Приведен код создания программы Python для автоматизации ручного процесса сбора и сортировки файлов Excel для отправки их по электронной почте сотням клиентов.

Есть бизнес-потребность в рассылке электронных писем с вложениями Excel по списку из большого количества клиентов. С помощью Python можно автоматизировать весь процесс и сэкономить время. Этот код хорошо справляется с автоматизацией ручного процесса, вместо того, чтобы много копировать, вставлять и манипулировать файлами вручную.

**import datetime**

**import os**

**import shutil**

**from pathlib import Path**

**import pandas as pd**

**import win32com.client as win32**

**## Установить формат даты**

**today\_string = datetime.datetime.today().strftime('%m%d%Y\_%I%p')**

**today\_string2 = datetime.datetime.today().strftime('%b %d, %Y')**

**## Установка имен папок для вложений и архивирования**

**attachment\_path = Path.cwd() / 'data' / 'attachments'**

**archive\_dir = Path.cwd() / 'archive'**

**src\_file = Path.cwd() / 'data' / 'Example4.xlsx'**

**df = pd.read\_excel(src\_file)**

**df.head()**

**customer\_group = df.groupby('CUSTOMER\_ID')**

**for ID, group\_df in customer\_group:**

**print(ID)**

**## Запишите каждый идентификатор, группу в отдельные файлы Excel и используйте идентификатор,**

**## чтобы назвать каждый файл с сегодняшней датой**

**attachments = []**

**for ID, group\_df in customer\_group:**

**attachment = attachment\_path / f'{ID}\_{today\_string}.xlsx'**

**group\_df.to\_excel(attachment, index=False)**

**attachments.append((ID, str(attachment)))**

**df2 = pd.DataFrame(attachments, columns=['CUSTOMER\_ID', 'FILE'])**

**email\_merge = pd.merge(df, df2, how='left')**

**combined = email\_merge[ ['CUSTOMER\_ID', 'EMAIL', 'FILE'] ].drop\_duplicates()**

**# Отправка индивидуальных отчетов по электронной почте соответствующим получателям**

**class EmailsSender:**

**def \_\_init\_\_(self):**

**self.outlook = win32.Dispatch('outlook.application')**

**def send\_email(self, to\_email\_address, attachment\_path):**

**mail = self.outlook.CreateItem(0)**

**mail.To = to\_email\_address**

**mail.Subject = today\_string2 + ' Report'**

**mail.Body = """Please find today's report attached."""**

**mail.Attachments.Add(Source=attachment\_path)**

**# Используйте это, чтобы показать электронную почту**

**#mail.Display(True)**

**# Раскомментировать для отправки**

**#mail.Send()**

**email\_sender = EmailsSender()**

**for index, row in combined.iterrows():**

**email\_sender.send\_email(row['EMAIL'], row['FILE'])**

**# Переместить файлы в архив**

**for f in attachments:**

**shutil.move(f[1], archive\_dir)**