

Реализация игры «Ханойские башни» на языке программирования Python

Кизьянов Антон Олегович

*Приамурский государственный университет имени Шолом-Алейхема
Студент*

Аннотация

В данной статье описан принцип головоломки «Ханойские башни» и создана для нее игра. Для создания используется язык программирования Python со стандартной библиотекой. Созданная игра может служить для наглядного примера решения головоломки «Ханойские башни» и тестирования различных способов ее решения.

Ключевые слова: Python, Ханойские башни

Implementation of the game "Towers of Hanoi" in the Python programming language

Kizyanov Anton Olegovich

*Sholom-Aleichem Priamursky State University
student*

Abstract

This article describes the principle of the Tower of Hanoi puzzle and created a game for it. For creation, the Python programming language with the standard library is used. The created game can serve as a visual example of solving the "Towers of Hanoi" puzzle and testing various ways to solve it.

Keywords: Python, Towers of Hanoi

Башня Ханоя, также называемая Башнями Ханоя или Башнями Брахмы, представляет собой головоломку, состоящую из трех вертикальных колышков и набора дисков разного размера с отверстиями в центре. Считается, что Ханойская башня была изобретена в 1883 году французским математиком Эдуардом Лукасом, хотя его роль в его изобретении оспаривается.

Диски имеют разные размеры и расположены в порядке возрастания, то есть меньшее находится над большим. Есть и другие варианты головоломки, в которых количество дисков увеличивается, но количество башен остается прежним.

Задача - переместить все диски в какую-нибудь другую башню, не нарушая порядок расстановки. Несколько правил, которым необходимо следовать для Ханойской башни:

- В любой момент времени между башнями можно перемещать только один диск.
- Снять можно только «верхний» диск.
- Никакой большой диск не может поместиться поверх маленького.

Цель исследования – написать игру «Ханойские башни» на языке программирования Python.

Ранее этим вопросом интересовались Д.С. Коняева, И.В. Дворянинова развивали тему «Пример использования рекурсивного алгоритма, задача "Ханойская башня"» [1] в которой рассмотрен пример использования рекурсивного алгоритма при решении задачи «Ханойская башня», приводится пример расчёта времени работы алгоритма для решения головоломки «Ханойская башня». Д.А. Петренков, А.В. Семёнов с темой «Вычислимость по тьюрингу частично рекурсивных функций и математическое решение задачи о ханойской башне» [2], а подробнее про алгебраический способ доказательства вычислимости по Тьюрингу частично рекурсивных функций, уделяется внимание конструированию машин Тьюринга и выполнению операций над ними, уточняется понятие вычислимых функций. В дополнение разбирается вариант применения рекурсивных алгоритмов и преобразования их в нерекурсивные на примере рекурсивного решения популярной головоломки «Ханойская башня». Л.Л. Голубева, А.Э. Малевич, Н.Л. Щеглова опубликовали статью «Компьютерное моделирование в mathematica на примере задачи о ханойских башнях» [3] описали идеи о объектно-ориентированного подхода на примере решения классической головоломки Ханойская башня. В качестве среды моделирования предложен символьный пакет Mathematica®.

Был написан код на языке программирования Python, в виде класса Disks. В классе имеются методы для перемещения диска «Move», для запуска игры «run» и отображении всех 3 башен с количеством на них дисков. Полный код представлен ниже.

```
import copy
import sys

class Disks():
    def __init__(self):
        self.Count_disks = 5
        self.Full_tower = list(range(self.Count_disks, 0, -1))

    def run(self):
        towers = {'A': copy.copy(self.Full_tower), 'B': [], 'C': []}

        while True:
            self.Show_tower(towers)
```

```
fromTower, toTower = self.Move(towers)

disk = towers[fromTower].pop()
towers[toTower].append(disk)
if self.Full_tower in (towers['B'], towers['C']):
    self.Show_tower(towers)
    print('Вы решили головоломку!')
    sys.exit()

def Move(self, towers):
    while True:
        print('Вводите буквы откуда и куда, или QUIT для выхода.')
        response = input('> ').upper().strip()

        if response == 'QUIT':
            print('Спасибо за игру!')
            sys.exit()

        if response not in ('AB', 'AC', 'BA', 'BC', 'CA', 'CB'):
            print('Введите одно из AB, AC, BA, BC, CA, или CB.')
            continue

        fromTower, toTower = response[0], response[1]

        if len(towers[fromTower]) == 0:
            print('На выбранной башне нет дисков')
            continue

        elif len(towers[toTower]) == 0:
            return fromTower, toTower
        elif towers[toTower][-1] < towers[fromTower][-1]:
            print('Вы не можете положить больший диск на меньший')
            continue
        else:
            return fromTower, toTower

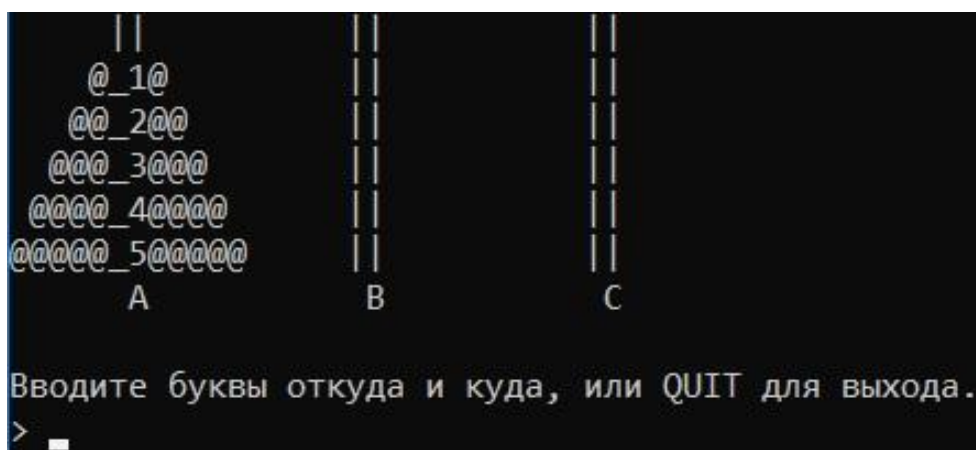
def Show_tower(self, towers):
    for level in range(self.Count_disks, -1, -1):
        for tower in (towers['A'], towers['B'], towers['C']):
            if level >= len(tower):
                self.Show_disk(0)
            else:
                self.Show_disk(tower[level])
    print()
    emptySpace = ' ' * (self.Count_disks)
```

```
print('{0} A{0}{0} B{0}{0} C\n'.format(emptySpace))
```

```
def Show_disk(self, width):  
    emptySpace = ' ' * (self.Count_disks - width)  
    if width == 0:  
        print(emptySpace + '||' + emptySpace, end="")  
    else:  
        disk = '@' * width  
        numLabel = str(width).rjust(2, '_')  
        print(emptySpace + disk + numLabel + disk + emptySpace, end="")
```

```
game = Disks()  
game.run()
```

При первом запуске, отображаются три башни с именами А, В и С. На башне А нанизаны диски в порядке убывающем порядке, количество дисков можно менять в переменной `self.Count_disks`, в данном случае их 5, первый экран можно увидеть на рисунке 1.



```
  ||  
  @_1@  
  @@_2@@  
  @@@_3@@@  
  @@@@_4@@@@  
  @@@@@_5@@@@@  
  A          B          C  
  
Вводите буквы откуда и куда, или QUIT для выхода.  
> _
```

Рис. 1 Первый запуск игры

Пользователя просят ввести 2 буквы, сочетание имен башен, с какой башни на какую перенести диск, если это возможно, например АВ, перенести диск с башни А на башню В как на рисунке 2.

Вывод

В этой статье было приведено описание головоломки «Ханойские башни» и создана игра на языке программирования Python. Игра позволяет визуализировать перемещение дисков между башнями, что делает ее нагляднее. Также есть возможность установки собственного количества дисков. Благодаря этой игре, можно проанализировать стратегии решения головоломки наглядным способом.

Библиографический список

1. Коняева Д.С., Дворянинова И.В. Пример использования рекурсивного алгоритма, задача "ханойская башня" // В сборнике: Студенческая наука для развития информационного общества. сборник материалов VI Всероссийской научно-технической конференции. 2017. С. 322-324. URL: <https://elibrary.ru/item.asp?id=29388821> (Дата обращения: 19.06.2021)
2. Петренков Д.А., Семёнов А.В. Вычислимость по Тьюрингу частично рекурсивных функций и математическое решение задачи о ханойской башне // В сборнике: Молодежный научный форум. сборник статей по материалам XLVII студенческой международной научно-практической конференции. 2019. С. 55-63. URL: <https://elibrary.ru/item.asp?id=37538158> (Дата обращения: 19.06.2021)
3. Голубева Л.Л., Малевич А.Э., Щеглова Н.Л. Компьютерное моделирование в Mathematica на примере задачи о ханойских башнях // В сборнике: Веб-программирование и Интернет-технологии WebConf2012. Материалы 2-ой Международной научно-практической конференции. 2012. С. 78. URL: <https://elibrary.ru/item.asp?id=42402696> (Дата обращения: 19.06.2021)