



### Методические рекомендации по теме

«Проверка делимости и другие алгоритмы для решения математических задач с циклами»

#### Цель:

- применение цикла для решения простейших математических и программных прикладных задач на языке Python.

#### Задачи:

- расширение кругозора обучающихся в области информатики и программирования;
- знакомство с решением простейших математических задач с циклами на языке Python;
- решение программных прикладных задач с циклами 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 применении циклов для решения математических задач в языке Python; об основных операциях с конечными циклами на языке программирования «Python»; о прикладном использовании операций с циклами с ограниченным числом повторений в программных проектах; о возможностях и особенностях применения циклов с ограниченным числом повторений в практике работы программиста.

*Метапредметные:* обучающиеся получат возможность владения обще предметными понятием «цикл», «повторение», «конечный цикл», «задача», «решение»; 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 проверки натурального числа на простоту (дополнительно).

### **Ход проведения урока**

#### **1. Организационный момент.**

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

Приветствие учащихся, сообщение темы и целей занятия (мы уже знаем, про циклы с заданным числом повторений (итераций) - «конечными циклами»; научимся использовать конечные циклы для решения различных математических задач).

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

- Что такое цикл с ограниченным числом повторений. Приведите примеры из жизни?
- Для решения каких математических задач мы могли бы использовать «конечный цикл»?
- Почему решение данных задач с «конечным циклом» проще чем стандартное решение?

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

- конечные циклы могут быть использованы для решения математических задач;
- например, циклы могут быть использованы для математических операций с определенной последовательностью чисел;
- если требуется провести подобную операцию, то конечный цикл позволяет не прописывать конкретные числа, а просто задать последовательность и количество повторений операции.

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.

##### Конечный цикл. Обобщение.

Продолжение демонстрации ролика с дальнейшим обсуждением вопросов:

- Какие команды используются при работе с конечными циклами? Что они значат?
- Что такое «счетчик» и как он связан с «конечным циклом»?
- Что такое восходящие и нисходящие циклы?

*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. сцена 8*

#### 5. Блок заданий.

##### Проекты: «Сумма числового ряда», «Четные числа», «Числовая лестница».

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

Блок включает **3 практических задания** для учеников с последующим разбором. **Задания 1 и 2** представляют собой 2 небольших программных проекта с использованием «конечных циклов» для решения математических задач.

**После выполнения заданий 1 и 2** ученики получают два работающих программных продукта – программа для расчета суммы числового ряда и программа для нахождения четных чисел в числовом ряду.

**Задание 3** включает обсуждение проекта по вопросам:

- Что такое «числовая лестница»?
- Для чего мы используем «конечный цикл» в числовой лестнице?

*Задание 3* представляет собой небольшой программный проект с использованием «конечных циклов» для решения математических задач.

*После выполнения задания 3* ученики получают программу для создания графической лестницы с помощью вывода на экран последовательности чисел.

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

*\*см. сцены 9 – 19*

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

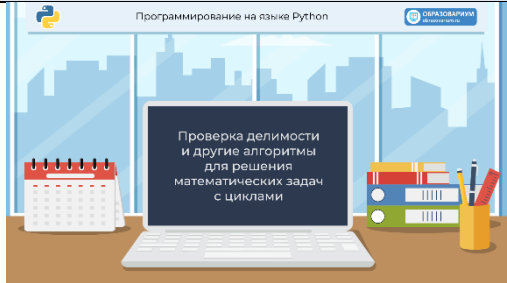
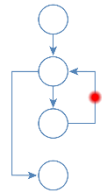
Завершаем демонстрацией ролика и кратким обобщением материалов занятия. Преподаватель возвращается к зафиксированному в ходе дискуссии в начале урока предположениям учеников и обсуждает насколько их предположения были правильными, делаются выводы.

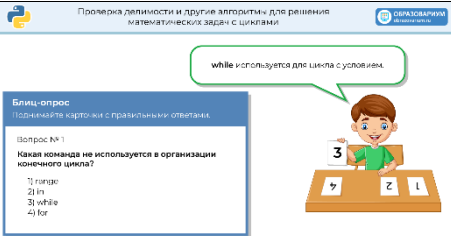
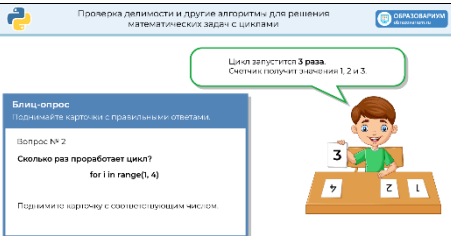
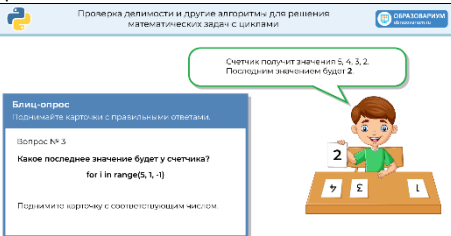
Преподаватель дает ученикам домашнее задание к следующему занятию (*Приложение 2*). *\*см. сцена 20*

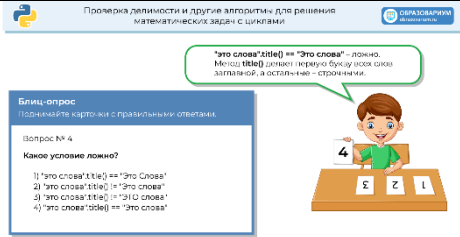
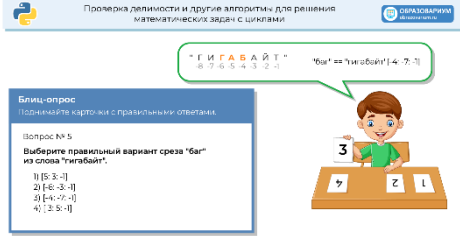
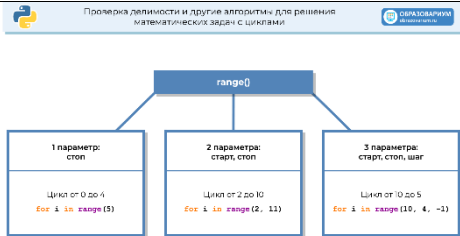
## Сценарный план видеоролика

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.

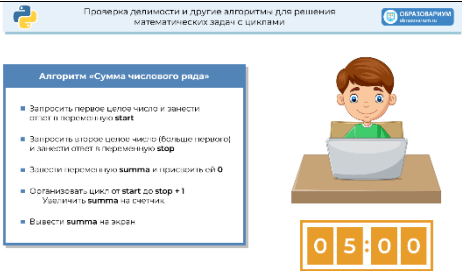
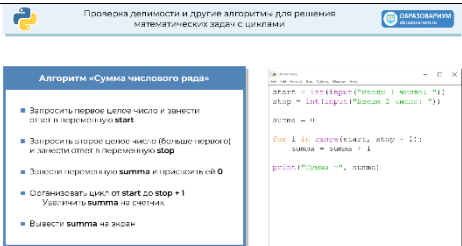
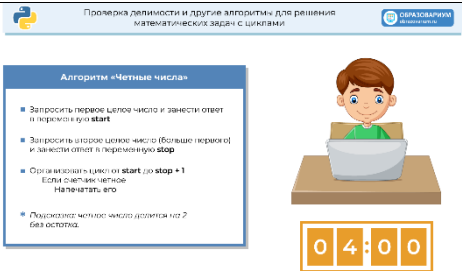
Таблица. Содержание видеоролика

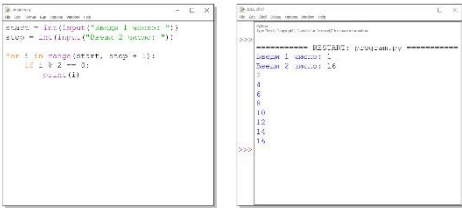
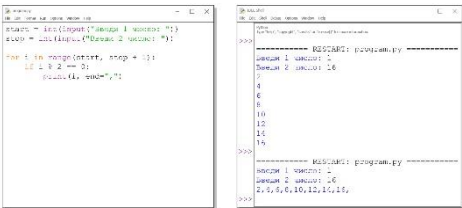
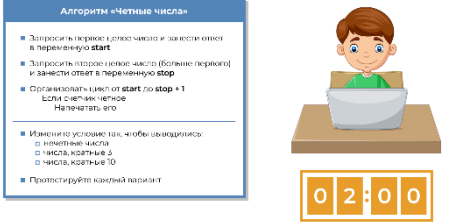
Название блока	Содержание блока и комментарии	Фрагменты из видеоролика	№ сцен
Вводный блок. Мы узнаем	<p>Обозначаем ученикам тему и цели урока.</p> <p>Проверка делимости и другие алгоритмы для решения математических задач с циклами.</p>	 <p>Сцена 1</p>	1 2
	<p>На прошлом занятии мы познакомились с конечным циклом, который имеет определенное заданное количество повторений.</p> <p>Теперь узнаем о практическом применении этого инструмента и какие задачи решаются с его помощью.</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> Какая команда не используется в организации конечного цикла?</p> <p>1) range 2) in 3) while 4) for</p> <p><i>Ответ 3. while используется для цикла с условием.</i></p>	 <p>Сцена 3</p>	<p>3 4 5 6 7</p>
	<p><b>Второй вопрос.</b> Поднимите карточку, сколько раз проработает цикл.</p> <p><b>for i in range (1, 4)</b></p> <p><i>Ответ 3. Цикл запустится 3 раза. Счетчик получит значения 1, 2 и 3.</i></p>	 <p>Сцена 4</p>	
	<p><b>Третий вопрос.</b> Поднимите карточку, какое последнее значение будет у счетчика?</p> <p><b>for i in range (5, 1, -1)</b></p> <p><i>Ответ 2. Счетчик получит значения 5, 4, 3, 2. Последним значением будет 2.</i></p>	 <p>Сцена 5</p>	

	<p><b>Четвертый вопрос.</b> Какое условие ложно?</p> <ol style="list-style-type: none"> <li>1) "это слова".title() == "Это Слова"</li> <li>2) "это слова".title() != "Это слова"</li> <li>3) "это слова".title() != "ЭТО слова"</li> <li>4) "это слова".title() == "Это слова"</li> </ol> <p><i>Ответ 4. "это слова".title() == "Это слова" – ложно. Метод <b>title()</b> делает первую букву всех слов заглавной, а остальные – строчными.</i></p>	 <p>Сцена 6</p>	
	<p><b>Пятый вопрос.</b> Выберите правильный вариант среза "баг" из слова "гигабайт"</p> <ol style="list-style-type: none"> <li>1) [5: 3: -1]</li> <li>2) [-6: -3: -1]</li> <li>3) [-4: -7: -1]</li> <li>4) [3: 5: -1]</li> </ol> <p><i>Ответ 3. "баг" == "гигабайт"[-4: -7: -1]</i></p>	 <p>Сцена 7</p>	
<p>Теоретический блок.</p> <p><b>Конечный цикл.</b></p>	<p><i>При необходимости преподаватель может поставить ролик на паузу и дать дополнительные пояснения по материалу</i></p> <p>Давайте вспомним, что для организации конечного цикла нужны команды <b>for</b>, <b>in</b>, <b>range</b>. Нужна переменная-счетчик, в роли которой чаще всего используют букву <b>i</b></p> <p>Для восходящего цикла, внутри функции <b>range</b>, может быть от одного до трех параметров: <b>старт</b>, <b>стоп</b>, <b>шаг</b>.</p> <p>У нисходящего цикла параметров всегда три, иначе он работать не будут</p>	 <p>Сцена 8</p>	8



<p>Блок заданий.</p> <p><b>Практические задания:</b></p> <p>Задание 1</p> <p>Задание 2</p> <p>Задание 3</p>	<p><i>После окончания дикторского текста запускается таймер на 5 мин.</i></p> <p><b>Задание 1. Алгоритм «Сумма числового ряда»</b></p> <ul style="list-style-type: none"> <li>Запросить первое целое число и занести ответ в переменную <b>start</b></li> <li>Запросить второе целое число (больше первого) и занести ответ в переменную <b>stop</b></li> <li>Завести переменную <b>summa</b> и присвоить ей 0</li> <li>Организовать цикл от <b>start</b> до <b>stop</b> Увеличить <b>summa</b> на счетчик</li> <li>Вывести <b>summa</b> на экран</li> </ul>	 <p>Сцена 9</p>	<p>9</p> <p>10</p> <p>11</p> <p>12</p> <p>13</p> <p>14</p> <p>15</p> <p>16</p> <p>17</p> <p>18</p> <p>19</p>
	<p><b>Разбор задания 1.</b> Наш код может выглядеть так.</p> <pre>start = int(input("Введи 1 число: ")) stop = int(input("Введи 2 число: ")) summa = 0 for i in range(start, stop + 1):     summa = summa + i print("Сумма =", summa)</pre> <p><i>После окончания времени провести разбор решения, анализируя код.</i></p>	 <p>Сцена 10</p>	
	<p><i>После окончания дикторского текста запускается таймер на 4 мин.</i></p> <p><b>Задание 2. Алгоритм «Четные числа»</b></p> <ul style="list-style-type: none"> <li>Запросить первое целое число и занести ответ в переменную <b>start</b></li> <li>Запросить второе целое число (больше первого) и занести ответ в переменную <b>stop</b></li> <li>Организовать цикл от <b>start</b> до <b>stop + 1</b> Если счетчик четное Напечатать его</li> </ul> <p><i>Подсказка: четное число при деление на 2 дает остаток 0.</i></p>	 <p>Сцена 11</p>	

<p><b>Разбор задания 2.</b> Наш код может выглядеть так.</p> <pre>start = int(input("Введи 1 число: ")) stop = int(input("Введи 2 число: "))  for i in range(start, stop + 1):     if i % 2 == 0:         print(i)</pre> <p><i>После окончания времени провести разбор решения, анализируя код.</i></p>	<div></div> <p>Сцена 12</p>
<p><b>Разбор задания 2 (дополнение)</b></p> <p>Но у нашего проекта есть один недостаток – числа выводятся в столбик и занимают много места.</p> <p>Давайте запретим курсору «перепрыгивать». Для этого немного изменим код последней строки и добавим туда параметр <b>end</b> со значением в виде запятой. Теперь числа будут выводиться на одной строке через запятую</p>	<div></div> <p>Сцена 13</p>
<p><i>После окончания дикторского текста запускается таймер на 2 мин.</i></p> <p><b>Задание 2. Алгоритм «Четные числа» (продолжение)</b></p> <ul style="list-style-type: none"><li>– Измените условие, чтобы выводились<ul style="list-style-type: none"><li>■ нечетные числа</li><li>■ числа, кратные 3</li><li>■ числа, кратные 10</li></ul></li></ul> <p>Протестируйте каждый вариант</p>	<div></div> <p>Сцена 14</p>

**Проект «Числовая лестница».**  
Необходимо вывести на экран вот такую лестницу из чисел.  
При всей видимой сложности – задача достаточно простая, если ее внимательно проанализировать.  
Имеется цикл от 1 до 9 включительно.  
Каждое новое число — это предыдущее, умноженное на десять плюс значение счетчика.

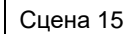
При всей видимой сложности – задача достаточно простая, если ее внимательно проанализировать.

Имеется цикл от 1 до 9 включительно.

Каждое новое число — это предыдущее, умноженное на десять плюс значение счетчика.

Имеется цикл от 1 до 9 включительно.  
Каждое новое число — это предыдущее, умноженное на десять плюс значение счетчика.

Каждое новое число — это предыдущее, умноженное на десять плюс значение счетчика.



### Задание 3. Алгоритм «Числовая лестница».

- Задание 3. Алгоритм «Числовая лестница».**
- Создать переменную **num** и присвоить ей 0
  - Организовать цикл от 1 до 10
    - num = num \* 10 + счетчик
    - вывести num на экран

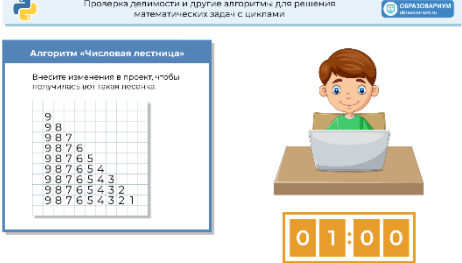
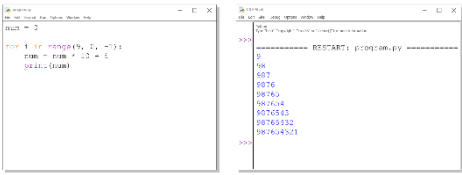
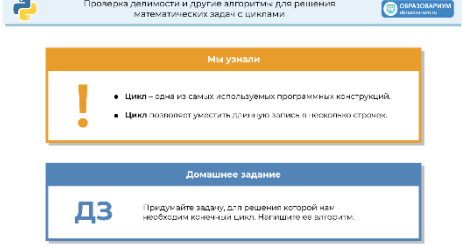


```
num = 0
for i in range(1, 10):
    num = num * 10 + i
print(num)
```

```
num = 0
for i in range(1, 10):
    num = num * 10 + i
print(num)
```

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



	<p><i>После окончания дикторского текста запускается таймер на 1 мин.</i></p> <p><b>Задание 3. Алгоритм «Числовая лестница» (дополнение).</b></p> <p>Внесите изменение в проект, чтобы получилась вот такая лесенка.</p>	 <p>Сцена 18</p>	
	<p><b>Разбор задания 3 (дополнение).</b></p> <p>Надо просто поменять параметры цикла и сделать его нисходящим от девяти до одного включительно</p> <p><i>После окончания времени провести разбор решения, анализируя код.</i></p>	 <p>Сцена 19</p>	
<p>Блок завершения занятия.</p> <p><b>Рефлексия. 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 цикл – одна из самых используемых программных конструкций;</li> <li>– цикл позволяет уместить длинную запись в несколько строчек.</li> </ul> <p><i>Преподаватель дает ученикам домашнее задание к следующему занятию (Приложение 2).</i></p>	 <p>Сцена 20</p>	20

### **Домашнее задание**

Придумайте и создайте свою программу, в которой для решения математической задачи используются «конечные циклы».

### **Практика**

Проект «Наименьший делитель».

Запросите целое число. Найдите и выведите на экран наименьший делитель этого числа, отличный от единицы.

Проект «Сумма введенных чисел»

Цель проекта: посчитать сумму чисел, введенных пользователем.

Придумайте и выведите на экран условие окончания ввода чисел. Запрашивайте у пользователя числа до тех пор, пока условие окончания ввода не будет выполнено.

Выведите на экран сумму введенных чисел и их количество.

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

«Проверка делимости и другие алгоритмы для решения математических задач с циклами».

**В начале занятия** можно рассмотреть домашнее задание – параметры для лифта. Если бы мы делали это через цикл, то вверх указали бы от 1 до 17 (последнее число не попадает в диапазон), а вниз с 16 до 1 (то же самое правило).

Повторите как создается конечный цикл, какими командами, какие есть варианты цикла (восходящий, нисходящий) и как это влияет на параметры.

Все повторения лучше сопровождать примерами на доске

**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.

**Перед написанием кода** для каждой задачи рекомендуется решить ее без помощи компьютера.

Так в первом задании можно задать два числа и попросить ребят в уме или с калькулятором сделать вычисления, чтобы понимать логику кода.

**Перед началом второго задания** надо напомнить – как узнать в Python четное или нечетное число (через деление без целой части)

**Перед началом третьего задания** можно записать числовую лесенку на доске и предложить ребятам самостоятельно найти закономерность в ее создании. Если не получится – тогда объяснение в ролике.

Объяснения нового материала в уроке практически нет, поэтому если останется время – можно самостоятельно придумать еще какую ни будь задачу, связанную с конечными циклами.

## Приложение 4

### Алгоритм проверки натурального числа на простоту.

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

#### Условие

Дано натуральное число. Задача: определить является ли заданное число простым.

## Алгоритм «Простое число»

- Запросить целое число и занести его в переменную **a**
- Создать переменную **k** со значением, равным нулю
- Организовать цикл в диапазоне от **2** до **a // 2**
  - Если **a** делится на значение счетчика цикла без остатка
    - Увеличить **k** на **1**
- Если **k** равно нулю
  - Вывести «**Число простое**»
- Иначе
  - Вывести «**Число не является простым**»