

**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  
Степанов Д. А. «    » \_\_\_\_\_ 20\_\_ г.  
Подпись студента \_\_\_\_\_  
Работа защищена «    » \_\_\_\_\_ 20\_\_ г.  
Проверил Воронкин Р.А. \_\_\_\_\_  
(подпись)

Ставрополь 2023

## Выполнение работы

The screenshot shows the JupyterLab file browser interface. At the top, there is a header bar with the Jupyter logo and the word "jupyter". To the right of the header are two buttons: "Quit" and "Logout". Below the header is a navigation bar with three tabs: "Files", "Running", and "Clusters". The "Files" tab is currently selected. Below the navigation bar is a message: "Select items to perform actions on them." To the right of this message are three buttons: "Upload", "New", and a refresh icon. Below these buttons is a table with a header row and six data rows. The header row has three columns: "Name", "Last Modified", and "File size". The data rows list folders: "VirtualBox VMs", "Videos", "source", "Searches", "Saved Games", and "Pictures". Each folder has a checkbox to its left and a timestamp to its right. The "VirtualBox VMs" folder has a timestamp of "2 месяца назад". The "Videos" folder has a timestamp of "год назад". The "source" folder has a timestamp of "2 месяца назад". The "Searches" folder has a timestamp of "год назад". The "Saved Games" folder has a timestamp of "год назад". The "Pictures" folder has a timestamp of "год назад".

	Name	Last Modified	File size
<input type="checkbox"/>	0		
<input type="checkbox"/>	VirtualBox VMs	2 месяца назад	
<input type="checkbox"/>	Videos	год назад	
<input type="checkbox"/>	source	2 месяца назад	
<input type="checkbox"/>	Searches	год назад	
<input type="checkbox"/>	Saved Games	год назад	
<input type="checkbox"/>	Pictures	год назад	

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.

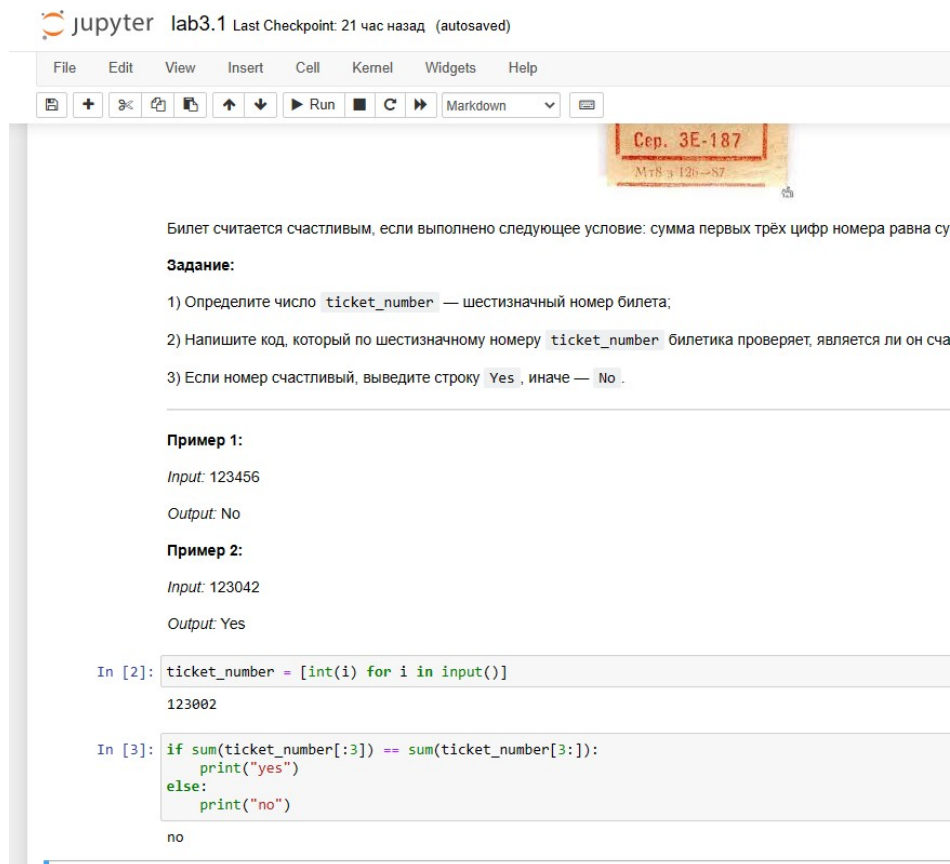
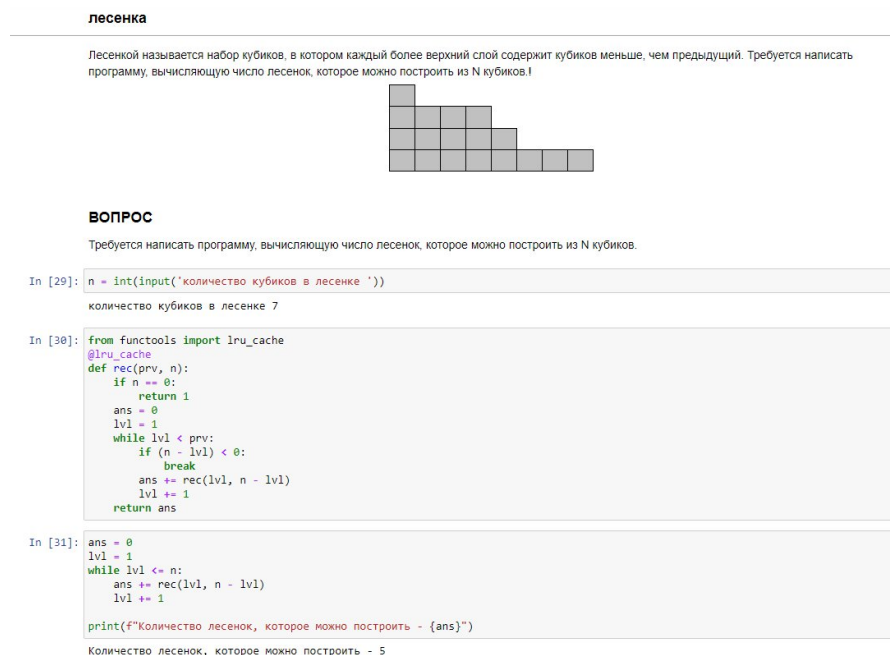


Рисунок 3 – задания в ноутбуках

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.



## 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 о среднем значении трёх наиболее быстрых прогонов

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

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

