

You are a helpful assistant. You can help me by answering my questions. You can also ask me questions.

model: unknown, word count: 19, token estimate: 22

```
def start_processing(entry, input_root):
    folder = entry.get()
    if not folder:
        messagebox.showerror("错误", "请先选择或输入一个文件夹路径")
        return

    pdf_files = get_pdf_files(folder)
    # print(pdf_files)

    values = []
    renamed_pdf_files = []
    for index, pdf_file in enumerate(pdf_files):
        pdf_text = read_pdf_content(pdf_file)
        pattern = r'[¥¥](\d+(?:\.\d{1,2})?)'
        result = re.findall(pattern, pdf_text)
        # print(result)

        if result:
            max_value = max([float(i) for i in result])
            values.append(max_value)
            new_pdf_file = rename_pdf_file(pdf_file, max_value, index)
            new_pdf_file = new_pdf_file.replace("\\", "/")
            renamed_pdf_files.append(new_pdf_file)

    total_amount = sum(values)
    display_results(values, total_amount, renamed_pdf_files, input_root, folder)

def main():
    root = tk.Tk()
    center_window(root)
    root.withdraw()
    root.title("发票金额统计")

    root.iconbitmap(icon_path)

    main_frame = ttk.Frame(root, padding="10 10 10 10")
    main_frame.grid(column=0, row=0, sticky=(tk.W, tk.E, tk.N, tk.S))

    folder_entry = ttk.Entry(main_frame, width=40)
    folder_entry.grid(column=0, row=0, padx=5, pady=5, sticky=(tk.W, tk.E))

    browse_button = ttk.Button(main_frame, text="打开", command=lambda: browse_folder(folder_entry))
    browse_button.grid(column=1, row=0, padx=5, pady=5, sticky=tk.W)

    start_button = ttk.Button(main_frame, text="开始处理", command=lambda: start_processing(folder_entry, root))
    start_button.grid(column=1, row=1, padx=5, pady=5, sticky=tk.W)

    label = ttk.Label(main_frame, text="请输入发票所在文件夹...", )
    label.grid(column=0, row=1, padx=4, pady=4, sticky=tk.W)

    def change_label_text(event):
        if folder_entry.get():
            label.config(text="正在处理发票 ... ")

    start_button.bind("<Button-1>", change_label_text)

    # 调整 folder_entry 组件的宽度以适应窗口大小
    def resize_folder_entry(event):
        new_width = event.width
        folder_entry.configure(width=int(new_width * 0.9 / 10))

    root.bind("<Configure>", resize_folder_entry)

    main_frame.columnconfigure(0, weight=1)
    # main_frame.rowconfigure(0, weight=1)
    root.columnconfigure(0, weight=1)
    # root.rowconfigure(0, weight=1)

    # 显示窗口
    root.deiconify()
    root.mainloop()

我希望 r'¥¥' 这个表达式可能在窗口中输入
```

model: unknown, word count: 337, token estimate: 1236

如果您希望用户在窗口中输入正则表达式，可以添加一个输入框（`ttk.Entry`）和一个标签（`ttk.Label`）来提示用户输入正则表达式。以下是修改后的代码：

PYTHON

```
def main():
    root = tk.Tk()
    center_window(root)
    root.withdraw()
    root.title("发票金额统计")

    root.iconbitmap(icon_path)

    main_frame = ttk.Frame(root, padding="10 10 10 10")
    main_frame.grid(column=0, row=0, sticky=(tk.W, tk.E, tk.N, tk.S))

    folder_entry = ttk.Entry(main_frame, width=40)
    folder_entry.grid(column=0, row=0, padx=5, pady=5, sticky=(tk.W, tk.E))

    browse_button = ttk.Button(main_frame, text="打开", command=lambda: browse_folder(folder_entry))
    browse_button.grid(column=1, row=0, padx=5, pady=5, sticky=tk.W)

    start_button = ttk.Button(main_frame, text="开始处理", command=lambda: start_processing(folder_entry, root))
    start_button.grid(column=1, row=1, padx=5, pady=5, sticky=tk.W)

    label = ttk.Label(main_frame, text="请输入发票所在文件夹...", )
    label.grid(column=0, row=1, padx=4, pady=4, sticky=tk.W)

    # 添加正则表达式输入框和标签
    regex_entry = ttk.Entry(main_frame, width=40)
    regex_entry.grid(column=0, row=2, padx=5, pady=5, sticky=(tk.W, tk.E))
    regex_label = ttk.Label(main_frame, text="请输入正则表达式: ")
    regex_label.grid(column=0, row=3, padx=5, pady=5, sticky=tk.W)

    def change_label_text(event):
        if folder_entry.get():
            label.config(text="正在处理发票 ... ")

    start_button.bind("<Button-1>", change_label_text)

    # 调整 folder_entry 组件的宽度以适应窗口大小
    def resize_folder_entry(event):
        new_width = event.width
        folder_entry.configure(width=int(new_width * 0.9 / 10))

    root.bind("<Configure>", resize_folder_entry)

    main_frame.columnconfigure(0, weight=1)
    root.columnconfigure(0, weight=1)

    # 显示窗口
    root.deiconify()
    root.mainloop()
```

复制

现在，您可以在 `start_processing` 函数中使用 `regex_entry.get()` 获取用户输入的正则表达式。例如：

PYTHON

```
def start_processing(entry, input_root):
    # ...
    regex_pattern = regex_entry.get()
    if not regex_pattern:
        messagebox.showerror("错误", "请先输入一个正则表达式")
        return
    # 使用用户输入的正则表达式
    result = re.findall(regex_pattern, pdf_text)
    # ...
```

复制

model: gpt-4, word count: 405, token estimate: 1266