**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П.О.СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

Дисциплина «скриптованные языки программирования»

ОТЧЕТ

Лабораторная работа № 5

Применение скриптовых языков для обработки данных

Выполнил студент

Группы ИТД-21

Чайдаков И.М.

Принял преподаватель

Малиновский И. Л.

Гомель 2024

Применение скриптовых языков для обработки данных

**Цель работы:** изучит средства *Python* для анализа данных; агрегирование и преобразование данных; исследование и классификацию данных.

**Задание 1:** Создать объект Series, содержащий список группы.

В качестве индексов использовать номер в журнале старосты.

Индекс старосты задать 'Starosta'.

Вывести:

1) Список всех элементов

2) Каждый второй элемент списка

3) Набор индексов

Создать DataFrame со столбацами Name, IsStarosta,

Phone, Gender, YearBirth, внести 10 записей.

Столбец Age должен рассчитываться как разница текущего года и даты рождения.

Индексы должны содержать первые символы фамилии, имени и отчества в латинице

Вывести:

1) Первые 3 записи DataFrame

2) Столбец с именами

3) Столбцы имени, пола и возраста

4) Данные о старосте

5) Минимальный и максимальный возраст

6) Сводная таблица из количества девушек и юношей и их среднего возраста

7) Распределение студентов по возрасту

Добавить функцию Сохранения DataFrame в формате csv-файла.

Добавить функцию создания нового DataFrame на основе данных.

**Ход выполнения задания:**

**1.** Создаем приложение для работы с базой данных класса (Приложение А).

**2.** Создаем пользовательский интерфейс для доступа к соответствующим функциям и проверяем их выполнение запустив программу в окне консоли (Рисунок 1).

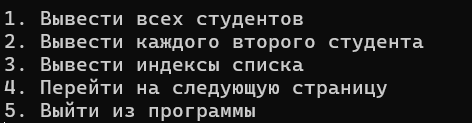


Рисунок 1 — Меню программы.

**3.** Осмотрим всех студентов выводя их список на экран (рисунок 2).

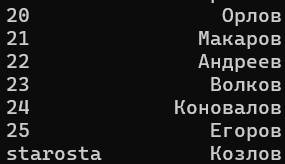


Рисунок 2 — Таблица данных.

**4.** Перейдем на следующую страницу меню открыв доступ к новым функциям (рисунок 3).

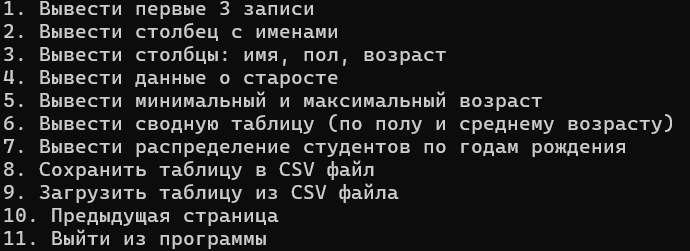


Рисунок 3 — Второе меню программы.

**5.** Выведем все наши данные используя заданный формат (рисунок 4).

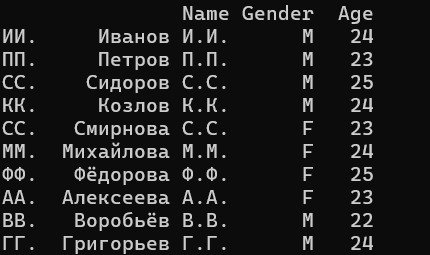


Рисунок 4 — Вывод данных (*fio*, *gender*, *age*).

**Задание 1:** Импортировать файл polit.csv и описать на его основе DataFrame

ctry - страна

fh09 - индекс Freedom House за 2009 год

polity09 - индекс polity2 за 2009 год

gini - индекс Джини за 2000-2010 года

fparl08 - доля женщин в парламенте в 2008 году

mena - регион Middle East and North Africa

lati - регион Latin America

cari - регион Caribbean, former British, French, Dutch colony

east - регион East Asia

sovi - регион Former Soviet Bloc

afri - регион Africa

corr0509 - индекс Control Corruption за 2005-2009 годы

Загрузить таблицу и создать датафрейм,

удалить строки с пропущенными значениями.

Сохранить изменения в файле polit.csv

Сохранить в NotFree страны с hf09 > 5.

Сохранить в afW страны afri с fparl08 > 30.

Сохранить в laDem страны afri и lati с Polity09 >= 8.

Добавить столбец CorrRound в котором значения corr0509 будут округлены до 2 знаков.

Добавить столбец statusFH в котором будут храниться типы стран в зависимости от значений fh09

Сгруппировать страны по столбах statusFH и вывести минимальное и максимальное

значение gini по каждой группе.

Записать разные группы statusFH в разные csv-файлы

**Ход выполнения задания:**

**1.** Описываем код для чтения данных с заданного файла и работы с ним (Приложение А).

**2.** Запускаем программу для выполнения ее функций и вывода данных на консоли (Рисунок 5).

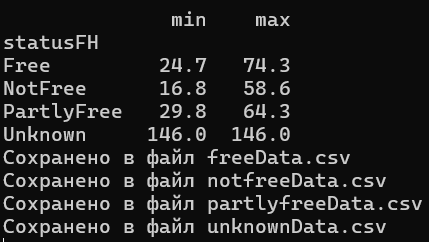


Рисунок 5 — Консольный вывод программы.

**3.** Откроем созданные файлы чтобы проверить правильность выполнения функций (рисунок 6).

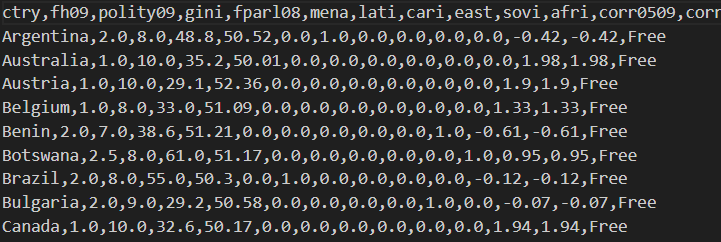


Рисунок 6 — Файл *FreeData.csv*.

**Вывод:** Библиотека *Pandas* позволяет работать с файлами данных и редактировать их данные что позволяет значительно оптимизировать процесс работы с табличными данными.

**ПРИЛОЖЕНИЕ А**

**Код программ**

**Задание 1:**

import pandas as pd

from datetime import datetime

def seriesMenu(series):

    while True:

        print("1. Вывести всех студентов")

        print("2. Вывести каждого второго студента")

        print("3. Вывести индексы списка")

        print("4. Перейти на следующую страницу")

        print("5. Выйти из программы")

        choice = input()

        match choice:

            case "1":

                print(series)

            case "2":

                print(series.iloc[1::2])

            case "3":

                print(series.index)

            case "4":

                dataframe\_menu(createDataFrame())

                break

            case "5":

                break

            case \_:

                print("Input error. Please, try again.")

def dataframe\_menu(df):

    while True:

        print("1. Вывести первые 3 записи")

        print("2. Вывести столбец с именами")

        print("3. Вывести столбцы: имя, пол, возраст")

        print("4. Вывести данные о старосте")

        print("5. Вывести минимальный и максимальный возраст")

        print("6. Вывести сводную таблицу (по полу и среднему возрасту)")

        print("7. Вывести распределение студентов по годам рождения")

        print("8. Сохранить таблицу в CSV файл")

        print("9. Загрузить таблицу из CSV файла")

        print("10. Предыдущая страница")

        print("11. Выйти из программы")

        choice = input()

        match choice:

            case "1":

                print(df.head(3))

            case "2":

                print(df["Name"])

            case "3":

                print(df[["Name", "Gender", "Age"]])

            case "4":

                print(df[df["IsStarosta"]])

            case "5":

                print("Минимальный возраст:", df["Age"].min())

                print("Максимальный возраст:", df["Age"].max())

            case "6":

                print("Количество студентов по полу:")

                print(df["Gender"].value\_counts())

                print("\nСредний возраст по полу:")

                print(df.groupby("Gender")["Age"].mean())

            case "7":

                print(df["YearBirth"].value\_counts())

            case "8":

                df.to\_csv("studentsData.csv")

                print("Таблица сохранена в файл studentsData.csv")

            case "9":

                df = pd.read\_csv("studentsData.csv", index\_col=0)

                print("Таблица загружена из файла studentsData.csv")

            case "10":

                seriesMenu(createSeries())

                break

            case "11":

                break

            case \_:

                print("Input error. Please, try again.")

def createSeries():

    namesList = [

        "Иванов", "Петров", "Сидоров", "Смирнов", "Кузнецов", "Попов", "Васильев",

        "Фёдоров", "Михайлов", "Александров", "Соколов", "Лебедев", "Козлов",

        "Новиков", "Морозов", "Захаров", "Зайцев", "Борисов", "Кириллов",

        "Орлов", "Макаров", "Андреев", "Волков", "Коновалов", "Егоров"

    ]

    seriesData = {i+1: namesList[i] for i in range(25)}

    seriesData["starosta"] = seriesData.pop(13)

    return pd.Series(seriesData)

def createDataFrame():

    data = {

        "Name": ["Иванов И.И.", "Петров П.П.", "Сидоров С.С.", "Козлов К.К.", "Смирнова С.С.",

                "Михайлова М.М.", "Фёдорова Ф.Ф.", "Алексеева А.А.", "Воробьёв В.В.", "Григорьев Г.Г."],

        "IsStarosta": [False, False, False, True, False, False, False, False, False, False],

        "Phone": ["+79160000001", "+79160000002", "+79160000003", "+79160000004", "+79160000005",

                "+79160000006", "+79160000007", "+79160000008", "+79160000009", "+79160000010"],

        "Gender": ["M", "M", "M", "M", "F", "F", "F", "F", "M", "M"],

        "YearBirth": [2000, 2001, 1999, 2000, 2001, 2000, 1999, 2001, 2002, 2000]

    }

    df = pd.DataFrame(data)

    currentYear = datetime.now().year

    df["Age"] = currentYear - df["YearBirth"]

    df.index = [name.split()[0][0] + name.split()[1][0] + name.split()[1][1] for name in df["Name"]]

    return df

seriesMenu(createSeries())

**Задание 2:**

import pandas as pd

polit = pd.read\_csv('polit.csv', delimiter=';', decimal=',', on\_bad\_lines='skip', encoding='utf-8')

polit.dropna(inplace=True)

notFree = polit[polit['fh09'] > 5]

afW = polit[(polit['afri'] == 1) & (polit['fparl08'] > 30)]

laDem = polit[((polit['afri'] == 1) | (polit['lati'] == 1)) & (polit['polity09'] >= 8)]

polit['corrRound'] = polit['corr0509'].round(2)

def determineStatusFH(fh09):

    if 7.0 >= fh09 >= 5.5:

        return "NotFree"

    elif 5.5 > fh09 > 2.5:

        return "PartlyFree"

    elif 2.5 >= fh09 >= 1.0:

        return "Free"

    return "Unknown"

polit['statusFH'] = polit['fh09'].apply(determineStatusFH)

giniStats = polit.groupby('statusFH')['gini'].agg(['min', 'max'])

print(giniStats)

for status, groupDF in polit.groupby('statusFH'):

    filename = f"{status.replace(' ', '\_').lower()}Data.csv"

    groupDF.to\_csv(filename, index=False)

    print(f"Сохранено в файл {filename}")

input()