# МИНИCTEPCTBO НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ

**ФЕДЕРАЦИИ**

# Федеральное государственное автономное образовательное учреждение высшего образования

**«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

# Кафедра инфокоммуникаций Институт цифрового развития

**ОТЧЁТ**

# по лабораторной работе №9

Дисциплина: «Основы программной инженерии»

Тема: «Работа со словарями в языке Python»

Выполнил: студент 2 курса группы Пиж-б-о-21-1

Рязанцев Матвей Денисович

Ставрополь 2022

Цель работы: приобретение навыков по работе со списками при написании программ с помощью языка программирования Python версии 3.x.

**Выполнение работы**

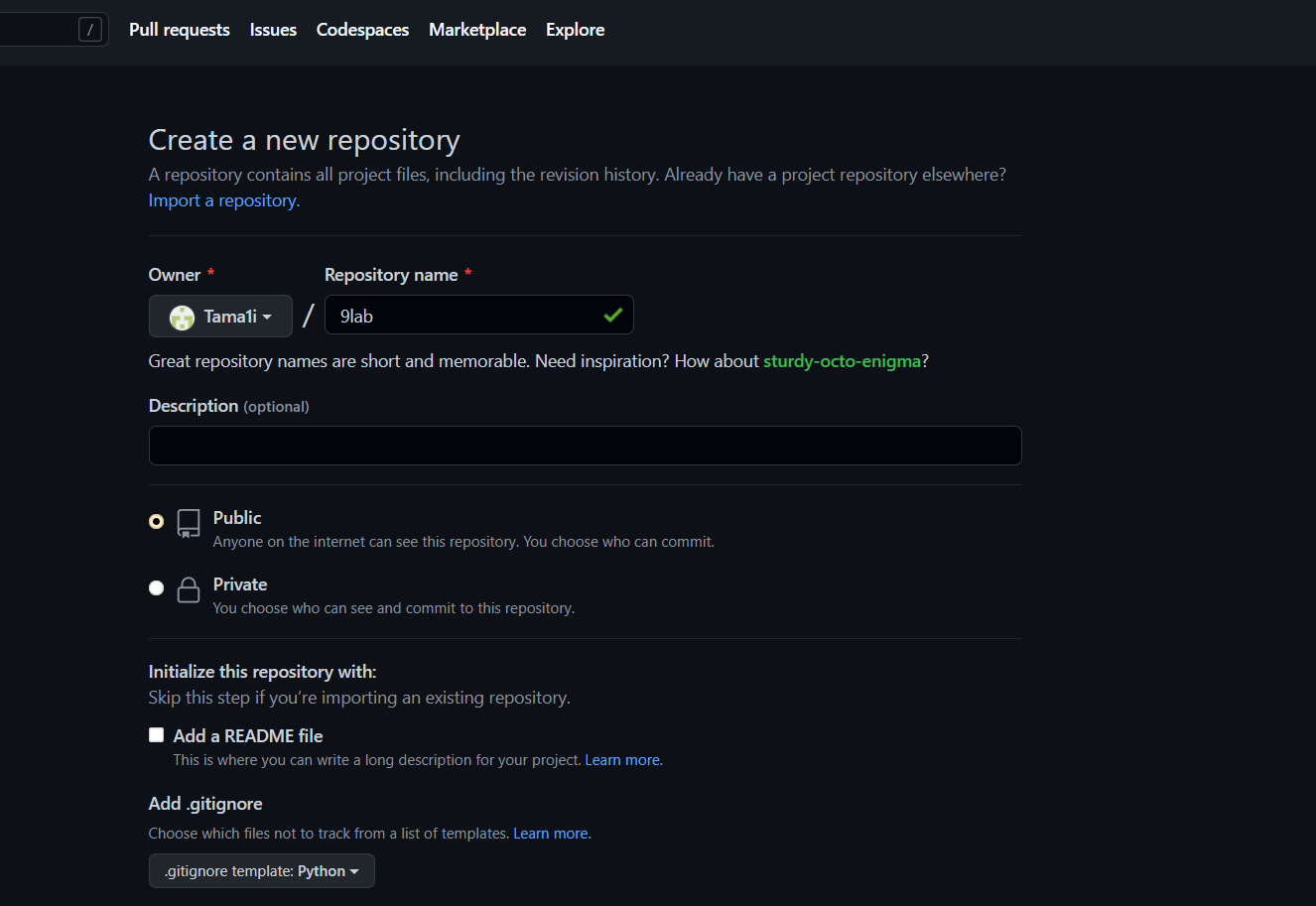


Рисунок 1 -создание репозитория

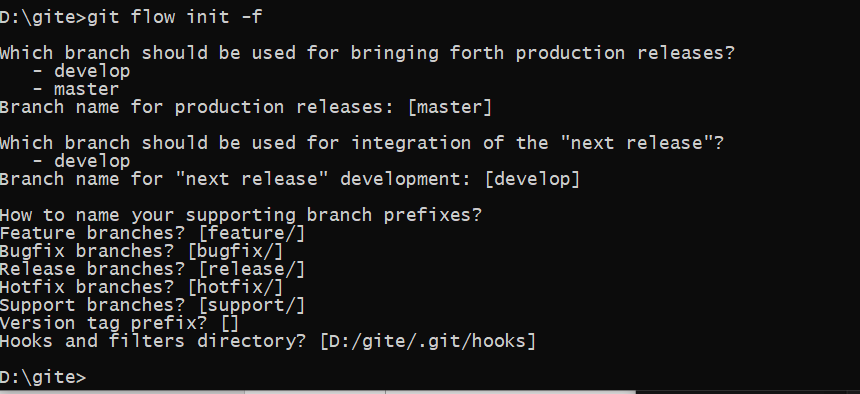


Рисунок 2 - Организация репозитория по модели ветвления git flow

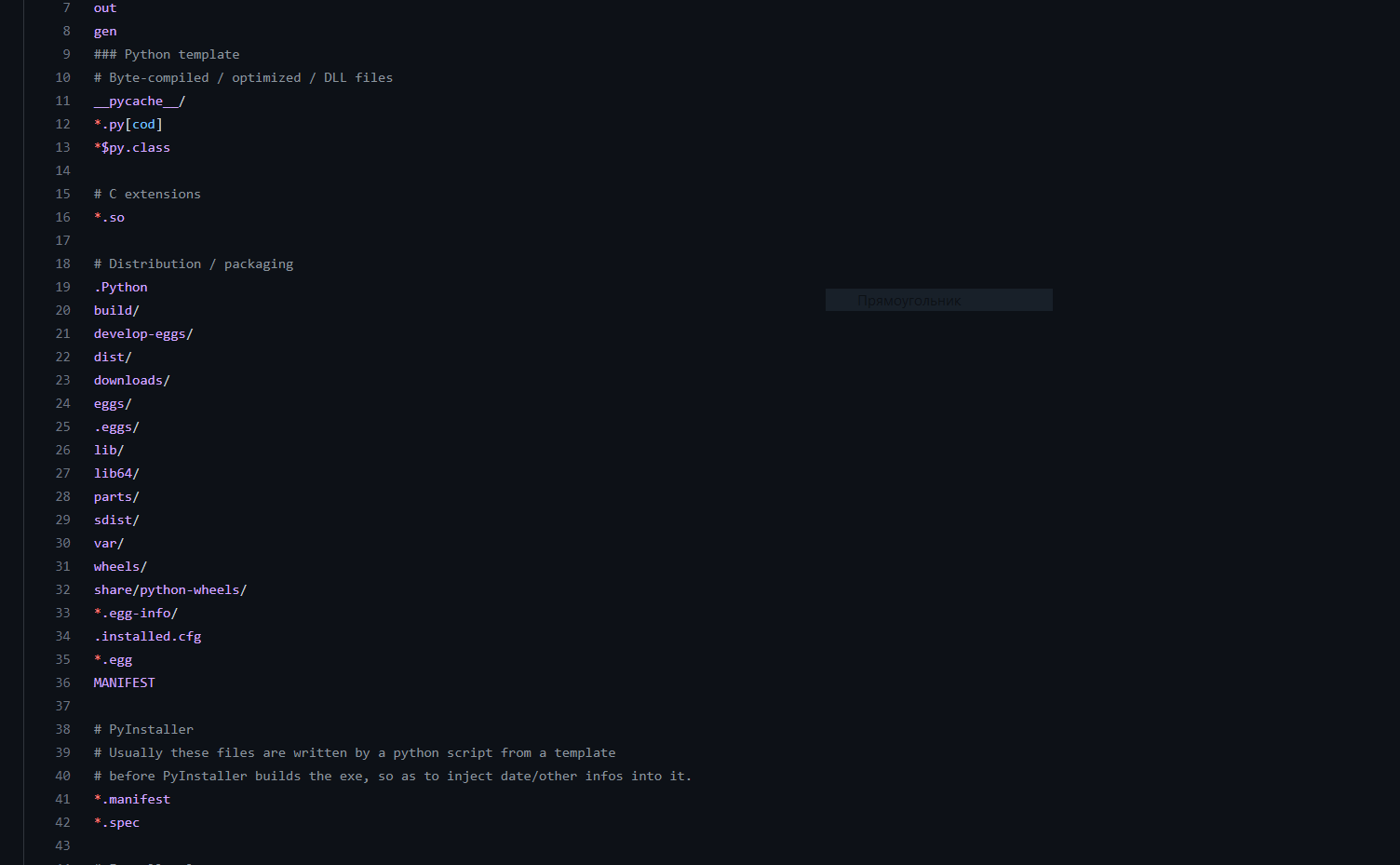


Рисунок 4 – изменение файла gitignor

Код общего задания:

#!/usr/bin/env python3

# -\*- coding: utf-8 -\*-

import sys

from datetime import date

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

# Список работников.

workers = []

# Организовать бесконечный цикл запроса команд.

while True:

# Запросить команду из терминала.

command = input(">>> ").lower()

# Выполнить действие в соответствие с командой.

if command == 'exit':

break

elif command == 'add':

# Запросить данные о работнике.

name = input("Фамилия и инициалы? ")

post = input("Должность? ")

year = int(input("Год поступления? "))

# Создать словарь.

worker = {

'name': name,

'post': post,

'year': year,

}

# Добавить словарь в список.

workers.append(worker)

# Отсортировать список в случае необходимости.

if len(workers) > 1:

workers.sort(key=lambda item: item.get('name', ''))

elif command == 'list':

# Заголовок таблицы.

line = '+-{}-+-{}-+-{}-+-{}-+'.format(

'-' \* 4,

'-' \* 30,

'-' \* 20,

'-' \* 8

)

print(line)

print(

'| {:^4} | {:^30} | {:^20} | {:^8} |'.format(

"№",

"Ф.И.О.",

"Должность",

"Год"

)

)

print(line)

# Вывести данные о всех сотрудниках.

for idx, worker in enumerate(workers, 1):

print(

'| {:>4} | {:<30} | {:<20} | {:>8} |'.format(

idx,

worker.get('name', ''),

worker.get('post', ''),

worker.get('year', 0)

)

)

print(line)

elif command == 'select':

# Получить текущую дату.

today = date.today()

# Разбить команду на части для выделения номера года.

parts = command.split(' ', maxsplit=1)

# Получить требуемый стаж.

period = int(parts[1])

# Инициализировать счетчик.

count = 0

# Проверить сведения работников из списка.

for worker in workers:

if today.year - worker.get('year', today.year) >= period:

count += 1

print(

'{:>4}: {}'.format(count, worker.get('name', ''))

)

# Если счетчик равен 0, то работники не найдены.

if count == 0:

print("Работники с заданным стажем не найдены.")

elif command == 'help':

# Вывести справку о работе с программой.

print("Список команд:\n")

print("add - добавить работника;")

print("list - вывести список работников;")

print("select <стаж> - запросить работников со стажем;")

print("help - отобразить справку;")

print("exit - завершить работу с программой.")

else:

print(f"Неизвестная команда {command}", file=sys.stderr)

Код программы идз 1:

#!/usr/bin/env python3

# -\*- coding: utf-8 -\*-

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

school = {

"1А": 32,

"1Б": 25,

"2А": 22,

"2Б": 31,

"3А": 32,

"3Б": 18,

}

for key, value in school.items():

print(f" В {key} классе количество детей = {value}.")

# Часть а)

print("\nВ одном из классов поменялось количество детей, теперь:\n")

school['2Б'] = 16

for key, value in school.items():

print(f" В {key} классе количество детей = {value}.")

# Часть б)

print("\nПоявился новый класс, теперь:\n")

school.setdefault("3В", 31)

for key, value in school.items():

print(f" В {key} классе количество детей = {value}.")

# Часть с)

print("\nРасформировали один класс, теперь:\n")

school.pop("1А")

for key, value in school.items():

print(f" В {key} классе количество детей = {value}.")

count = 0

for value in school.values():

count += value

print(f"\nВсего учеников в школе - {count}")

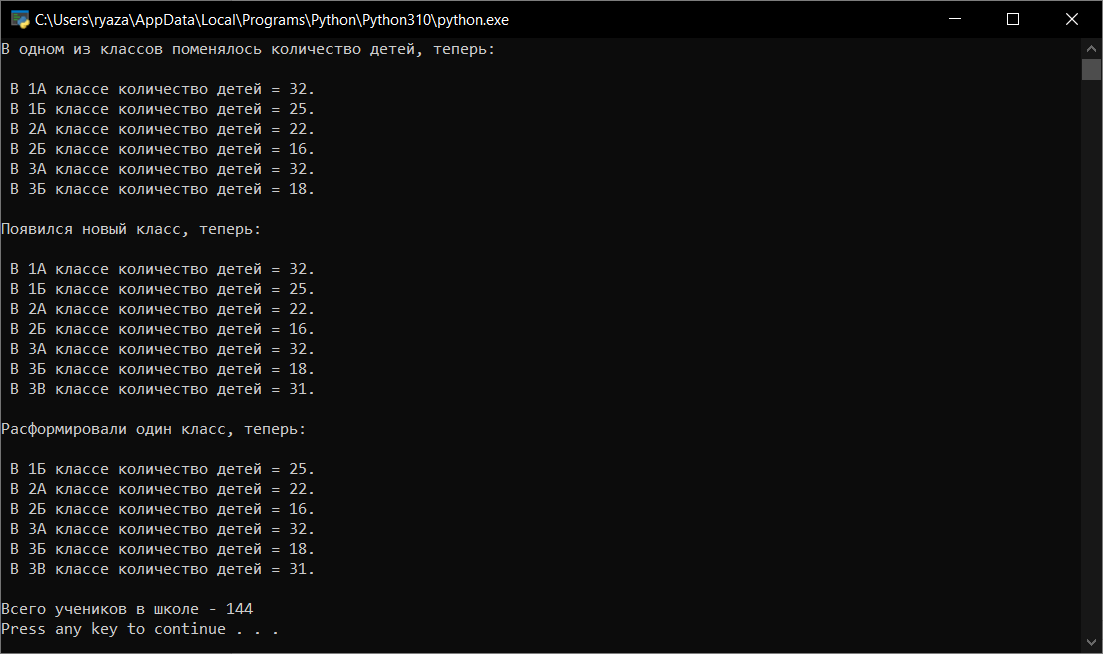


Рисунок 5 – результат работы программы

Код программы идз 2:

#!/usr/bin/env python3

# -\*- coding: utf-8 -\*-

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

some\_dict = {

1: "abc",

2: "home",

3: "test",

4: "task"

}

print(f"Словарь до изменений:\n{some\_dict}")

dict\_items = some\_dict.items()

changed\_dict = {i: j for j, i in dict\_items}

print(f"Словарь после изменений:\n{changed\_dict}")

Рисунок 12 – код программы идз 2

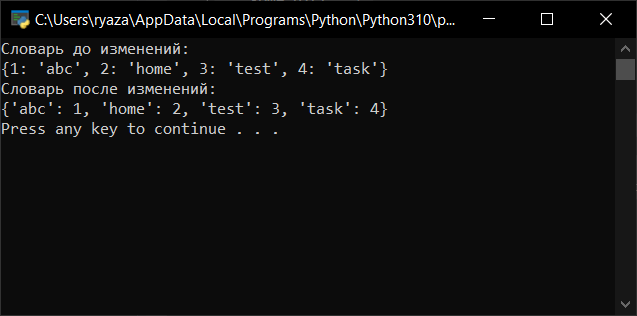
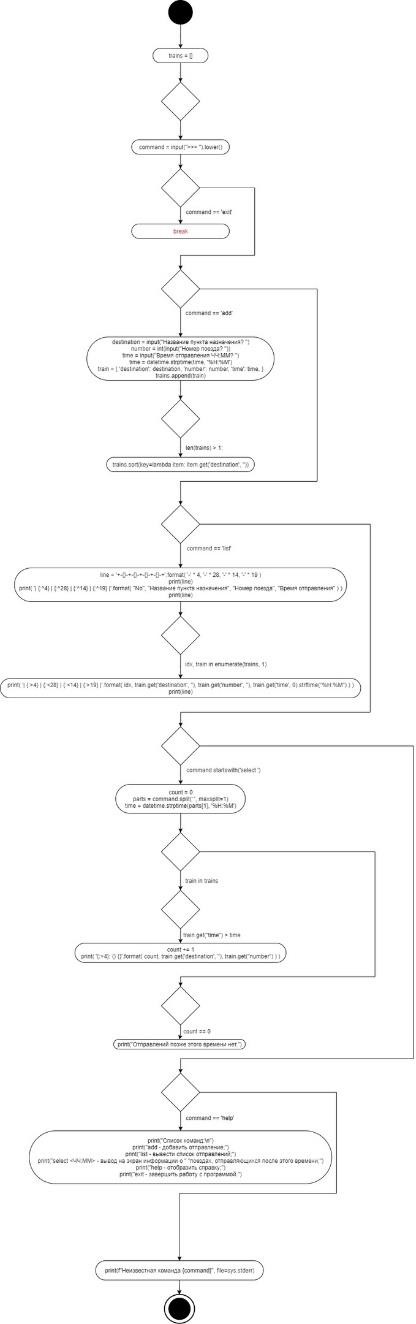


Рисунок 6 – результат работы программы

Индивидуальное задание

Рисунок 7 – uml диаграмма

Код программы:

#!/usr/bin/env python3

# -\*- coding: utf-8 -\*-

import sys

from datetime import date

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

# Список работников.

pep = []

# Организовать бесконечный цикл запроса команд.

while True:

# Запросить команду из терминала.

command = input(">>> ").lower()

# Выполнить действие в соответствие с командой.

if command == 'exit':

break

elif command == 'add':

# Запросить данные о работнике.

name = input("name faname? ")

num = int(input("number? "))

br = int(input("burftday? "))

# Создать словарь.

chel = {

'name': name,

'num': num,

'br': br,

}

# Добавить словарь в список.

pep.append(chel)

# Отсортировать список в случае необходимости.

if len(pep) > 1:

pep.sort(key=lambda item: item.get('br',''))

elif command == 'list':

# Заголовок таблицы.

line = '+-{}-+-{}-+-{}-+-{}-+'.format(

'-' \* 4,

'-' \* 30,

'-' \* 20,

'-' \* 8

)

print(line)

print(

'| {:^4} | {:^30} | {:^20} | {:^8} |'.format(

"№",

"F.I.O.",

"NUMBER",

"BRDAY"

)

)

print(line)

# Вывести данные о всех сотрудниках.

for idx, chel in enumerate(pep, 1):

print(

'| {:>4} | {:<30} | {:<20} | {:>8} |'.format(

idx,

chel.get('name', ''),

chel.get('num', ''),

chel.get('br', 0)

)

)

print(line)

elif command == 'select':

# Получить требуемый стаж.

zapros = int(input("zapros po numeru "))

# Инициализировать счетчик.

count = 0

# Проверить сведения работников из списка.

for chel in pep:

if chel.get('num') == zapros:

count += 1

print(

'{:>4}: {}'.format(count, chel.get('name', ''))

)

# Если счетчик равен 0, то работники не найдены.

if count == 0:

print("cheela s takim nomerom net")

elif command == 'help':

# Вывести справку о работе с программой.

print("Список команд:\n")

print("add - add chel;")

print("list - show list of pep;")

print("select <стаж> - запросить работников со стажем;")

print("help - отобразить справку;")

print("exit - завершить работу с программой.")

else:

print(f"Неизвестная команда {command}", file=sys.stderr)

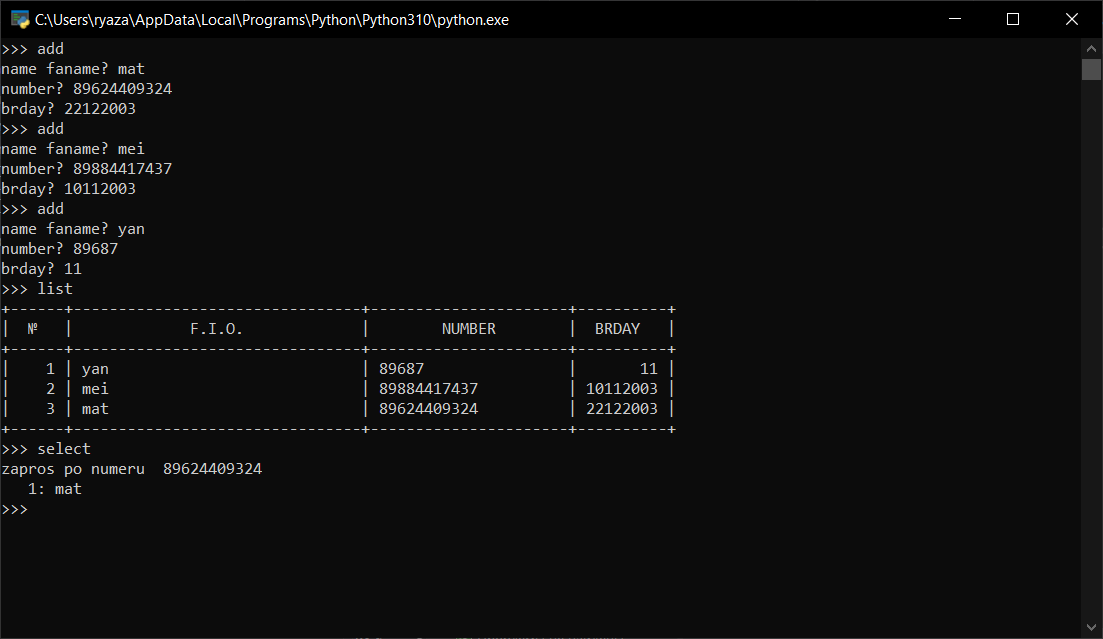


Рисунок 8 – результат работы программы идз

**Контрольные вопросы**

1. Что такое словари в языке Python?

Словари в Python – это изменяемые отображения ссылок на объекты, доступные по ключу.

2. Может ли функция len() быть использована при работе со словарями?

Функция len() возвращает длину (количество элементов) в объекте. Аргумент может быть последовательностью, такой как строка, байты, кортеж, список или диапазон или коллекцией (такой как словарь, множество или неизменяемое множество).

3. Какие методы обхода словарей Вам известны?

Самый очевидный вариант обхода словаря — это попытаться напрямую запустить цикл for по объекту словаря, так же как мы делаем это со списками, кортежами, строками и любыми другими итерируемыми объектами. for something in currencies: print(something)

4. Какими способами можно получить значения из словаря по ключу?

С помощью метода .get()

5. Какими способами можно установить значение в словаре по ключу?

С помощью функции dict.update()

6. Что такое словарь включений?

Словарь включений аналогичен списковым включениям, за исключением того, что он создаёт объект словаря вместо списка.

7. Самостоятельно изучите возможности функции zip() приведите примеры ее использования.

Функция zip() в Python создает итератор, который объединяет элементы из нескольких источников данных. Эта функция работает со списками, кортежами, множествами и словарями для создания списков или кортежей, включающих все эти данные. Предположим, что есть список имен и номером сотрудников, и их нужно объединить в массив кортежей. Для этого можно использовать функцию zip(). Вот пример программы, которая делает именно это:

employee\_numbers = [2, 9, 18, 28]

employee\_names = ["Дима", "Марина", "Андрей", "Никита"]

zipped\_values = zip(employee\_names, employee\_numbers)

zipped\_list = list(zipped\_values)

print(zipped\_list)

Функция zip возвращает следующее:

[('Дима', 2), ('Марина', 9), ('Андрей', 18), ('Никита', 28)]

8. Самостоятельно изучите возможности модуля datetime. Каким функционалом по работе с датой и временем обладает этот модуль? Datetime — важный элемент любой программы, написанной на Python. Этот модуль позволяет управлять датами и временем, представляя их в таком виде, в котором пользователи смогут их понимать.

datetime включает различные компоненты. Так, он состоит из объектов следующих типов:

 date — хранит дату

 time — хранит время

 datetime — хранит дату и время

Как получить текущие дату и время?

import datetime

dt\_now = datetime.datetime.now()

print(dt\_now)

Результат:

2022-09-11 15:43:32.249588

Получить текущую дату:

from datetime import date

current\_date = date.today()

print(current\_date)

Результат:

2022-09-11

Получить текущее время:

import datetime

current\_date\_time = datetime.datetime.now()

current\_time = current\_date\_time.time()

print(current\_time)

Результат:

15:51:05.627643