



### Методические рекомендации по теме

«Рисование символами: вывод строк с помощью цикла с неограниченным количеством повторений.  
Создание эффекта анимации»

#### Цель:

- дать представление об использовании циклов с неограниченным количеством повторений в динамической псевдографике.

#### Задачи:

- практика применения циклов с неограниченным количеством повторений в Python;
- практика применения вложенных циклов;
- анализ программного кода с целью определения, что выведет программа при конкретных исходных данных;
- исправление ошибок и дописывание программного кода;
- написание программного кода.

#### Планируемые результаты

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

*Предметные:* обучающиеся получают представления об использовании циклов с неограниченным количеством повторений в псевдографике, о применении вложенных циклов в Python.

*Метапредметные:* обучающиеся получат возможность владения общепредметными понятиями «цикл», «цикл с неограниченным числом повторений», «вложенный цикл»; информационно-логическими умениями; умениями самостоятельно планировать пути достижения целей; владения умениями принятия решений и осуществления осознанного выбора; повышения уровня ИКТ – компетентности и расширение кругозора в области информатики и программирования; знакомство с профессиональной деятельностью программиста в рамках ранней профориентации; развитие интеллектуальных способностей, а также логического и критического мышления.

### **Материалы к занятию**

Приложение 1: Сценарный план видеоролика

Приложение 2: Домашнее 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 узнаем, об интересном виде искусства «псевдографике» – рисовании и создании анимации с помощью символов, научимся автоматизировать процесс создания «шедевров» в технике «псевдографики»).

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

- Может ли псевдографика быть динамической?
- Каков по вашему мнению механизм работы динамической псевдографики?
- Для каких целей можно использовать динамическую псевдографику?

- Возможно ли использование циклов для создания динамической псевдографики?
- В каком параметре (**время**) различие между статичной и динамической псевдографикой?

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

- псевдографические изображения могут быть динамическими;
- динамика работает за счет изменения или отображения новых символов через определенные промежутки времени.

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.**

### **Псевдографика.**

Новый материал излагается в сопровождении видеоролика, рекомендуется разместить на доске или флип-чарте изображения объектов псевдографики, в которых используется большое количество повторяющихся символов, при необходимости обращаться к ним для иллюстрации использования циклов.

Обсуждением вопросов по просмотренным материалам:

- Как должен работать проект «Горнолыжный слалом»?
- Как мы планируем в нем использовать циклы 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:** «Горнолыжный слалом», «Анимация лица».

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

**«Горнолыжный слалом»:** включает *практическое задание 1* «Горнолыжный слалом» по использованию циклов для создания «динамической лыжни», которая постоянно меняет свою траекторию («лыжня» будет постоянно петлять из стороны в сторону).

**После выполнения задания** ученики получают работающий программный продукт – программа выводит псевдографику (динамическое изображение «лыжни» для горнолыжного слалома) с помощью организации трех циклов: внешний – бесконечный, и два внутренних – конечных, которые и будут отвечать за повороты. Есть возможность регулировать, чаще или реже менять направления.

«Анимация лица»: включает *практические задания 2 и 3* для учеников с последующим разбором. Задание представляет собой небольшой программный проект для создания трех картинок псевдографики лица: профиль, вид справа и вид слева, организованный бесконечный цикл будет анимировать эти картинки.

*После выполнения задания* ученики получают работающую программу для вывода анимации лица, изображенного в стиле псевдографики. Есть возможность регулировать время задержки после каждой картинки на указанное время.

Блок включает теоретическую вставку «модуль управления временем **time** и команда **sleep**».

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

*\*см. сцены 10 – 23*

## **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*).

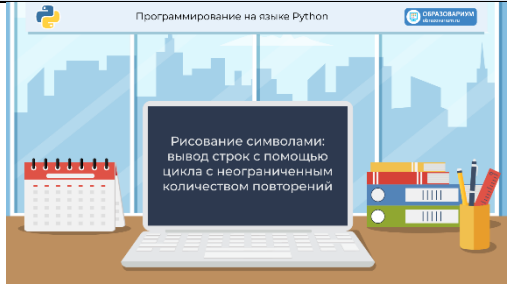
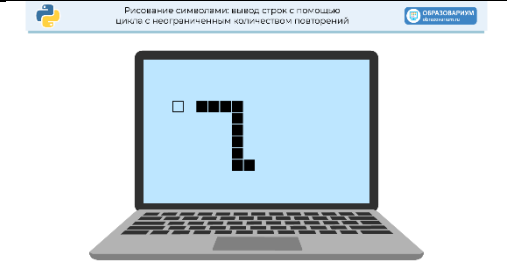
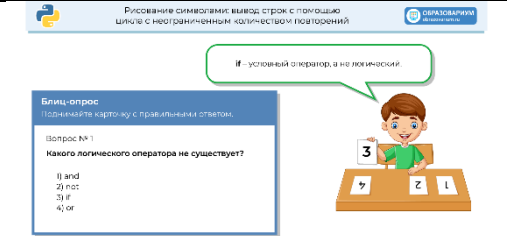
*\*см. сцена 24*

## **Приложение 1**

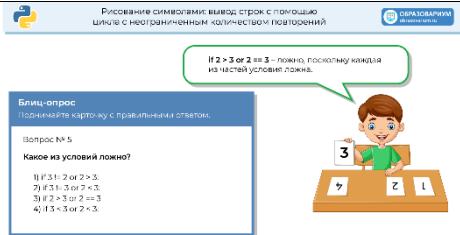
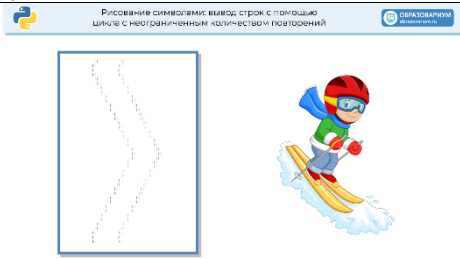
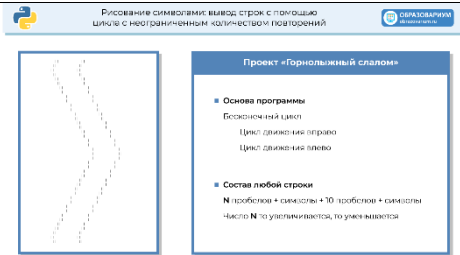
### **Сценарный план видеоролика**

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	<p>Обозначаем ученикам тему и цели урока.</p> <p>Рисование символами вывод строк с помощью цикла с неограниченным количеством повторений.</p> <p>Создание эффекта анимации</p>	 <p>Сцена 1</p>	1 2
	<p>В это теме узнаем, как можно из текстовых символов создавать рисунки, в которые подразумевают постоянное изменение сюжета.</p> <p>Этот тип псевдографики был очень популярен на заре появления первых компьютерных игр.</p> <p>Например, змейка из символов или бесконечно движущаяся дорога.</p> <p>Рассмотрим некие аналоги тех легендарных проектов.</p>	 <p>Сцена 2</p>	
Блок повторения. Блиц-опрос	<p>Повторение материала предыдущего урока; на столе имеются пронумерованные карточки; после каждого вопроса выбираем ту, номер которой, совпадает с правильным ответом.</p> <p><b>Первый вопрос.</b> Какой логический оператор отсутствует?</p> <ol style="list-style-type: none"> <li>1) and</li> <li>2) not</li> <li>3) if</li> <li>4) or</li> </ol> <p>Ответ 3. <i>if</i> – условный оператор, а не логический.</p>	 <p>Сцена 3</p>	3 4 5 6 7

	<p><b>Второй вопрос.</b> Сколько ошибок в данном примере? Поднимите карточку с номером ответа.</p> <pre>if 2 = 2     блок команд elif 2 != 2:     блок команд</pre> <p><i>Ответ 3. Правильно писать так:</i></p> <pre>if 2 == 2:     блок команд elif 2 != 2:     блок команд.</pre>	<div data-bbox="1509 231 1960 466"> <p>Рисование символических выводов строк с помощью циклов с неограниченным количеством повторений</p> <p>Блиц-опрос Поднимайте карточку с правильными ответами.</p> <p>Вопрос № 2 Сколько ошибок в данном примере?</p> <pre>if 2 = 2     блок команд elif 2 != 2:     блок команд</pre> <p>Поднимите карточку с соответствующим числом</p> <p>Правильно писать так: if 2 == 2:     блок команд elif 2 != 2:     блок команд</p> </div> <p>Сцена 4</p>
	<p><b>Третий вопрос.</b> Сколько раз проработает цикл? Поднимите карточку с номером ответа.</p> <pre>for i in range(-1, 2)</pre> <p><i>Ответ 3. Счетчик примет значения: -1, 0, 1. Цикл проработает 3 раза</i></p>	<div data-bbox="1509 694 1960 928"> <p>Рисование символических выводов строк с помощью циклов с неограниченным количеством повторений</p> <p>Блиц-опрос Поднимайте карточку с правильными ответами.</p> <p>Вопрос № 3 Сколько раз проработает цикл?</p> <pre>for i in range(-1, 2)</pre> <p>Поднимите карточку с соответствующим числом</p> <p>Счетчик примет значения: -1, 0, 1. Цикл проработает 3 раза.</p> </div> <p>Сцена 5</p>
	<p><b>Четвертый вопрос.</b> Выберите правильный вариант среза "чай" из слова "случайность".</p> <ol style="list-style-type: none"> <li>1) [3:7]</li> <li>2) [4:7]</li> <li>3) [3:6]</li> <li>4) [4:8]</li> </ol> <p><i>Ответ 3. "чай" = "случайность"[3: 6].</i></p>	<div data-bbox="1509 1005 1960 1240"> <p>Рисование символических выводов строк с помощью циклов с неограниченным количеством повторений</p> <p>Блиц-опрос Поднимайте карточку с правильными ответами.</p> <p>Вопрос № 4 Выберите правильный вариант среза "чай" из слова "случайность".</p> <ol style="list-style-type: none"> <li>1) [3:7]</li> <li>2) [4:7]</li> <li>3) [3:6]</li> <li>4) [4:8]</li> </ol> <p>"случайность" 0 1 2 3 4 5 6 7 8 9 10</p> <p>"чай" = "случайность"[3:6]</p> </div> <p>Сцена 6</p>

	<p><b>Пятый вопрос.</b> Какое из условий ложно?</p> <p>1) if 3 != 2 or 2 &gt; 3:  2) if 3 != 3 or 2 &lt; 3:  3) if 2 &gt; 3 or 2 == 3  4) if 3 &lt; 3 or 2 &lt; 3:</p> <p><i>Ответ 3. if 2 &gt; 3 or 2 == 3 – ложно, поскольку каждая из частей условия ложна.</i></p>	 <p>Сцена 7</p>	
<p>Теоретический блок.</p> <p><b>Псевдо-графика.</b></p>	<p>Перейдем к теме урока.  Первый проект имеет название «Горнолыжный слалом».  Если вы катались на горных лыжах или видели это по телевизору, то наверняка знаете, что спортсмен едет не по прямой, а наискосок, постоянно меняя свою траекторию. Так и наша «лыжня» будет постоянно петлять из стороны в сторону.  В основе ее будет три цикла: внешний – бесконечный, и два внутренних – конечных, которые и будут отвечать за повороты.</p>	 <p>Сцена 8</p>	8 9
	<ul style="list-style-type: none"> <li>• Основа программы</li> <li>Бесконечный цикл</li> <li>Цикл движения вправо</li> <li>Цикл движения влево</li> <li>• Состав любой строки</li> </ul> <p>N пробелов + символы + 10 пробелов + символы  Число N то увеличивается, то уменьшается</p>	 <p>Сцена 9</p>	



Блок заданий.  
**Практические задания:**  
Задание 1  
Задание 2  
Задание 3  
Задание 4

*После окончания дикторского текста запускается таймер на 4 мин.*

**Задание 1. Алгоритм «Горнолыжный слалом»**

- `text = "||        ||"`
- Организовать бесконечный цикл
  - o Организовать конечный восходящий цикл от 0 до 20  
Напечатать пробел \* счетчик + `text`
  - o Организовать конечный нисходящий цикл от 20 до 0  
Напечатать пробел \* счетчик + `text`

*После запуска таймера учащиеся записывают код программы и проверяют её работу.*

**Разбор задания 1. Ваш код может выглядеть так.**

```
text = "||        ||"  
while True:  
    for i in range(20):  
        print(" " * i + text)  
    for i in range(20, 0, -1):  
        print(" " * i + text)
```

*Здесь впервые создан проект, в котором встречаются циклы внутри цикла. Такие конструкции называются **вложенные**.*

Алгоритм «Горнолыжный слалом»

- `text = "||        ||"`
- Организовать бесконечный цикл
  - Организовать конечный восходящий цикл от 0 до 20  
Напечатать пробел \* счетчик + `text`
  - Организовать конечный нисходящий цикл от 20 до 0  
Напечатать пробел \* счетчик + `text`








Сцена 10

```
text = "||        ||"  
while True:  
    for i in range(20):  
        print(" " * i + text)  
    for i in range(20, 0, -1):  
        print(" " * i + text)
```

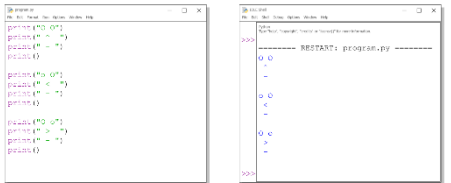

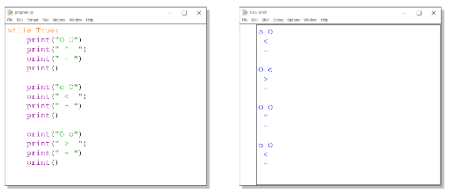
Сцена 11


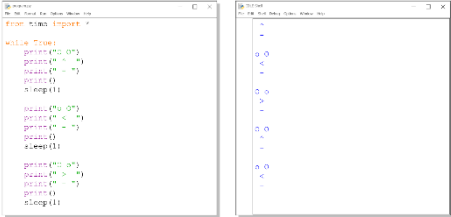

10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23

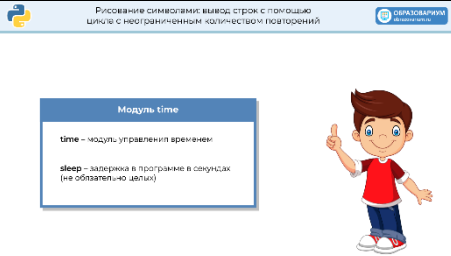
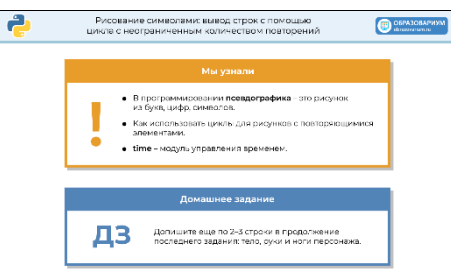
	<p><i>После окончания дикторского текста запускается таймер на 1 мин.</i></p> <p><b>Задание 1. Алгоритм «Горнолыжный слалом» (продолжение)</b></p> <p>Число «двадцать» в наших циклах можно поменять на большее или меньшее. В зависимости от этого, наша «<b>лыжня</b>» будет чаще или реже менять направление.</p>	<div data-bbox="1509 228 1962 501"><div><div>Рисование симметричных выводов строк с помощью цикла с неограниченным количеством повторений</div><div>Алгоритм «Горнолыжный слалом»</div><div><ul style="list-style-type: none"><li>■ <code>text = "    "</code></li><li>■ Организовать бесконечный цикл<ul style="list-style-type: none"><li>□ Организовать конечный вложенный цикл от 0 до 20<ul style="list-style-type: none"><li>■ Напечатать <code>пробел * счетчик + text</code></li></ul></li><li>□ Организовать конечный вложенный цикл от 20 до 0<ul style="list-style-type: none"><li>■ Напечатать <code>пробел * счетчик + text</code></li></ul></li></ul></li><li>■ Изменить число в конечных циклах на свое усмотрение</li></ul></div><div></div></div></div> <div data-bbox="1485 531 1590 560">Сцена 12</div>
	<p><i>Постановка задачи</i></p> <p>Второй проект будет посвящен анимации лица. Мы создадим три картинки в стиле псевдографики: профиль, вид справа и вид слева. После чего активируем бесконечный цикл, который будет их менять В этом проекте каждый рисунок представлен тремя строчками с участием команды <b>print</b>. Плюс команда <b>print</b> с пустыми скобками между блоками кода. Итого <b>12 строк</b>.</p>	<div data-bbox="1509 655 1962 911"><div><div>Рисование симметричных выводов строк с помощью цикла с неограниченным количеством повторений</div><div>Проект «Анимация лица»</div><div><div><div>Каждый рисунок:</div><div><ul style="list-style-type: none"><li>□ 3 строки с участием команды <code>print()</code></li><li>□ между пустой строкой <code>print()</code></li></ul></div></div></div><div></div></div></div> <div data-bbox="1485 959 1590 987">Сцена 13</div>

<p><i>После окончания дикторского текста запускается таймер на 2 мин.</i></p> <p><b>Задание 2. Алгоритм «Анимация лица»</b></p> <ul style="list-style-type: none"> <li>Создайте первую картинку из трех строчек с командой <b>print()</b></li> <li>Добавьте пустой <b>print()</b></li> </ul> <p><i>После запуска таймера учащиеся записывают код программы и проверяют её работу.</i></p>	<div data-bbox="1512 231 1960 263">  <span>Рисование символическим выводом строк с помощью цикла с неограниченным количеством повторений</span>  </div> <div data-bbox="1512 295 1937 502"> <div data-bbox="1518 300 1736 430"> <p>Алгоритм «Анимация лица»</p> <ul style="list-style-type: none"> <li>Создать первую картинку из трех строк с командой <b>print()</b></li> <li>Добавить пустой <b>print()</b></li> </ul> </div>  <div data-bbox="1792 446 1926 502"> <div>0</div><div>2</div><div>:</div><div>00</div> </div> </div>
---	--

Сцена 14

	<p><b>Разбор задания 2 (продолжение).</b> Добавленный код программы может выглядеть так:</p> <pre>print("o O") print(" &lt; ") print(" - ") print()  print("O o") print(" &gt; ") print(" - ") print()</pre>	<div data-bbox="1512 231 1960 470"><p>Рисование символами: вывод строк с помощью цикла с неограниченным количеством повторений</p></div> <p>Сцена 17</p>
	<p><i>После окончания дикторского текста запускается таймер на 1 мин.</i></p> <p><b>Задание 2. Алгоритм «Анимация лица» (продолжение).</b></p> <ul style="list-style-type: none"><li>Создайте бесконечный цикл</li></ul> <p>Вставьте весь код внутрь</p>	<div data-bbox="1512 710 1960 981"><p>Рисование символами: вывод строк с помощью цикла с неограниченным количеством повторений</p><div data-bbox="1512 766 1736 957"><p>Алгоритм «Анимация лица»</p><ul style="list-style-type: none"><li>Создать первую картинку из трех строк с командой <code>print()</code></li><li>Добавить пустой <code>print()</code></li><li>Создать вторую и третью картинку из трех строк с командой <code>print()</code></li><li>Добавить новую строку пустой <code>print()</code></li><li>Создать бесконечный цикл и разместить весь код внутри него</li></ul></div></div> <p>Сцена 18</p>
	<p><i>Постановка задачи</i></p> <p>Код программы будет выглядеть примерно так, как на рис. <b>Сцена 19</b></p> <p>Однако картинки меняются слишком быстро, и мы не успеваем их увидеть. Как бы нам замедлить программу?</p> <p>В этом нам поможет модуль управления временем <b>time</b> и команда <b>sleep</b>.</p>	<div data-bbox="1512 1069 1960 1340"><p>Рисование символами: вывод строк с помощью цикла с неограниченным количеством повторений</p><p>Модуль <code>time</code> управляет временем. Команда <code>sleep</code> делает задержку.</p></div> <p>Сцена 19</p>

	<p><i>После окончания дикторского текста запускается таймер на 2 мин.</i></p> <p><b>Задание 3. Алгоритм «Анимация лица» (окончание)</b></p> <ul style="list-style-type: none"> <li>Подключите модуль времени <b>from time import *</b></li> <li>Вставьте после каждой картинки задержку на 1 секунду <b>sleep (1)</b></li> </ul>	<p>Рисование символами: вывод строк с помощью циклов с неограниченным количеством повторений</p> <p>Алгоритм «Анимация лица»</p> <ul style="list-style-type: none"> <li>Подключить модуль времени: <b>from time import *</b></li> <li>Вставить после каждой картинке задержку на 1 секунду: <b>sleep()</b></li> </ul>  <p>Сцена 20</p>
	<p><b>Разбор задания 3.</b> Дописанный код программы может выглядеть так, как рис. <b>Сцена 21:</b></p> <p>Перед внешним циклом появилась строка кода: <b>from time import *</b> После каждого внутреннего цикла появилась строка команды: <b>sleep(1)</b></p> <p><i>Обратите внимание – картинки меняются уже на так быстро как вначале.</i></p>	<p>Рисование символами: вывод строк с помощью циклов с неограниченным количеством повторений</p>  <p>Сцена 21</p>
	<p><i>После окончания дикторского текста запускается таймер на 1 мин.</i></p> <p><b>Дополнительно к проекту «Анимация лица»</b></p> <p>Время задержки в команде <b>sleep</b> не обязательно должно быть целым числом. Попробуйте поменять аргумент внутри команды, используя дробное значение.</p>	<p>Рисование символами: вывод строк с помощью циклов с неограниченным количеством повторений</p> <p>Команда <b>sleep</b> может принимать и дробные значения.</p> <p>Алгоритм «Анимация лица». Задержка времени</p> <ul style="list-style-type: none"> <li>Показывать значения задержки на свое усмотрение.</li> </ul>  <p>Сцена 22</p>

	<p><i>Обратите внимание!</i></p> <p>Модуль <b>time</b></p> <p><b>time</b> – модуль управления временем.</p> <p><b>sleep</b> – задержка в программе в секундах (не обязательно целых)</p>	 <p>Сцена 23</p>	
<p>Блок завершения занятия.</p> <p><b>Рефлексия.</b></p> <p><b>Сообщение домашнего задания</b></p>	<p><i>Завершаем демонстрацией ролика и кратким обобщением материалов занятия.</i></p> <p><b>Подведем итоги.</b></p> <p>Мы узнали:</p> <ul style="list-style-type: none"> <li>▪ в программировании псевдографика – это рисунок из букв, цифр, символов;</li> <li>▪ как использовать циклы для рисунков с повторяющимися элементами;</li> <li>▪ <b>time</b> – модуль управления временем.</li> </ul> <p><i>Преподаватель дает ученикам домашнее задание к следующему занятию (Приложение 2).</i></p>	 <p>Сцена 24</p>	24

## Приложение 2

### Домашнее задание

Создайте дополнение к проекту занятия «Анимация лица», добавив анимацию тела, рук и ног персонажа.

*Задание можно выполнить на компьютере и представить результат и код в виде файла или снимка экрана, или распечатки.*

## Практика

### Проект «Узор»

Выведите на экран изображение ромба. Поместите вывод ромба в бесконечный цикл. Символ для рисования и размер ромба задайте самостоятельно.

Пример изображения ромба.



### Проект «Пингвин»

Запрограммируйте вывод на экран трех изображений пингвина (каждое изображение по отдельности).



Меняя их по очереди и используя задержку времени сделайте анимацию шагающего пингвина. Можете продолжить проект, дополнив его своими идеями.

### Приложение 3

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

«Рисование символами: вывод строк с помощью цикла с неограниченным количеством повторений. Создание эффекта анимации».

**В начале занятия** можно поинтересоваться, какие картинки в стиле псевдографика ребятам с элементами цикличности или математических закономерностей ребятам удалось найти. Если есть техническая возможность – можно их показать и предложить проанализировать. Также необходимо повторить синтаксис и принцип работы конечного цикла, а заодно рассказать о вложенных циклах. Как пример – минуты внутри часа, час внутри суток. Данная информация пригодится при выполнении первого 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.

**При создании первого проекта** можно менять вводные данные как в текстовой переменной, так и в конечных циклах. Аналогичным образом можно поступить и в задании № 2 Если у ребят возникнут свои идеи по анимации, то это можно только приветствовать.

**Если останется время** – можно расширить второй проект, за счет увеличения количества картинок. Также можно проработать дополнения к проекту – например другие части тела.

**Рекомендуем** заострить внимание на схожести подключения разных модулей, когда меняется только название самого модуля.