Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ МОРДОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ им. Н.П. ОГАРЁВА»

(ФГБОУ ВО «МГУ им. Н.П. Огарёва»)

Факультет математики информационных технологий

Кафедра фундаментальной информатики

ОТЧЁТ

по учебной практике

технологическая (проектно-технологическая) практика

Студента 3 курса

Направление подготовки 02.03.02 Фундаментальная информатика и информационные технологии

Место прохождения практики г. Саранск, ФГБОУ ВО «МГУ им. Н. П. Огарева», корпус 1, аудитория 243

Срок прохождения практики 01.09.2022 – 11.01.2023

Автор отчёта А. Е. Конышев

подпись, дата

Обозначение отчёта ОП–02069964–У–02.03.02–09–23

Руководитель практики

от университета,

канд. физ.-мат. наук А. Н Андронов

подпись, дата

Отчёт защищён Оценка

дата

Саранск 2023

**ЗАДАНИЕ**

**на учебную практику**

**Технологическая (проектно-технологическая) практика**

Студенту Конышеву Артёму Евгеньевичу 3 курса 302 группы

Направление подготовки 02.03.02 Фундаментальная информатика и информационные технологии

Место прохождения практики г. Саранск, ФГБОУ ВО «МГУ им. Н. П. Огарева», корпус 1, аудитория 243

Срок прохождения практики 01.09.2022 – 11.01.2023

Срок представления отчёта студентом на защиту 11.01.2023

1 Цели и задачи практики состоят в практическом применении обучающимися теоретических знаний и практических умений при работе со средствами вычислительной техники и использовании различных языков и методов программирования для решения прикладных задач. Научиться основам профессиональной работы с Python; закрепить знания об алгоритмизации и показать связь базовых алгоритмических конструкций с операторами языка программирования; закрепить на практике имеющиеся теоретические знания о разработке программ в Python.

2 Компетенции студента, формируемые в результате прохождения практики: ОПК -5.

3 Задание на практику:

1. Изучить особенности и приёмы выполнения практических задач с использованием языка программирования Python;

2. Для закрепления полученных знаний выполнить лабораторные работы с использованием языка программирования Python

Руководитель практики

от университета,

канд. физ.-мат. наук А. Н Андронов

подпись, дата

Задание к исполнению принял А. Е. Конышев

подпись, дата

ДНЕВНИК

по учебной практике

Технологическая (проектно-технологическая) практика

Студента Конышева Артёма Евгеньевича 3 курса 302 группы

Направление подготовки 02.03.02 Фундаментальная информатика и информационные технологии

ЗАПИСИ

о работах, выполненных в период практики

|  |  |  |
| --- | --- | --- |
| Дата | Содержание | Замечания |
| 01.09.22-01.10.22 | Выполнение Лабораторной работы № 1 |  |
| 01.10.22-01.11.22 | Выполнение Лабораторной работы № 2 |  |
| 01.11.22-01.12.22 | Выполнение Лабораторной работы № 3 |  |

Руководитель практики

от университета,

канд. физ.-мат. наук А. Н Андронов

подпись, дата

**Лабораторная работа 1**

Задание 1. С клавиатуры вводятся пять чисел: a, b, c, d и e. Вывести на экран сумму и разность максимального из них и их среднего арифметического.

Код программы:

from random import \*

a = int(input("Введите a "))

b = int(input("Введите b "))

c = int(input("Введите c "))

d = int(input("Введите d "))

e = int(input("Введите e "))

avg: float = (a + b + c + d + e) / 5

maxNumber = max(a, b, c, d, e)

print(avg, maxNumber, avg + maxNumber, avg - maxNumber)

Результат:

Введите a 8

Введите b 3

Введите c 6

Введите d 9

Введите e 1

Среднее = 5.4

Максимальное = 9

Их сумма = 14.4

Их разность = -3.5999999999999996

Задание 2. С клавиатуры вводятся три числа – x, a и d. Проверить, является ли x членом арифметической прогрессии, в которой первый элемент равен (a+3), а разность равна 2d.

Код программы:

x: int = int(input("Введите x "))

a: int = int(input("Введите a "))

d: int = int(input("Введите d "))

print("первый член = ", a + 3, ", разность = ", d \* 2, ", x = ", x, sep="")

print(x - (a + 3) % (d \* 2) < 0.0001 and (a + 3) <= x)

Результат:

Введите x 4

Введите a 1

Введите d 4

первый член = 4, разность = 8, x = 4

True

Задание 3. «Замена». Пусть есть два списка – основной состав команды (11 человек) и запасной (5 человек). Заменить три случайно выбранных элемента первого списка тремя случайно выбранными элементами второго списка. Затем удалить 1 элемент (номер которого задать с клавиатуры) из первого списка.

Код программы:

from random import \*

from collections import Counter

mainTeam = [i for i in range(1, 12)]

reserveTeam = [i for i in range(12, 17)]

print("основная команда: ", mainTeam)

print("запасная команда: ", reserveTeam)

tempTeam = sample(mainTeam, 3)

tempTeam.sort()

print("список игроков из основной команды на замену: ", tempTeam)

mainTeam = list((Counter(mainTeam) - Counter(tempTeam)).elements())

tempTeam = sample(reserveTeam, 3)

tempTeam.sort()

print("список игроков из запасной команды на замену: ", tempTeam)

mainTeam.extend(tempTeam)

print("основная команда поле замены: ", mainTeam)

playerNumber: int = int(input("введите номер игрока основной команды для удаления "))

if playerNumber in mainTeam:

mainTeam.remove(playerNumber)

print("основная команда поле удаления игрока: ", mainTeam)

else:

print("игрока с таким номером в основной команде нет")

Результат:

основная команда: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]

запасная команда: [12, 13, 14, 15, 16]

список игроков из основной команды на замену: [1, 10, 11]

список игроков из запасной команды на замену: [12, 15, 16]

основная команда поле замены: [2, 3, 4, 5, 6, 7, 8, 9, 12, 15, 16]

введите номер игрока основной команды для удаления 8

основная команда поле удаления игрока: [2, 3, 4, 5, 6, 7, 9, 12, 15, 16]

Задание 4. «Карманные деньги». У студента есть n рублей (вводится с клавиатуры). Кусок пиццы стоит 85 рублей, хлеб – 35 рублей, лапша быстрого приготовления – 13 рублей, молоко – 65 рублей, шоколадный батончик – 55 рублей, шаурма – 170 рублей. Вывести на экран продукты, которые он может купить (пицца и хлеб, пицца и лапша, пицца+хлеб+молоко и т.д.) и сколько денег у него останется при каждом варианте покупки.

Код программы:

from itertools import combinations

products = {

'пицца' : 85,

'хлеб' : 35,

'лапша' : 13,

'молоко' : 65,

'батончк' : 55,

'шаверма' : 170

}

cash: int = 100

print("если у студента ", cash, "рублей, то он сможет купить: ")

for i in range(1, 7):

comb = list(combinations(products, i))

for j in comb:

purchaseAmount = 0

for k in j:

purchaseAmount += products[k]

change = cash - purchaseAmount

if change > 0:

for k in j:

print(k, end=" ")

print(", тогда сдача =", change, "рублей")

Результат:

если у студента 100 рублей, то он сможет купить:

пицца , тогда сдача = 15 рублей

хлеб , тогда сдача = 65 рублей

лапша , тогда сдача = 87 рублей

молоко , тогда сдача = 35 рублей

батончк , тогда сдача = 45 рублей

пицца лапша , тогда сдача = 2 рублей

хлеб лапша , тогда сдача = 52 рублей

хлеб батончк , тогда сдача = 10 рублей

лапша молоко , тогда сдача = 22 рублей

лапша батончк , тогда сдача = 32 рублей

Задание 5. Вычислить, сколько секунд прошло с начала первой пары до конца четвертой пары.

Код программы:

import datetime

start = datetime.timedelta(hours=9)

finish = datetime.timedelta(hours=15, minutes=45)

print(f'прошло {(finish - start).seconds} секунд')

Результат:

прошло 24300 секунд

Задание 6. Определить, через сколько месяцев банковский вклад S тыс. рублей, проценты по которому начисляются ежемесячно из расчета i % годовых, превысит величину (S+200).

Код программы:

from math import \*

depositAmount: float = 300

depositPercentage: float = 2.5

monthlyMultiplier: float = pow((1 + depositPercentage / 100), 1/12)

numberOfMonths = ceil(log(1 + 200 / depositAmount, monthlyMultiplier))

print(f'сумма вклада: {depositAmount} тыс. рублей')

print(f'ставка: {depositPercentage} % годовых')

print(f'через {numberOfMonths} месяцев депозит увеличится на 200 тыс. рублей')

Результат:

сумма вклада: 300 тыс. рублей

ставка: 2.5 % годовых

через 249 месяцев депозит увеличится на 200 тыс. рублей

Лабораторная работа 2

Задание. Построить круговую и столбовую диаграммы распределения подписчиков групп vk г. Уфа по полу, возрасту, вузу. (Использовать VK API)

Код программы:

import json

from datetime import date, datetime

from time import sleep

import itertools

import requests

from requests import Response

import matplotlib as mpl

import matplotlib.pyplot as plt

from acsess import token

def getAllMembers(group\_id: int) -> list:

address: str = 'https://api.vk.com/method/'

method: str = 'groups.getMembers'

version: float = 5.131

count: int = 1000

fields: str = 'bdate, universities, sex'

offset: int = 0

params: dict = dict(access\_token=token, group\_id=group\_id, v=version, fields=fields, count=count)

all\_members: list = []

while True:

params['offset'] = offset

response: Response = requests.get(address + method, params)

count\_of\_members: int = response.json()["response"]['count']

all\_members.extend(response.json()['response']['items'])

print("\rКоличество подписчиков:", count\_of\_members, "\tОбработано:", offset, end="")

sleep(0.200)

offset += 1000

if offset > count\_of\_members:

break

return all\_members

def writeDataInJsonFile(data: list, fileName: str) -> None:

with open(fileName, "w", encoding='utf-8') as jsonfile:

json.dump(data, jsonfile, indent=4, ensure\_ascii=False)

jsonfile.close()

def readDataFromJsonFile(fileName: str) -> list:

with open(fileName, "r", encoding='utf-8') as jsonfile:

data: list = json.load(jsonfile)

return data

def getAge(birthdate: str) -> int:

if len(birthdate) > 7:

today: date = date.today()

birthdate = datetime.strptime(birthdate, "%d.%m.%Y")

return today.year - birthdate.year - ((today.month, today.day) < (birthdate.month, birthdate.day))

return -1

def updateAgeDict(birthdate: str, ages: dict) -> None:

age: int = getAge(birthdate)

if age != -1:

if age <= 18:

ages["0 - 18"] += 1

elif age <= 25:

ages["19 - 25"] += 1

elif age <= 35:

ages["26 - 35"] += 1

elif age <= 45:

ages["36 - 45"] += 1

elif age <= 55:

ages["46 - 55"] += 1

else:

ages["55 and older"] += 1

else:

ages["unspecified"] += 1

def updateUniversityDict(universities\_of\_user: list, all\_universities: dict) -> None:

count\_of\_universities\_of\_user = len(universities\_of\_user)

if count\_of\_universities\_of\_user == 0:

all\_universities["unspecified"] += 1

else:

for university in universities\_of\_user:

university\_id: str = university["id"]

if university\_id in all\_universities:

all\_universities[university\_id] += 1

else:

all\_universities[university\_id] = 1

def processingData(dataList: list, \_ageDict: dict, \_sexDict: dict, \_universityDict: dict) -> None:

for data in dataList:

if 'bdate' in data:

updateAgeDict(data['bdate'], \_ageDict)

else:

\_ageDict["unspecified"] += 1

if 'sex' in data:

\_sexDict[str(data["sex"])] += 1

else:

\_sexDict["0"] += 1

if 'universities' in data:

updateUniversityDict(data['universities'], \_universityDict)

else:

\_universityDict["unspecified"] += 1

def getUniversitiesTitles(city\_id: int, universities\_id: list) -> dict:

address: str = "https://api.vk.com/method/"

method: str = "database.getUniversities"

params: dict = dict(access\_token=token, country\_id=1, city\_id=city\_id, v=5.131, count=100)

response: Response = requests.get(address + method, params)

all\_universities: dict = response.json()['response']['items']

universities\_titles: dict = {}

for university\_id in universities\_id:

for university in all\_universities:

if university\_id == university['id']:

universities\_titles[university\_id] = university['title']

return universities\_titles

def getModifiedUniversityDict(city\_id: int, universities: dict, count\_of\_MP\_universities: int) -> dict:

MP\_universities: dict = dict(itertools.islice(universities.items(), 1, count\_of\_MP\_universities))

MP\_universities\_titles: dict = getUniversitiesTitles(city\_id, [\*MP\_universities])

count\_of\_other\_universities = 0

for count in list(universities.values())[count\_of\_MP\_universities:]:

count\_of\_other\_universities += count

print("Количество остальных университетов:", count\_of\_other\_universities)

top\_and\_other\_universities = dict(zip(MP\_universities\_titles.values(),

MP\_universities.values()))

top\_and\_other\_universities["other"] = count\_of\_other\_universities

return top\_and\_other\_universities

def buildingBarChart(title: str, filename: str, keys: list, values: list) -> None:

dpi = 160

fig = plt.figure(dpi=dpi, figsize=(1024 / dpi, 1024 / dpi))

mpl.rcParams.update({'font.size': 10})

ax = plt.axes()

ax.yaxis.grid(True, zorder=0)

plt.title(title)

plt.bar(keys, values, zorder=1)

plt.xticks(keys)

fig.autofmt\_xdate(rotation=25)

fig.savefig(filename)

def buildingPieChart(title: str, filename: str, keys: list, values: list) -> None:

dpi = 160

fig = plt.figure(dpi=dpi, figsize=(1024 / dpi, 1024 / dpi))

mpl.rcParams.update({'font.size': 10})

plt.title(title)

plt.pie(values, autopct='%.1f', radius=1.1)

plt.legend(

bbox\_to\_anchor=(0.3, 0.1),

loc=10,

fontsize=7, labels=keys)

fig.savefig(filename)

cities\_ids: dict[str, int] = {

"Нижний Новгород": 95,

"Чебоксары": 157,

"Казань": 60,

"Пенза": 109,

"Ульяновск": 149,

"Самара": 123,

"Рязань": 122,

"Саратов": 125,

"Уфа": 151,

"Челябинск": 158

}

age\_groups = {

"0 - 18": 0,

"19 - 25": 0,

"26 - 35": 0,

"36 - 45": 0,

"46 - 55": 0,

"55 and older": 0,

"unspecified": 0

}

sex\_dict = {

"0": 0, # пол не указан

"1": 0, # женский

"2": 0 # мужской

}

university\_dict = {

"unspecified": 0

}

file\_path = "allMembers.json"

# allMembers = getAllMembers(62298989)

# writeDataInJson(allMembers, filePath)

group\_members: list = readDataFromJsonFile(file\_path)

processingData(group\_members, age\_groups, sex\_dict, university\_dict)

university\_dict = {k: v for k, v in sorted(university\_dict.items(), key=lambda item: item[1], reverse=True)}

university\_dict = getModifiedUniversityDict(151, university\_dict, 11)

charts\_titles = (

'Распределение подписчиков по возрасту',

'Распределение подписчиков по полу',

'Распределение подписчиков по университетам'

)

buildingBarChart(charts\_titles[0], 'barsAge.png', [\*age\_groups.keys()][:6], list(age\_groups.values())[:6])

buildingBarChart(charts\_titles[1], 'barsSex.png', ["Женщины", "Мужчины"], list(sex\_dict.values())[1:])

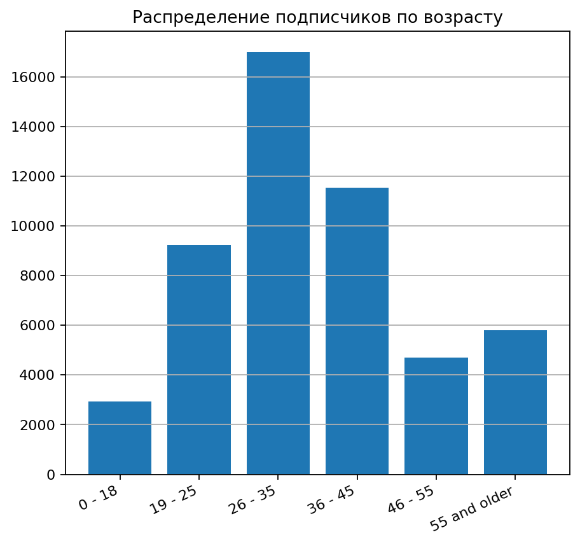
buildingBarChart(charts\_titles[2], 'barsUniversity.png', [\*university\_dict.keys()], list(university\_dict.values()))

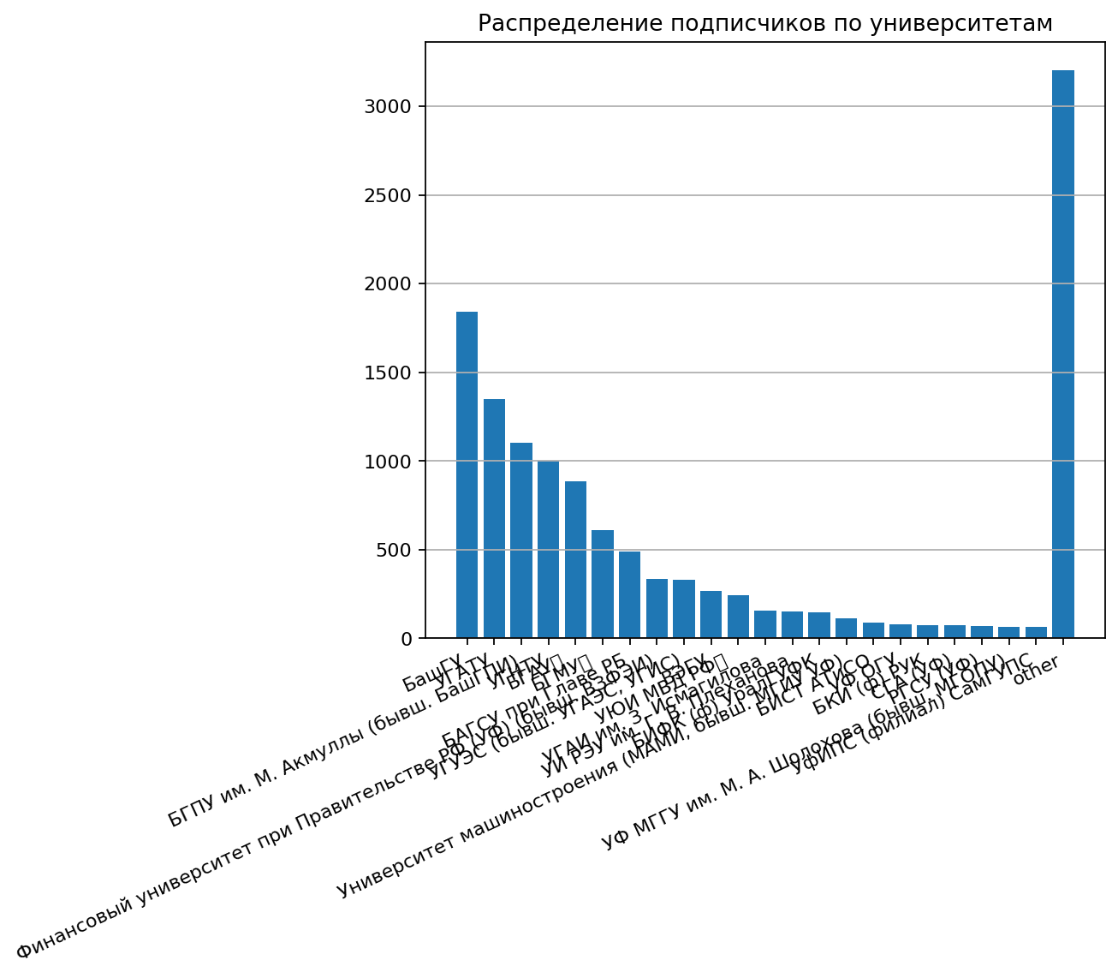
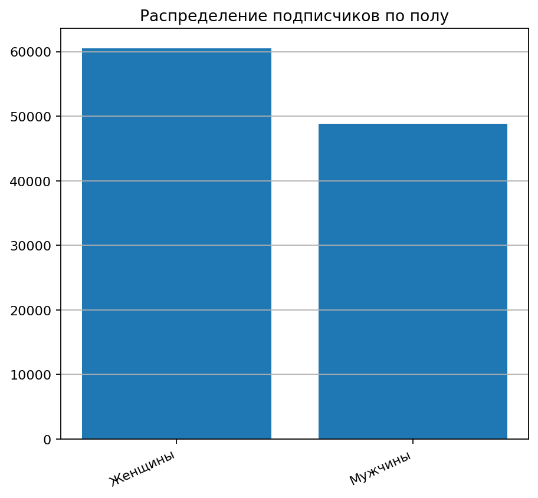
buildingPieChart(charts\_titles[0], 'pieAge.png', [\*age\_groups.keys()][:6], list(age\_groups.values())[:6])

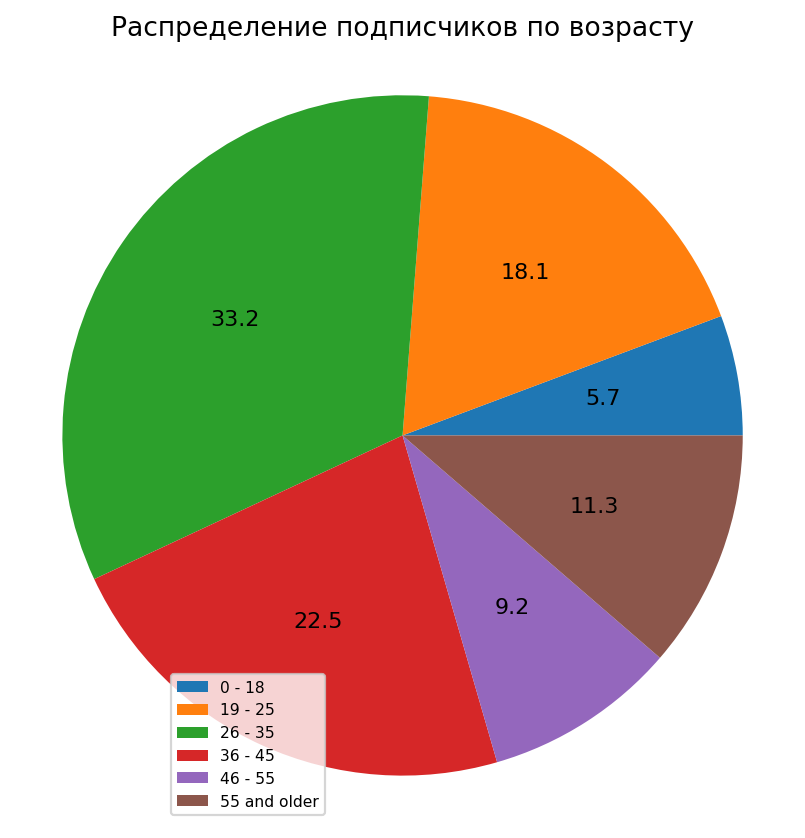
buildingPieChart(charts\_titles[1], 'pieSex.png', ["Женщины", "Мужчины"], list(sex\_dict.values())[1:])

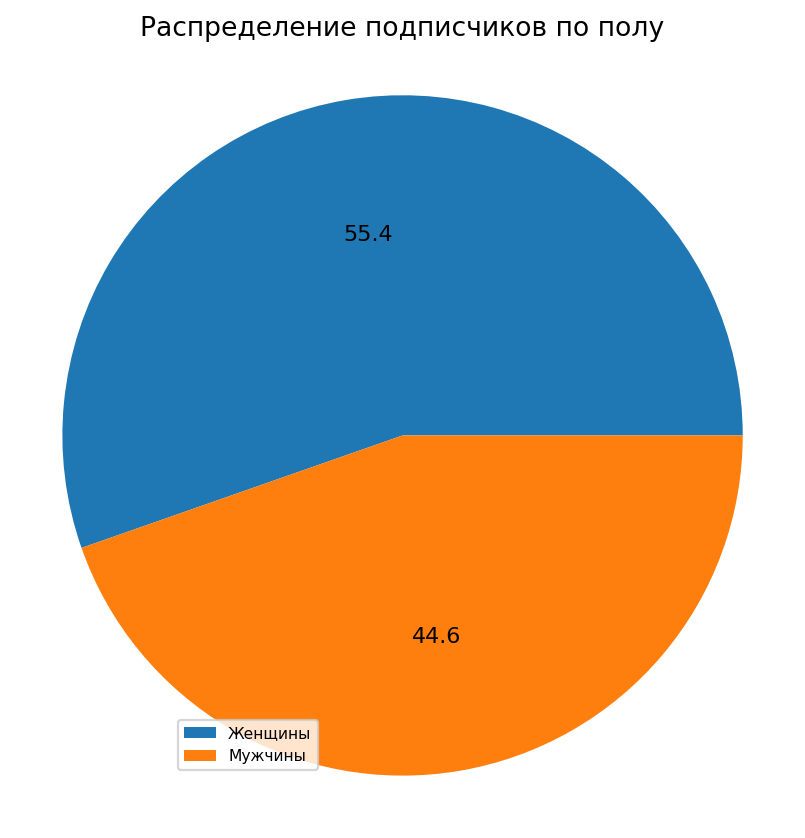
buildingPieChart(charts\_titles[2], 'pieUniversity.png', [\*university\_dict.keys()], list(university\_dict.values()))

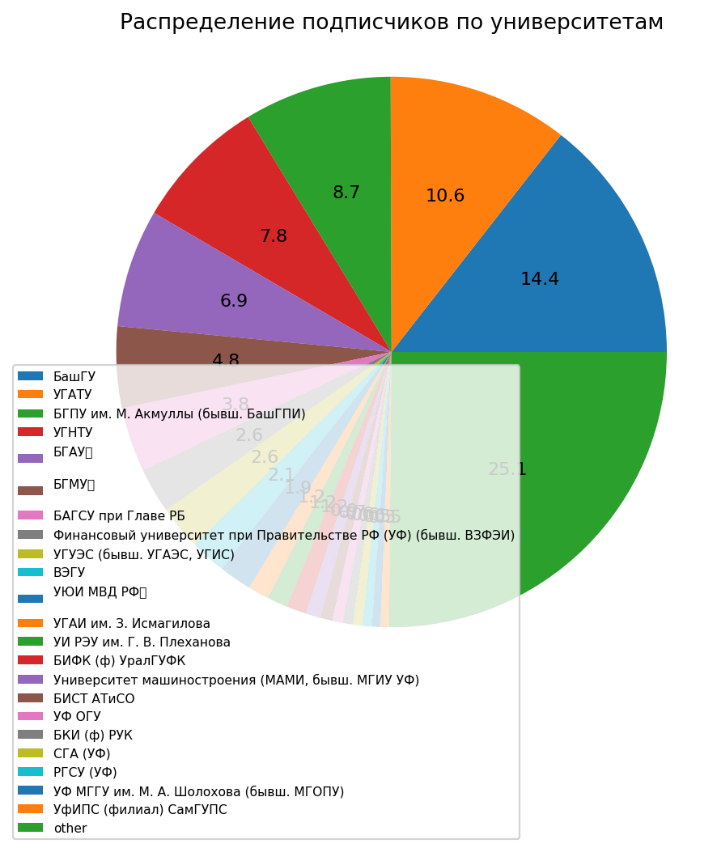
Результат:











Лабораторная работа 3

Задание. Средствами Python узнать, в каких городах России с населением свыше n человек в настоящее время идет дождь/снег/облачно.

Использовать OpenWeatherMap web API (обертка PyOWM).

Код программы:

import json

from pyowm import OWM

def printWeatherInformation(cities: list, population: int, status: str) -> None:

apiKey = "8543aaba717a4093d9776b21f90aeca3"

owm = OWM(apiKey)

mgr = owm.weather\_manager()

cities = [city for city in cities if city["population"] >= population]

count = 0

while count < cities.\_\_len\_\_():

citiesList = cities[count:count + 20:]

citiesIDList = [city["cityID"] for city in citiesList]

observation = mgr.weather\_at\_ids(citiesIDList)

for i in range(len(observation)):

if observation[i].weather.status == status:

city\_name = citiesList[i]["name"]

city\_population = citiesList[i]["population"]

correct\_word\_form = "человек" + "a" if str(city\_population)[-1] in ["2", "3", "4"] else ""

weather\_status = statusList[status]

print(f"В городе {city\_name} c населением {city\_population} {correct\_word\_form} сейчас {weather\_status}")

count += 20

statusList = {"Rain": "идёт дождь", "Snow": "снег", "Clouds": "облачно"}

with open("russian\_cities\_update.json", "r", encoding='utf-8') as jsonfile:

ru\_cities = json.load(jsonfile)

printWeatherInformation(ru\_cities, 20000, "Clouds")

Результат:

В городе Барнаул c населением 631124 человекa сейчас снег

В городе Иркутск c населением 617515 сейчас снег

В городе Махачкала c населением 604266 сейчас снег

В городе Омск c населением 1139897 сейчас снег

В городе Томск c населением 568508 сейчас снег

**ЗАКЛЮЧЕНИЕ**

Учебная практика была проведена в период с 01.09.2022 по 11.01.2023. В установленный период практики были выполнены лабораторные работы №1 «Основы программирования на Python», №2 «Парсинг vk», №3 «Получение погодных данных». Отчет по практике был подготовлен и сдан в установленные сроки.

АНКЕТА

Учебная практика

Технологическая (проектно-технологическая) практика

1. Удовлетворены ли Вы местом прохождения практики? (один ответ)

□ да □ нет

1. Удовлетворены ли Вы качеством разработки методических указаний, содержащихся в программе практики? (один ответ)

* в методических указаниях недостаточно информации для составления отчета
* методические указания не подходят к профильной организации
* не ознакомлен(а) с программой практики
* методические указания мне понятны

1. Считаете ли Вы достаточными для выполнения работ, предусмотренных программой практики, те теоретические знания, которые Вы получили в Университете? (один ответ)

* да
* скорее да, чем нет
* скорее нет, чем да
* нет

1. Дала ли производственная практика возможность применить и развить навыки, необходимые для дальнейшей профессиональной деятельности (в соответствии с Вашим направлением подготовки/Вашей специальностью)? (один ответ)

* да
* скорее да, чем нет
* скорее нет, чем да
* нет

1. Хотели бы Вы в дальнейшем продолжить свою трудовую деятельность в организации, в которой проходили практику? (один ответ)

* да
* нет
* со мной заключили договор о дальнейшем трудоустройстве

1. Как Вы оцениваете итоги практики с точки зрения ее результативности? (один ответ)

* на практике я еще больше убедился(ась) в правильности выбора профессии
* практика разочаровала меня в выбранной профессии
* практика обнаружила пробелы в моей специальной подготовке
* практика носила формальный характер

1. Оцените степень удовлетворенности местом прохождения практики (материально- техническая оснащенность, кадровый состав) по пятибалльной шкале (1 – очень плохо, 5 – отлично)
2. Оцените степень удовлетворенности взаимоотношениями с руководителем практики от Университета по пятибалльной шкале (1 – очень плохо, 5 – отлично)
3. Оцените удовлетворенность в целом условиями прохождения практики (один ответ)

* полностью удовлетворен(а)
* удовлетворен(а) частично
* полностью не удовлетворен(а)

1. Ваши предложения по организации практики:

Предложений нет