Давайте представим, что в 2020 году в Москве проводили опрос и выявили, к какому классу люди себя относят. По результатам опроса все люди разделились на сладкоежек, вегетарианцев и любителей мяса. Давайте напишем программу, которая поможет нам подвести итоги опроса. Для создания программы нужно:

1. Создать родительский класс **Initialization**, который состоит из:

- метода инициализации, в который поступают аргументы: capacity - целое число, food - список из строковых названий еды. Если в значение capacity передается не целое число, вывести надпись ‘Количество людей должно быть целым числом’ и не создавать для таких экземпляров атрибуты capacity и food.

2. Создать дочерний класс **Vegetarian** от класса **Initialization**, который состоит из:

- метода инициализации, принимающего аргументы capacity, food. Нужно создать одноименные атрибуты через вызов родительского метода __init__.
- метода __str__, который возвращает строку формата "<capacity> людей предпочитают не есть мясо! Они предпочитают <food>"

3. Создать дочерний класс **MeatEater** от класса **Initialization**, который состоит из:

- метода инициализации, принимающего аргументы capacity, food. Нужно создать одноименные атрибуты через вызов родительского метода __init__.
- метода __str__, который возвращает строку формата "<capacity> мясоедов в Москве! Помимо мяса они едят еще и <food>"

4. Создать дочерний класс **SweetTooth** от класса **Initialization**, который состоит из:

- метода инициализации, принимающего аргументы capacity, food. Нужно создать одноименные атрибуты через вызов родительского метода __init__.
- магического метода **__str__**, который возвращает строку формата ‘Сладкоежек в Москве <capacity>. Их самая любимая еда: <food>’;
- магического метода **__eq__**, который будет позволять сравнивать экземпляры класса SweetTooth с числами и другими нашими классами. Если сравнение происходит с целым числом и атрибут capacity с ним совпадает, то необходимо вернуть **True**, в противном случае - **False**. Если же сравнение идет с другим нашим классом(**Vegetarian** или **MeatEater**) и значения атрибутов capacity равны, то возвращается **True**, в противном случае - **False**. А если же сравнивается с другим типом данных, верните ‘Невозможно сравнить количество сладкоежек с <значение>’;
- магического метода **__lt__**. Если сравнение происходит с целым числом и количество сладкоежек (атрибут capacity) меньше, необходимо вернуть **True**, в противном случае - **False**. Если сравнение происходит с экземпляром одного из наших классов **Vegetarian** или **MeatEater** и сладкоежек меньше, то верните **True**, в противном случае верните **False**. В случае если сравнение идет с остальными типами данных, верните ‘Невозможно сравнить количество сладкоежек с <значение>’
- магического метода **__gt__**. Если сравнение происходит с целым числом и количество сладкоежек больше, необходимо вернуть значение **True**, в противном же случае - **False**. Если сравнение происходит с другим нашим классом **Vegetarian** или **MeatEater** и сладкоежек больше, то верните True, в противном случае - False. В случае если сравнение идет с остальными типами данных, верните ‘Невозможно сравнить количество сладкоежек с <значение>’

```
v_first = Vegetarian(10000, ['Орехи', 'овощи', 'фрукты'])
print(v_first)  # 10000 людей предпочитают не есть мясо! Они предпочитают ['Орехи', 'овощи', 'фрукты']
v_second = Vegetarian([23], ['nothing'])  # Количество людей должно быть целым числом
m_first = MeatEater(15000, ['Жареную картошку', 'рыба'])
print(m_first)  # 15000 мясоедов в Москве! Помимо мяса они едят еще и ['Жареную картошку', 'рыба']
s_first = SweetTooth(30000, ['Мороженое', 'Чипсы', 'ШОКОЛАД'])
print(s_first)  # Сладкоежек в Москве 30000. Их самая любимая еда: ['Мороженое', 'Чипсы', 'ШОКОЛАД']
print(s_first > v_first)  # True
print(30000 == s_first)  # True
print(s_first == 25000)  # False
print(100000 < s_first)  # False
print(100 < s_first)  # True
```