Практическое занятие №3

Задание №1

Тема: Условный оператор

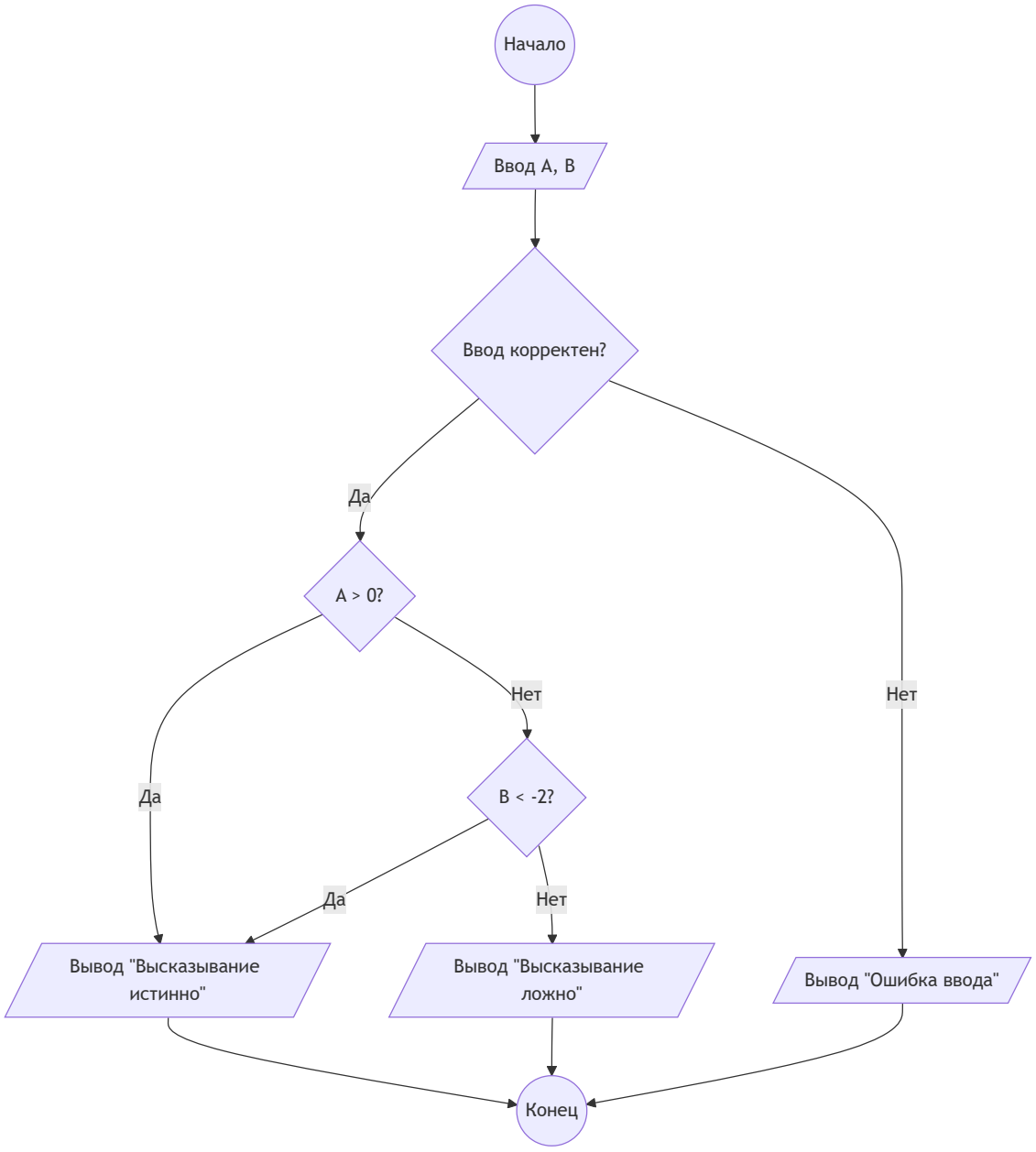
Цель: Научиться использовать условный оператор для проверки истинности высказываний.

Постановка задачи:

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

Тип алгоритма: алгоритм с ветвлением.

Блок-схема алгоритма:



Текст программы:

```
try:
    a = int(input("Введите число A: "))
    b = int(input("Введите число B: "))
    if a > 0 or b < -2:
        print("Высказывание истинно")
    else:
        print("Высказывание ложно")
except ValueError:
    print("Ошибка: Введите целые числа.")
```

Протокол работы программы (примеры):

```
Введите число A: 5
Введите число B: -1
Высказывание истинно

Введите число A: -3
Введите число B: -1
Высказывание ложно

Введите число A: -1
Введите число B: -3
Высказывание истинно

Введите число A: 1
Введите число B: 1
Высказывание истинно
```

Вывод:

В ходе выполнения практического задания были закреплены навыки использования условного оператора `if-else` и логического оператора `or` для проверки истинности сложных высказываний. Также была добавлена обработка исключений для повышения надежности программы.

Задание №2

Тема: Условный оператор, работа со словарями

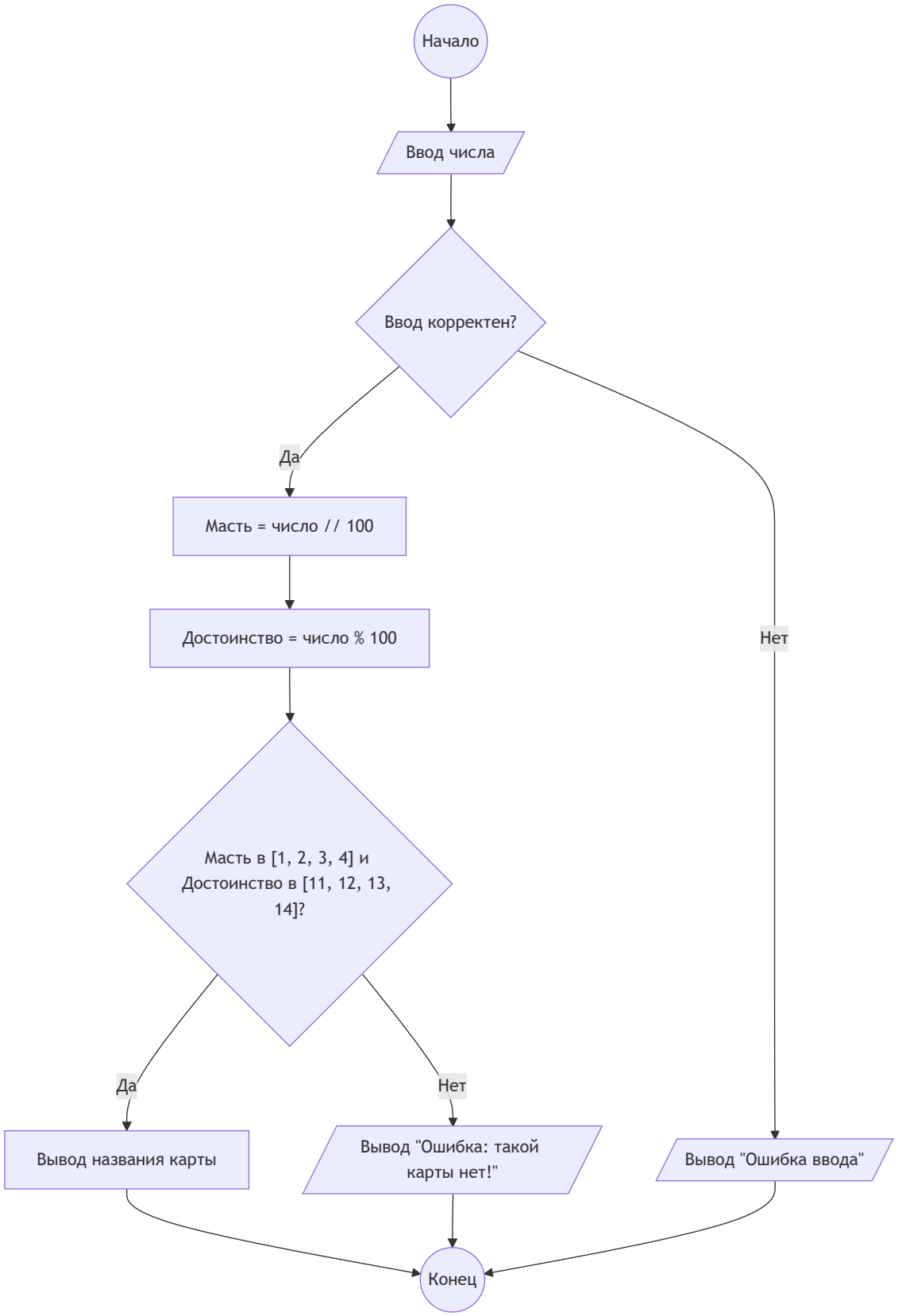
Цель: Научиться использовать условный оператор и словари для обработки данных.

Постановка задачи:

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: алгоритм с ветвлением.

Блок-схема алгоритма:



Текст программы:

```
suits = {
    1: "пики",
    2: "трефы",
    3: "бубны",
    4: "червы"
}

dignities = {
    11: "Валет",
    12: "Дама",
    13: "Король",
    14: "Туз"
}

try:
    num = int(input("Введите трехзначное число: "))
    suit = num // 100
    dignity = num % 100

    if 1 <= suit <= 4 and 11 <= dignity <=14:
        print(dignities.get(dignity), suits.get(suit))  # Использование .get() для
предотвращения KeyError
    else:
        print("Ошибка: такой карты нет!")
except ValueError:
    print("Ошибка: Введите целое число.")
```

Протокол работы программы (примеры):

```
Введите трехзначное число: 111
Валет пики

Введите трехзначное число: 414
Туз червы

Введите трехзначное число: 512
Ошибка: такой карты нет!

Введите трехзначное число: 310
Ошибка: такой карты нет!
```

Вывод:

В ходе выполнения практического задания были закреплены навыки использования условного оператора, словарей и обработки исключений для корректного определения названия игральной карты по ее числовому коду. Использование `get()` для доступа к значениям словаря повышает устойчивость программы к ошибкам.