

Домашнее задание 2

Камень, ножницы, бумага, ящерица, Спок

«Камень, ножницы, бумага» это популярная детская игра. В этом задании Вам нужно разработать программу, с которой можно играть в расширенный вариант этой игры, известной как «камень, ножницы, бумага, ящерица, Спок». Правила расширенного варианта следующие:

Ножницы режут бумагу. Бумага заворачивает камень. Камень давит [ящерицу](#), а ящерица травит [Спока](#), в то время как Спок ломает ножницы, которые, в свою очередь, отрезают голову ящерице, которая ест бумагу, на которой улики против Спока. Спок испаряет камень, а камень, разумеется, затупляет ножницы.

Графически, они представлены на рисунке 1 (стрелка показывает, кто кого побеждает).

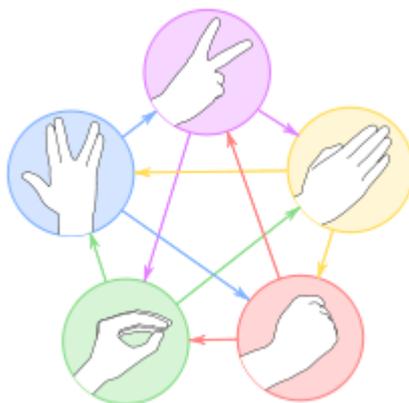


Рис. 1. По часовой стрелке с верхнего: ножницы, бумага, камень, ящерица, Спок

В задании необходимо реализовать функцию `rpsls(name)`, которая получает на вход строку `name`, содержащую выбор пользователя, случайным образом делает свой выбор и печатает результат игры.

Если попробовать закодировать правило выигрыша напрямую используя конструкцию `if/elif/else`, то потребуется 25 условных выражений. Однако есть более простой способ. Если закодировать каждый выбор следующим образом:

- камень – 0;
- Спок – 1;
- бумага – 2;
- ящерица – 3;
- ножницы – 4;

то каждый вариант выигрывает у двух предыдущих и проигрывает двум следующим за ним (если использовать арифметику по модулю).

Как выполнять задание

Для решения задания используйте прилагаемый файл с шаблоном `rpsls_template.py`.

1. Реализуйте вспомогательную функцию `name_to_number(name)`, которая преобразует строку `name` в число в диапазоне от 0 до 4, как описано выше. Эта функция должна использовать конструкцию `if/elif/else`. Используйте условия вида `name == 'бумага'`, чтобы различить случаи. Для упрощения отладки добавьте в конце условного оператора раздел `else`, для случая, когда `name` не соответствует ни одной из правильных входных строк, и распечатайте сообщение об ошибке. Функцию `name_to_number()` можно проверить, используя шаблон `name_to_number_test.py`.
2. Реализуйте вторую вспомогательную функцию, `number_to_name(number)`, которая преобразует число в диапазоне от 0 до 4 в строку с соответствующим названием. В этой функции тоже можно использовать конструкцию `if/elif/else` с печатью сообщения об ошибке в разделе `else`. Проверьте эту функцию с помощью шаблона `number_to_name_test.py`.
3. Реализуйте первую часть основной функции `rpsls(player_choice)`. Распечатайте пустую строку (чтобы разделить последовательные игры), затем – строку с сообщением о выборе игрока (переданном в параметре `player_choice`). Затем, вычислите значение `player_number`, соответствующее выбору игрока, вызвав функцию `name_to_number()` с аргументом `player_choice`. Проверьте, что первая часть функции работает, с помощью вызовов в конце шаблона.
4. Реализуйте вторую часть функции `rpsls()`, которая генерирует выбор компьютера и печатает соответствующее сообщение. Используйте функцию `random.randrange()` чтобы сгенерировать случайное значение `comp_number` между 0 и 4, которое соответствует выбору компьютера. Поэкспериментируйте с `randrange` в интерактивном режиме, чтобы лучше понять как она работает и понять, как генерировать числа в правильном диапазоне. После этого вычислите строку `comp_name`, соответствующую выбору компьютера, используя функцию `number_to_name()`. Напечатайте соответствующее сообщение. Проверьте, что добавленная часть функции работает.
5. Реализуйте последнюю часть функции `rpsls()`, которая определяет и печатает сообщение. Для этого, вычислите разницу между `comp_number` и `player_number` по модулю 5. Затем напишите оператор `if/elif/else`, который проверяют различные возможные значения разности и печатает соответствующее сообщение о победителе.

Проверьте, что функция работает. Так как мы ещё не познакомились со средствами, которые позволяют получить ввод от пользователя, просто вызывайте Вашу функцию, передавая ей текстовые строки со своим выбором.

Функция `random.randrange()`

Модуль `random` в Python содержит функции для генерации случайных чисел. Функция `randrange()` генерирует и возвращает случайное число из заданного диапазона:

- вызов `randrange(stop)` даст число из диапазона от 0 до `stop-1` включительно;
- `randrange(start,stop)` – число из диапазона от `start` до `stop-1` включительно;
- `randrange(start,stop,step)` – числа из диапазона `start` до `stop-1` включительно с шагом `step`;
- если `stop < start` и `step` отрицателен, то числа в диапазоне идут в сторону уменьшения.

Примеры диапазонов:

```
randrange(10) – случайное число из [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
randrange(1,11) – случайное число из [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

`randrange(0,10,3)` – случайное число из [0, 3, 6, 9]
`randrange(0,-10,-1)` – случайное число из [0, -1, -2, -3, -4, -5, -6, -7, -8, -9]

Пример использования `randrange()`:

```
import random
print(random.randrange(10), random.randrange(10), random.randrange(10))
```

Возможный вариант вывода (запустите несколько раз – значения будут разными):

```
>>> import random
>>>
print(random.randrange(10), random.randrange(1,11), random.randrange(0,10,3))
3 9 0
>>>
print(random.randrange(10), random.randrange(1,11), random.randrange(0,10,3))
0 2 6
>>>
print(random.randrange(10), random.randrange(1,11), random.randrange(0,10,3))
7 8 0
>>>
print(random.randrange(10), random.randrange(1,11), random.randrange(0,10,3))
2 7 9
```