Учреждение образования

«Белорусский государственный университет

информатики и радиоэлектронике»

Кафедра инженерной психологии и эргономики

Отчет по

ЛАБОРАТОРНОЙ РАБОТЕ №2

Разработка и реализация классов. Работа с данными с использованием Pandas

Подготовил:

Студент гр. 010101

Житкович Р.В.

Проверила:

Василькова А.Н.

Минск 2023

Задача №1 – Вариант 3

Условие:

Создайте класс ПЕРСОНА с методами, позволяющими вывести на экран информацию о персоне, а также определить ее возраст (в текущем году). Создайте дочерние классы: АБИТУРИЕНТ (фамилия, дата рождения, факультет), СТУДЕНТ (фамилия, дата рождения, факультет, курс), ПРЕПОДАВАТЕЛЬ (фамилия, дата рождения, факультет, должность, стаж), со своими методами вывода информации на экран и определения возраста. Создайте список из п персон, выведите полную информацию из базы на экран, а также организуйте поиск персон, чей возраст попадает в заданный диапазон.

Решение:

class Person():  
 def \_\_init\_\_(self, name='Руслан', surname='Житкович', birth\_year = 2003):  
 self.name = name  
 self.surname = surname  
 self.birth\_year = birth\_year  
  
 def get\_name(self):  
 return self.name  
  
 def get\_surname(self):  
 return self.surname  
  
 def get\_birth\_year(self):  
 return self.birth\_year  
  
 def set\_name(self, name):  
 self.name = name  
  
 def set\_surname(self, surname):  
 self.surname = surname  
  
 def set\_birth\_year(self, birth\_year):  
 self.birth\_year = birth\_year  
  
 def get\_age(self):  
 return datetime.datetime.today().year - self.birth\_year  
  
 def enter\_data(self):  
 self.name = input('Введите имя')  
 self.surname = input('Введите фамилию')  
 self.birth\_year = int(input('Введите год рождения'))  
  
 def print\_info(self):  
 return self.name,self.surname,self.birth\_year  
  
class Enrolle(Person):  
 def \_\_init\_\_(self, name='Петя', surname='Булатов', birth\_year = 2005, faculty = 'FCP'):  
 self.name = name  
 self.surname = surname  
 self.birth\_year = birth\_year  
 self.faculty = faculty  
  
class Student(Enrolle):  
  
 def \_\_init\_\_(self, name='Петя', surname='Булатов', birth\_year = 2005, faculty = 'FCP', course = 2):  
 self.name = name  
 self.surname = surname  
 self.birth\_year = birth\_year  
 self.faculty = faculty  
 self.course = course

if \_\_name\_\_ == '\_\_main\_\_':

match(input('1 - для работы с классами, 2 - для работы с pandas\n> ')):

case '1':

lst = []

petya = Enrolle()

misha = Enrolle('Миша', 'Угрумов', 2004, 'FKSIS')

katya = Enrolle('Катя', 'Задорнова', 2004, 'FKSIS')

vasya = Student()

zhenya = Student('Евгений', 'Мальцев', 2000, 'FCP', 4)

pasha = Student('Павел', 'Кириянов', 2001, 'FCSIS', 3)

lst.append(petya)

lst.append(vasya)

lst.append(misha)

lst.append(katya)

lst.append(pasha)

board1 = int(input('Введите левую границу возраста: '))

board2 = int(input('Введите правую границу возраста: '))

for i in range(len(lst)):

if (lst[i].get\_age() >= board1 and lst[i].get\_age() <= board2):

print(lst[i].print\_info())

Результат работы программы отображен на рисунке 1.

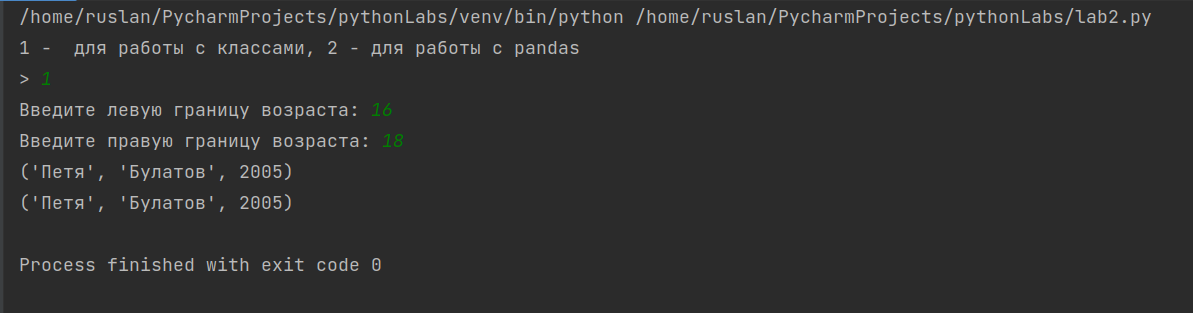


Рисунок 1 – Результат первой задачи

Задача №2

Условие:

Реализовать средствами Python и Pandas возможности Excel-функций IF и VLOOKUP.

Решение:

def pandas\_task():

states = pd.read\_excel('https://github.com/datagy/mediumdata/raw/master/pythonexcel.xlsx', sheet\_name='states')

sales = pd.read\_excel('https://github.com/datagy/mediumdata/raw/master/pythonexcel.xlsx', sheet\_name='sales')

print('Список из Excel:')

print(sales,'\n')

print('Список с операцией IF:')

print()

sales['MoreThan500'] = ['Yes' if x > 500 else 'No' for x in sales['Sales']]

print(sales,'\n')

print('Список с операцией связывания таблиц:')

sales = pd.merge(sales, states, how='left', on='City')

print(sales)

Результат работы программы отображен на рисунке 2.

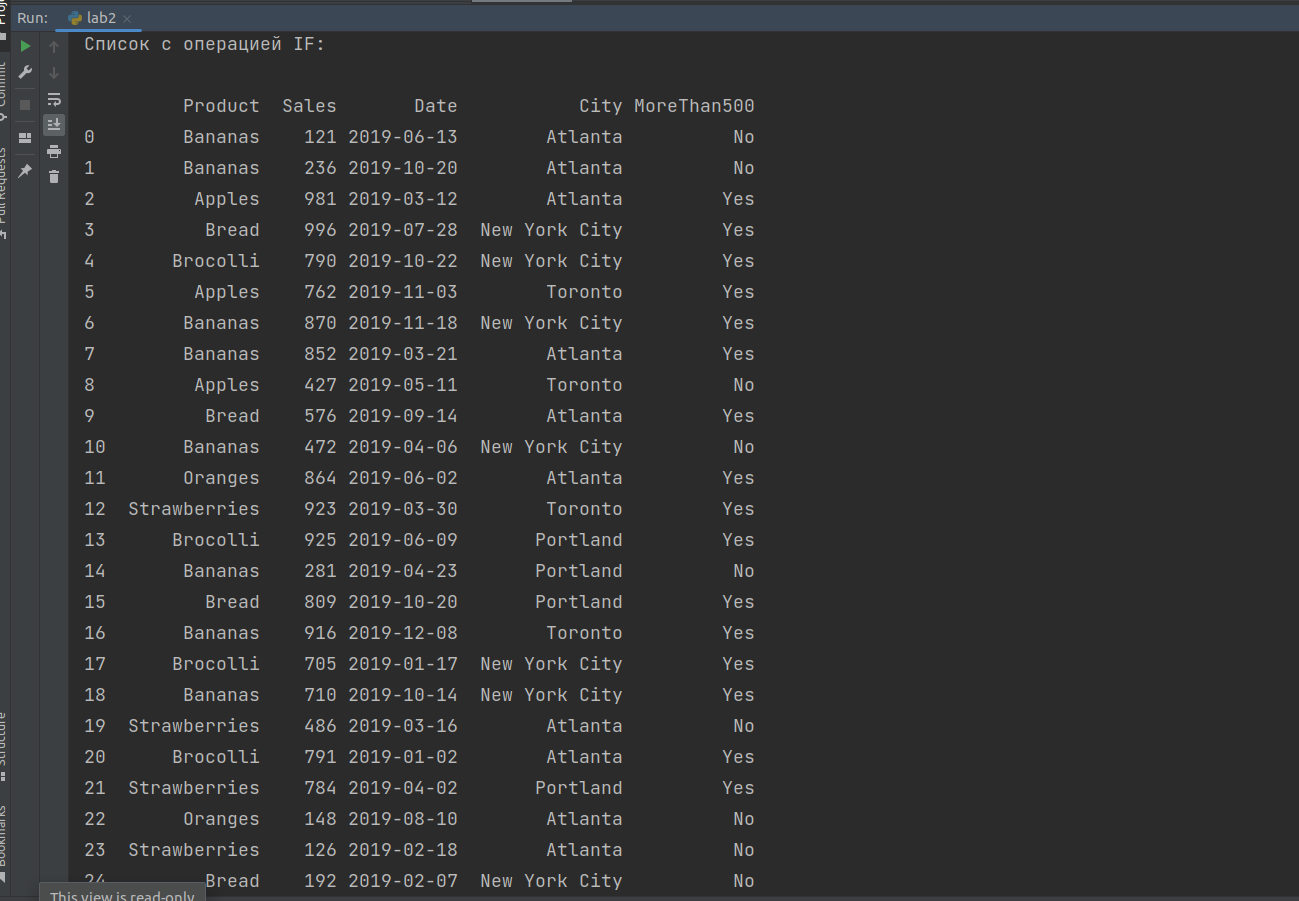


Рисунок 2 – Результат второй задачи