

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

**Основы кроссплатформенного
программирования Отчет по лабораторной
работе №2.16**

Тема: «Работа с данными формата JSON в языке Python»

Выполнил студент группы

ИВТ-б-о-21-1

Назаров Н.Ю. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил доцент

Кафедры инфокоммуникаций, старший
преподаватель

Воронкин Р.А.

(подпись)

Выполнение работы



Рисунок 1 – редактирование gitignore

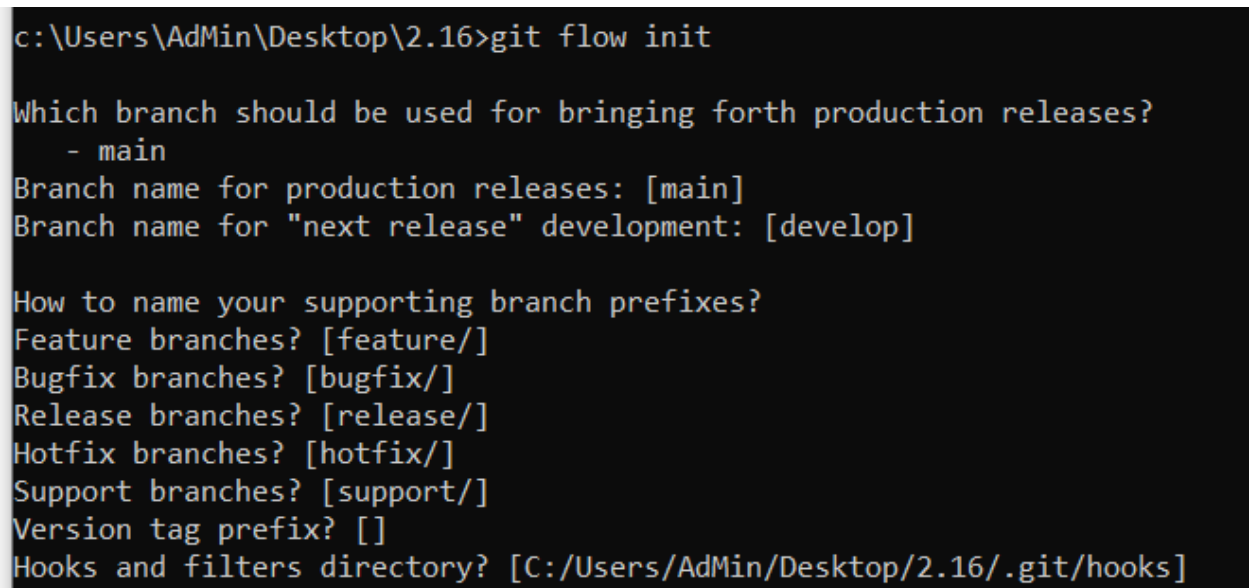


Рисунок 2 – организация репозитория в соответствии с моделью git-flow

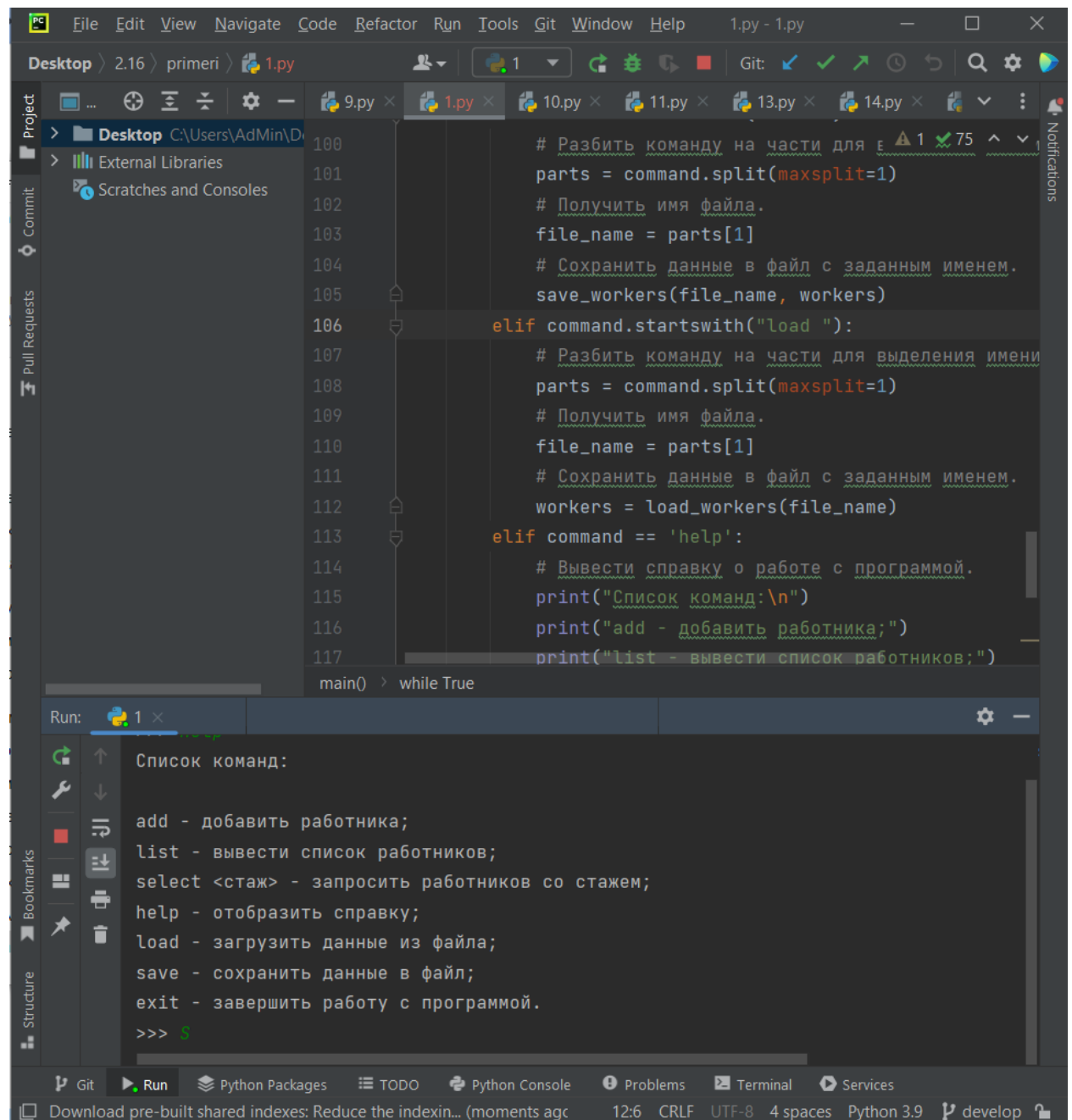


Рисунок 3 – проработал 1 пример

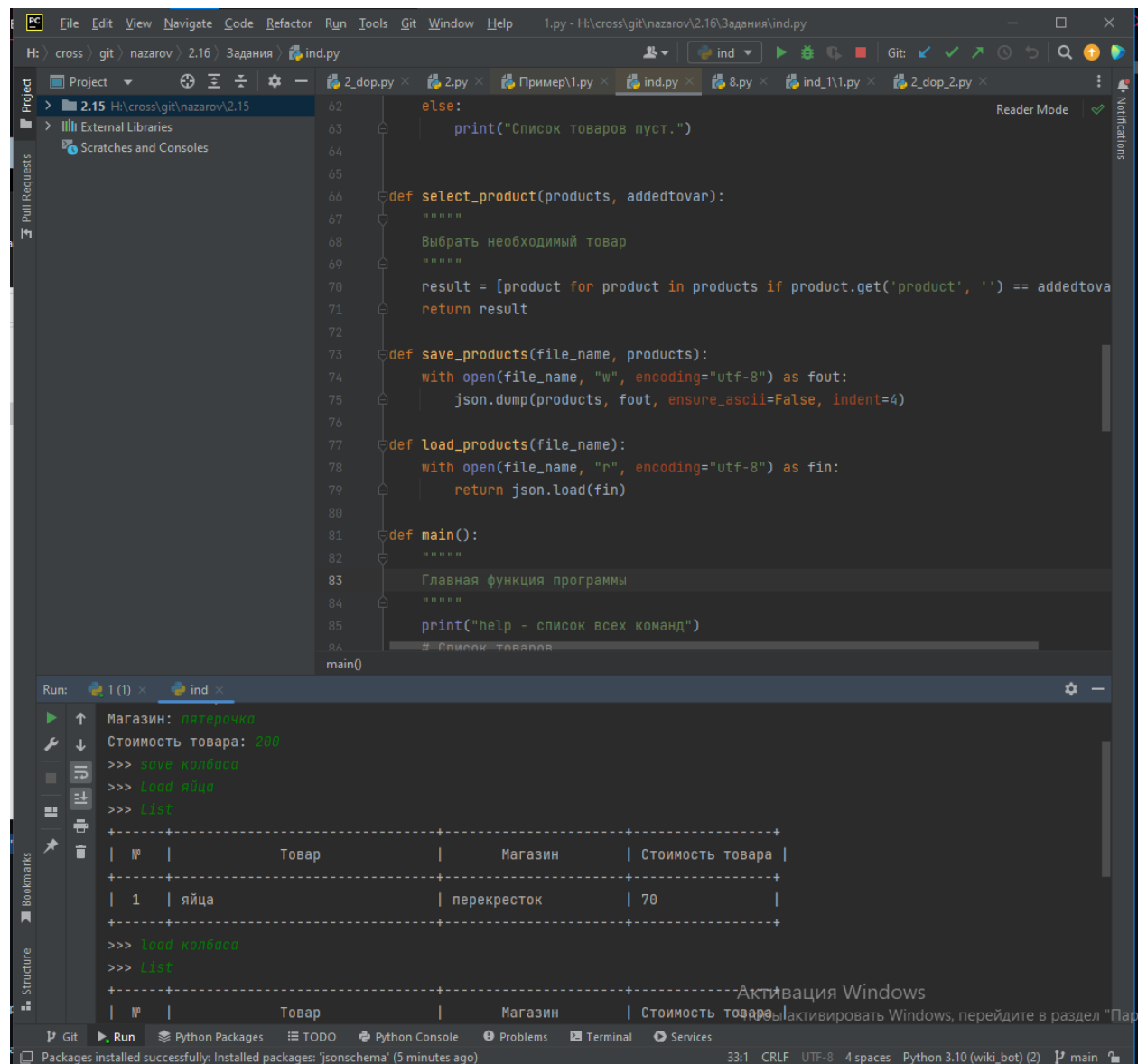


Рисунок 5 – выполнил 1 индивидуальное задание

The screenshot shows the PyCharm IDE with a Python script named `ind_2.py` open. The script defines functions for saving and loading products. The `load_products` function uses a JSON schema to validate the loaded data. The Run window at the bottom shows the execution of the script, which successfully loads and displays a list of products.

```
70 result = [product for product in products if product.get('product',  
71 return result  
72  
73 def save_products(file_name, products):  
74     with open(file_name, "w", encoding="utf-8") as fout:  
75         json.dump(products, fout, ensure_ascii=False, indent=4)  
76  
77 def load_products(file_name):  
78     schema = {  
79         "type": "array",  
80         "items": [  
81             {  
82                 "type": "object",  
83                 "properties": {  
84                     "product": {  
85                         "type": "string"  
86                     },  
87                     "shop": {  
88                         "type": "string"  
89                     },  
90                     "cost": {  
91                         "type": "integer",  
92                     }  
93                 }  
94             }  
95         ]  
96     }  
97     load_products()
```

Run: 1 (1) × ind_2 ×

```
>>> add  
Название товара: колбаса  
Магазин: пятерочка  
Стоимость товара: 220  
>>> save колбаса  
>>> load колбаса  
Валидация прошла успешно  
>>> list  
+-----+-----+-----+-----+  
| № | Товар | Магазин | Стоимость товара |  
+-----+-----+-----+-----+  
| 1 | колбаса | пятерочка | 220 |  
+-----+-----+-----+-----+  
>>> |
```

Пакеты установлены успешно: Installed packages: 'jsonschema' (25 minutes ago) 16:5 CRLF UTF-8 4 spaces Python 3.10 (wiki_bot) (2) main

Рисунок 6 – выполнил 2 индивидуальное задание

Ответы на контрольные вопросы:

1. Для чего используется JSON?

JSON (англ. JavaScript Object Notation, обычно произносится как JAY-sən) – текстовый формат обмена данными, основанный на JavaScript. Как и многие другие текстовые форматы, JSON легко читается людьми.

2. Какие типы значений используются в JSON?

Набор пар ключ: значение. Упорядоченный набор значений.

3. Как организована работа со сложными данными в JSON?

JSON может содержать другие вложенные объекты в JSON, в дополнение к вложенным массивам. Такие объекты и массивы будут передаваться, как

значения назначенные ключам и будут представлять собой связку ключ-значение.

4. Самостоятельно ознакомьтесь с форматом данных JSON5? В чем отличие этого формата от формата данных JSON?

Формат обмена данными JSON5 (JSON5) - это надмножество JSON, целью которого является смягчение некоторых ограничений JSON путем расширения его синтаксиса для включения некоторых продуктов из ECMAScript 5.1. Эта библиотека JavaScript является официальной эталонной реализацией библиотек синтаксического анализа и сериализации JSON5.

Краткое описание возможностей. Следующие функции ECMAScript 5.1, которые не поддерживаются в JSON, были расширены до JSON5. Объекты.

Ключи объекта могут быть идентификатором ECMAScript 5.1.

5. Какие средства языка программирования Python могут быть использованы для работы с данными в формате JSON5?

Реализация Python формата данных JSON5.

JSON5 расширяет формат обмена данными JSON, чтобы сделать его более удобным для использования в качестве языка конфигурации:

Комментарии в стиле JavaScript (как однострочные, так и многострочные) разрешены.

Ключи объектов могут быть без кавычек, если они являются допустимыми идентификаторами ECMAScript.

Объекты и массивы могут заканчиваться запятыми.

Строки могут заключаться в одинарные кавычки, допускаются многострочные строковые литералы.

Есть еще несколько более мелких расширений JSON; см. полную информацию на странице выше.

Этот проект реализует реализацию чтения и записи для Python; где возможно, он отражает стандартный пакет Python JSON API для простоты использования.

Есть одно заметное отличие от JSON api: методы `load()` и `load()` поддерживают опциональную проверку (и отклонение) повторяющихся ключей объекта; `pass allow_duplicate_keys = False` для этого (по умолчанию разрешены дубликаты).

Это ранний выпуск. Это было достаточно хорошо протестировано, но это *МЕДЛЕННО*. Он может быть в 1000-6000 раз медленнее, чем модуль JSON, оптимизированный для C, и в 200 раз (или более) медленнее, чем модуль JSON на чистом Python.

6. Какие средства предоставляет язык Python для сериализации дан-ных в формате JSON?

`json.dump()` # конвертировать python объект в json и записать в файл
`json.dumps()` # тоже самое, но в строку.

7. В чем отличие функций `json.dump()` и `json.dumps()`?

Dumps записывает в строку, а dump в файл.

8. Какие средства предоставляет язык Python для десериализации данных из формата JSON?

`json.load()` # прочитать json из файла и конвертировать в python объект
`json.loads()` # тоже самое, но из строки с json (s на конце от string/строка)

9. Какие средства необходимо использовать для работы с данными формата JSON, содержащими кириллицу?

```
import codecs  
  
json.load(codecs.open('sample.json', 'r', 'utf-8-sig'))
```

10. Самостоятельно ознакомьтесь со спецификацией JSON Schema? Что такое схема данных?

Схема JSON - это словарь, который позволяет аннотировать и проверять документы JSON.

Преимущества:

- Описывает ваш существующий формат (ы) данных.
- Предоставляет понятную документацию, читаемую человеком и машиной.
- Проверяет данные, которые полезны для:

- Автоматизированное тестирование.
- Обеспечение качества предоставленных клиентом данных.