



### Методические рекомендации по теме «Вложенное условие»

#### Цель:

- объяснение понятия вложенное условие, применение вложенных условий для решения задач на языке Python.

#### Задачи:

- расширение кругозора обучающихся в области информатики и программирования;
- знакомство с вложенными условиями на языке Python;
- решение программных прикладных задач с использованием вложенных условий на языке Python;
- ранняя профориентация школьников, профессиональная деятельность программиста;
- развитие интеллектуальных способностей, логического и критического мышления.

#### Планируемые результаты

*Личностные:* обучающиеся получают навыки активной коммуникации в группе, осознанной ориентировки в мире ИТ профессий, постановки собственных образовательных задач и владение первичными навыками деятельностного анализа и критической оценки получаемой информации.

*Предметные:* обучающиеся получают представления: о программировании вложенных условий на языке Python; об округлении чисел в языке программирования Python; о положительных и отрицательных сторонах применения вложенных условий в языке программирования «Python»; о прикладном использовании вложенных условий в программных проектах; о возможностях и особенностях применения вложенных условий в практике работы программиста.

*Метапредметные:* обучающиеся получат возможность владения обще предметными понятиями «вложенное условие»; владение информационно-логическими умениями; умениями самостоятельно планировать пути достижения целей; умениями принятия решений и осуществления осознанного выбора; повысят уровень ИКТ-компетентности.

## **Материалы к занятию**

Приложение 1: Сценарный план ролика

Приложение 2: Домашнее задание и практика

Приложение 3: Краткие организационно-методические рекомендации по организации работы на занятии.

Приложение 4: Алгоритм решения квадратного уравнения, имеющего вещественные корни (дополнительно).

## **Ход проведения урока**

### **1. Организационный момент**

#### **Мотивация на учебную деятельность.**

Приветствие учащихся, сообщение темы и целей занятия (мы узнаем, что такое вложенное «условие» в программировании и как работать с такими условиями на языке Python; научимся использовать вложенные условия для создания сложных многоуровневых алгоритмов и сделаем несколько программных проектов, чтобы понять как вложенные условия используются программистами, познакомимся с сильными и слабыми сторонами применения вложенных условий).

#### ***Проблемная дискуссия*** по вопросам:

- Как бы вы могли описать вложенное условие?
- Приведите примеры вложенных условий из реальной жизни?

- Зачем нужны вложенные условия в программировании?

**Итоги дискуссии** (обобщаются преподавателем и фиксируются ответы учеников на доске, чтобы вернуться к ним и оценить правильность предположений учеников на этапе рефлексии):

- в реальной жизни мы часто встречаемся с ситуациями, где выбор происходит на нескольких уровнях;
- использование вложенных условий позволяет программисту реализовать в коде такие сложные условные конструкции.

Преподаватель называет ученикам тему и цели урока.

## **2. Вводный блок.**

### **Тема.**

Преподаватель при необходимости останавливая трансляцию, комментируя дополнительно тему занятия).

*\*см. сцены 1 – 2 (здесь и далее приводится Таблица «Содержание видеоролика». Приложение 1)*

## **3. Блок повторения.**

### **Блиц-опрос.**

Преподаватель предлагает ученикам ответить на 5 вопросов по предыдущей теме; задания выполняются в сопровождении видеоролика с использованием таймера; ученики выполняют задания, голосуют, обсуждают результаты. Процедура голосования определяется инструкцией в сцене 3; учитель должен убедиться, что всем понятна процедура голосования.

*Преподаватель может поставить ролик на паузу и обсудить результаты голосования; объяснить правильный ответ руководствуясь материалами предыдущего занятия*

*\*см. сцены 3 – 7*

## **4. Теоретический блок.**

### **Вложенное условие.**

Продолжение демонстрации ролика с дальнейшим обсуждением вопросов:

- Почему нам не всегда подходит простая связка условных операторов (**if-elif-else**) и возникает необходимость использовать вложенные условия?
- Какой синтаксис используется в Python для вложенных условий?
- Почему не стоит использовать в коде излишне сложные конструкции с вложенными условиями?

*При необходимости преподаватель может поставить ролик на паузу и дать дополнительные пояснения по материалу; если ответы на вопросы вызывают у учеников затруднения, преподаватель может вывести нужную сцену ролика на экран для помощи ученикам.*

*\*см. сцены 8 – 10*

### **5. Блок заданий.**

#### **Проект «Чат-бот. Алгоритм русской части».**

К началу демонстрации блока заданий ученики должны занять рабочие места и запустить Python (терминал IDLE) на своих компьютерах.

**«Чат-бот. Алгоритм русской части»:** включает **3 практических задания** с таймером; после завершения работы таймера демонстрируется разбор задания. Задания представляют собой этапы создания программного проекта с использованием вложенных условий.

**После выполнения задания** ученики получают работающий продукт – двуязычный чат-бот (русский и английский языки). *На сцене разбора задания преподаватель ставит ролик на паузу и вместе с учениками проводит разбор задания.*

*\*см. сцены 11 – 16*

### **6. Рефлексия. Сообщение домашнего задания.**

Завершаем демонстрацией ролика и кратким обобщением материалов занятия. Преподаватель возвращается к зафиксированным в ходе дискуссии в начале урока предположениям учеников и обсуждает насколько их предположения были правильными, делаются выводы.

Преподаватель дает ученикам домашнее задание к следующему занятию (*Приложение 2*).

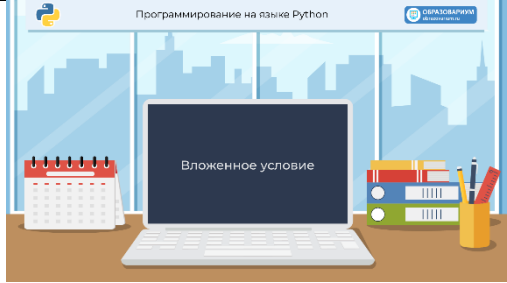
*\*см. сцена 17*

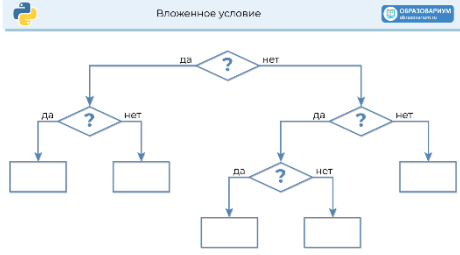
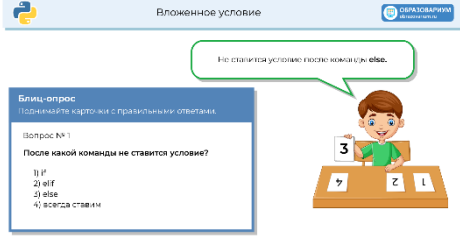
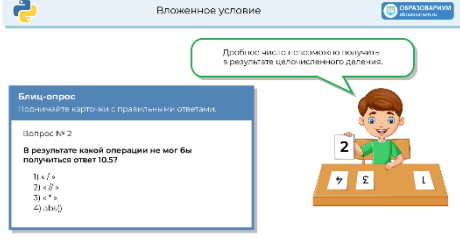
## Приложение 1

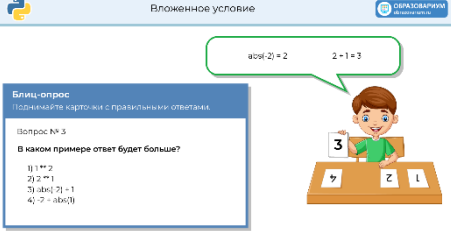
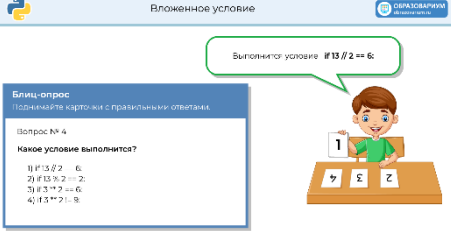
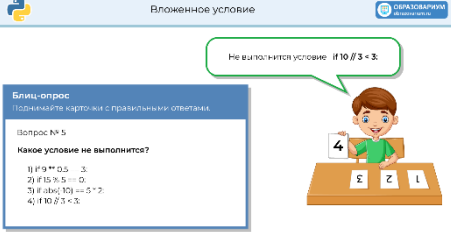

### Сценарный план видеоролика

В таблице «Содержание видеоролика» представлен дикторский текст из видеоролика, примеры заданий и задач, которые будут демонстрироваться на экране. Учитель при подготовке к уроку может ознакомиться с содержанием видеоролика в текстовом формате, при необходимости распечатать фрагменты текста или примеры заданий и задач для использования в работе с учениками. Распечатанные тексты и задания из таблицы также можно применять в качестве раздаточного материала как на уроке, так и для домашних заданий.

Таблица. Содержание видеоролика

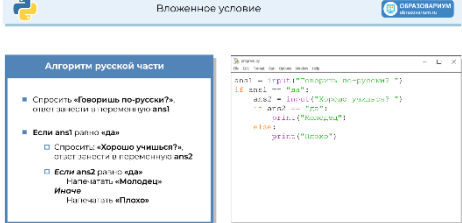
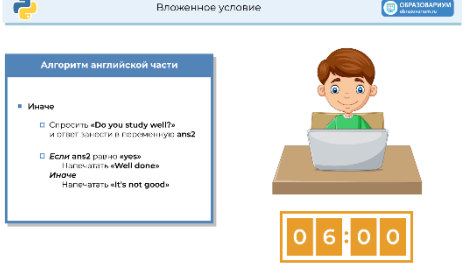
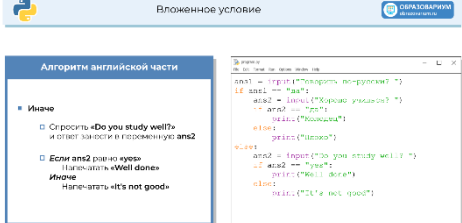
Название блока	Содержание блока и комментарии	Фрагменты из видеоролика	№ сцен
Вводный блок. <b>Мы узнаем</b>	<p><i>Обозначаем ученикам тему и цели урока.</i></p> <p>Вложенное условие</p>	 <p>Сцена 1</p>	<p>1</p> <p>2</p>

	<p>Условные операторы обеспечивают определенное поведение программы в зависимости от выполнения/не выполнения условий. В некоторых ситуациях условий несколько, и они зависят друг от друга. В таком случае на помощь придут вложенные условия, которые позволяют создавать более сложные алгоритмы.</p>	 <p>Сцена 2</p>	
<p>Блок повторения.</p> <p><b>Блиц-опрос</b></p>	<p><i>Повторение материала предыдущего урока; на столе имеются пронумерованные карточки; после каждого вопроса выбираем ту, номер которой, совпадает с правильным ответом.</i></p> <p><b>Первый вопрос.</b> После какой команды не ставится условие?</p> <ol style="list-style-type: none"> <li>1) if</li> <li>2) elif</li> <li>3) else</li> <li>4) всегда ставим</li> </ol> <p><i>ОТВЕТ № 3: не ставится условие после команды <b>else</b>.</i></p>	 <p>Сцена 3</p>	<p>3</p> <p>4</p> <p>5</p> <p>6</p> <p>7</p>
	<p><b>Второй вопрос.</b> В результате какой операции не мог бы получиться 10.5?</p> <ol style="list-style-type: none"> <li>1) &lt;&lt;/&gt;</li> <li>2) &lt; &lt;/&gt;</li> <li>3) &lt;&lt;*&gt;</li> <li>4) abs()</li> </ol> <p><i>ОТВЕТ № 2: дробное число невозможно получить в результате целочисленного деления.</i></p>	 <p>Сцена 4</p>	

	<p><b>Третий вопрос.</b> В каком примере ответ будет больше?</p> <ol style="list-style-type: none"> <li>1) <math>1 ** 2</math></li> <li>2) <math>2 ** 1</math></li> <li>3) <math>\text{abs}(-2) + 1</math></li> <li>4) <math>-2 + \text{abs}(1)</math></li> </ol> <p><i>ОТВЕТ № 3: <math>\text{abs}(-2) = 2</math>; <math>2 + 1 = 3</math>.</i></p>	<p>Вложенное условие</p>  <p>Сцена 5</p>	
	<p><b>Четвертый вопрос.</b> Какое условие выполнится?</p> <ol style="list-style-type: none"> <li>1) <math>\text{if } 13 // 2 == 6:</math></li> <li>2) <math>\text{if } 13 \% 2 == 2:</math></li> <li>3) <math>\text{if } 3 ** 2 == 6:</math></li> <li>4) <math>\text{if } 3 ** 2 != 9:</math></li> </ol> <p><i>ОТВЕТ № 1: выполнится условие <math>\text{if } 13 // 2 == 6:</math></i></p>	<p>Вложенное условие</p>  <p>Сцена 6</p>	
	<p><b>Пятый вопрос.</b> Какое условие не выполнится?</p> <ol style="list-style-type: none"> <li>1) <math>\text{if } 9 ** 0.5 == 3:</math></li> <li>2) <math>\text{if } 15 \% 5 == 0:</math></li> <li>3) <math>\text{if } \text{abs}(-10) == 5 * 2:</math></li> <li>4) <math>\text{if } 10 // 3 &lt; 3:</math></li> </ol> <p><i>ОТВЕТ № 4: Не выполнится условие <math>\text{if } 10 // 3 &lt; 3:</math></i></p>	<p>Вложенное условие</p>  <p>Сцена 7</p>	
<p>Теоретический блок.</p> <p><b>Вложенное условие.</b></p>	<p>При необходимости преподаватель может поставить ролик на паузу и дать дополнительные пояснения по материалу</p> <p>Рассмотрим следующую ситуацию. Я хочу пойти в кино, но не знаю есть ли билеты или нет. Также я готов пойти на каток, но не знаю: работает он ли закрыт.</p> <p>При любом выборе (кино или каток) меня ждет новый выбор (получится или нет).</p>	<p>Вложенное условие</p>  <p>Сцена 8</p>	<p>8 9 10</p>

	<p>Это алгоритм с <b>вложенными условиями</b>. Вложенные условия позволяют добавить к основным условиям второстепенные.</p> <p>Выглядеть это может следующим образом. В первом случае используется <b>if</b>, а во втором коде – <b>elif</b>.</p> <p>Первый код. Сначала могу пойти в кино. А после кино могу пойти на каток. Второй код. Я могу пойти или в кино, или на каток. Совместить не получится. Но в любом случае мне предстоит повторный выбор – получится или нет.</p> <p>Количество вложенных условий не ограничено, однако увлекаться ими особенно на первых порах не следует.</p> <p>Вложенные условия располагаются после основных условий с отступом в 4 пробела. Ставить их можно после любого оператора.</p> <p>Команды <b>if-elif-else</b>, относящиеся к одному условию, находятся ровно друг под другом! После вложенных условий мы также в конце ставим двоеточие и блок команд начинаем с отступом в еще 4 пробела</p>	<div data-bbox="1507 341 1960 564"> <p>Вложенное условие</p> <div> <pre> if пойду в кино:     if есть билеты:         посмотрю фильм     else:         вернусь домой if пойду на каток:     if он открыт:         покатюсь     else:         вернусь домой </pre> <pre> if пойду в кино:     if есть билеты:         посмотрю фильм     else:         вернусь домой elif пойду на каток:     if он открыт:         покатюсь     else:         вернусь домой </pre> </div> </div> <p>Сцена 9</p> <div data-bbox="1507 671 1960 906"> <p>Вложенное условие</p> <div> <pre> if пойду в кино:     if есть билеты:         посмотрю фильм     else:         вернусь домой elif пойду на каток:     if он открыт:         покатюсь     else:         вернусь домой </pre> </div> <p>Отступ – 4 пробела</p> </div> <p>Сцена 10</p>	
<p>Блок заданий. <b>Практические задания:</b> Задание 1</p>	<p><i>После окончания дикторского текста запускается таймер на 6 мин.</i></p> <p><b>Задание 1. Чат-бот. Алгоритм русской части.</b></p> <p>Спросить: «Говоришь по-русски?», ответ занести в переменную ans1.</p> <p><b>Если ans1 равно «да»</b> Спросить: «Хорошо учишься?», ответ занести в переменную ans2. <b>Если ans2 равно «да»</b> Напечатать «Молодец!» <b>Иначе</b> Напечатать «Плохо.»</p>	<div data-bbox="1507 1011 1960 1281"> <p>Вложенное условие</p> <div> <p>Алгоритм русской части</p> <ul style="list-style-type: none"> <li>Спросить «Говоришь по-русски?» ответ занести в переменную ans1</li> <li>Если ans1 равно «да» <ul style="list-style-type: none"> <li>Спросить «Хорошо учишься?» ответ занести в переменную ans2</li> <li>Если ans2 равно «да» Напечатать «Молодец!»</li> <li>Иначе Напечатать «Плохо»</li> </ul> </li> </ul> </div> <div> </div> </div> <p>Сцена 11</p>	<p>11 12 13 14 15 16</p>



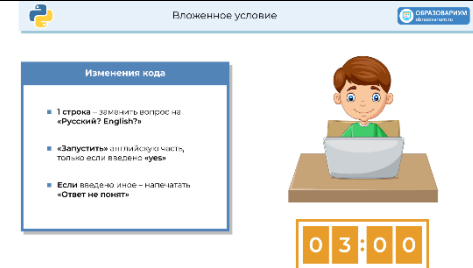
	<p><b>Разбор задания 1.</b></p> <p>Код программы может выглядеть так.</p> <pre>ans1 = input("Говоришь по-русски? ") if ans1 == "да":     ans2 = input("Хорошо учишься? ")     if ans2 == "да":         print("Молодец!")     else:         print("Плохо.")</pre> <p><i>После окончания времени провести разбор решения, анализируя код.</i></p>	 <p>Сцена 12</p>	
	<p><i>После окончания дикторского текста запускается таймер на 6 мин. В продолжении задания вопрос будет задан на английском языке.</i></p> <p><b>Задание 1. Чат-бот. Алгоритм английской части. Продолжение. Иначе</b></p> <p>Спросить «Do you study well?» и ответ занести в переменную ans2 <b>Если</b> ans2 равно «yes» Напечатать «Well done!» <b>Иначе</b> Напечатать «It's not good.»</p>	 <p>Сцена 13</p>	
	<p><b>Разбор задания 1. Продолжение</b></p> <p>Код продолжения программы может выглядеть так.</p> <pre>else:     ans2 = input("Do you study well?")     if ans2 == "yes":         print("Well done!")     else:         print("It's not good.")</pre> <p><i>После окончания времени провести разбор решения, анализируя код.</i></p>	 <p>Сцена 14</p>	

*После окончания дикторского текста запускается таймер на 3 мин.  
В окончании задания внесем изменения в код.*

### Задание 1. Чат-бот. Окончание

Изменения кода

- 1 строка – заменить вопрос на «Говоришь по-русски or English? »
- «запустить» английскую часть только если введено «yes»
- если введено иное – напечатать «Ответ не понят.»



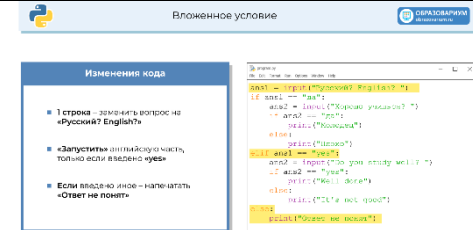
Сцена 15

### Разбор задания 1. Окончание

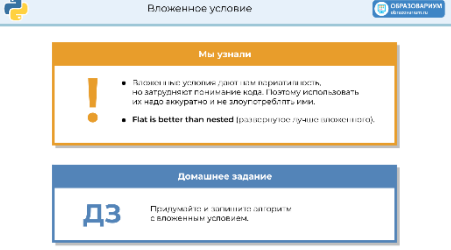
Код продолжения программы может выглядеть так.

```
ans1 = input("Говоришь по русски or English? ")
if ans1 == "да":
    ans2 = input("Хорошо учишься? ")
    if ans2 == "да":
        print("Молодец!")
    else:
        print("Плохо.")
elif ans1 == "yes":
    ans2 = input("Do you study well? ")
    if ans2 == "yes":
        print("Well done!")
    else:
        print("It's not good.")
else:
    print ("ответ не понят")
```

*После окончания времени провести разбор решения, анализируя код.*



Сцена 16

Блок завершения занятия. <b>Рефлексия.</b> <b>Сообщение домашнего задания</b>	<i>Завершаем демонстрацией ролика и кратким обобщением материалов занятия.</i>  <b>Подведем итоги:</b>  - мы узнали, что вложенные условия дают нам вариативность, но затрудняют понимание кода; - использовать вложенные условия надо аккуратно и не злоупотреблять ими.  <i>Преподаватель дает ученикам домашнее задание к следующему занятию (Приложение 2).</i>	 <p>Сцена 17</p>	17
---	--	---	----

## Приложение 2

### Домашнее задание

Придумать и записать свой алгоритм, в котором будут использованы вложенные условия.

### Практика

#### Проект «Дни недели»

Спросите у пользователя есть ли сейчас каникулы. Если нет каникул, то узнайте какой сегодня день: выходной или рабочий.

Обработайте все возможные варианты. На каждый вариант придумайте и выведите на экран свое сообщение о количестве свободного времени, дел или возможности отдохнуть.

Обработайте варианты, когда пользователь ввел ответ, которого вы не ожидали, выведите в этих случаях соответствующие сообщения.

*\* Используйте в проекте двойное вложенное условие.*

## Проект «Покупка»

Запросите у пользователя количество имеющихся денег. Обработайте вариант, когда у пользователя нет денег, выведите соответствующее сообщение.

В остальных случаях узнайте стоимость товара. Определите, достаточно ли денег и запросите у пользователя подтверждение покупки. Обработайте все возможные варианты. На каждый вариант придумайте и выведите на экран свое сообщение.

*\* Используйте в проекте тройное вложенное условие.*

## Приложение 3

### Краткие организационно-методические рекомендации по организации работы на занятии

«Вложенное условие».

**В начале занятия** необходимо вспомнить три команды условного оператора – **if/elif/else**: чем они отличаются друг от друга, последовательность применения, сколько их может быть в одном условии. Особое внимание уделите синтаксису: двоеточия и отступы.

Для наглядности можно написать несколько примеров на доске используя алгоритмы, которые придумали ребята в качестве домашнего задания.

**Перед просмотром блока повторения** из ролика необходимо раздать дидактический материал для выполнения заданий из блока повторение (по 4 пронумерованных карточки)

Во время голосований карточками можно останавливать ролик и вести учет правильных ответов. По окончании блока – отметить тех, у кого наилучший результат.

Далее карточки необходимо собрать.

**После теоретического блока** можно остановить ролик и попросить привести примеры ситуаций, где нам понадобится вложенное условие. При формулировках алгоритмов попросите ребят использовать слова «**если**», «**иначе если**», «**иначе**»

**Блок заданий** представлен одной программой, которая выполняется в три этапа. После каждого этапа код можно тестировать на предмет обнаружения ошибок или некорректной работы.

Если останется время – можно сделать еще один вариант «языка». Для этого измените поясняющий текст в строке № 1 и добавьте еще один блок с условием **elif**, взяв за образец «английскую часть».

#### **Приложение 4**

Дополнительно

**Алгоритм решения квадратного уравнения, имеющего вещественные корни.**

Изучите предложенный алгоритм и напишите код программы. Вы можете сверить свой код с образцом решения.

**Условие**

Дано квадратное уравнение  $ax^2 + bx + c = 0$ . Задача: найти корни данного уравнения или убедиться в отсутствии корней.

## Алгоритм «Решение квадратного уравнения»

- Запросить число и занести его в переменную **a**
- Запросить число и занести его в переменную **b**
- Запросить число и занести его в переменную **c**
- Вычислить **d** по формуле  **$b^2 - 4ac$**
- Если **d** больше нуля
  - Вычислить  $x_1$  и  $x_2$  по формуле  $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$
  - Вывести значения  $x_1$  и  $x_2$
- Иначе если **d** равен нулю
  - Вычислить  $x$  по формуле  $\frac{-b}{2a}$
  - Вывести значение  $x$
- Иначе
  - Вывести фразу «Корней нет»