

Урок № 5

Класс: 10

Тема урока: Вложенные и каскадные ветвления

Цель урока:

1) обучающий аспект: продолжить знакомить учащихся с условным оператором и научить применять его при составлении программ на языке программирования Python.

2) развивающий аспект: развитие мыслительной деятельности, речи, алгоритмического стиля мышления.

3) воспитательный аспект: воспитание эмоционально-положительной направленности на практическую деятельность, интереса к информатике, личной ответственности за результаты своей работы.

Тип урока: усвоение новых знаний.

Вид урока: урок-презентация, урок-практическое занятие.

Ресурсы: компьютеры, проектор, среда программирования Python IDLE, презентация.

Межпредметные связи: математика.

Основные понятия: условные операторы IF, ELSE, ELIF.

Ход урока (дидактическая структура урока)

1. Организационный этап (2 мин)	<u>УУД:</u> - Личностные - Коммуникативные
Приветствие.	
2. Проверка и по необходимости разбор домашнего задания (5 мин)	<u>УУД:</u> - Личностные - Познавательные - Регулятивные - Знаково-символические - Коммуникативные
Заданное на прошлом уроке домашнее задание: Улучшить созданную на прошлом уроке викторину, добавив в нее свои (не менее двух вопросов) задания. Дополнительное условие: при написании программы не пользоваться копированием текста, а весь код набирать самостоятельно.	
3. Теоретическая часть (15 мин)	<u>УУД:</u> - Личностные - Познавательные - Знаково-символические - Коммуникативные
Откройте тетради и запишите тему урока: «Вложенные и каскадные ветвления».	

На прошлом уроке мы научились составлять программы на языке Python с применением конструкции «ветвление» или «условный оператор if».

Напоминаю, если перевести на русский язык, конструкция условного оператора означает следующее:

Если <выполняется условие> делать: какие-то действия. Давайте еще раз посмотрим на общий вид условного оператора. Видим какими могут быть логические выражения.

Общий вид условного оператора

Полная форма условного оператора:

```
if <условие> : <оператор_1>
```

```
else : <оператор_2>
```

Неполная форма условного оператора:

```
if <условие> : <оператор>
```



Знаки отношений:

> больше

< меньше

== равно

>= больше или равно

<= меньше или равно

!= не равно

Сложные условия.

Чтобы составить сложное условие используются операторы:

and - «и», or - «или», not - «не»,

Например:

```
if a>0 and a<10 or a==100:
```

```
    print(a)
```

Будет ли напечатано a, если a равно 7? А если a равно 20?

Приоритет :

1) отношения (<, >, <=, >=, ==, !=)

2) not («НЕ»)

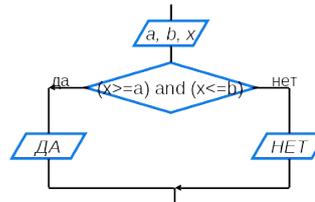
3) and («И»)

4) or («ИЛИ»)

Рассмотрим еще раз несколько примеров использования условного оператора.

Условный оператор

```
print ('Определение принадлежности точки  
отрезку')  
a = int (input ('Введите a: '))  
b = int (input ('Введите b: '))  
x = int (input ('Введите x: '))  
if x >= a and x <= b:  
    print ('Точка принадлежит отрезку')  
else:  
    print ('Точка не принадлежит отрезку')
```



Программа определения принадлежит ли точка отрезку. Вспоминаем из курса математики в каком случае точка принадлежит отрезку.

Ответ (Точка принадлежит отрезку, если значение ее координаты располагается между значениями, задающими концы отрезка).

Верно. Для проверки достаточно поверить попадает ли заданное значение для точки в промежуток между концами отрезка или нет. Это мы видим на слайде.

Итак, переходим к вложенному ветвлению.

Вложенные ветвления

```
if <условие1> :  
    if <условие2> :  
        <оператор1>  
    else: <оператор2>  
else: <оператор3>
```

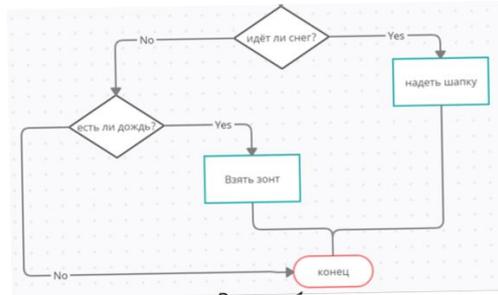
Вот пример схемы с вложенными ветвлениями. Когда после проверки условия необходимо либо его дополнительно скорректировать, либо дополнить. Т.е. фактически сделать выбор внутри выбора.

Записываем в тетрадь.

Теперь рассмотрим каскадное ветвление на примере.

Каскадное ветвление

Если после **else** сразу следует еще один оператор **if**, можно использовать каскадное ветвление со служебным словом **elif** (сокращение от **else-if**): если очередное условие ложно, выполняется проверка следующего условия и т. д.



Итак, зададим себе вопрос: «Идет ли снег на улице?» Если ответ «Да», то мы наденем шапку, если же «нет», то тут у нас сразу возникает следующий вопрос: «Идет ли дождь», от ответа на него будет зависеть брать нам зонт или нет. Так вот, если сразу за **ELSE** следует новое условие **IF** можно использовать каскадное ветвление со служебным словом **ELIF**.

4. Работа на компьютерах
(19 мин)

УУД:

- Личностные
- Регулятивные
- Познавательные
- Знаково-символические
- Коммуникативные

Учащиеся пишут программы на компьютерах под руководством учителя, согласно заданным блок схемам

Задачи:

Для примера напишем программу определяющую возрастную группу. Пусть пользователь вводит свой возраст а программа определит к какой группе он относится.

```
age = int(input("введите возраст"))
if age > 0 and age <= 10:
    print("вы ребёнок")
```

Продолжим добавлять условия

```

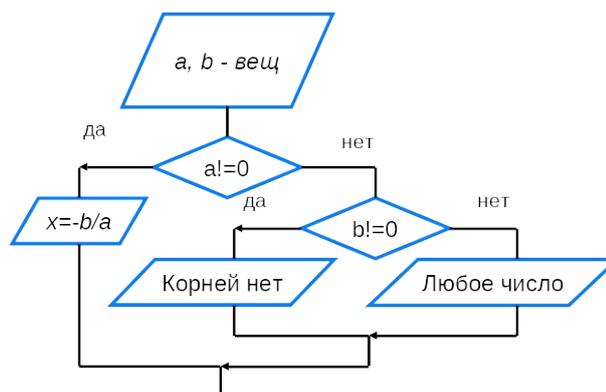
age = int(input("введите возраст"))
if age > 0 and age <= 10:
    print("вы ребёнок")
elif age > 10 and age <= 16:
    print("вы подросток")

```

Самостоятельно, добавьте еще несколько условий, избегая копирования текста.

Задание 2.

Решение линейного уравнения



Пример решения

Решение линейного уравнения

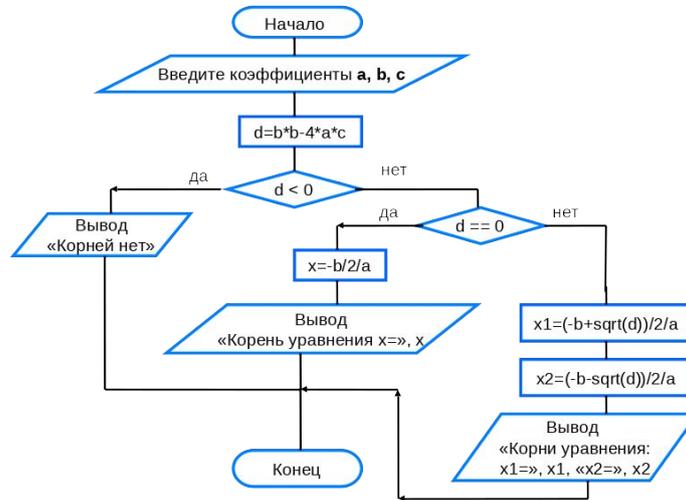
```

# Решение линейного уравнения
a = float( input ('Введите коэффициент
a>>'))
b = float( input ('Введите коэффициент
b>>'))
if a != 0:
    x = -b/ a
    print ('Корень уравнения x=', x)
elif b != 0:
    print ('Корней нет')
else:
    print(' x-любое число')

```

Задание 3.

Блок-схема решения КВУР



Пример решения

```
from math import * # подключение модуля для вычисления
квadrатного корня
# Решение квадратного уравнения
print ('Введите коэффициенты a, b, c >>')
a = float (input ('a='))
b = float (input ('b='))
c = float (input ('c='))
d = b * b - 4 * a * c
if d < 0:
    print ('Корней нет')
elif d == 0:
    x = - b/2/ a
    print ('Корень уравнения x=', "{:6.4f}". format(x))
else:
    x1 = (-b+sqrt(d))/2/ a
    x2 = (-b - sqrt(d))/2/ a
    print ('Корни уравнения:')
    print ('x1=', "{:6.4f}". format(x1))
    print ('x2=', "{:6.4f}". format(x2))
```

--

5. Домашнее задание (1 мин)	<u>УУД:</u> - Личностные - Коммуникативные
<p>Домашнее задание:</p> <p>Написать программы:</p> <ol style="list-style-type: none">1) Доделать программу нахождения корней квадратного уравнения.На повторение условного оператора.2) Дано целое число. Если оно является положительным, то умножить его на 3; в противном случае вычесть из него 100. Вывести полученное число.3) Определить, является ли число a делителем числа b.4) Определить возможность существования треугольника по сторонам. (Треугольник существует только тогда, когда сумма любых двух его сторон больше третьей).	