



Методические рекомендации по теме

«Циклы с неограниченным количеством повторений для решения математических задач»

Цель:

- применение «бесконечного 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 «бесконечными циклами» на языке Python;
- решение программных прикладных задач с «бесконечными 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 «бесконечными циклами» на языке программирования «Python»; о прикладном использовании операций с бесконечными циклами в программных проектах; о возможностях и особенностях применения бесконечных циклов в практике работы программиста.

Метапредметные: обучающиеся получат возможность владения обще предметными понятием «цикл», «повторение», «бесконечный 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

Ход проведения урока

1. Организационный момент.

Мотивация на учебную деятельность.

Приветствие учащихся, сообщение темы и целей занятия (мы уже знаем, про циклы с неограниченным числом повторений (итераций) - «бесконечными циклами», теперь познакомимся с «бесконечным циклом» и научимся использовать бесконечные циклы для решения математических задач).

Проблемная дискуссия по вопросам:

- Какие основные особенности цикла с неограниченным числом повторений?
- Для решения каких математических задач мы могли бы использовать «бесконечный цикл»?
- Каким образом мы можем задать остановку «бесконечного цикла»?

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

- циклы с неограниченным числом повторений могут быть использованы для решения математических задач;
- поскольку «бесконечные циклы» связаны с условиями, то это позволяет использовать их для решения сложных задач.

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.

Задача «Улитка на столбе».

Продолжение демонстрации ролика с дальнейшим обсуждением вопросов:

- Что требуется для решения задачи «Улитка на столбе»?
- Что в этой задаче является циклом?

- Почему следует использовать цикл с неограниченным числом повторений?

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 – 9*

5. Блок заданий.

Проекты: «Улитка на столбе», «Подготовка к марафону».

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

«Улитка на столбе»: включает практическое *задание 1* по использованию «бесконечных циклов» для решения математических задач.

После выполнения задания ученики получают работающий программный продукт – программа для решения задач типа «улитка на столбе» с различными вводными.

«Подготовка к марафону»: включает *практическое задание 2* для учеников с последующим разбором. Задание представляет собой небольшой программный проект с использованием «бесконечных циклов» для решения математических задач.

После выполнения задания ученики получают программу для расчета нагрузки спортсмена для подготовки к марафону при известной норме увеличения нагрузки.

Задание включает теоретическую вкладку с обсуждением вопросов:

- В чем суть проекта «подготовка к марафону»?

- Для чего мы используем «бесконечный цикл» в этом проекте?

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

**см. сцены 10 – 18*

6. Рефлексия. Сообщение домашнего задания.

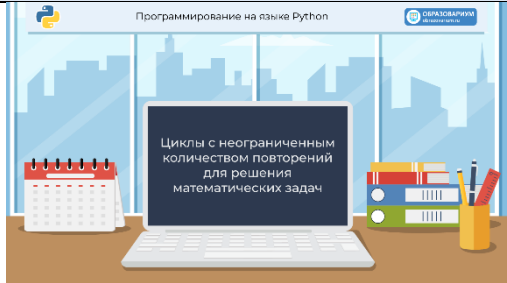
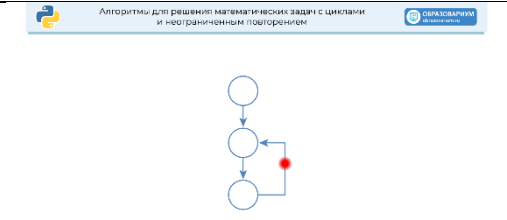
Завершаем демонстрацией ролика и кратким обобщением материалов занятия. Преподаватель возвращается к зафиксированным в ходе дискуссии в начале урока предположениям учеников и обсуждает насколько их предположения были правильными, делаются выводы.

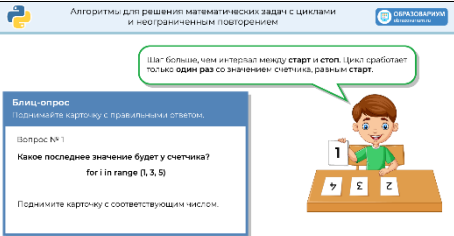
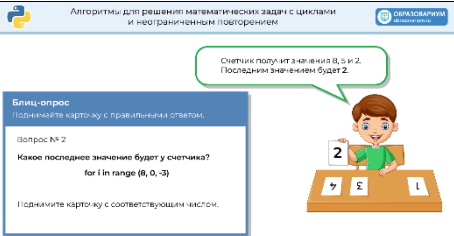
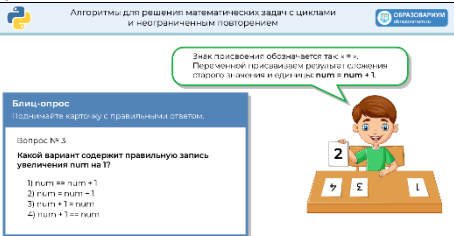
Преподаватель дает ученикам домашнее задание к следующему занятию (*Приложение 2*). **см. сцена 19*

Сценарный план видеоролика

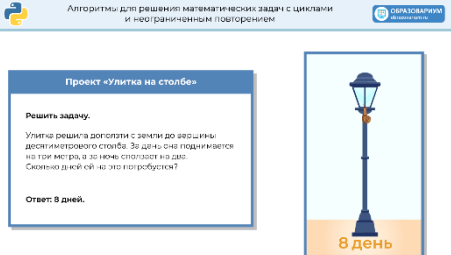
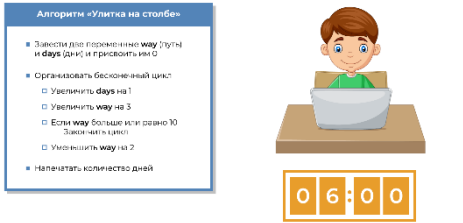
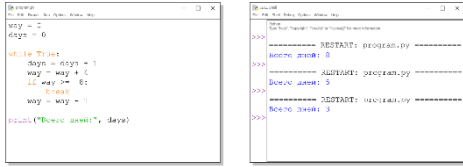
В таблице «Содержание видеоролика» представлен краткий текст из видеоролика, примеры заданий и задач, которые будут демонстрироваться на экране. Учитель при подготовке к уроку может ознакомиться с содержанием видеоролика в текстовом формате, при необходимости распечатать фрагменты текста или примеры 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. Циклы с неограниченным количеством повторений для решения математических задач	 Сцена 1	1 2
	Мы продолжаем знакомиться с циклическими конструкциями, при помощи которых можно упростить программы, избежав повторения в написании одного и того же кода.	 Сцена 2	

<p>Блок повторения.</p> <p>Блиц-опрос</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>Первый вопрос. Какое последнее значение будет у счетчика? Поднимите карточку с номером ответа.</p> <p>for i in range (1,3,5)</p> <p><i>Ответ 1. Шаг больше, чем интервал между старт и стоп. Цикл сработает только один раз со значением счетчика, равным старт..</i></p>	<div data-bbox="1509 228 1960 464">  <p>Алгоритмы для решения математических задач с циклами и неограниченным повторением</p> <p>Блиц-опрос: Поднимайте карточку с правильным ответом.</p> <p>Вопрос №1: Какое последнее значение будет у счетчика? for i in range (1, 3, 5)</p> <p>Поднимите карточку с соответствующим числом.</p> <p>1) 1 2) 3 3) 5</p> <p>Шаг больше, чем интервал между старт и стоп. Цикл сработает только один раз со значением счетчика, равным старт.</p> </div> <p>Сцена 3</p>	<p>3</p> <p>4</p> <p>5</p> <p>6</p> <p>7</p>
	<p>Второй вопрос. Какое последнее значение будет у счетчика? Поднимите карточку с номером ответа.</p> <p>for i in range (8 ,0, -3)</p> <p><i>Ответ 2. Счетчик получит значения 8, 5 и 2. Последним значением будет 2.</i></p>	<div data-bbox="1509 608 1960 844">  <p>Алгоритмы для решения математических задач с циклами и неограниченным повторением</p> <p>Блиц-опрос: Поднимайте карточку с правильным ответом.</p> <p>Вопрос №2: Какое последнее значение будет у счетчика? for i in range (8, 0, -3)</p> <p>Поднимите карточку с соответствующим числом.</p> <p>1) 8 2) 5 3) 2</p> <p>Счетчик получит значения 8, 5 и 2. Последним значением будет 2.</p> </div> <p>Сцена 4</p>	
	<p>Третий вопрос. Какой вариант содержит правильную запись увеличения num на 1?</p> <p>1) num == num + 1</p> <p>2) num = num + 1</p> <p>3) num + 1 = num</p> <p>4) num + 1 == num</p> <p><i>Ответ 2. Знак присвоения обозначается так « = ». Переменной присваиваем результат сложения старого значения и единицы: num = num + 1.</i></p>	<div data-bbox="1509 919 1960 1155">  <p>Алгоритмы для решения математических задач с циклами и неограниченным повторением</p> <p>Блиц-опрос: Поднимайте карточку с правильным ответом.</p> <p>Вопрос №3: Какой вариант содержит правильную запись увеличения num на 1?</p> <p>1) num == num + 1 2) num = num + 1 3) num + 1 = num 4) num + 1 == num</p> <p>Знак присвоения обозначается так « = ». Переменной присваивается результат сложения старого значения и единицы: num = num + 1.</p> </div> <p>Сцена 5</p>	

	<p>Четвертый вопрос. Какой цикл не сработает ни разу, если <code>count = 1</code>?</p> <ol style="list-style-type: none"> 1) <code>while "1" == count:</code> 2) <code>while "1" != count:</code> 3) <code>while 1 == count:</code> 4) <code>while count == count:</code> <p><i>Ответ 1. Текст не может быть равен числу. "1" == count – ложно и цикл не работает.</i></p>	<p>Сцена 6</p>	
	<p>Пятый вопрос. Число 6.5 округлили до 6. Какая функция могла это сделать?</p> <ol style="list-style-type: none"> 1) <code>int</code> 2) <code>round</code> 3) <code>floor</code> 4) <code>все</code> <p><i>Ответ 4. Все три функции могут это делать.</i></p>	<p>Сцена 7</p>	
<p>Теоретический блок.</p> <p>Задача «Улитка на столбе»</p>	<p><i>При необходимости преподаватель может поставить ролик на паузу и дать дополнительные пояснения по материалу</i></p> <p>Рассмотрим проект «Улитка на столбе» и решим его без помощи компьютера.</p> <p>Улитка решила доползти с земли до вершины десятиметрового столба. За день она поднимается на три метра, а за ночь сползает на два. Сколько дней ей на это потребуется?</p>	<p>Сцена 8</p>	<p>8 9</p>

	<p>Правильный ответ – восемь дней. Посчитайте сами: на восьмой день она начнет свое восхождение с отметки в семь метров. И к вечеру достигнет вершины.</p> <p>То, что за ночь она опять сползет вниз уже не имеет значения. Цель достигнута.</p> <p>По факту – улитка совершает циклические движения: вверх и вниз. Поэтому для создания алгоритма потребуется цикл.</p>	 <p>Сцена 9</p>	
<p>Блок заданий.</p> <p>Практические задания:</p> <p>Задание 1</p>	<p><i>После окончания дикторского текста запускается таймер на 6 мин.</i></p> <p>Задание 1. Алгоритм «Улитка на столбе»</p> <ul style="list-style-type: none"> • Завести две переменные way и days и присвоить им 0 • Организовать бесконечный цикл <ul style="list-style-type: none"> ◦ увеличить days на 1 ◦ увеличить way на 3 ◦ если way больше или равно 10 закончить цикл ◦ уменьшить way на 2 • Напечатать количество дней 	 <p>Сцена 10</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>Разбор задания 1. Наш код может выглядеть так.</p> <pre> way = 0 days = 0 while True: days = days + 1 way = way + 3 if way >= 10: break way = way - 2 print("Всего дней", days) </pre> <p><i>Обратите внимание на отсутствие команды else в условии (оно не нужно, ведь если цикл не прервется, то строка с уменьшением пути и так</i></p>	 <p>Сцена 11</p>	

	<p>выполнится). Можно попробовать другие параметры высоты столба и скорости движения улитки вверх и вниз.</p>		
	<p>Задание 2. Проект «Подготовка к марафону»</p> <p>Этот проект уже более сложный. Лыжник, готовясь к 50-ти километровому марафону в первый день пробежал 10 км. Каждый следующий день он решил пробегать на 10 % больше, чтобы выйти на уровень 50 км в день. Успеет ли лыжник подготовиться к соревнованиям если у него в запасе 20 дней?</p>	<div data-bbox="1507 339 1962 600"><div>Алгоритмы для решения математических задач с циклами и неограниченным повторением</div><div>Проект «Подготовка к марафону»</div><div>Решите задачу.</div><div>Лыжник, готовясь к марафону, в первый день пробежал 10 км. Каждый следующий день он решил пробегать на 10 % больше, чтобы выйти на уровень 50 км в день. Успеет ли лыжник подготовиться к соревнованиям, если у него в запасе 20 дней?</div><div><div>1 день — 10 км</div><div>2 день — 11 км</div><div>3 день — 12 км</div><div>20 дней — ? 50 км</div></div><div></div></div> <p>Сцена 12</p>	
	<p><i>После окончания дикторского текста запускается таймер на 7 мин.</i></p> <p>Задание 2. Алгоритм «Подготовка к марафону»</p> <ul style="list-style-type: none">■ Завести переменную days и присвоить ей 0■ Завести переменную dist и присвоить ей 10■ Организовать бесконечный цикл<ul style="list-style-type: none">◦ увеличить days на 1◦ если days равно 20<ul style="list-style-type: none">напечатать «не успеет»прервать цикл◦ увеличить dist на 10 процентов◦ если dist больше 50<ul style="list-style-type: none">напечатать «успеет»прервать цикл	<div data-bbox="1507 770 1962 1042"><div>Алгоритмы для решения математических задач с циклами и неограниченным повторением</div><div>Алгоритм «Подготовка к марафону»</div><div><ul style="list-style-type: none">■ Завести переменную days (дни) и присвоить ей 0■ Завести переменную dist (дистанция) и присвоить ей 10■ Организовать бесконечный цикл<ul style="list-style-type: none">□ Увеличить days на 1□ Если days равно 20<ul style="list-style-type: none">• Напечатать «не успеет»• Прервать цикл□ Увеличить dist на 10 процентов□ Если dist больше 50<ul style="list-style-type: none">• Напечатать «успеет»• Прервать цикл</div><div></div><div><div>07:00</div></div></div> <p>Сцена 13</p>	

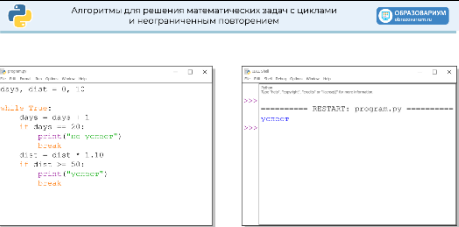
Разбор задания 2. Код проекта может быть таким:

```
days, dist = 0, 10
while True:
    days = days + 1
    if days == 20:
        print("не успел")
        break
    dist = dist * 1.10
    if dist >= 50:
        print("успел")
        break
```

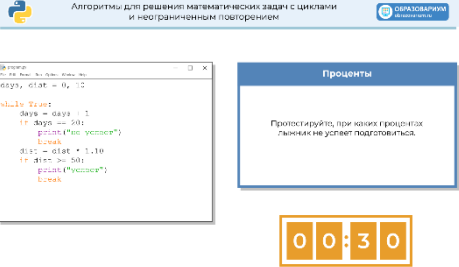
Запустим проект и получим ответ: успеет! Напомните, чтобы увеличить число на 10 процентов надо умножить его на 1.10.

После окончания дикторского текста запускается таймер на 30 сек.
Задание 2. Алгоритм «Подготовка к марафону» (продолжение)

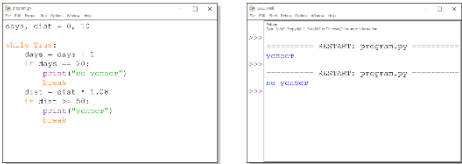
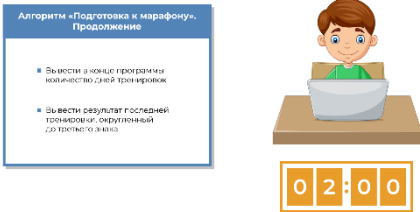
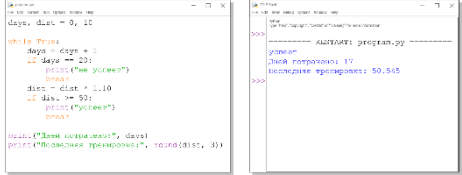
– Протестируйте, при каких процентах он не успеет?

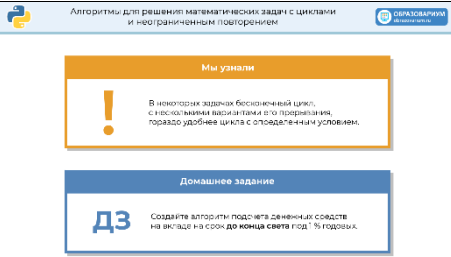


Сцена 14



Сцена 15

<p>Разбор задания 2 (продолжение).</p> <p>Ответ – если будет 8 и меньше процентов</p> <p><i>Здесь присутствуют два условия для окончания цикла, и каждое сопровождается выводом пояснительной информации. Создать подобный проект с использованием условного цикла было бы проблематично.</i></p>	 <p>Сцена 16</p>
<p><i>После окончания дикторского текста запускается таймер на 2 мин.</i></p> <p>Задание 2. Алгоритм «Подготовка к марафону» (окончание).</p> <ul style="list-style-type: none"> Вывести в конце программы количество дней тренировок. Вывести результат последней тренировки, округленный до третьего знака 	 <p>Сцена 17</p>
<p>Разбор задания 2 (окончание).</p> <p>Код программы дополняется строками:</p> <pre>print("Дней потрачено:", days) print("Последняя тренировка:", round(dist, 3))</pre> <p><i>После этого, программа будет не только сообщать результат (успеет/не успеет), а также выводит количество потраченных дней и длину пути для последней тренировки.</i></p>	 <p>Сцена 18</p>

<p>Блок завершения занятия. Рефлексия. Сообщение домашнего задания</p>	<p><i>Завершаем демонстрацией ролика и кратким обобщением материалов занятия.</i></p> <p>Подведем итоги.</p> <p>– Пришли к выводу:</p> <ul style="list-style-type: none"> • в некоторых задачах бесконечный цикл, с несколькими вариантами его прерывания, гораздо удобнее цикла с определенным <p><i>Преподаватель дает ученикам домашнее задание к следующему занятию (Приложение 2).</i></p>	 <p>Сцена 19</p>	<p>19</p>
--	---	---	-----------

Приложение 2

Домашнее задание

В качестве домашнего задания создайте алгоритм подсчета денежных средств на вкладе на срок до «конца света» под 1 % годовых.

Практика

Проект «Алгоритм 1»

Найдите ответ задачи из ОГЭ путем перебора чисел.

У исполнителя **Алго** две команды, которым присвоены номера:

1. прибавь 1
2. умножь на b

(b – неизвестное натуральное число; $b \geq 2$).

Выполняя первую команду, **Алго** увеличивает число на экране на 1; выполняя вторую команду, умножает это число на b.

Программа для исполнителя Алго – это последовательность номеров команд. Известно, что программа **11211** переводит число 8 в число 62.

Определите значение b .

Проект «Алгоритм 2»

Найдите ответ задачи из ОГЭ путем перебора чисел.

У исполнителя Алго две команды, которым присвоены номера:

1. прибавь 1
2. умножь на b

(b – неизвестное натуральное число; $b \geq 2$).

Выполняя первую команду, Алго увеличивает число на экране на 1; выполняя вторую команду, умножает это число на b .

Программа для исполнителя Алго – это последовательность номеров.

Известно, что программа 11121 переводит число 5 в число 73.

Определите значение b .

Приложение 3

Краткие организационно-методические рекомендации по организации работы на занятии

«Циклы с неограниченным количеством повторений для решения математических задач».

В начале занятия можно рассмотреть домашнее задание – примеры бесконечных циклов в окружающем нас мире и условий для их прерывания. Также проговорить про сходства и отличия «бесконечного цикла» и цикла с условием (операторы, синтаксис).

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.

Оба проекта представляют собой классические олимпиадные задачи, в которых приведенные данные достаточно условны. Вы можете после создания проектов прописать новые вводные параметры и посмотреть, как программы будут их вычислять.

Если останется время:

1. В проекте номер один, можно сделать вывод на экран (в конце цикла) параметра «путь», чтобы видеть, где находится улитка в конце дня.
2. В проекте номер два, можно создать переменную «общий путь», чтобы подсчитать общий километраж спортсмена за время подготовки. Также можно сделать чтобы программа выводила километраж за каждый день.