Учреждение образования

«Белорусский государственный университет

информатики и радиоэлектроники»

### «СОВРЕМЕННЫЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ»

***Лабораторная работа № 2***

«Разработка и реализация классов. Работа с данными с использованием Pandas»

Проверила: Выполнил:

Василькова А.Н. Островский Я.А.

Минск 2024

Вариант № 2

Цель: Освоить разработку и реализацию классов с применением языка программирования Python. Освоить работу с данными с использованием Pandas.

Задание: Часть 1: Николай – оригинальный человек. Он решил создать класс Nikola, принимающий при инициализации 2 параметра: имя и возраст. Но на этом он не успокоился. Не важно, какое имя передаст пользователь при создании экземпляра, оно всегда будет содержать «Николая». В частности - если пользователя на самом деле зовут Николаем, то с именем ничего не произойдет, а если его зовут, например, Максим, то оно преобразуется в “Я не Максим, а Николай”. Более того, никаких других атрибутов и методов у экземпляра не может быть добавлено, даже если кто-то и вздумает так поступить (т.е. если некий пользователь решит прибавить к экземпляру свойство «отчество» или метод «приветствие», то ничего у такого хитреца не получится).

Часть 2: Реализовать средствами Python и Pandas возможности Excelфункций IF и VLOOKUP, а также о том, как воспроизвести средствами Pandas функционал сводных таблиц Excel.

Файл micola.py

class Micola():  
 def \_\_init\_\_(self, name, age):  
 if "Мікола" in name:  
 self.name = name  
 else:  
 self.name = f"Я не {name}, а Мікола"  
 self.age = age  
  
 def \_\_setattr\_\_(self, name, value):  
 try:  
 if name not in ("name", "age"):  
 raise AttributeError("Нельга дадаваць новыя ўласцівасці або метады да класу Mikola")  
 super().\_\_setattr\_\_(name, value)  
 except AttributeError as e:  
 print(f"Памылка: {e}")  
  
 def \_\_repr\_\_(self):  
 return f"{self.\_\_class\_\_.\_\_name\_\_}({self.name}; {self.age})"  
  
  
micola = Micola("Мікола", 18)  
print(micola)  
  
fake\_micola = Micola("Міця", 52)  
print(fake\_micola)  
  
micola.patronymic = "Аляксандравіч"

Файл pnds.py

import pandas as pd  
  
# реалізаваць функцыю IF Excel праз pandas  
sales\_df1 = pd.read\_excel('pythonexcel.xlsx', sheet\_name='sales')  
sales\_df1["MoreThan500"] = ["Yes" if x > 500 else "No" for x in sales\_df1["Sales"]]  
print(sales\_df1.head(5), "\n")  
  
# рэалізаваць функцыю VLOOKUP Excel праз pandas  
sales\_df2 = pd.read\_excel('pythonexcel.xlsx', sheet\_name='sales')  
states\_df = pd.read\_excel('pythonexcel.xlsx', sheet\_name='states')  
sales = sales\_df2.merge(states\_df, on="City", how="left")  
print(sales.head(5), "\n")  
  
# рэалізаваць функцыяна зводных табліцаў Excel праз pandas  
sales\_df3 = pd.read\_excel('pythonexcel.xlsx', sheet\_name='sales')  
pivot = sales\_df3.pivot\_table(values="Sales", index="City", aggfunc="sum")  
print(pivot, "\n")  
  
# задача на дыапазон датаў у pandas  
# можна замест periods указаца end = "2024-01-01"  
dates = pd.date\_range(start="2023-12-12", periods=21, freq="D")  
print(dates, "\n")  
  
# задача на знвходжанне месяца даты ў pandas  
# dates = pd.Series(['2020-05-18', '2020-08-20', '2020-12-21'])  
# dates = pd.to\_datetime(dates)  
# months = dates.dt.month  
# print(months, "\n")  
  
# спосаб вышэй -- рашэнне дадзенай задачы праз серыю, гэты праз датафрэйм  
df = pd.DataFrame({'sales\_date': ['2020-05-18', '2020-08-20', '2020-12-21'], 'total\_sales': [675, 500, 575]})  
df['month'] = pd.DatetimeIndex(df['sales\_date']).month  
print(df, "\n")  
  
  
# пераўтварэнне адметкі часу ў дату і час  
timestamps = pd.Series([1577836800, 1581609600, 1583625600, 1585699200, 1588876800])  
datetimes = pd.to\_datetime(timestamps, unit="s")  
print(datetimes)

Результат выполнения программы micola показан на рисунке 1.

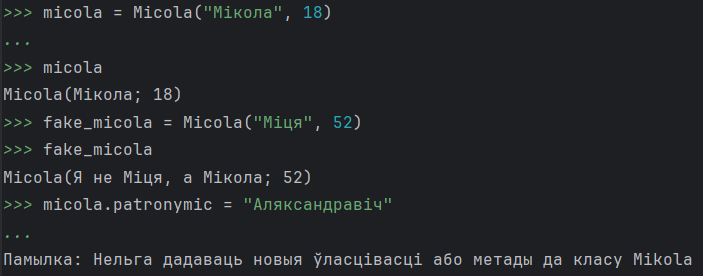


Рисунок 1 – результат выполнения программы micola

Результат выполнения программы pnds показан на рисунке 2.

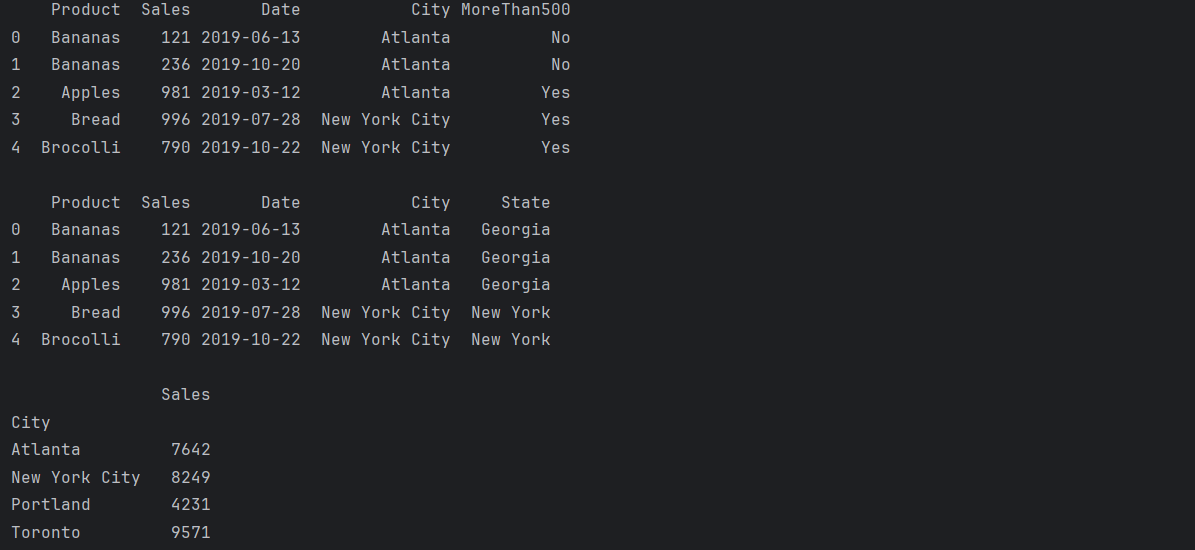


Рисунок 2 – результат выполнения программы pnds