

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Северо-Кавказский федеральный университет»**

**Кафедра инфокоммуникаций**

**Отчёт по практическому занятию №3.1  
«Работа с IPython и Jupyter Notebook»**

**по дисциплине «Теории распознавания образов»**

Выполнил студент группы ПИЖ-б-о-21-1  
Образцова М.Д. « » \_\_\_\_\_ 20\_\_ г.  
Подпись студента \_\_\_\_\_  
Работа защищена « » \_\_\_\_\_ 20\_\_ г.  
Проверил Воронкин Р.А. \_\_\_\_\_  
(подпись)

Ставрополь 2023

## Выполнение работы

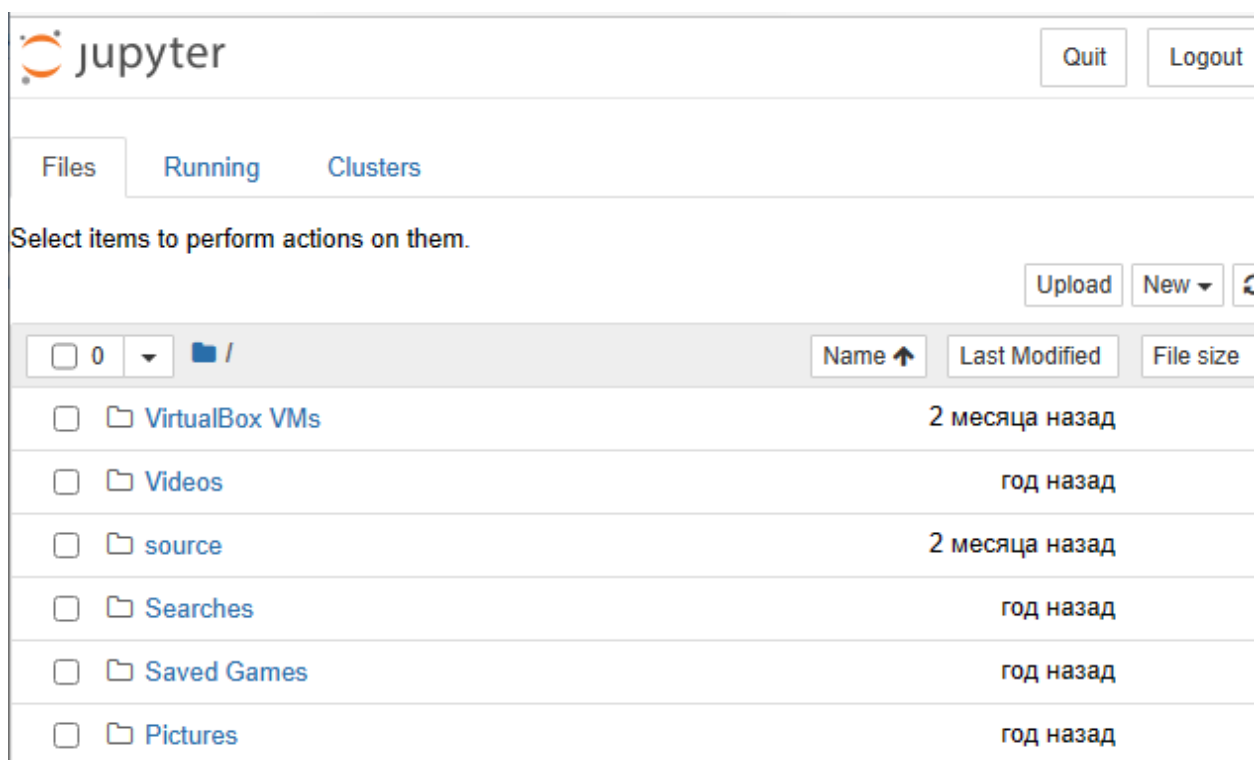


Рисунок 1 – Пример установки и запуска ноутбука

```
In [1]: 2+3
```

```
Out[1]: 5
```

```
In [2]: f = 5  
a = 3  
print(a+f)
```

```
8
```

```
In [3]: n = 7  
for i in range(n):  
    print(i*10)
```

```
0  
10  
20  
30  
40  
50  
60
```

---

```
In [1]: i = 0  
while True:  
    i += 1  
    if i > 5:  
        break  
    print("test while")
```

```
test while  
test while  
test while  
test while  
test while
```

Рисунок 2 – Примеры

1. Решить задания в ноутбуках, выполненных преподавателем.

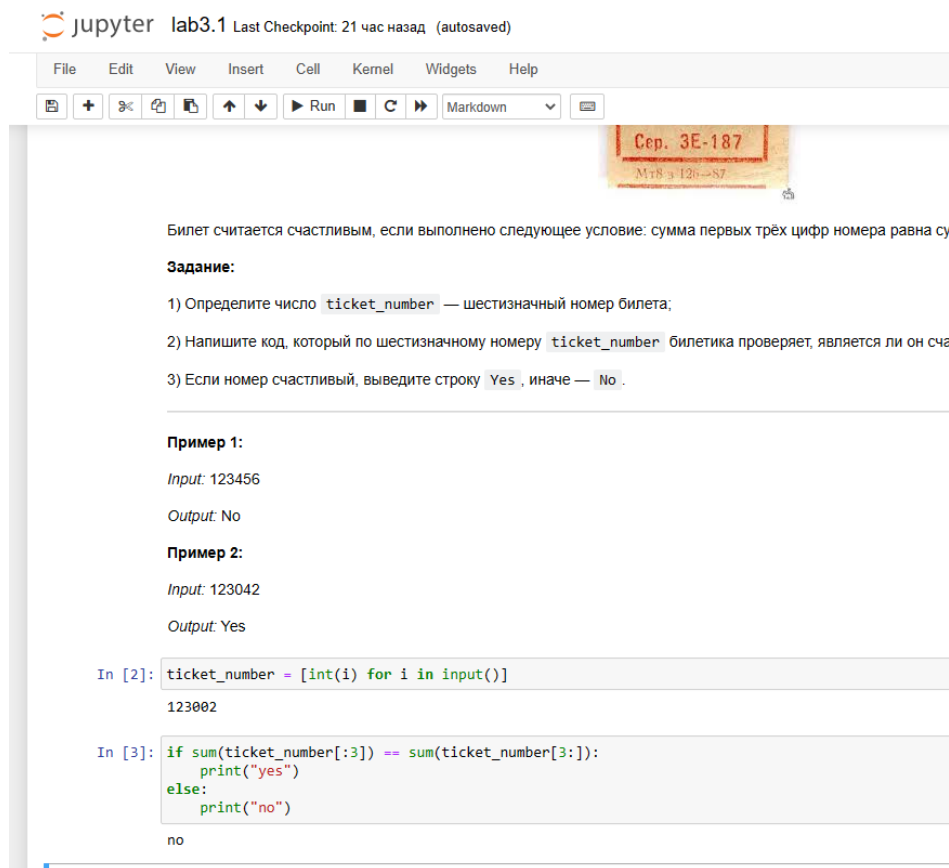


Рисунок 3 – задания в ноутбуках

2. Создать ноутбук, в котором выполнить решение вычислительной задачи (например, задачи из области физики, экономики, математики, статистики и т. д.), условие которой предварительно необходимо согласовать с преподавателем.

коллекционер номеров семья заехала в отель с совершенно разными номерами и разными условиями(темами). суть в том, что нельзя выбрать номер. очередной номер комнаты выбирается произвольно. цель семьи - побывать во всех номерах отеля.

ВОПРОС сколько раз нужно приехать, чтобы побывать во всех номерах (есть вероятность заселения в один номер более одного раза)

```
In [1]: import random
```

```
In [5]: n = int(input('количество номеров отеля '))
```

```
has_been_visited = {}
all_count = ind_count = 0
for i in range(n):
    has_been_visited[i] = False
```

количество номеров отеля 1000

```
In [6]: while ind_count < n:
        num = random.randint(0, n-1)
        if has_been_visited[num] == False:
            ind_count += 1
            has_been_visited[num] = True
            all_count += 1

print('были просмотрены все номера отеля')
print('кол-во попыток = ', all_count)
```

были просмотрены все номера отеля  
кол-во попыток = 6936

---

## Вопросы для защиты работы

### 1. Как осуществляется запуск Jupyter notebook?

Для запуска Jupyter Notebook перейдите в папку Scripts (она находится внутри каталога, в котором установлена Anaconda) и в командной строке наберите: `ipython notebook`

### 2. Какие существуют типы ячеек в Jupyter notebook?

Code

Markdown

Raw NBConvert

Heading

### 3. Как осуществляется работа с ячейками в Jupyter notebook?

Перед первой строкой написано `In []`. Это ключевое слово значит, что дальше будет ввод. Попробуйте написать простое выражение вывода.

Вывод должен отобразиться прямо в notebook. Это и позволяет заниматься программированием в интерактивном формате, имея возможность отслеживать вывод каждого шага.

Также обратите внимание на то, что `In []` изменилась и вместе нее теперь `In [1]`. Число в скобках означает порядок, в котором эта ячейка будет запущена. В первой цифра 1, потому что она была первой запущенной ячейкой. Каждую ячейку можно запускать индивидуально и цифры в скобках будут менять соответственно.

Если есть несколько ячеек, то между ними можно делиться переменными и импортами. Это позволяет проще разбивать весь код на связанные блоки, не создавая переменную каждый раз. Главное убедиться в запуске ячеек в правильном порядке, чтобы переменные не использовались до того, как были созданы.

### 4. Что такое "магические" команды Jupyter notebook? Какие "магические" команды Вы знаете?

`%ismagic` – список доступных магических команд

`%env` – для работы с переменными окружения

`%run` – для запуска файлов с расширением «.ру»

`%%time` – позволяет получить информацию о времени работы кода в рамках одной ячейки

`%timeit` – запускает переданный ей код 1000000 (по умолчанию) и выводит информацию среднем значении трёх наиболее быстрых прогонах

5. Самостоятельно изучите работу с Jupyter notebook и IDE PyCharm и Visual Studio Code. Приведите основные этапы работы с Jupyter notebook в IDE PyCharm и Visual Studio Code

IDE, которая играет важную роль при разговоре о Python, — это Jupyter Notebook. Ранее известный как IPython Notebook, Jupyter Notebook особенно важен

для придания формы тому, что Дональд Кнут, ученый-компьютерщик из Стэнфорда, назвал «грамотным программированием».

Грамотное программирование — это стандартная форма программирования, ориентированная на удобочитаемость кода. Это позволяет программистам придавать форму логическим единицам своего кода, значению этих единиц кода и их результатам. Скомпилированный блокнот представляет код как законченный и понятный мыслительный процесс и его технологическое воплощение.

Для поддержки грамотного программирования в Jupyter Notebook есть множество доступных инструментов, которые обеспечивают полную свободу редактирования кода с его соответствующей поддерживающей прозой.

Начиная с базового уровня, записные книжки (файлы, в которых написан код) могут разделять код на «ячейки». Ячейки позволяют легко различать определенные функции.

Помимо ячеек кода, доступны ячейки разметки, в которых легко ввести описание кода, значение или результаты. Возможности редактирования ячеек разметки безграничны; вы можете поиграть с текстовыми форматами, изображениями и даже математическими уравнениями и диаграммами.

Обширная поддержка интеграции Jupyter Notebook в PyCharm позволяет разработчикам создавать, выполнять и отлаживать исходные коды, одновременно изучая их выходные данные.

PyCharm позволяет вносить изменения в исходный документ разными способами. Это включает:

- Редактирование и предварительный просмотр.
- Использование записной книжки как исходного кода с определениями в виде текстов.
- Предоставление предварительных просмотров в реальном времени вместе с отладкой.
- Параметры автосохранения вашего кода.
- Выделение всех типов синтаксических ошибок и ошибок.
- Возможность добавлять комментарии к строкам.
- Возможность одновременного выполнения и предварительного просмотра результатов.
- Разрешения на использование специального отладчика Jupyter Notebook Debugger.
- Распознавайте файлы.ipynb по значку.

Для работы с Python в записных книжках Jupyter необходимо активировать среду Anaconda в VS Code или другую среду Python, в которой установлен пакет Jupyter. Для выбора среды используйте команду Python: Select Interpreter из командной палитры (Ctrl+Shift+P).

После активации соответствующей среды можно создать и открыть записную книжку Jupyter, подключиться к удаленному серверу



Jupyter для запуска ячеек кода и экспортировать записную книжку Jupyter в виде файла Python.

