечивать организацию доступа  пользователей к разработанному продукту |  |  |

Руководитель практики от организации

Сидоров Иван Сергеевич

Дата «\_\_\_»\_\_\_\_\_\_\_\_\_2023 г. Подпись \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Зачет по практике принят

Кирилл Андреевич Лихачёв

Дата «\_\_\_»\_\_\_\_\_\_\_\_\_2023г. Подпись \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент освоил (не освоил) основной вид профессиональной деятельности (ВПД)

Руководитель практики от организации

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

Руководитель практики от учебного заведения

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ А. К. Лихачев