|  |  |  |
| --- | --- | --- |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное автономное  образовательное учреждение высшего образования  «Пермский государственный национальный  исследовательский университет» | | |
|  | Институт компьютерных наук и технологий | |
| **ОТЧЁТ**  по индивидуальной работе №1  по дисциплине «Языки программирования»  Вариант 7 | | |
|  | | Работу выполнил  студент группы ПМИ-9-2023 1 курса  Васильева Елизавета Владиславовна  «18» июня 2023 г. |
| Работу проверил  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Рубцова М. Б.  «\_\_\_» \_\_\_\_\_\_\_\_\_ 2023 г. |
| Пермь 2024 | | |

СОДЕРЖАНИЕ

[Постановка задачи](#_Toc153130027) 3

[Алгоритм решения 3](#_Toc153130028)

[Тестирование](#_Toc153130029) 4-5

[Код программы](#_Toc153130030) 6-8

# Постановка задачи

# **Игра в «пьяницу». В игре в «пьяницу» карточная колода раздается поровну двум игрокам. Далее они вскрывают по одной верхней карте, и тот, чья карта старше, забирает себе обе вскрытые карты, которые кладутся под низ его колоды. Тот, кто остается без карт – проигрывает. Для простоты будем считать, что все карты различны по номиналу, а также, что самая младшая карта побеждает самую старшую карту ("шестерка берет туза"). Игрок, который забирает себе карты, сначала кладет под низ своей колоды карту первого игрока, затем карту второго игрока (то есть карта второго игрока оказывается внизу колоды).**

# **Напишите программу, которая моделирует игру в пьяницу и определяет, кто выигрывает. В игре участвует 10 карт, имеющих значения от 0 до 9, большая карта побеждает меньшую, карта со значением 0 побеждает карту 9. Входные данные Программа получает на вход две строки: первая строка содержит 5 чисел, разделенных пробелами—номера карт первого игрока, вторая – аналогично 5 карт второго игрока. Карты перечислены сверху вниз, то есть каждая строка начинается с той карты, которая будет открыта первой. Выходные данные. Программа должна определить, кто выигрывает при данной раздаче, и вывести слово first или second, после чего вывести количество ходов, сделанных до выигрыша. Если на протяжении 106 ходов игра не заканчивается, программа должна вывести слово botva. Примеры входные данные: 1 3 5 7 9 2 4 6 8 0 выходные данные: second 5**

# Алгоритм решения

1. Внимательно изучить задание проекта. Далее выявить структуры данных, используя которые получится сделать работу кода более эффективной.
2. Продумать и прописать функции основного механизма игры. Прописать поля ввода данных.
3. Прописать примеры корректных входных данных. Прописать вывод информации пользователю, сопроводив комментариями.

# Тестирование

1. Корректные входные данные (повторяющиеся числа)

INPUT P1: 2 4 6 8 5

INPUT P2: 1 3 6 9 3

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

OUTPUT: botva

1. Корректные входные данные (не повторяющиеся числа)

INPUT P1: 0 1 2 3 4

INPUT P2: 5 6 7 8 9

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

OUTPUT:

second 5

1. Некорректные входные данные

INPUT P1: f 2 3 4 5

INPUT P2: “ ”

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

OUTPUT:

Некорректный ввод: invalid literal for int() with base 10: 'f'

1. Пустой ввод

INPUT P1:’ ’

INPUT P2: ‘ ’

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

OUTPUT: Некорректный ввод: Каждый игрок должен иметь ровно 5 карт.

1. Частично пустой ввод

INPUT P1: 0 1 2 3 4

INPUT P2:” ”

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

OUTPUT:

Некорректный ввод: Каждый игрок должен иметь ровно 5 карт.

Код программы

#Основной класс, управляющий игрой(Логика игры):

classCardGame:

MAX\_TURNS =10\*\*6

def\_\_init\_\_(self,player1\_cards,player2\_cards):

self.player1 =Player("first",player1\_cards)

self.player2 =Player("second",player2\_cards)

self.turns =0

defplay(self):

whileself.player1.has\_cards()andself.player2.has\_cards()andself.turns <self.MAX\_TURNS:

self.turns +=1

card1 =self.player1.draw\_card()

card2 =self.player2.draw\_card()

ifself.is\_card1\_winner(card1,card2):

self.player1.add\_cards([card1,card2])

else:

self.player2.add\_cards([card1,card2])

ifself.turns >=self.MAX\_TURNS:

print("botva")

elifself.player1.has\_cards():

print(f"{self.player1.name}{self.turns}")

else:

print(f"{self.player2.name}{self.turns}")

#Специальное правило, где карта 0 побеждает карту 9

defis\_card1\_winner(self,card1,card2):

# Special rule where 0 beats 9

if(card1 ==0andcard2 ==9):

returnTrue

elif(card2 ==0andcard1 ==9):

returnFalse

returncard1 >card2

#Класс, описывающий игрока и его колоду карт(в т.ч. действия с ней)

classPlayer:

def\_\_init\_\_(self,name,cards):

self.name =name

self.cards =cards

defhas\_cards(self):

returnlen(self.cards)>0

defdraw\_card(self):

returnself.cards.pop(0)

defadd\_cards(self,cards):

self.cards.extend(cards)

#Функция проверяет корректность ввода

defread\_input():

try:

player1\_cards =list(map(int,input("Введите карты первого игрока: ").split()))

player2\_cards =list(map(int,input("Введите карты второго игрока: ").split()))

iflen(player1\_cards)!=5orlen(player2\_cards)!=5:

raiseValueError("Каждый игрок должен иметь ровно 5 карт.")

returnplayer1\_cards,player2\_cards

exceptValueErrorase:

print(f"Некорректный ввод: {e}")

returnread\_input()

#Функция запускает игру, читая входные данные и создавая объект CardGame

defmain():

player1\_cards,player2\_cards =read\_input()

game =CardGame(player1\_cards,player2\_cards)

game.play()

if\_\_name\_\_ =="\_\_main\_\_":

main()