```python
import pandas as pd
from docx import Document
from docx.enum.table import WD_TABLE_ALIGNMENT
from docx.enum.text import WD_PARAGRAPH_ALIGNMENT
from docx.oxml.ns import qn
from docx.shared import Pt
from openpyxl import load_workbook

# 处理日期
# 打开文件


workbook = load_workbook(filename='日化.xlsx', data_only=True)
# 选择工作表
ws = workbook['销售订单表']
nrows = ws.max_row
# 遍历第二列单元格
for i in range(2, nrows + 1):
    # 处理单元格数据
    if isinstance(ws.cell(i, 2).value, str):
        print(ws.cell(i, 2).value)
        ws.cell(i, 2).value = ws.cell(i, 2).value.replace("#", "/")

# 保存文件
workbook.save(filename='日化_1.xlsx')
# 关闭文件
workbook.close()

# 读取"销售订单表"到DataFrame
df_dd = pd.read_excel('日化_1.xlsx', sheet_name='销售订单表')
print(df_dd.head(20))

# 读取"商品信息表"到DataFrame
df_xx = pd.read_excel('日化_1.xlsx', sheet_name='商品信息表')
print(df_xx.head(20))

# 处理缺失值
print(df_dd)
df_dd.dropna(inplace=True)
print(df_dd)

# 根据商品编号，增加商品小类列
dict_xx = dict()
for index, row in df_xx.iterrows():
    print(row['商品编号'], row['商品小类'])
    dict_xx[row['商品编号']] = row['商品小类']
print(dict_xx)
df_dd['商品小类'] = df_dd['商品编号'].replace(dict_xx)
print(df_dd.head(20))

# 排序，按商品小类排序也可以
df_dd.sort_values(by=['所在地市', '商品编号'], inplace=True)
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
print(df_dd.head(200))

# 查看分组后的信息，此操作是为了便于同学们理解，可以直接遍历df_dd输出到word
print(df_dd.groupby(['所在地市', '商品编号']).sum())
print((df_dd.groupby(['所在地市', '商品编号']).sum()).describe())
# 保存到文件，此操作是为了便于同学们理解，可以直接遍历df_dd输出到word
df_dd.to_excel('日化_2.xlsx', index=False)

# 创建word文档
doc = Document()
# 设置正文字体
doc.styles['Normal'].font.name = '宋体'
doc.styles['Normal']._element.rPr.rFonts.set(qn('w:eastAsia'), '宋体')


# 定义段落样式设置函数
def set_paragraph_style(paragraph):
    # 设置段落对齐方式
    paragraph.alignment = WD_PARAGRAPH_ALIGNMENT.LEFT
    # 设置段落中第一个运行(run) 字号为12
    paragraph.runs[0].font.size = Pt(12)
    # 设置字体
    paragraph.runs[0].name = '宋体'
    # 设置首行缩进为2个字符，每个字符字号为12
    paragraph.paragraph_format.first_line_indent = Pt(24)
    # 设置段落的段前间距
    paragraph.paragraph_format.space_before = Pt(18)
    # 设置段落的段后间距
    paragraph.paragraph_format.space_after = Pt(18)


row1 = None
```

```python
list_rows = []
total_price = 0
for i, (index, row2) in enumerate(df_dd.iterrows()):
    index, row2 = (index, row2)
    print(i)
    if (row1 is None):
        row1 = row2
        list_rows.append(row1)
        total_price += row1['金额']

    else:
        if row2['商品小类'] == row1['商品小类']:
            list_rows.append(row2)
            total_price += row2['金额']
        if row2['商品小类'] != row1['商品小类']:

            # 添加标题
            heading = doc.add_heading(row1['所在地市'] + row1['商品小类'] + '销售表', level=0)
            # 标题居中显示
            heading.alignment = WD_PARAGRAPH_ALIGNMENT.CENTER

            table = doc.add_table(rows=len(list_rows) + 1, cols=4)
            table.style = 'Table Grid'

            # 设置表头行
            table_header = table.rows[0]
            table_header.cells[0].text = '订单日期'
            table_header.cells[1].text = '商品编号'
            table_header.cells[2].text = '订购数量'
            table_header.cells[3].text = '订购单价'

            # 设置表头行单元格内文本样式
            for cell in table_header.cells:
                paragraph = cell.paragraphs[0]
                paragraph_format = paragraph.paragraph_format
                paragraph_format.alignment = WD_TABLE_ALIGNMENT.CENTER
                run = paragraph.runs[0]
                run.bold = True

            num_row = 1

            for row_samecity in list_rows:
                table.rows[num_row].cells[0].text = row_samecity['订单日期'].strftime('%Y%m%d')
                table.rows[num_row].cells[1].text = row_samecity['商品编号']
                table.rows[num_row].cells[2].text = str(row_samecity['订购数量'])
                table.rows[num_row].cells[3].text = str(row_samecity['订购单价'])
                num_row = num_row + 1
            print(total_price)
            paragraph = doc.add_paragraph("合计：{0}元".format(total_price))
            set_paragraph_style(paragraph)
            if row2['所在地市'] != row1['所在地市']:
                doc.add_page_break()

            row1 = row2
            list_rows.clear()
            total_price = 0
            list_rows.append(row1)
            total_price += row1['金额']
    if i == 1000:
        break
doc.save("销售表.docx")

```