

3.3 Методы `__str__`, `__repr__`, `__len__`, `__abs__` 6 out of 11 steps passed 14 out of 29 points received

Подвиг 9. Объявите класс `Recipe` для представления рецептов. Отдельные ингредиенты рецепта должны определяться классом `Ingredient`. Объекты этих классов должны создаваться командами:

```
ing = Ingredient(name, volume, measure)
recipe = Recipe()
recipe = Recipe(ing_1, ing_2,..., ing_N)
```

где `ing_1`, `ing_2`,..., `ing_N` - объекты класса `Ingredient`.

В каждом объекте класса `Ingredient` должны создаваться локальные атрибуты:

`name` - название ингредиента (строка);

`volume` - объем ингредиента в рецепте (вещественное число);

`measure` - единица измерения объема ингредиента (строка), например, литр, чайная ложка, грамм, штук и т.д.;

С объектами класса `Ingredient` должна работать функция:

```
str(ing) # название: объем, ед. изм.
```

и возвращать строковое представление объекта в формате:

"название: объем, ед. изм."

Например:

```
ing = Ingredient("Соль", 1, "столовая ложка")
s = str(ing) # Соль: 1, столовая ложка
```

Класс `Recipe` должен иметь следующие методы:

`add_ingredient(ing)` - добавление нового ингредиента `ing` (объект класса `Ingredient`) в рецепт (в конец);

`remove_ingredient(ing)` - удаление ингредиента по объекту `ing` (объект класса `Ingredient`) из рецепта;

`get_ingredients()` - получение кортежа из объектов класса `Ingredient` текущего рецепта.

Также с объектами класса `Recipe` должна поддерживаться функция:

`len(recipe)` - возвращает число ингредиентов в рецепте.

Пример использования классов (эти строчки в программе писать не нужно):

```
recipe = Recipe()
recipe.add_ingredient(Ingredient("Соль", 1, "столовая ложка"))
recipe.add_ingredient(Ingredient("Мука", 1, "кг"))
recipe.add_ingredient(Ingredient("Мясо баранины", 10, "кг"))
ings = recipe.get_ingredients()
n = len(recipe) # n = 3
```

P.S. На экран ничего выводить не нужно, только объявить классы.

