## Задание 1. Структуры данных

Реализуйте структуру данных "Односвязный список" на языке программирования по вашему выбору. Необходимо реализовать создание пустого списка, добавление элемента списка, удаление элемента, поиск элемента по номеру и по значению. Желательно использовать ООП. По возможности обеспечить все возможные проверки.

from typing import Any

class Node:

def \_\_init\_\_(self, value: Any, next: 'Node' = None):

self.value = value

self.next = next

class LinkedList:

def \_\_init\_\_(self):

self.\_len = 0

self.first: Node = None

def \_\_len\_\_(self):

return self.\_len

def add(self, value: Any, index: int = None):

if index is None: index = len(self)

if index > len(self): raise RuntimeError()

node = Node(value)

if not len(self):

self.first = node

self.\_len += 1

return

i = 0

cur = self.first

while cur.next is not None and i < index - 1:

cur = cur.next

i += 1

node.next = cur.next

cur.next = node

self.\_len += 1

def get\_by\_value(self, value: Any):

if not len(self): return

cur = self.first

while cur.next is not None and cur.value != value:

cur = cur.next

return cur if cur.value == value else None

def get\_by\_index(self, index: int):

if not len(self): return

cur = self.first

i = 0

while cur.next is not None and i != index:

cur = cur.next

i += 1

return cur if i == index else None

def \_\_repr\_\_(self):

s = '{'

cur = self.first

while cur:

s += f'{cur.value}'

if cur.next:

s += ' -> '

cur = cur.next

s += '}'

return s

a = LinkedList()

print(a)

a.add(1)

print(a)

a.add(2)

a.add(3)

print(a)

a.add(4, 1)

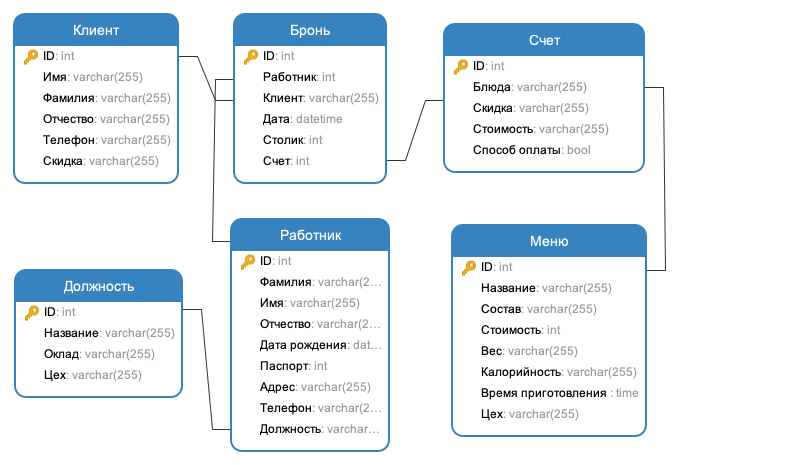
print(a)

print(a.get\_by\_index(2).value)

print(a.get\_by\_value(2).value)

## Задание 2. БД

Спроектировать схему базы данных по текстовому описанию. Обосновать нормальность отношений. Задача «РЕСТОРАН». Постоянным клиентам предоставляется возможность заказать столик заранее. Официант указывает столик, открывает гостевой счет и вводит заказы в соответствии с меню. Далее заказ автоматически обрабатывается, формируются марки на приготовление выбранных блюд и направляют их на производство, в соответствующие цеха кухни, в бар. Расчеты с посетителем сводятся к простой операции: на бланке печатается итоговый счет. Если клиент – постоянный посетитель, то соответствующие привилегии рассчитываются автоматически, затем указываются способ оплаты и полученная от клиента сумма.



В данном случае мы имеем многоуровневую (несколько линий) систему работы ресторана - задается таблицей **Бронь** (Бронирование столика).

Постоянный клиент имеет возможность заказать столик, официант в таблице “Бронь” указывает дату и номер столика, а также открывает клиенту счет. Помимо этого здесь указан сотрудник, обслуживающий стол. В таблице счет ведется учет заказных блюд и формируется их стоимость, с учетом скидки клиента, и указывается способ оплаты. Блюда заказываются из меню, где у каждого указан цех приготовления: кухня или бар.

Связей “многие ко многим” на схеме нет, пояснения по этой части не требуются.

Приведение модели в **1-**ю нормальную форму

Поскольку ещё в процессе анализа мы избегали использования многозначных атрибутов, то в нашей модели нет таблиц, в ячейках которых хранилось бы более одного значения. Следовательно, наша модель уже находится в 1-й нормальной форме (1НФ).

Приведение модели во **2-**ю нормальную форму

Согласно определению 2НФ, каждый неключевой атрибут каждой таблицы должен функционально зависеть от первичного ключа. Последовательно рассматривая каждую таблицу, убеждаемся, что это действительно так.

Приведение модели в **3-**ю нормальную форму

Рассмотрим, зависят ли атрибуты таблиц ТОЛЬКО от первичного ключа, или встречаются ещё и зависимости от других, неключевых атрибутов, называемые транзитивными. Связей с неключевыми полями отсутствуют. Схема приведена в 3НФ.

## Задание 3. Pandas

При помощи библиотеки pandas создать два датафрэйма с индексами 'Moscow', 'Tula', 'Yaroslavl', 'Tver' и 'Moscow', 'Tula', 'Volgograd', 'Novgorod' и случайными значениями в столбцах 'report' (от 1 до 10) и 'sales' (от 100 до 1000). Написать программный код расчёта суммы продаж и суммарное количество отчетов по двум таблицам.

import pandas as pd

import random

def randomizeData(min1, max1, min2, max2):

randomReport = random.sample(range(min1, max1), 4)

randomSales = random.sample(range(min2, max2), 4)

result = []

for i in range(len(randomReport)):

result.append([randomReport[i], randomSales[i]])

return result

df1 = pd.DataFrame(data=randomizeData(1,10,100,1000), index=['Moscow', 'Tula', 'Yaroslavl', 'Tver'], columns=['report','sales'])

print(df1)

df2 = pd.DataFrame(data=randomizeData(1,10,100,1000), index=['Moscow', 'Tula', 'Volgograd', 'Novgorod'], columns=['report','sales'])

print(df2)

df3 = df1.add(df2, fill\_value=0)

answer = df3.sum(axis=0)

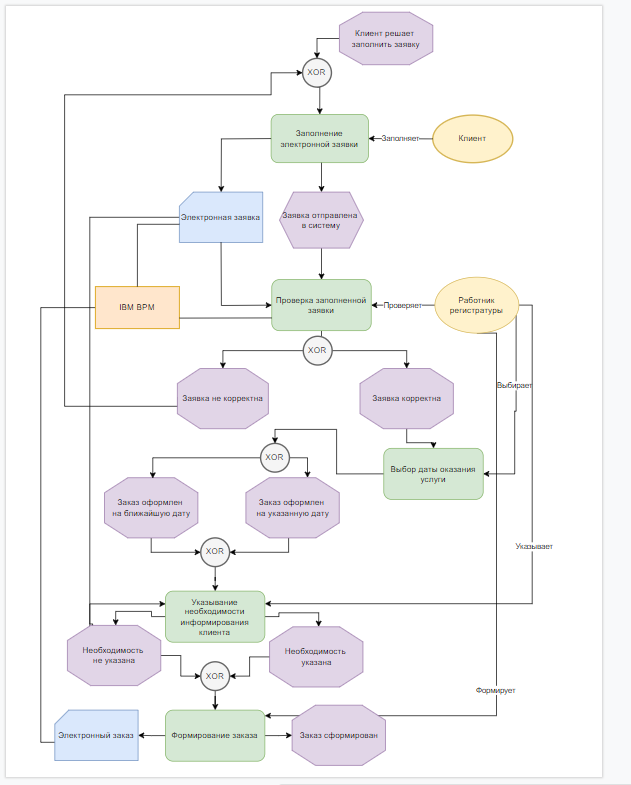
print(df3)

print(answer)

## Задание 4. Стрелочки

Смоделируйте процесс «Оказание услуги» в нотации EPC. Для получения услуги клиент должен заполнить заявку на сайте поставщика услуги, выбрав вид услуги, желаемое время оказания и указав контактные данные (ФИО, телефон). Работник регистратуры, работая в ИС IBM BPM, проверяет заполненную заявку на корректность и в случае правильного ее заполнения формирует заказ, указывая дату и время оказания услуги. Каждый рабочий день компании – поставщика разделен на временные интервалы, которые заполняются заказами на получение услуг. Если все временные интервалы желаемой даты заняты, работник регистратуры ищет свободные интервалы на ближайшие даты. Помимо этого, если клиент указал в заявке корректный номер телефона, то при формировании заказа работник регистратуры указывает в системе необходимость уведомления клиента по СМС (отправка уведомлений осуществляется через СМС - шлюз).

<https://drive.google.com/file/d/1Sd7ld5FngaOgKdcP3VWbTka_xNgOR2C1/view?usp=sharing>



Про нотацию, на экзе будет другая сто проц

[https://www.businessstudio.ru/wiki/docs/v4/doku.php/ru/csdesign/bpmodeling/epc\_notation#:~:text=Нотация%20EPC%20(Event%2DDriven%20Process,упорядоченную%20комбинацию%20событий%20и%20функций](https://www.businessstudio.ru/wiki/docs/v4/doku.php/ru/csdesign/bpmodeling/epc_notation#:~:text=%D0%9D%D0%BE%D1%82%D0%B0%D1%86%D0%B8%D1%8F%20EPC%20(Event%2DDriven%20Process,%D1%83%D0%BF%D0%BE%D1%80%D1%8F%D0%B4%D0%BE%D1%87%D0%B5%D0%BD%D0%BD%D1%83%D1%8E%20%D0%BA%D0%BE%D0%BC%D0%B1%D0%B8%D0%BD%D0%B0%D1%86%D0%B8%D1%8E%20%D1%81%D0%BE%D0%B1%D1%8B%D1%82%D0%B8%D0%B9%20%D0%B8%20%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%B9)

## Задание 5. Сети

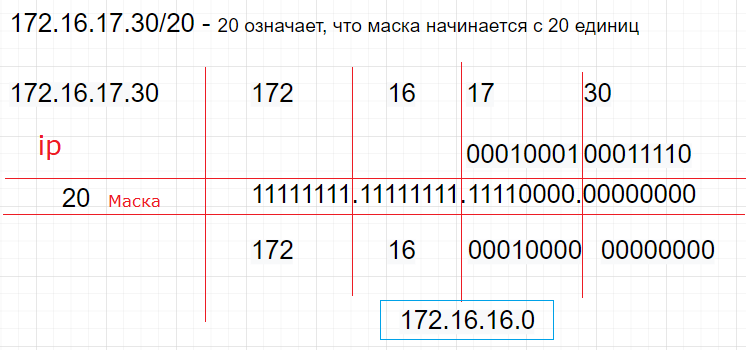
**Могут попросить разбить сеть на подсети, решение тут** <http://ciscotips.ru/subnetting-equal>

По сетевым адресам и маскам определить, находятся эти два устройства в одной подсети или в разных.

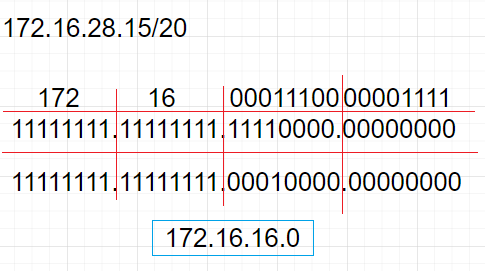
DeviceA: 172.16.17.30/20

DeviceB: 172.16.28.15/20

Определяем адрес сети у первого устройста



Аналогично с вторым



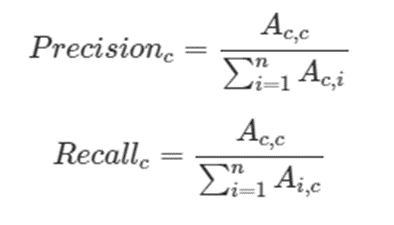
Адреса одинаковы => устройства в одной подсети

## Задание 6. Метрики в машинном обучении

Дана матрица результатов классификации в задачах диагностики болезни, где классы 1 и 2 представляют различные варианты виды диагностируемой болезни, а класс 0 — отсутствие болезни.

| Расчетный класс | 0 (Эмпирический) | 1 (Эмпирический) | 2 (Эмпирический) |
| --- | --- | --- | --- |
| 0 | 111 | 3 | 5 |
| 1 | 2 | 123 | 17 |
| 2 | 19 | 0 | 20 |

Необходимо рассчитать метрики полноты и точности, F-метрику по каждому классу в отдельности, средние и средневзвешенные. Обосновать выбор наиболее релевантной метрики учитывая специфику задачи, несбалансированность классов. Предложить основные подходы для усовершенствования классификации







<https://www.machinelearningmastery.ru/multi-class-metrics-made-simple-part-ii-the-f1-score-ebe8b2c2ca1/> - тут пример

Accuracy

Класс 0 = (111 + 123 + 20) / (111 + 123 +20+2+19+3+5) = 0.89

Класс 1 = (111+ 123 + 20) / (111 + 123 +20+3+0+2+ 17) = 0.92

Класс 2 = (111 + 123 + 20) / (111 + 123 + 20 +5 + 17 + 19 +0) = 0.86

Среднее = (0.89 + 0.92 + 0.86) / 3 =0.89

Средневзвешенное = (0.89 \* 132 + 0.92 \* 126 + 0.86 \* 42) / (132 + 126 + 42) = 0.898

Precision

Класс 0 = (111) / (111 +3+5) = 0.93

Класс 1 = (123) / (123 +2+ 17) = 0.87

Класс 2 = (20) / (20 + 19) = 0.51

Среднее = (0.93 + 0.87 + 0.51) / 3 = 0.77

Средневзвешенное = (0.93 \* 132 + 0.87 \* 126 + 0.51 \* 42) / (132 + 126 + 42) = 0.85

Recall

Класс 0 = (111) / (111 +2-+ 19) = 0.84

Класс 1 = (123) / (123 + 3) = 0.98

Класс 2 = (20) / (20 +5 + 17) = 0.48

Среднее = (0.84 + 0.98 + 0.48) / 3 =0.77

Средневзвешенное = (0.84 \* 132 +

0.98 \* 126 + 0.48 \* 42) / (132 + 126 + 42) = 0.85

F-мера

Класс 0 = 2 \* 0.93 \* 0.84 / (0.93 + 0.84)

= 0.88

Класс 1 =2 \* 0.87 \* 0.98 / (0.87 + 0.98)

= 0.92

Класс 2 =2 \* 0.51 \* 0.48 / (0.51 + 0.48)

= 0.49

Среднее = (0.88 + 0.92 + 0.49) / 3 = 0.76

Средневзвешенное = (0.88 \* 132 + 0.92 \* 126 + 0.49 \* 42) / (132 + 126 + 42) = 0.84