**Отчёт о проекте**

1. **Задание.**

Задача стоит в разработке алгоритма «*Игра в слова*». Необходимо учесть все тонкости данной игры: игрок обязан использовать только те слова, которые относятся к заданной в начале теме (животные, города, фрукты/овощи); игрок обязан использовать только те слова, которые начинаются с той буквы, с которой заканчивается слово программы. Сама программа должна действовать аналогично игроку.

1. **Программный код.**

import random  
  
  
# функционал игры.  
def game(words, randWord):  
 flag = 0  
 word\_game = randWord  
 # Флаг на победу.  
 while flag == 0:  
 print(word\_game)  
 wordsEnd.append(word\_game)  
 print("Your word!")  
 flag\_word = 0  
 # Флаг на верное слово.  
 while flag\_word == 0:  
 flag\_find = 0  
 word = input()  
 # Проход по всем словам из заданной темы.  
 for i in words:  
 # Занижение регистра для сравнения слов и букв.  
 i.lower()  
 word = word.lower()  
 # Проверка, есть ли слово в списке и сходится ли первая буква с последней.  
 if word\_game[-1] == word[0] and i == word:  
 # Проверка, есть ли слово в списке пройденных слов.  
 for j in wordsEnd:  
 # Найденное слово выкидывает в начало  
 if j == i:  
 break  
 # Добавляет в список подходящее слово и игра продолжается.  
 else:  
 flag\_word = 1  
 wordsEnd.append(word)  
 # Проверка всех слов с пройденными и поиск подходящего.  
 for k in words:  
 # Флаг на найденное ботом слово.  
 flag\_notfind = 1  
 for n in wordsEnd:  
 if k == n:  
 flag\_notfind = 0  
 if k[0] == word[-1] and flag\_notfind == 1:  
 word\_game = k  
 flag\_find = 1  
 break  
 if flag\_find == 0:  
 print("Ohh... I'm lose... By!")  
 flag = 1  
 flag\_word = 1  
 break  
 break  
 break  
 if flag\_word == 0:  
 print("No! This word isn't true.")  
  
  
# Инициализация слов в темах.  
wordsAnimal = ["тигр", "кошка", "кот", "собака", "страус", "аист", "зубр", "дикобраз", "зебра", "жираф", "рыба"]  
wordsCountry = ["москва", "екатеренбург", "воронеж", "краснодар", "калининград", "ростов", "вороново", "орел"]  
wordsFruits = ["ананас", "салат", "мандарин", "апельсин", "помидор", "огурец", "цукини", "баклажан", "вишня", "яблоко"]  
# Список пройденных слов.  
wordsEnd = []  
# Приветствие.  
print("Hello, let's go play!")  
print("Choose theme:")  
print("1. Animal.")  
print("2. Country.")  
print("3. Plant.")  
# Выбор темы.  
choose = int(input())  
print("I'm starting!")  
if choose == 1:  
 game(wordsAnimal, random.choice(wordsAnimal))  
elif choose == 2:  
 game(wordsCountry, random.choice(wordsCountry))  
elif choose == 3:  
 game(wordsFruits, random.choice(wordsFruits))  
else:  
 game(wordsCountry, random.choice(wordsCountry))

1. **Пояснительная записка.**
   1. **Название.**

**«***Игра в слова***»**

* 1. **Автор проекта.**

Костригин А. С.

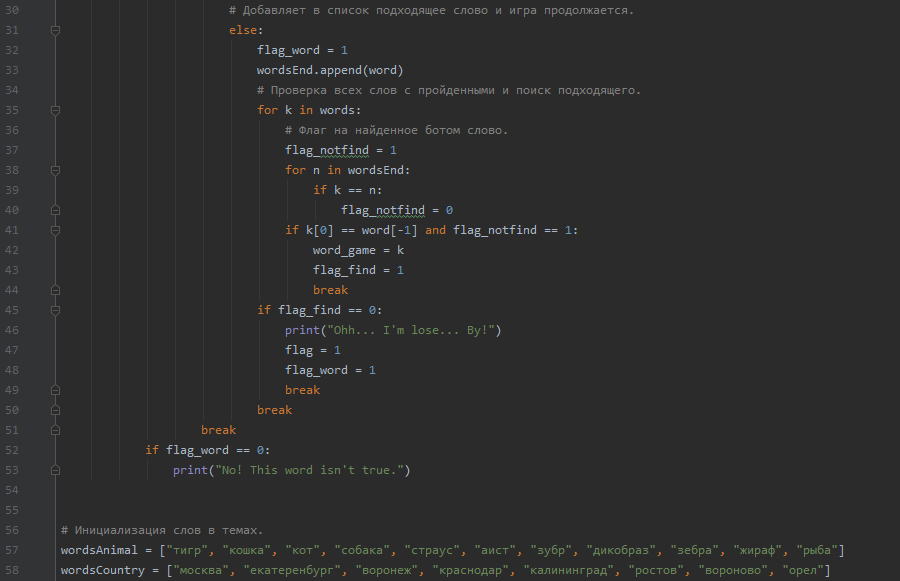
* 1. **Описание идеи.**

Идея заключается в тонкостях реализации игры. Необходимо учесть базовые правила игры: необходимую библиотеку слов; слово игрока или программы должно начинаться с той буквы, с которой заканчивается предыдущее; слово должно использоваться строго по той библиотеке и той теме, которая используется; в случае если у программы заканчивается библиотека (словарный запас), тогда программа проигрывает, а игрок, в то же время может использовать только те слова, которые используются библиотекой программы.

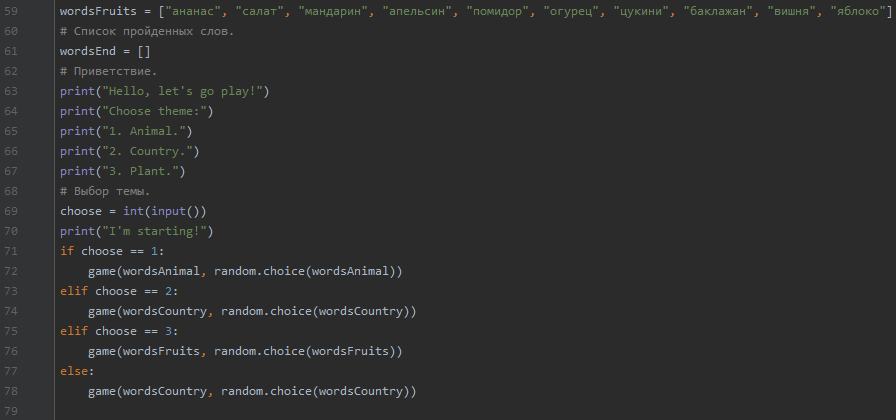
* 1. **Описание реализации.**



*Рис. 1. Часть кода программы 1.*



*Рис. 2. Часть кода программы 2.*



*Рис. 3. Часть кода программы 3.*

Для начала необходимо задать библиотеку слов. А также, так как в программе нужно сделать не менее 3 тем игры, важно описать не менее 3-х библиотек. Реализация идеи описана в *Рис. 2* и *Рис. 3* в строках *57-59*.

В игре есть правило: слова не должны повторяться. Чтобы реализовать это правило необходимо создать список, который будет хранить те слова, которые уже были использованы как программой, так и игроком. Создание списка можно отследить в *Рис. 3* строке *61.*

Далее необходимо выбрать тему. Программа должна изначально приветствовать пользователя и дать право выбора, для этого будет выведено несколько строк с приветствием и темами. Чтобы выбрать тему используются цифры, затем в функцию, где будет сама игра передается библиотека слов и на удачу выбранное слово программой. Реализация описана в *Рис. 3* в строках *63-78.*

Осуществляется переход к самому функционалу игры. Во-первых, игра должна продолжаться до тех пор, пока вся библиотека заданной темы не будет использована. Для этого инициализируется флаг, равное 0, чтобы затем в цикле поменять флаг, в случае окончания списка слов. Это можно увидеть на *Рис. 1* в строке *6*

Далее необходимо напечатать слово, выбранное программой и добавить его в список использованных слов (*Рис. 1* строки *10,11*). Затем нужно учесть случай, когда игрок будет вводить не те слова или вовсе не слова, а набор букв. Для этого ставится флаг, используемый в цикле, который будет подтверждать правильность слова (*Рис. 1* строки *13,15*). В дальнейшем программа столкнется с такой же проблемой, прежде чем она напечатает слово, оно должно быть из библиотеки данной темы и не быть использовано. Для этого ставится ещё один флаг, который будет это условие соблюдать. В противном случае, программа сознается в проигрыше (*Рис. 1* строка *16*). Затем пользователь вводит слово, которое будет проходить по всему списку слов. Для удобства и в связи с большой вариантностью введения слова пользователем, опускается регистр. Если такое слово найдено, а также первая буква введенного слова сходится с последней буквой слова, напечатанным программой, программа идет дальше (*Рис. 1* строки *17-24*). Далее слово проверяется с использованными, если оно находится в списке, пользователю даётся возможность исправиться (*Рис. 1* строки *26-29*). Если игрок смог соблюсти все перечисленные условия, то флаг на проверку слова поднимается и пройденное слова записывается в список использованных (*Рис. 2* строки *31-33*). Затем программа выбирает себе слово из библиотеки, которое ещё не использовалось. Добавляется ещё один флаг, который проверяет слова на использование. Если все подходящее слово не находится в использованном, то флаг поднятый (*Рис. 2* строки *35-40*). Если слово нашлось, то флаг найденного слова поднимается, и программа переходит в начало с дальнейшим продолжением игры (*Рис. 2* строки *41-44*). В противном случае программа сознается в поражении (*Рис. 2* строки *45-49*).

* 1. **Описание технологий.**

В программе используется библиотека «*random*». Она используется для выбора слова в начале игры программой.