Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет имени Г.В. Плеханова»

**МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ**

специальность 09.02.07 «Информационные системы и программирование»

Квалификация: Программист

ПРАКТИЧЕСКАЯ РАБОТА №1

ПО ОП.04 «Основы алгоритмизации и программирования»

|  |  |
| --- | --- |
| Выполнил студент  группы П-3-23  Гончаров С.М. | Проверил преподаватель  \_\_\_\_\_\_\_\_\_Т.Д. Артамонова  «\_\_\_» \_\_\_\_\_\_\_\_\_ 2024 года |

Москва 2024

Цель работы: разработать текстовую игру, используя основы программирования на Python, с упором на работу с коллекциями данных (списки, кортежи, множества, словари).

1. Первым делом используя основную команду print, которая пригодится нам больше всего раз, создал небольшое описание игры, далее добавил int(input()) -–это будет кнопкой для продолжения на следующий этап, что бы все не сливалось в единый текст.

Создал словарь inventar = {}, который будет играть роль инвентаря, в него в процессе будем добавлять предметы, print(list(inventar.values())) – выводит в консоль все значения словаря. int(input()) – кнопочка для продолжения.

Прописываем print в котором хранится определенный текст и далее прописываем inventar[“”] = “” – тут добавляются значения в словарь, далее print и потом выводим в консоль инвентарь - print(list(inventar.values())).

Так же принт с определенным текстом и инт инпут для продолжения.

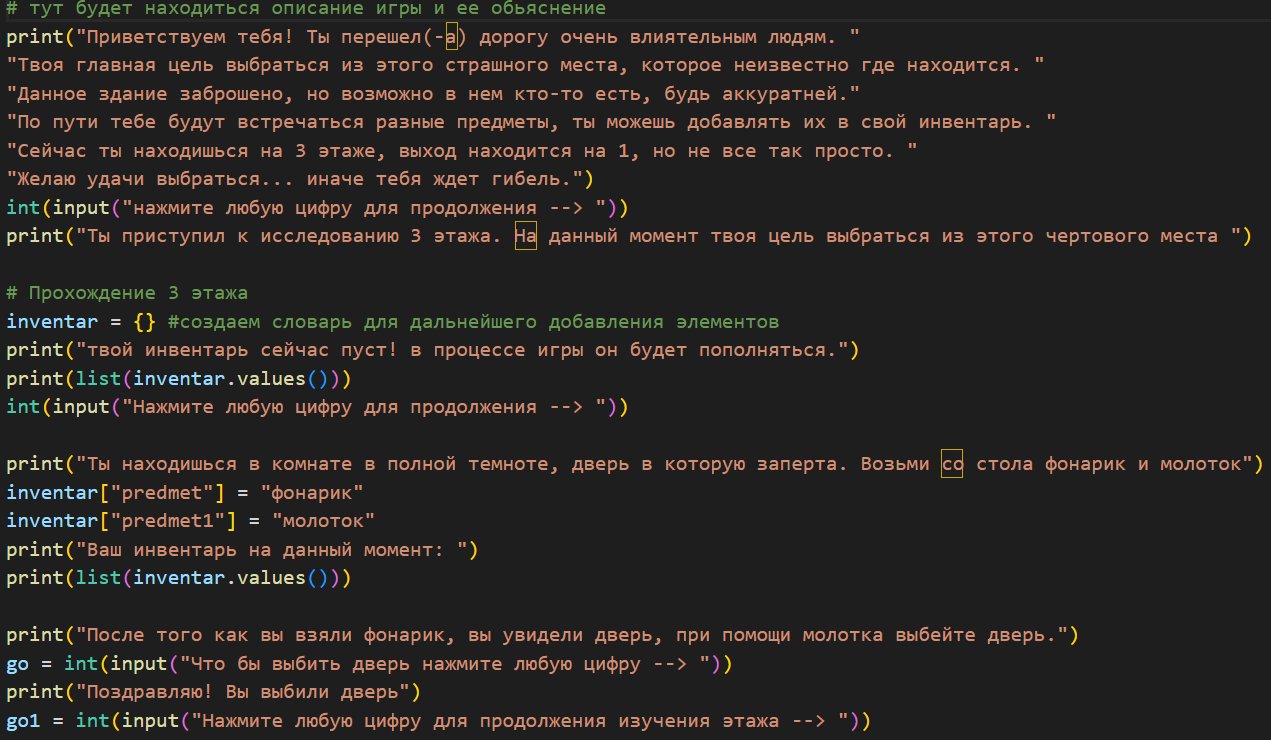


Рисунок 1 - начало создания текстовой игры

1. Создаем пользователю выбор при помощи условного оператора if/elif/else. Для этого создали отдельную переменную которой присвоили значения для выбора. В if, переменная = 1 и так же там хранятся: print – который хранит определенный текст int(input()) – который служит пропуском, так же обратились к словарю для добавления элементов inventar[“”] = “”, прописали print и создали команду для вывода словаря в терминал.

В elif переменная = 2, там хранятся буквально те же значения и те же действия, но там добавили команду для удаления элемента из словаря – del inventar[“”].

Else – прописал на случай если пользователь не умеет читать и решил нажать на другую цифру и что бы ему сразу говорило GAME OVER.

Ну и далее по мелочи print и int(input()).

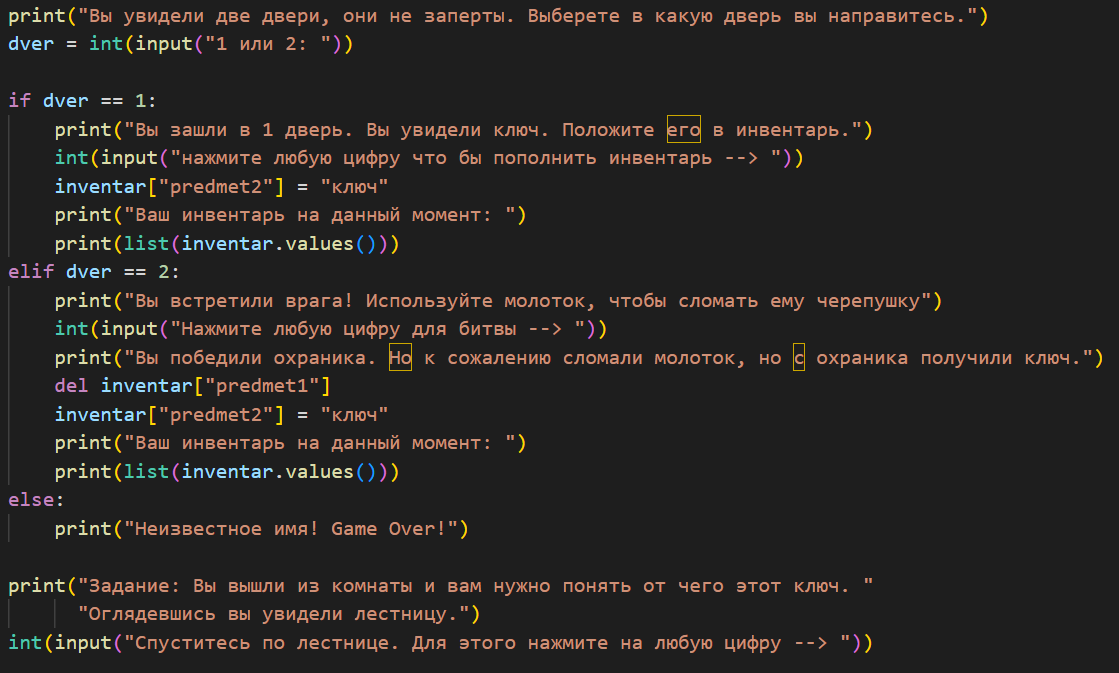


Рисунок 2 - работа с условным оператором и словарем

1. Перешел к следующему этапу, а именно действиям на 2 этаже. Первым тут идут так же, как и во всем коде print и int(input()), от которых скоро клавиши сотрутся.

И тут я добавил множества (mnojestvo = {}), в них будут хранится задания, которые будет выполнять игрок или нет, все зависит от его решений.

Далее работа с условными операторами и создание выбора для игрока и проверка на его удачу (угадывание кода).

В if добавлена команда для удаления инвентаря - del inventar[“”] ну и конечно команда для вывода инвентаря в терминал - print(list(inventar.values())). Ну и по классике print и int(input()).

В elif добавляем в множества элементы при помощи –mnojestvo.add() и внутри создаем еще один цикл в котором будет угадывание пароля и если пароль угадан, то будет удаляться одно задание из множества при помощи mnojestvo.remove() и далее выводим в консоль оставшиеся задания при помощи print(mnojestvo).

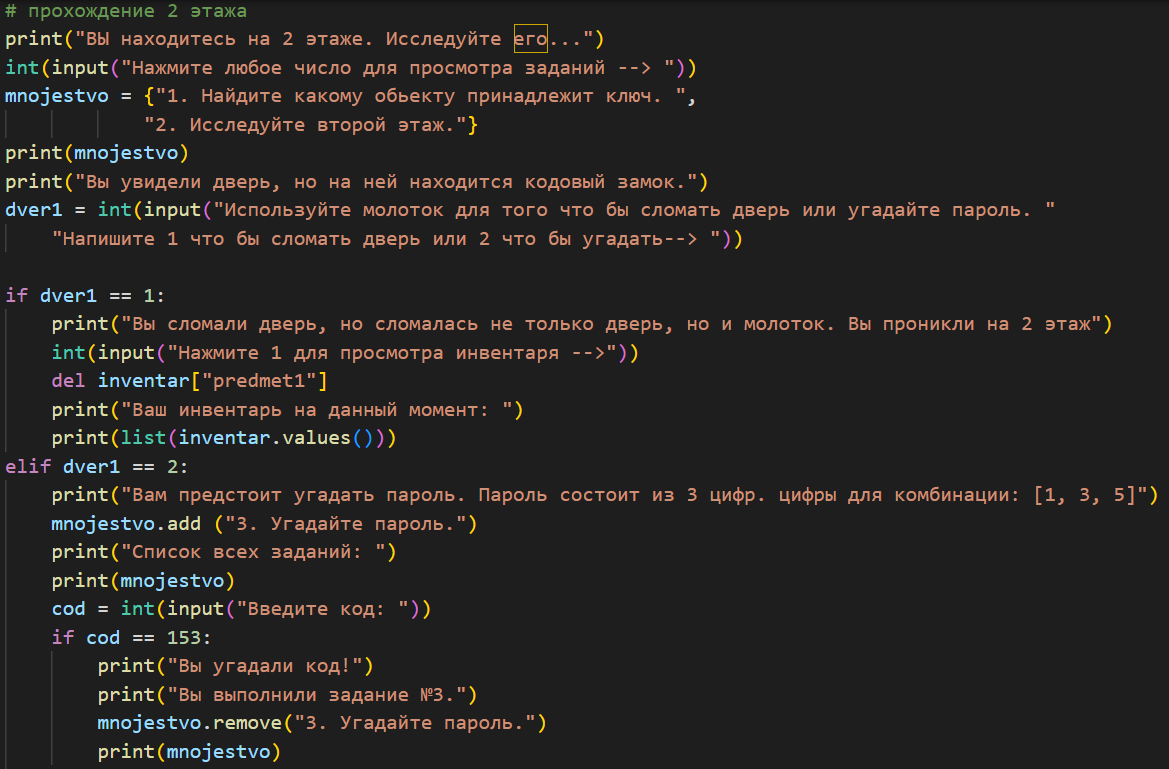


Рисунок 3 - работа с множеством и условным оператором

1. В цикле подобрали все возможные варианты, если не угадал пароль, т.к. пароль состоит 3 трех цифр, то вариантов не так много в обратном случае пишем else если пользователь где-то накосячил.

Ну и соответственно далее идет print в котором просто какой-то текст.

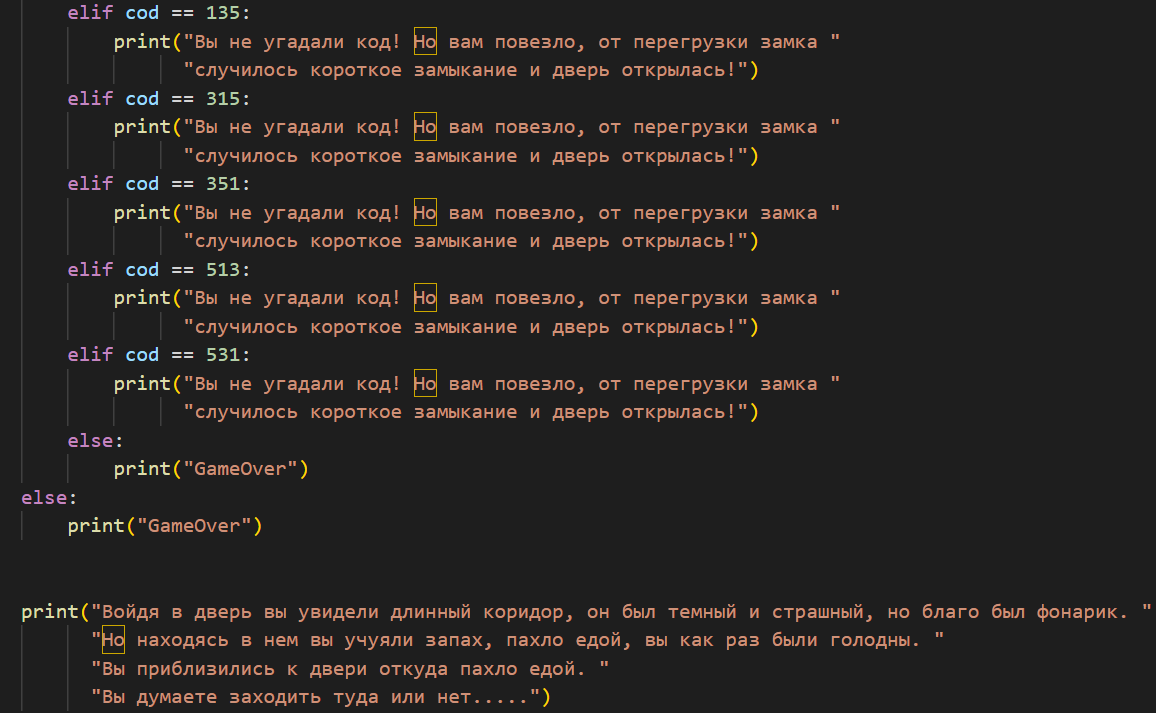


Рисунок 4 – цикл

1. Создаем еще один цикл при помощи условных операторов, который служит так же для выбора, в данном случае заходить в комнату или нет.

Если пользователь зайдет, то он выполнит 1 задание и следовательно оно удалиться при помощи команды mnojestvo.discard (), и выведеться в консоль оставшиеся задания при помощи print(mnojestvo), так же добавляем элементы в инвентарь при помощи inventar[“”] = “” и удаляем del inventar[“”]. Ну и соответственно много print и int(input()).

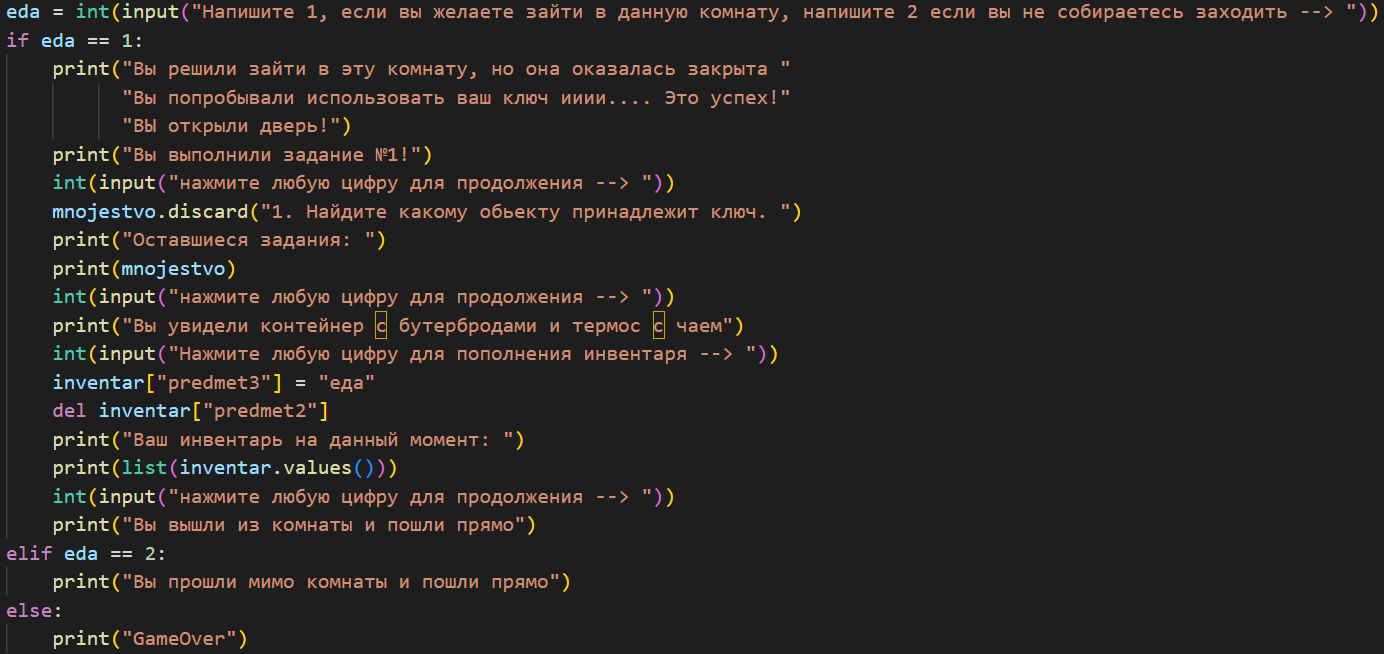


Рисунок 5 - цикл, работа с множеством и словарем

1. Первым делом прописываем print в котором находиться сюжетная линия игры, потом int(input()) – который отделяет сюжетные линии и образно создает кнопку продолжения. Далее удаляем элемент из множества при помощи mnojestvo.discard(“”), т.к. пользователь выполнил задание и выводим оставшееся множество в терминал при помощи print(mnojestvo), далее прописываются printы с сюжетной линией и int(input()).

Создаем выбор для пользователя, так же при помощи условных операторов. В 1 условии игрок проигрывает, но зато удаляются элементы из инвентаря т.к. он избавился от данного элемента, во втором условии пользователь выигрывает, но этот элемент остается у него в словаре.

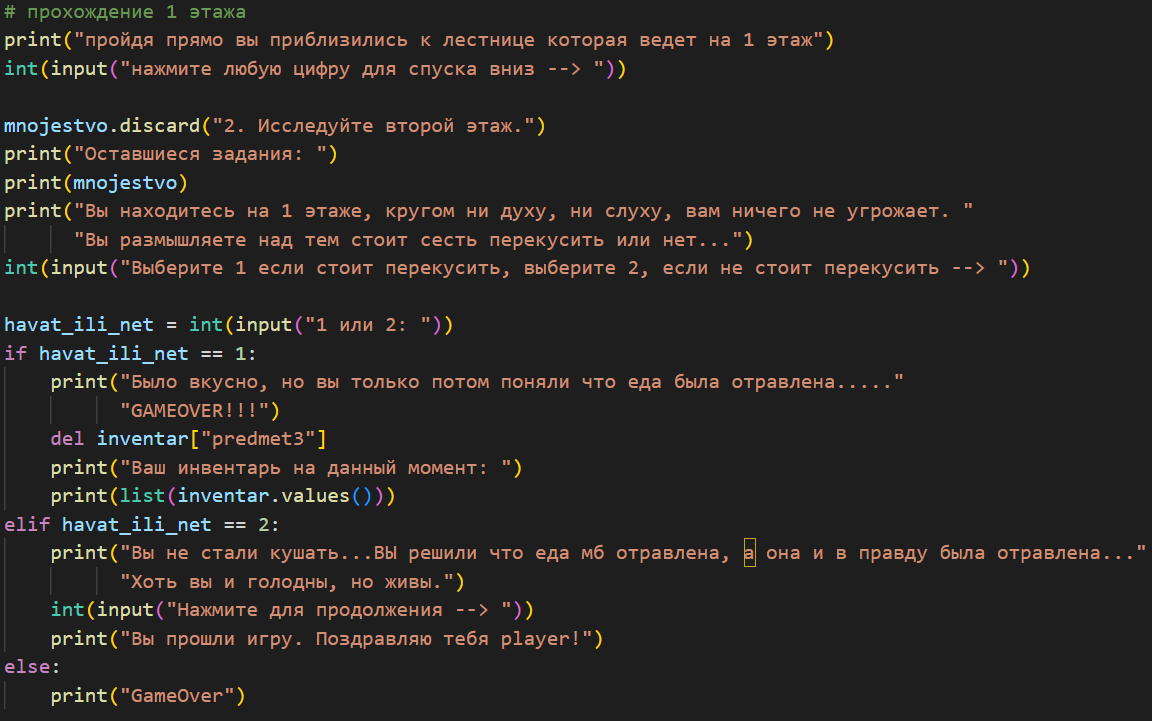


Рисунок 6 - работа с циклом, множеством и словарем

1. Тут я добавил в конце результаты того, сколько заданий выполнил пользователь и сколько предметов у него в инвентаре осталось.

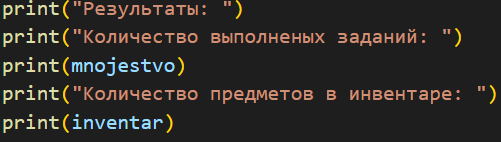


Рисунок 7 – результаты

1. Результаты работы программы.

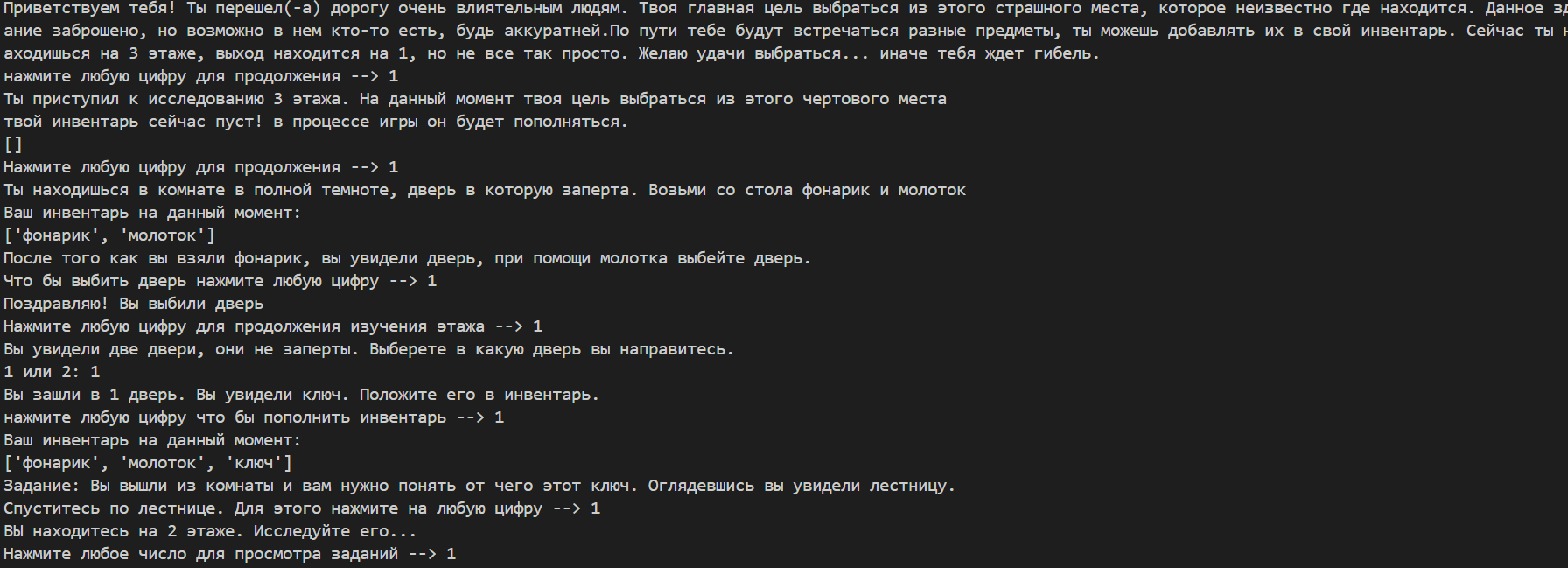


Рисунок 8 – результат работы программы

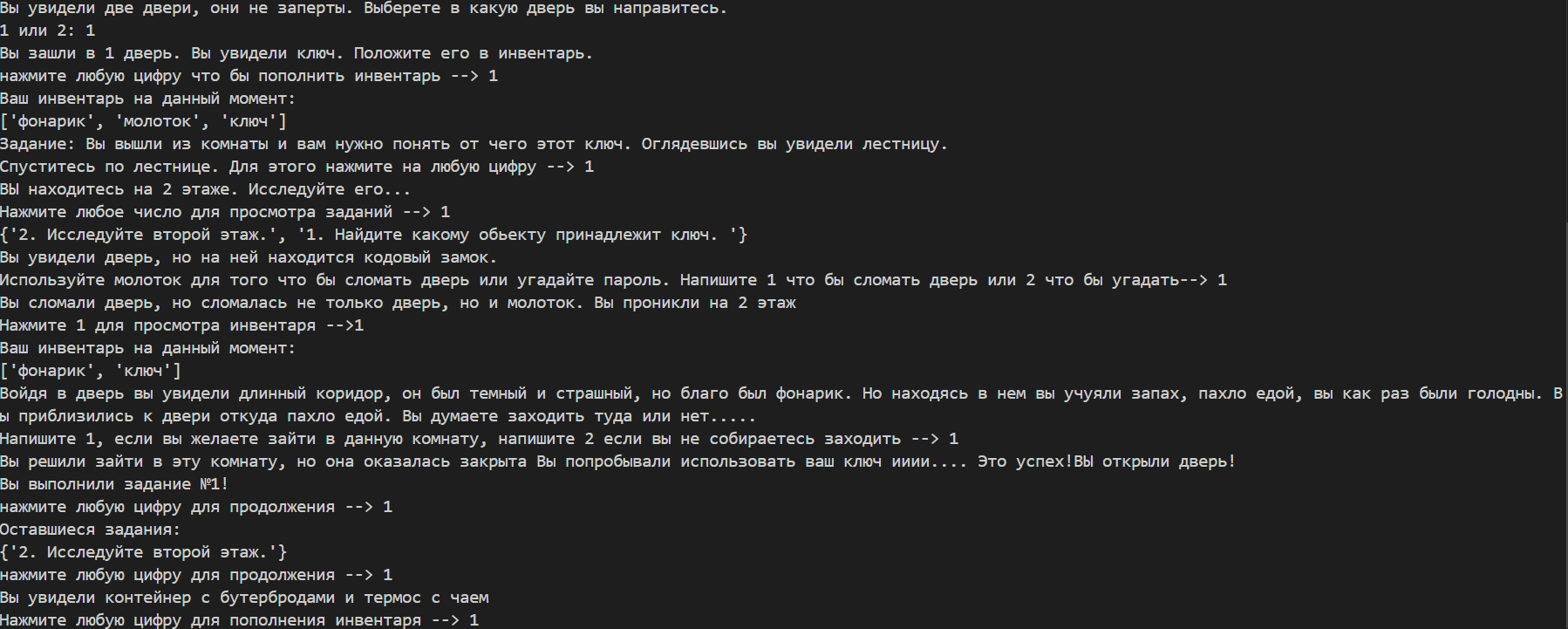


Рисунок 9 – результат работы программы

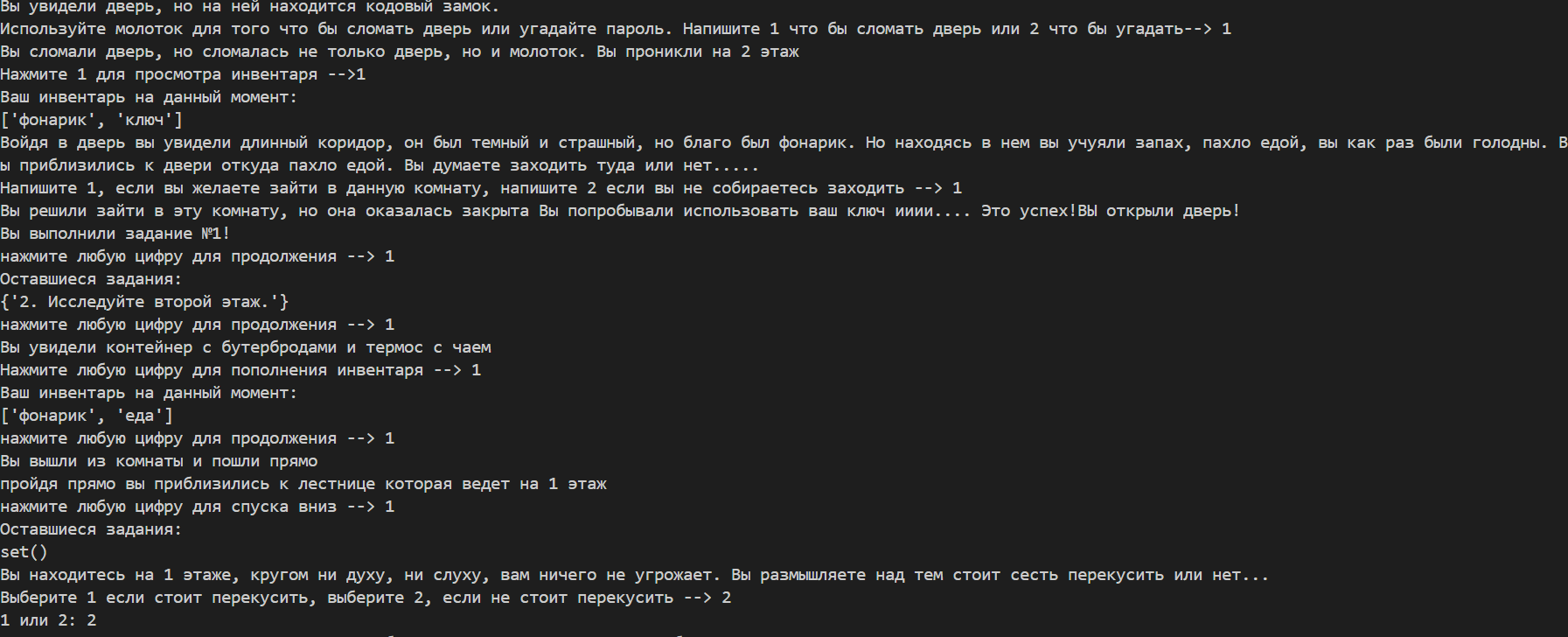


Рисунок 10 – результат работы программы

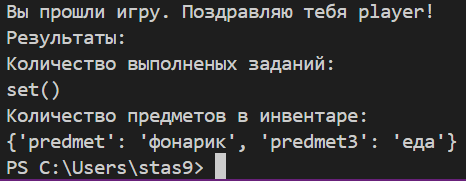


Рисунок 11 – финальный результат работы программы

Вывод: разработал текстовую игру, используя основы программирования на Python, с упором на работу с коллекциями данных (списки, кортежи, множества, словари).