



### Методические рекомендации по теме «Цикл с условием»

#### Цель:

- знакомство с условиями в циклических алгоритмах на языке Python и применение для решения прикладных задач с использованием условий в циклических алгоритмах.

#### Задачи:

- расширение кругозора обучающихся в области информатики и программирования;
- знакомство с условиями в циклических алгоритмах на языке Python;
- решение программных прикладных задач с использованием условий в циклических алгоритмах;
- 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 циклических алгоритмах на языке Python; об использовании условий в циклах; о прикладном использовании условий в циклах в программных проектах; о возможностях и особенностях применения условий в цикле в практике работы программиста.

*Метапредметные:* обучающиеся получают возможность владения обще предметными понятиями «цикл», «условие», «условие в цикле»; владение информационно-логическими умениями; владение умениями самостоятельно

планировать пути достижения целей; умениями принятия решений и осуществления осознанного выбора; повысят уровень ИКТ-компетентности.

### **Материалы к занятию**

Приложение 1: Сценарный план видеоролика

Приложение 2: Домашние задание и практика

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

Приложение 4: Алгоритм Евклида для нахождения наибольшего общего делителя двух натуральных чисел  
(дополнительно)

### **Ход проведения урока**

#### **1. Организационный момент.**

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

Приветствие учащихся, сообщение темы и целей занятия (мы узнаем, что такое циклические алгоритмы с условием и как их использовать в языке Python; научимся использовать условия в циклических алгоритмах, такой цикл повторяется до тех пор, пока записанное в нем условие истинно).

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

- Как бы вы определили понятие «цикл»? Приведите примеры.
- Как бы вы думаете, что такое «цикл» в программировании и зачем они нужны?
- Что такое бесконечный цикл в жизни и в программировании?
- Где нам может понадобиться «цикл» с ограниченным числом повторений?

- Для чего, по вашему мнению, нужен цикл с условием? Приведите примеры

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

- «цикл» в программировании означает повторяющиеся действия;
- использование циклов позволяет оптимизировать код, избежав многократной записи повторяющихся действий.

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.**

**Определение и виды циклов. Цикл с условием.**

Продолжение демонстрации ролика с дальнейшим обсуждением вопросов:

- Какие виды циклов бывают?
- Для чего служит команда **while**?
- Что такое «тело цикла»?
- Какой синтаксис используется для записи цикла с условием?

*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 – 12*

## **5. Блок заданий.**

**Проекты «Положительные и отрицательные числа», «Искатель сокровищ».**

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

**«Положительные и отрицательные числа» и «Искатель сокровищ»:** включает **2 практических задания** с таймером; после завершения работы таймера демонстрируется разбор задания. Задания представляют собой программные проекты с использованием циклов с условием.

**После выполнения задания** ученики получают два работающих программных продукта – программа для определения положительных и отрицательных чисел, программа симулятор добычи сокровищ в компьютерной игре.

*На сцене разбора задания преподаватель ставит ролик на паузу и вместе с учениками проводит разбор задания.*

*\*см. сцены 13 – 21*

## **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

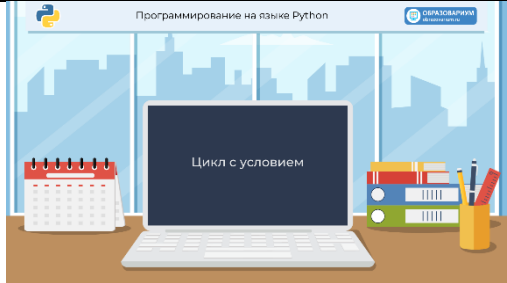
были правильными, делаются выводы. Преподаватель дает ученикам домашнее задание к следующему занятию (Приложение 2). \*см. сцена 22

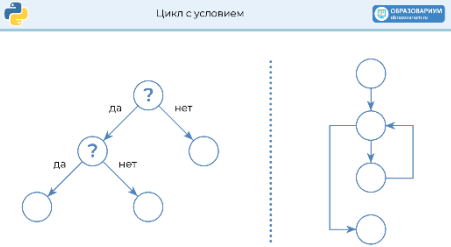
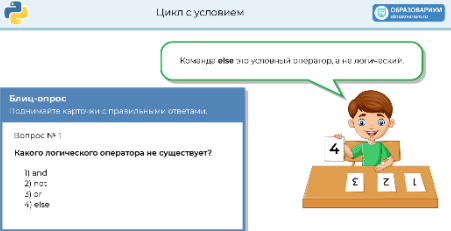
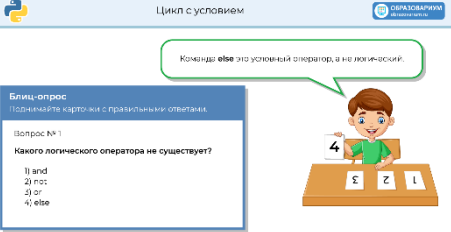
## Приложение 1

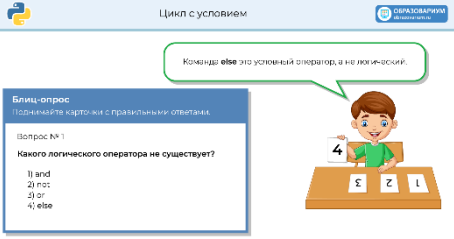
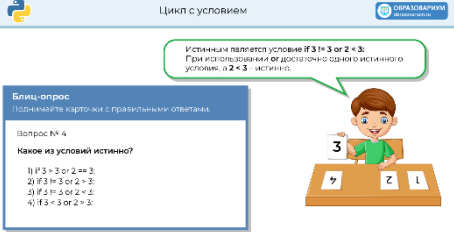
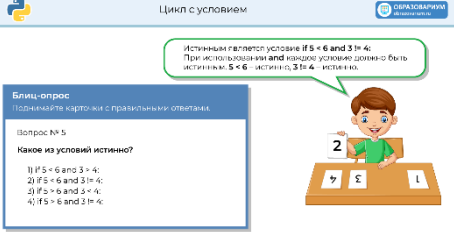
### Сценарный план видеоролика



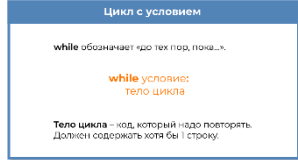
В таблице «Содержание видеоролика» представлен краткий текст из видеоролика, примеры заданий и задач, которые будут демонстрироваться на экране. Учитель при подготовке к уроку может ознакомиться с содержанием видеоролика в текстовом формате, при необходимости распечатать фрагменты текста или примеры 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

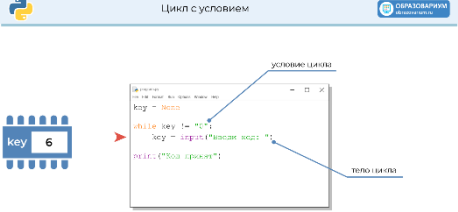
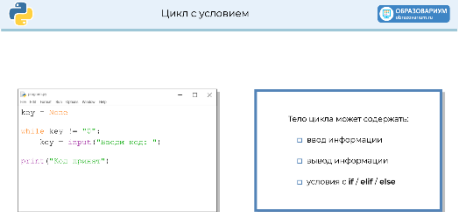
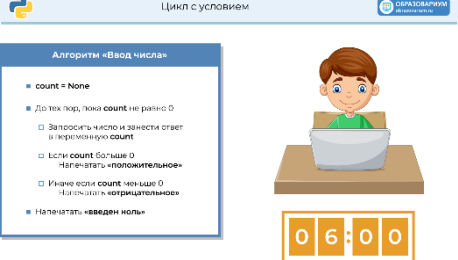
Название блока	Содержание блока и комментарии	Фрагменты из видеоролика	№ сцен
Вводный блок. Мы узнаем	Обозначаем ученикам тему и цели урока.  Цикл с условием	 Сцена 1	1 2

	<p>Все программы, которые мы создавали до сегодняшнего дня были одноразовыми.</p> <p>В зависимости от того, было условие истинным или ложным, срабатывал один или другой фрагмент кода.</p> <p>Но он срабатывал только один раз.</p> <p>Однако помимо условных и линейных алгоритмов имеется третий вид – циклические алгоритмы.</p> <p>И вот о них мы сегодня и поговорим.</p>	 <p>Цикл с условием</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> Какой логический оператор не существует?</p> <ol style="list-style-type: none"> <li>1) and</li> <li>2) not</li> <li>3) or</li> <li>4) else</li> </ol> <p><i>Ответ 4. Команда <b>else</b> это условный оператор, а не логический.</i></p>	 <p>Цикл с условием</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> Что может быть в одном условии?</p> <ol style="list-style-type: none"> <li>1) Несколько and</li> <li>2) Несколько or</li> <li>3) И and, и or</li> <li>4) Все вышеперечисленное?</li> </ol> <p><i>Ответ 4. Никаких ограничений на количество логических операторов в одном условии не существует.</i></p>	 <p>Цикл с условием</p> <p>Сцена 4</p>	

	<p><b>Третий вопрос.</b> В каком варианте условие записано правильно?</p> <ol style="list-style-type: none"> <li>1) <code>if 2 = 2:</code></li> <li>2) <code>if 2 &lt; 1:</code></li> <li>3) <code>elif 2 != 2:</code></li> <li>4) <code>elif 2 &gt;&lt; 1:</code></li> </ol> <p><i>Ответ 2. Правильно записано условие <code>if 2 &lt; 1</code>:. Во всех остальных вариантах есть ошибка синтаксиса.</i></p>	 <p>Цикл с условием</p> <p>Команда <code>else</code> это условный оператор, а не логический.</p> <p>Блиц-опрос Поднимайте карточки с правильными ответами.</p> <p>Вопрос № 1 Какого логического оператора не существует?</p> <ol style="list-style-type: none"> <li>1) <code>and</code></li> <li>2) <code>not</code></li> <li>3) <code>or</code></li> <li>4) <code>else</code></li> </ol> <p>Сцена 5</p>	
	<p><b>Четвертый вопрос.</b> Какое из условий истинно?</p> <ol style="list-style-type: none"> <li>1) <code>if 3 &gt; 3 or 2 == 3:</code></li> <li>2) <code>if 3 != 3 or 2 &gt; 3:</code></li> <li>3) <code>if 3 != 3 or 2 &lt; 3:</code></li> <li>4) <code>if 3 &lt; 3 or 2 &gt; 3:</code></li> </ol> <p><i>Ответ 3. Истинным является условие <code>if 3 != 3 or 2 &lt; 3</code>:. При использовании <code>or</code> достаточно одного истинного условия, а <code>2 &lt; 3</code> – истинно.</i></p>	 <p>Цикл с условием</p> <p>Истинным является условие <code>if 3 != 3 or 2 &lt; 3</code>:. При использовании <code>or</code> достаточно одного истинного условия, а <code>2 &lt; 3</code> – истинно.</p> <p>Блиц-опрос Поднимайте карточки с правильными ответами.</p> <p>Вопрос № 4 Какое из условий истинно?</p> <ol style="list-style-type: none"> <li>1) <code>if 3 &gt; 3 or 2 == 3:</code></li> <li>2) <code>if 3 != 3 or 2 &gt; 3:</code></li> <li>3) <code>if 3 != 3 or 2 &lt; 3:</code></li> <li>4) <code>if 3 &lt; 3 or 2 &gt; 3:</code></li> </ol> <p>Сцена 6</p>	
	<p><b>Пятый вопрос.</b> Какое из условий истинно?</p> <ol style="list-style-type: none"> <li>1) <code>if 5 &lt; 6 and 3 &gt; 4:</code></li> <li>2) <code>if 5 &lt; 6 and 3 != 4:</code></li> <li>3) <code>if 5 &gt; 6 and 3 &lt; 4:</code></li> <li>4) <code>if 5 &gt; 6 and 3 != 4:</code></li> </ol> <p><i>Ответ 2. Истинным является условие <code>if 5 &lt; 6 and 3 != 4</code>:. При использовании <code>and</code> каждое условие должно быть истинным. <code>5 &lt; 6</code> – истинно, <code>3 != 4</code> – истинно,</i></p>	 <p>Цикл с условием</p> <p>Истинным является условие <code>if 5 &lt; 6 and 3 != 4</code>:. При использовании <code>and</code> каждое условие должно быть истинным. <code>5 &lt; 6</code> – истинно, <code>3 != 4</code> – истинно.</p> <p>Блиц-опрос Поднимайте карточки с правильными ответами.</p> <p>Вопрос № 5 Какое из условий истинно?</p> <ol style="list-style-type: none"> <li>1) <code>if 5 &lt; 6 and 3 &gt; 4:</code></li> <li>2) <code>if 5 &lt; 6 and 3 != 4:</code></li> <li>3) <code>if 5 &gt; 6 and 3 &lt; 4:</code></li> <li>4) <code>if 5 &gt; 6 and 3 != 4:</code></li> </ol> <p>Сцена 7</p>	

<p>Теоретический блок.</p> <p><b>Определение и виды циклов.</b></p> <p><b>Цикл с условием.</b></p>	<p><i>При необходимости преподаватель может поставить ролик на паузу и дать дополнительные пояснения по материалу</i></p> <p>Чтобы заставить программу несколько раз повторить одно и тоже действие существует специальный инструмент. Он называется цикл. Слово цикл можно перевести как повторение</p>	<p>Цикл с условием</p>  <p>Сцена 8</p>	<p>8 9 10 11 12</p>
	<p>Существует три вида циклов:</p> <ul style="list-style-type: none"> <li>– цикл с условием,</li> <li>– конечный – работающий определенное количество раз</li> <li>– бесконечный</li> </ul>	<p>Цикл с условием</p>  <p>Сцена 9</p>	
	<p><i>Рассмотрите пример цикла с условием</i></p> <p>Цикл с условием – означает, что надо что-то делать пока условие истинно. В Python для организации цикла используют команду <b>while</b>. <b>while</b> обозначает «до тех пор, пока...».</p> <p><b>while</b> условие:     тело цикла</p> <p>Тело цикла – код, который надо повторять. Должен содержать хотя бы 1 строку.</p>	<p>Цикл с условием</p>  <p>Сцена 10</p>	



	<p>Рассмотрим пример программного кода.</p> <pre>key = None while key != "0":     key = input("Введи код: ")     print("Код принят")</pre> <p><i>Проведите разбор примера, анализируя код.</i></p>	<div></div> <p>Сцена 11</p>	
	<p>В первой строке создается переменная без значения. Все дело в том, что для записи условия в строке два нам нужно чтобы программа уже знала эту переменную. Иначе возникнет ошибка.</p> <p>Тело цикла может содержать:</p> <ul style="list-style-type: none"><li>– ввод информации,</li><li>– вывод информации,</li><li>– условия с if/elif/else</li></ul>	<div></div> <p>Сцена 12</p>	
<p>Блок заданий. <b>Практические задания:</b> Задание 1 Задание 2</p>	<p><i>После окончания дикторского текста запускается таймер на 6 мин.</i></p> <p><b>Задание 1. Алгоритм «Ввод чисел»</b></p> <pre>count = None</pre> <p><b>До тех пор, пока count не равно 0:</b></p> <p><b>Запросить число и занести ответ в переменную count</b></p> <p><b>Если count больше 0:</b></p> <p><b>Напечатать «положительное»</b></p> <p><b>Иначе если count меньше 0:</b></p> <p><b>Напечатать «отрицательное»</b></p> <p><b>Напечатать «введен ноль»</b></p>	<div></div> <p>Сцена 13</p>	<p>13 14 15 16 17 18 19 20 21</p>

### Разбор задания 1.

Код вашей программы может выглядеть так.

```
count = None
```

```
while count != 0:
```

```
    count = int(input("Введи число: "))
```

```
    if count > 0:
```

```
        print("положительное")
```

```
    elif count < 0:
```

```
        print("отрицательное")
```

```
    print("введен ноль")
```

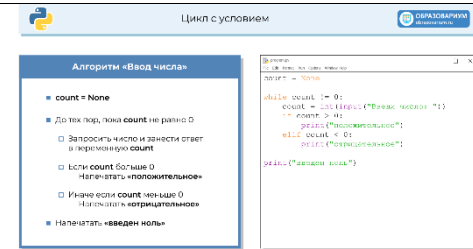
*После окончания времени провести разбор решения, анализируя код.*

### Вопрос

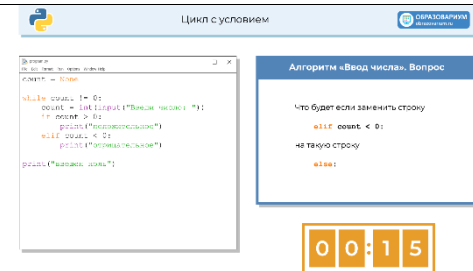
— что будет если в строке шесть убрать **elif** с условием и прописать **else**?

### Правильный ответ:

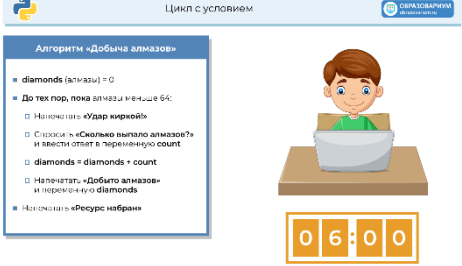
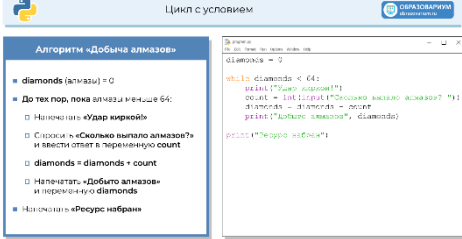
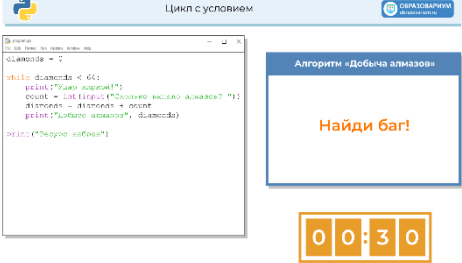
- если мы введем ноль, сначала программа напишет, что число отрицательное, потом снова перейдет на строку два. Поймет, что введен ноль, и закончит цикл фразой «введен ноль».

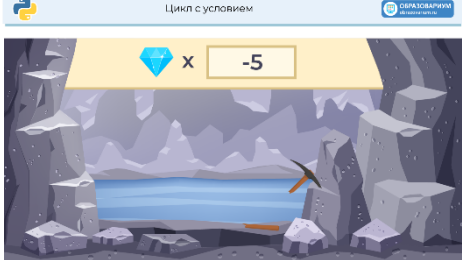
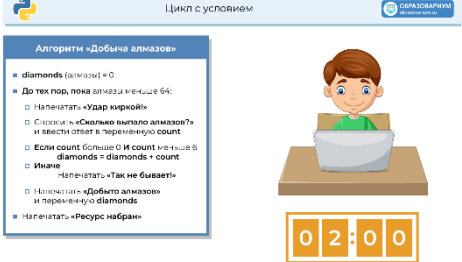
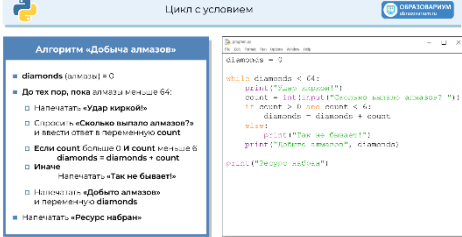


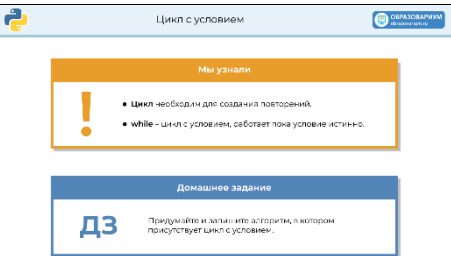
Сцена 14



Сцена 15

	<p><i>После окончания дикторского текста запускается таймер на 6 мин.</i></p> <p><b>Задание 2. Алгоритм «Добыча алмазов»</b></p> <p>diamonds (алмазы) = 0</p> <p>до тех пор, пока алмазы меньше 64:</p> <p>    <b>напечатать</b> «удар киркой»</p> <p>    <b>спросить</b> «сколько выпало алмазов» и <b>ввести</b> ответ в переменную count</p> <p>    diamonds = diamonds + count</p> <p>    <b>напечатать</b> «добыто алмазов» и переменную diamonds</p> <p>    <b>напечатать</b> «ресурс набран».</p> <p><i>После окончания времени провести разбор решения, анализируя код.</i></p>	 <p>Сцена 16</p>
	<p><b>Разбор задания 2.</b></p> <p>Код может выглядеть следующим образом</p> <pre> diamonds = 0 while diamonds &lt; 64:     print("Удар киркой!")     count = int(input("Сколько выпало алмазов? "))     diamonds = diamonds + count     print("Добыто алмазов", diamonds)     print("Ресурс набран") </pre> <p><i>После окончания времени провести разбор решения, анализируя код.</i></p>	 <p>Сцена 17</p>
	<p><i>После окончания дикторского текста запускается таймер на 0,5 мин.</i></p> <p>Что в нем необычного?</p> <p>Строка номер 5, где изменяется значение переменной diamonds.</p> <p>Это общее правило изменения переменных. Мы как бы перезаписываем ее значение.</p> <p>А теперь запустите программу с разными значениями и найдите баг!</p>	 <p>Сцена 18</p>

	<p><b>Разбор задания 2 (продолжение).</b>          Что неправильного в этой программе?</p> <p>Можно набрать необходимое количество с одного удара – для этого достаточно ввести число 64 или больше. Если ввести отрицательное значение, то алмазы, наоборот, уйдут в минус. Что необходимо изменить?</p>	 <p>Сцена 19</p>	
	<p><i>После окончания дикторского текста запускается таймер на 2 мин.</i></p> <p><b>Задание 2 (окончание). Алгоритм «Добыча алмазов»</b></p> <pre> diamonds (алмазы) = 0 до тех пор, пока алмазы меньше 64:     напечатать «удар киркой»     спросить «сколько выпало алмазов» и ввести ответ в переменную count     если count больше 0 И count меньше 6         diamonds = diamonds + count     иначе         напечатать «так не бывает»     напечатать «добыто алмазов» и переменную diamonds напечатать «ресурс набран»         </pre>	 <p>Сцена 20</p>	
	<p><b>Разбор задания 2 (окончание).</b>          Код имеет следующий вид, и он лишен бага, который был в 1-ой версии.</p> <pre> diamonds = 0 while diamonds &lt; 64:     print("Удар киркой!")     count = int(input("Сколько выпало алмазов? "))     if count &gt; 0 and count &lt; 6:         diamonds = diamonds + count         </pre>	 <p>Сцена 21</p>	

	<pre> else:     print("Так не бывает!")     print("Добыто алмазов", diamonds)     print("Ресурс набран") </pre> <p><i>После окончания времени провести разбор решения, анализируя изменения кода</i></p>		
<p>Блок завершения занятия. <b>Рефлексия.</b> <b>Сообщение домашнего задания</b></p>	<p><i>Завершаем демонстрацией ролика и кратким обобщением материалов занятия.</i></p> <p><b>Подведем итоги:</b></p> <ul style="list-style-type: none"> <li>- узнали, что такое инструмент цикл и для чего он служит;</li> <li>- какие три вида циклов используются;</li> <li>- познакомились с организацией цикла с условием, разобрали примеры.</li> </ul> <p><i>Преподаватель дает ученикам домашнее задание к следующему занятию (Приложение 2).</i></p>	 <p>Сцена 22</p>	22

## Приложение 2

### Домашнее задание

Придумать и записать свой алгоритм, в котором будут использован цикл с условием.

### Практика

Проект «Подтверждение пароля»

Цель проекта: реализовать имитацию регистрации нового пользователя в части ввода пароля и его подтверждения.

Запросите пароль.

Запрашивайте подтверждение пароля до тех пор, пока оригинал не совпадет с подтверждением. Если подтверждение пароля не совпадает с оригиналом, выведите соответствующее сообщение.

Посчитайте, сколько попыток ввода подтверждения пароля было произведено. Выведите это значение на экран.

Проект «Билеты»

Цель проекта: реализовать имитацию онлайн продажи билетов.

Задайте начальные значения:

- исходное количество билетов в кассе,
- максимальное количество билетов, которое можно купить за один раз,
- минимальный резерв билетов, который всегда должен оставаться в кассе.

Организуите «продажу» билетов до тех пор, пока количество билетов в кассе не сравняется с минимальным резервом.

Перед каждой покупкой выводите количество оставшихся в кассе билетов. Каждое действие с билетами сопровождайте сообщениями для пользователя.

### *Приложение 3*

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

«Цикл с условием».

**В начале занятия** необходимо повторить команды условного оператора – **if/elif/else**: чем они отличаются друг от друга, последовательность применения, синтаксис.

Также нужно вспомнить что из себя представляют **вложенные условия**, для чего они используются. А также повторить материал про логические переменные (**and, or, not**).

Разберите алгоритмы с логическими переменными, которые придумали ребята и попробуйте их озвучить (записать) еще и при помощи вложенных условий. Какие будут выводы?

*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 циклов, которые работают неопределенное количество раз.

Особое внимание уделите тому, что команда **while** не требует дополнительных условных операторов. А также тому, что при формулировке условия желательно проговаривать его как «**до тех пор, пока...**»

*Задание 1* частично состоит из проекта, который уже встречался нам в уроке 8 – при сравнении роста мальчиков. Поэтому можно сделать на это ссылку. После внесения исправление в строку 6 (**else** вместо **elif**) дайте ребятам протестировать проект и сделать вывод о выводе лишней информации. После этого – верните проекту первоначальный код.

*Задание 2* выполняется в два этапа. После разбора кода на первом этапе - дайте ребятам протестировать проект. Будет хорошо, если они самостоятельно найдут проблемы кода, связанные с вводом некорректной информации, и предложат варианты решения. Если этого не произойдет – тогда ролик вам в этом поможет.

## Приложение 4

Дополнительно

**Алгоритм Евклида для нахождения наибольшего общего делителя двух натуральных чисел**

Изучите предложенные алгоритмы и напишите код программ. Вы можете сверить свой код с образцом решения.

#### Условие

Дано два натуральных числа. Задача: найти наибольший общий делитель для данных чисел.

### Алгоритм Евклида нахождения НОД

- Запросить целое число и занести его в переменную **a**
- Запросить целое число и занести его в переменную **b**
- Организовать цикл с условием работы пока **a** и **b** отличны от нуля
  - Если **a** больше **b**
    - Присвоить **a** остаток деления **a** на **b**
  - Иначе
    - Присвоить **b** остаток деления **b** на **a**
- Вывести сумму **a** и **b**



**Алгоритм разбиения записи натурального числа на отдельные цифры в позиционной системе счисления с основанием 10**

**Условие**

Дано натуральное число. Задача: записать это число отдельными цифрами.

### **Алгоритм «Разбиение числа на отдельные цифры»**

- Запросить целое число и занести его в переменную **number**
- Организовать цикл с условием пока **number** больше нуля
  - Присвоить **d** остаток от деления **number** на 10
  - Присвоить **number** результат деления без остатка **number** на 10
  - Распечатать значение **d**