

**МИНИСТЕРСТВО 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 №2.4

Дисциплина: «Программирование на Python» Тема: «Работа со
словарями в языке Python»

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

Богдашов Артём
Владимирович

Ставрополь 2022



Выполнение работы:

1. Создал репозиторий в GitHub «rep 2.6» в который добавил .gitignore, который дополнил правила для работы с IDE PyCharm с ЯП Python, выбрал лицензию MIT, клонировал его на лок. сервер и организовал в соответствии с моделью ветвления git-flow.

Create a new repository


A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)


Owner * **Repository name ***

 Artem0622 / lab2_6 

Great repository names are short and memorable. Need inspiration? How about [furry-telegram](#)?

Description (optional)


☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.


Initialize this repository with:
Skip this step if you're importing an existing repository.


☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: Python 

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

License: MIT License 

 You are creating a public repository in your personal account.

[Create repository](#)

Рисунок 1.1 Создание репозитория

```
C:\Games\Программы\СУ\1.3\Программирование на Python\LR1_1>git clone https://github.com/Artem0622/lab2_6.git
Cloning into 'lab2_6'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 7 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (7/7), 4.36 KiB | 893.00 KiB/s, done.
Resolving deltas: 100% (1/1), done.
```

Рисунок 1.2 Клонирование репозитория

```

1 contributor
276 lines (220 sloc) | 5.57 KB

1  idea/
2  # Created by https://www.toptal.com/developers/gitignore/api/python,pycharm
3  # Edit at https://www.toptal.com/developers/gitignore?templates=python,pycharm
4
5  ### PyCharm ###
6  # Covers JetBrains IDEs: IntelliJ, RubyMine, PhpStorm, AppCode, PyCharm, CLion, Android Studio, WebStorm and Rider
7  # Reference: https://intellij-support.jetbrains.com/hc/en-us/articles/206544839
8
9  # User-specific stuff
10 .idea/**/workspace.xml
11 .idea/**/tasks.xml
12 .idea/**/usage.statistics.xml
13 .idea/**/dictionaries
14 .idea/**/shelf
15
16 # AWS User-specific
17 .idea/**/aws.xml
```

Рисунок 1.3 Изменение .gitignore

```

C:\Games\Программы\CV\1.3\Программирование на Python\LR1_1\lab2_6>git flow init
which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]
How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/Games/Программы/CV/1.3/Программирование на Python/LR1_1/lab2_6/.git/hooks]

```

Рисунок 1.4 Организация репозитория в соответствии с моделью ветвления git-flow

Пример 1. Использовать словарь, содержащий следующие ключи: фамилия и инициалы работника; название занимаемой должности; год поступления на работу. Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из заданных словарей; записи должны быть размещены по алфавиту; вывод на дисплей фамилий работников, чей стаж работы в организации превышает значение, введенное с клавиатуры; если таких работников нет, вывести на дисплей соответствующее сообщение.

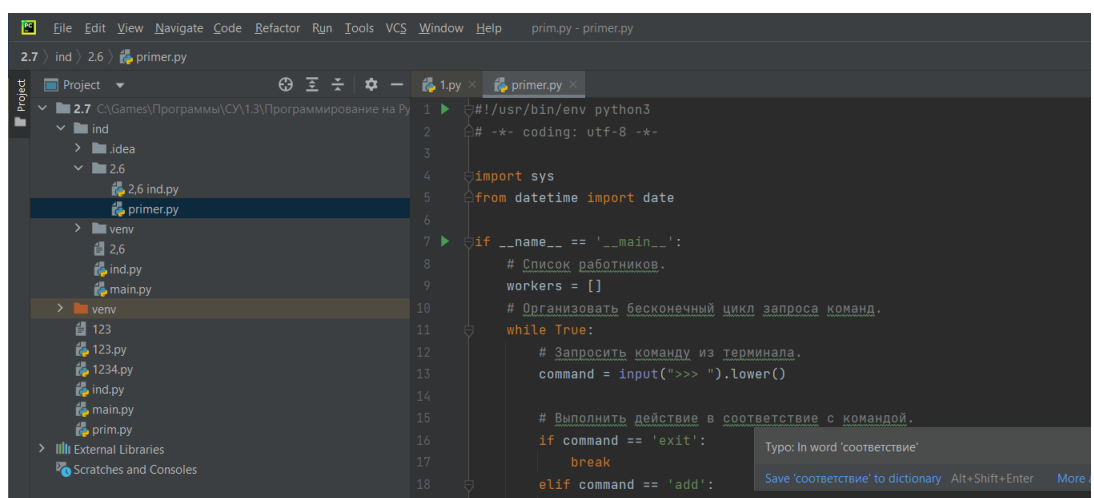


Рисунок 2.1 Создание проекта в PyCharm

```
>>> help
Список команд:

add - добавить работника;
list - вывести список работников;
select <стаж> - запросить работников со стажем;
help - отобразить справку;
exit - завершить работу с программой.

>>> add
Фамилия и инициалы? V Artue
Должность? Root
Год поступления? 2004
>>> list
+-----+-----+-----+-----+
| № |          Ф.И.О.          | Должность | Год |
+-----+-----+-----+-----+
| 1 | V Artue                  | Root      | 2004 |
+-----+-----+-----+-----+

>>> list
>>> Неизвестная команда 150
select 150
Работники с заданным стажем не найдены.
>>> select 10
1: V Artue
>>> |
```

Рисунок 2.2 Результат выполнения программы

Индивидуальное задание Вариант 3

3. Использовать словарь, содержащий следующие ключи: фамилия и инициалы; номер группы; успеваемость (список из пяти элементов). Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть упорядочены по алфавиту; вывод на дисплей фамилий и номеров групп для всех студентов, имеющих хотя бы одну оценку 2; если таких студентов нет, вывести соответствующее сообщение.

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import sys
5
6
7 if __name__ == '__main__':
8     print("help - список всех команд")
9     # список людей.
10    students = []
11    count = 0
12    # Организовать бесконечный цикл запроса команд.
13    while True:
14        # Запросить команду из терминала.
15        command = input(">>> ").lower()
16        # Выполнить действие в соответствие с командой.
17        if command == 'exit':
18            break
19
20        elif command == 'add':
21            # Запросить данные о студентах.
22            name = input("Фамилия и инициалы ")
23            number = input("Номер группы ")
24            performance = input("Успеваемость ")
25            # Создать словарь.
26            student = {
27                'name': name,
28                'number': number,
29                'performance': performance,
30            }
```

```

help - список всех команд
>>> Иванов Иван Иванович
>>> Неизвестная команда иванов иван иванович
add
Фамилия и инициалы Иванов Иван Иванович
Номер группы 01
Успеваемость 4
>>> Гуляницкий Александр Евгеньевич
>>> Неизвестная команда гуляницкий александр евгеньевич
add
Фамилия и инициалы Гуляницкий Александр Евгеньевич
Номер группы 02
Успеваемость 4
>>> select
* Гуляницкий Александр Евгеньевич группа № 02
>>> list
>>> Неизвестная команда list
list
+-----+-----+-----+-----+
| № |          Ф.И.О.          | Номер группы | Успеваемость |
+-----+-----+-----+-----+
| 1 | Гуляницкий Александр Евгеньевич | 02          | 4            |
| 2 | Иванов Иван Иванович          | 01          | 4            |
+-----+-----+-----+-----+
>>>

```

Рисунок 2.3 Вывод программы задания

```

>>> add
Фамилия и инициалы Гуляницкий Александр Евгеньевич
Номер группы 02
Успеваемость 5
>>> list
+-----+-----+-----+-----+
| № |          Ф.И.О.          | Номер группы | Успеваемость |
+-----+-----+-----+-----+
| 1 | Гуляницкий Александр Евгеньевич | 02          | 5            |
+-----+-----+-----+-----+
>>> select
Таких студентов нет
>>>

```

Рисунок 2.4 Вывод программы индивидуального задания

3.Сделал коммит, выполнил слияние с веткой main, и запустил изменения в уд. репозиторий.

```

C:\Games\Программы\СУ\1.3\Программирование на Python\2.6\lab2_6>git add .
C:\Games\Программы\СУ\1.3\Программирование на Python\2.6\lab2_6>git commit -m "save program"
[develop 3217f3e] save program
 3 files changed, 339 insertions(+), 3 deletions(-)
 create mode 100644 ind.py
 create mode 100644 primer.py
C:\Games\Программы\СУ\1.3\Программирование на Python\2.6\lab2_6>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

```

Рисунок 3.1 Коммит изменений

```

C:\Games\Программы\СУ\1.3\Программирование на Python\2.6\lab2_6>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

```

Рисунок 3.2 Переход на ветку main

```
C:\Games\Программы\СУ\1.3\Программирование на Python\2.6\lab2_6>git merge develop
Updating 2c28c7e..3217f3e
Fast-forward
 .gitignore | 153 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 ind.py      | 90 ++++++++++++++++++++++++++++++++++++++
 primer.py   | 99 ++++++++++++++++++++++++++++++++++++++
 3 files changed, 339 insertions(+), 3 deletions(-)
 create mode 100644 ind.py
 create mode 100644 primer.py
```

Рисунок 3.3 Слияние ветки main с develop

```
C:\Games\Программы\СУ\1.3\Программирование на Python\2.6\lab2_6>git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 4.86 KiB | 4.86 MiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/Artem0622/lab2_6.git
 2c28c7e..3217f3e main -> main
```

Рисунок 3.4 Пуш изменений

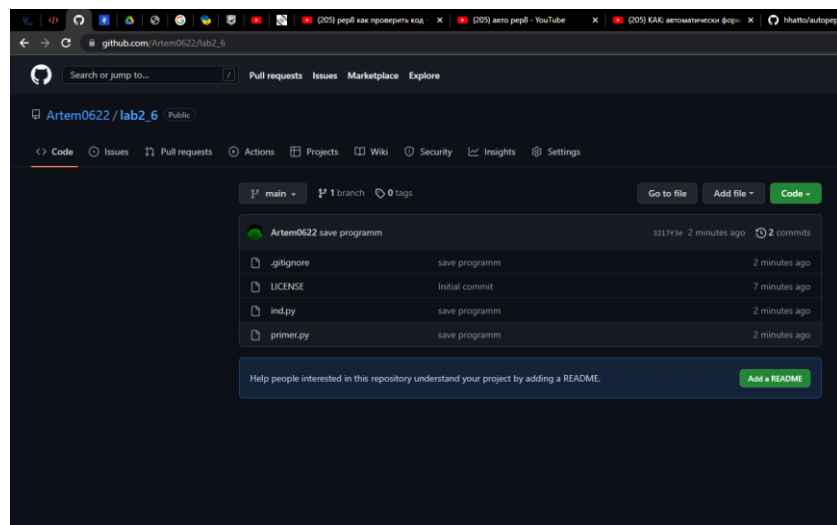


Рисунок 3.5 Изменение на уд сервере

Контр. вопросы и ответы на них:

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