

Практическое занятие № 3

Тема: составление программ ветвящейся структуры в IDE PyCharm Community.

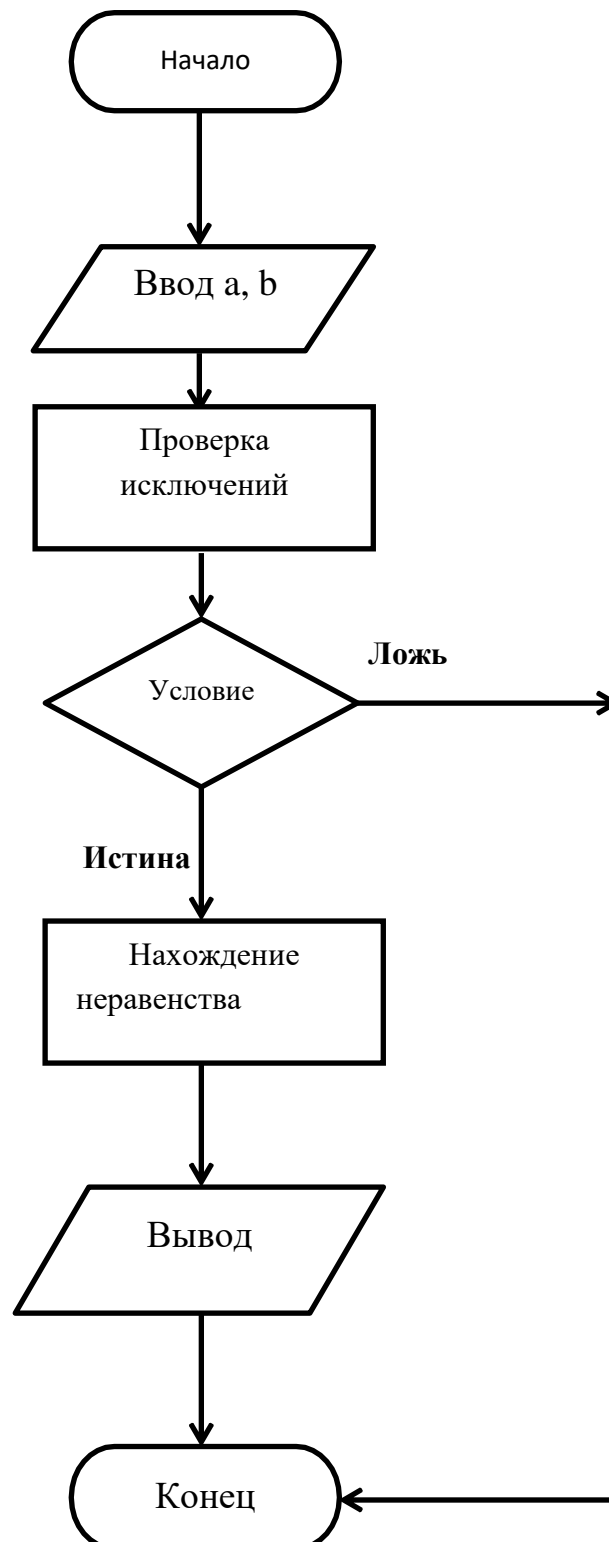
Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ ветвящейся структуры в IDE PyCharm Community.

Постановка задачи.

1) Даны два целых числа: А, В. Проверить истинность высказывания: «Справедливы неравенства $A > 0$ или $B < -2$ ».

Тип алгоритма: Ветвящийся

Блок схема алгоритма:



Текст программы:

#Вариант-5

#Даны два целых числа: А, В. Проверить истинность высказывания: «Справедливы
#неравенства $A > 0$ или $B < -2$ ».

```
a = input('Введите целое число А')
```

```
#Исключение для А
```

```
while type(a) != int:
```

```
    try:
```

```
        a = int(a)
```

```
    except ValueError:
```

```
        print('Введено не целое число!')
```

```
        a = input('Введите целое число А')
```

```
b = input('Введите целое число В')
```

```
#Исключение для В
```

```
while type(b) != int:
```

```
    try:
```

```
        b = int(b)
```

```
    except ValueError:
```

```
        print('Введено не целое число!')
```

```
        b = input('Введите целое число В')
```

```
#Выполнение задачи
```

```
if a > 0 or b < -2:
```

```
    print('Неравенство справедливо')
```

```
else:
```

```
    print('Неравенство несправедливо')
```

Протокол работы:

Введите целое число А 5

Введите целое число В -10

Неравенство справедливо

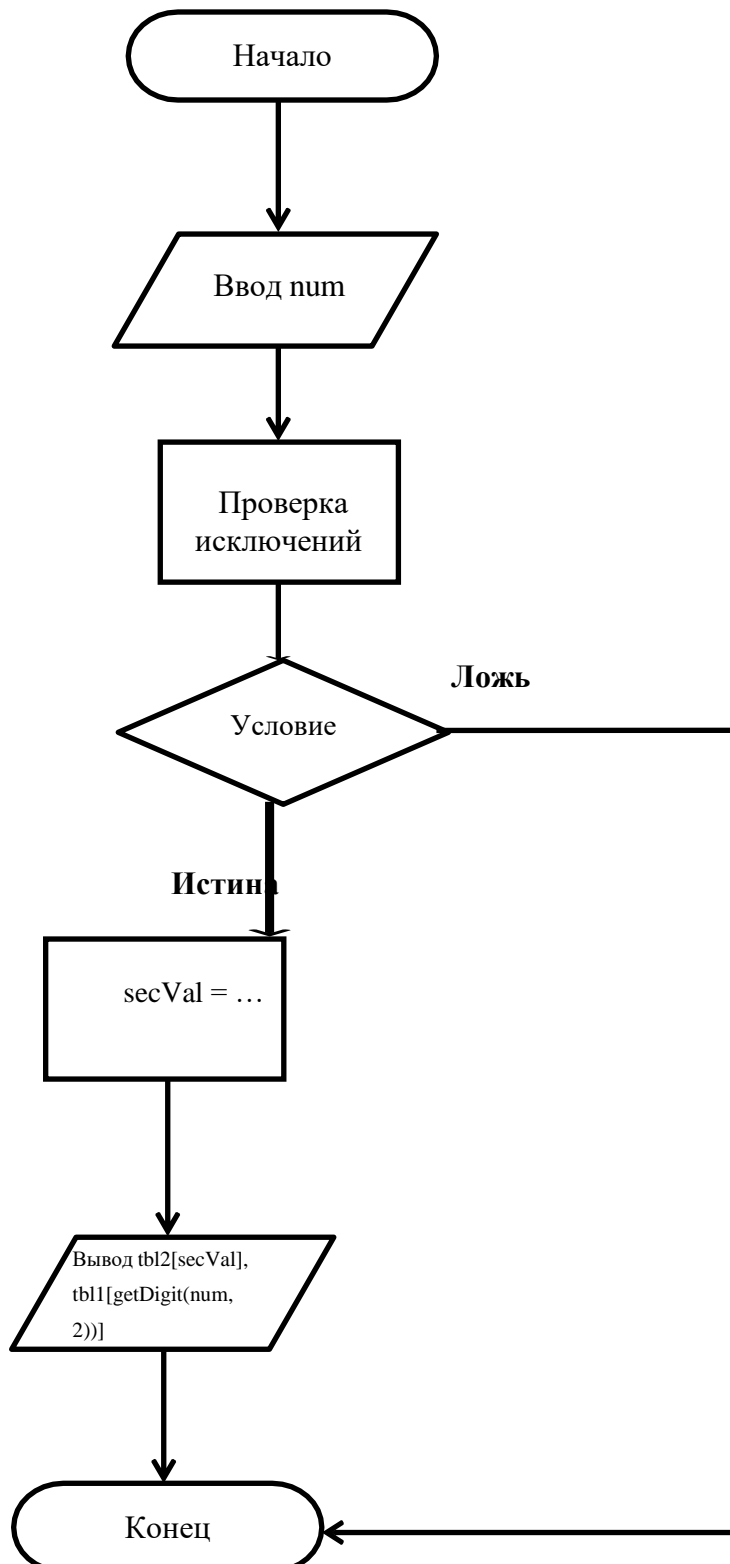
Process finished with exit code 0

Постановка задачи.

- 1) Мастям игральных карт присвоены порядковые номера: 1 – пики, 2 – трефы, 3 – бубны, 4 – червы. Достоинству карт, старших десятки, присвоены номера: 11 – валет, 12 – дама, 13 – король, 14 – туз. Дано трехзначное число, в котором первая цифра указывает на масть, а вторые две на достоинство карты. Вывести соответствующее название карты вида «дама червей», «туз треф» и т.п.

Тип алгоритма: Ветвящийся

Блок схема алгоритма:



Текст программы:

#Вариант-5

#Мастям игральных карт присвоены порядковые номера: 1 – пики, 2 – трефы, 3 – бубны, 4 – червы. Достоинству карт, старших десятки, присвоены номера: 11 – валет, 12 – дама, 13 – король, 14 – туз. Дано трехзначное число, в котором первая цифра указывает на масть, а вторые две на достоинство карты. Вывести соответствующее название карты вида «дама червей», «туз треф» и т.п.

```
tbl1 = {
    1: 'пики',
    2: 'трефы',
    3: 'бубны',
    4: 'червы'
}
```

```
tbl2 = {
    11: 'валет',
    12: 'дамы',
    13: 'король',
    14: 'туз'
}
```

```
minVal, maxVal = 111, 444
```

```
# Функция исполняет определение числа
# из трехзначного числа.
# Идет градация 0 = 1-ое число с конца
# Идет градация 1 = 2-ое число и т.д.
```

```
def getDigit(number, n):
    return number // 10 ** n % 10
```

```
num = input('Введите трехзначное число: ')
while type(num) != int:
    try:
        num = int(num)
    except ValueError:
        print('Введено не целое число!')
        num = input('Введите трехзначное число: ')
```

```
if minVal < num < maxVal:
```

```
    #Переводим последние две цифры в строку, для получения двузначного числа и снова переводим
    в число
```

```
    #для вывода по ключу таблицы значения
    secVal = int(str(getDigit(num, 1)) + str(getDigit(num, 0)))
```

```
    #Проверка на наличие ключа в таблице
```

```
    if secVal not in tbl2:
        print("Ключа {} не существует в таблице".format(secVal))
    else:
        print(tbl2[secVal], tbl1[getDigit(num, 2)])
```

```
elif num > maxVal:
```

```
    print('Введено слишком большое число - {}. Число должно быть больше {}'.format(num, maxVal))
```

```
elif minVal > num:
```

```
    print('Введено слишком маленькое число - {}. Число должно быть больше {}'.format(num, minVal))
```

Протокол работы:

Введите трехзначное

число: 113

король пики

Process finished with exit

code 0

Вывод:

В процессе выполнения практического занятия я закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрел навыки составления программ ветвящейся структуры в IDE PyCharm Community.

Были использованы языковые конструкции try, if, elif, while.