



### Методические рекомендации по теме

#### Игра «Угадай число»

##### Цель:

- дать представление о прикладном применении случайного выбора в программировании при создании компьютерных игр на языке Python.

##### Задачи:

- практика применения случайного выбора при программировании игр в Python;
- понимание случайного выбора в программировании как основы игрового проекта;
- 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 случайного выбора и генерации случайных целых и дробных чисел в Python при разработке игр.

*Метапредметные:* обучающиеся получают возможность владения общепредметными понятиями «случайный выбор», «случайное число», «генерация случайных чисел»; 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 используем модуль **random**, бесконечный цикл, операторы условия).

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

- Какие игры с использованием случайного выбора вы внесли в свои списки?
- Где в этих играх используется случайный выбор?
- Как могли бы быть использованы в этих играх методы **random** и **uniform**?

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

- случайный выбор является основой множества компьютерных и реальных игр;

– достаточно сложно представить игру, в которой бы отсутствовал элемент случайного выбора.  
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.**

### **Рандомизация выбора числа.**

Новый материал излагается в сопровождении видеоролика.

Занятие полностью посвящено разработке игрового проекта, поэтому теоретический блок представлен в виде небольших вставок, разделяющих этапы работы над проектом и содержащих описание следующего этапа.

Обсуждение вопросов по просмотренным материалам:

- В чем смысл проекта?

- Какова механика работы проекта?
- Как определяется выигрыш в игре?

*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.

### Проект: Игра «Угадай число».

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

«Угадай число»: включает **часть 1** для учеников с последующим разбором. В этой части организуется переменная и ей присваивается значение сгенерированного случайного числа. Это число нужно будет угадать.

«Угадай число»: включает **часть 2** для учеников с последующим разбором. В этой части, создается бесконечный цикл, который позволяет пользователю сделать несколько попыток в угадывании числа.

«Угадай число»: включает **часть 3** для учеников с последующим разбором. В этой части, дописывается программный код, который помогает пользователю угадывать число с помощью подсказок.

«Угадай число»: включает **часть 4** для учеников с последующим разбором. В этой части, дописывается программный код, который ограничивает количество попыток и в случае их исчерпания сообщает загаданное число.

***После выполнения задания*** ученики получают работающую программу для игры «Угадай число»

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

*\*см. сцены 9 – 17*

#### **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.

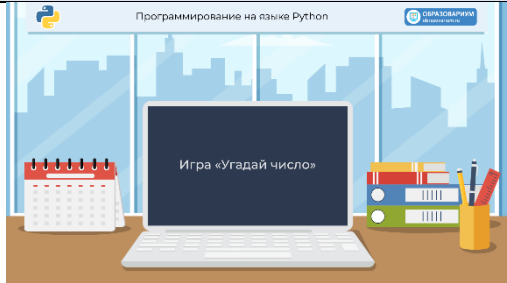
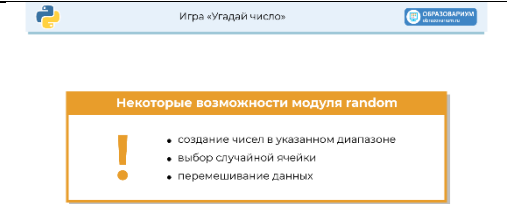
Преподаватель дает ученикам домашнее задание к следующему занятию (*Приложение 2*).

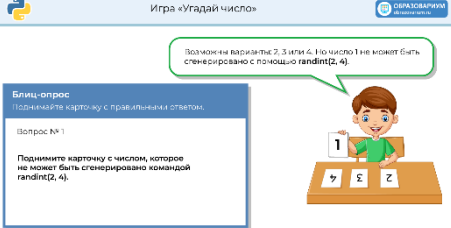
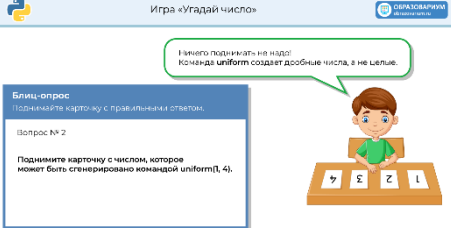
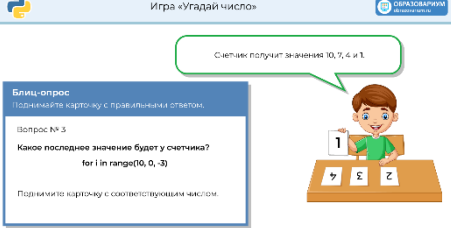
*\*см. сцена 18*

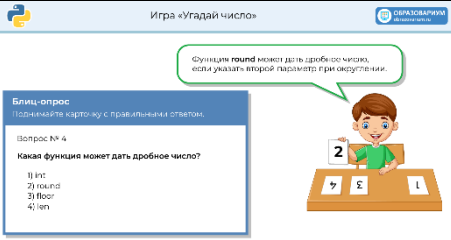
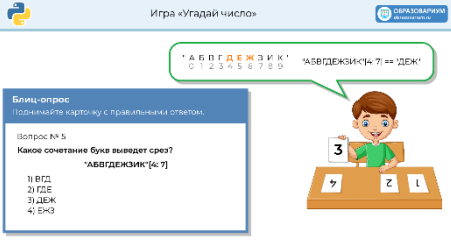
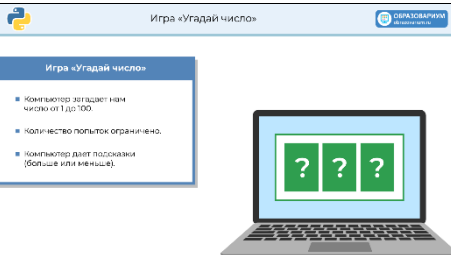
## Сценарный план видеоролика

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.

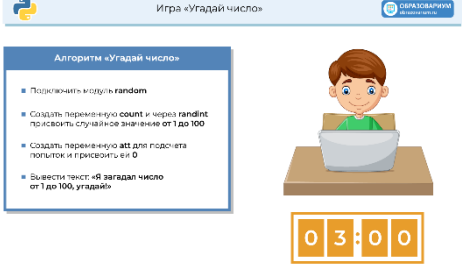
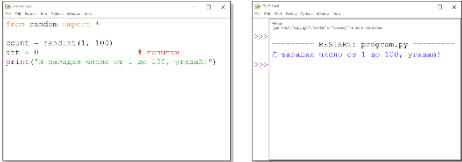

Таблица. Содержание видеоролика

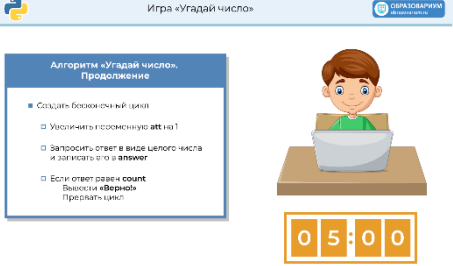
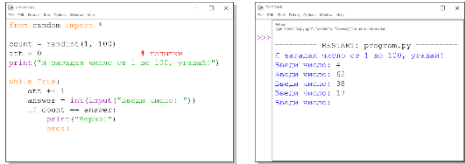
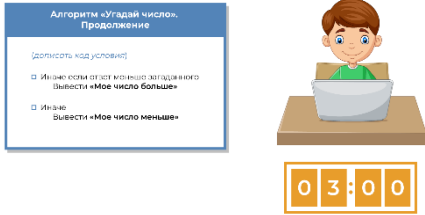
Название блока	Содержание блока и комментарии	Фрагменты из видеоролика	№ сцен
Вводный блок. Мы узнаем	Обозначаем ученикам тему и цели урока.  Игра «Угадай число».	 Сцена 1	1 2
	С использованием модуля <b>random</b> , который осуществляет генерацию случайных событий, создаются многие проекты. В том числе такие, как игры с числами.  Некоторые возможности модуля <b>random</b> : <ul style="list-style-type: none"> <li>• создание чисел в указанном диапазоне;</li> <li>• выбор случайной ячейки;</li> <li>• перемешивание данных.</li> </ul>	 Сцена 2	

<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></p> <p>Поднимите карточку с номером числа, которое не сгенерирует <b>randint</b> (2,4)</p> <p><i>Ответ 1. Возможны варианты: 2, 3 или 4.</i></p>	 <p>Сцена 3</p>	<p>3 4 5 6 7</p>
	<p><b>Второй вопрос.</b></p> <p>Поднимите карточку с номером числа, которое может сгенерировать <b>uniform</b> (1, 4)</p> <p><i>Ответ: ничего поднимать не надо! Команда uniform создает дробные числа, а не целые.</i></p>	 <p>Сцена 4</p>	
	<p><b>Третий вопрос.</b> Какое последнее значение будет у счетчика? Поднимите карточку с номером ответа.</p> <p><b>for i in range (10, 0, -3)</b></p> <p><i>Ответ 1. Счетчик получит значения 10, 7, 4, 1.</i></p>	 <p>Сцена 5</p>	

	<p><b>Четвертый вопрос.</b> Какая функция может дать дробное число?</p> <ol style="list-style-type: none"> <li>1) int</li> <li>2) round</li> <li>3) floor</li> <li>4) len</li> </ol> <p><i>Ответ 2. Функция <b>round</b> может дать дробное число, если указать второй параметр при округлении.</i></p>	 <p>Сцена 6</p>	
	<p><b>Пятый вопрос.</b> Какое сочетание букв выведет срез "АБВГДЕЖЗИК"[4: 7]</p> <ol style="list-style-type: none"> <li>1) ВГД</li> <li>2) ГДЕ</li> <li>3) ДЕЖ</li> <li>4) ЕЖЗ</li> </ol> <p><i>Ответ 3. "АБВГДЕЖЗИК"[4: 7] = "ДЕЖ".</i></p>	 <p>Сцена 7</p>	
<p>Теоретический блок.</p> <p><b>Игра</b></p>	<p>Создадим свой первый игровой проект, в котором исходные данные будет создавать компьютер на основе модуля <b>random</b>. Это будет проект «Угадай число».</p> <p><b>Игра «Угадай число»</b></p> <ul style="list-style-type: none"> <li>• Компьютер загадает нам число от 1 до 100</li> <li>• Количество попыток - ограничено</li> <li>• Компьютер дает подсказки (больше или меньше)</li> </ul>	 <p>Сцена 8</p>	8



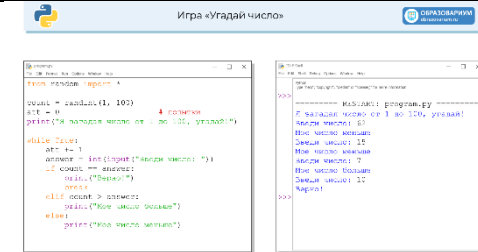
<p>Блок заданий. <b>Практические задания:</b> Часть 1 Задание 2 Задание 3 Задание 4</p>	<p><i>После окончания дикторского текста запускается таймер на 3 мин.</i></p> <p><b>Алгоритм «Угадай число». Часть 1.</b></p> <ul style="list-style-type: none"> <li>Подключить модуль <b>random</b></li> <li>Создать переменную count и через randint присвоить случайное значение от 1 до 100</li> <li>Создать переменную att для подсчета попыток и присвоить ей 0</li> <li>Вывести текст: «Я загадал число от 1 до 100, угадай!»</li> </ul>	 <p>Сцена 9</p>	<p>9 10 11 12 13 14 15 16 17</p>
	<p><b>Разбор Часть 1.</b> Ваш код может выглядеть так.</p> <pre>from random import *  count = randint(1, 100) att = 0 # попытки print("Я загадал число от 1 до 100, угадай!")</pre> <p><i>Создаем переменную со случайным числом и переменную для подсчета попыток.</i></p>	 <p>Сцена 10</p>	
	<p><i>Немного теории:</i></p> <p>Давайте ответим на вопросы: будет ли наша программа одноразовой или нет?</p> <p>Нет, предполагается несколько попыток.</p> <p>Значит нам понадобится цикл, какой - конечный, с условием или бесконечный?</p> <p>Поскольку мы не знаем, с какой попытки сможем угадать, то конечный отпадает.</p> <p>Сколько у нас условий для выхода из цикла?</p> <p>Два. Если отгадали число или если закончились отведенные попытки.</p> <p>Значит, лучшим вариантом для нас станет бесконечный цикл, с выходом по команде <b>break</b></p>	 <p>Сцена 11</p>	

	<p><i>После окончания дикторского текста запускается таймер на 5 мин.</i></p> <p><b>Алгоритм «Угадай число». Часть 2.</b></p> <ul style="list-style-type: none"> <li>Создать бесконечный цикл <ul style="list-style-type: none"> <li>Изменить переменную <b>att</b> на 1</li> <li>Запросить ответ в виде целого числа и записать его в <b>answer</b></li> <li>Если ответ равен <b>count</b> <p>Написать: «верно»</p> <p>Завершить цикл</p> </li> </ul> </li> </ul>	 <p>Сцена 12</p>	
	<p><b>Разбор Часть 2.</b> Код добавленных строк программы может выглядеть так.</p> <pre>while True:     att += 1     answer = int(input("Введи число: "))     if count == answer:         print("Верно!")         break</pre>	 <p>Сцена 13</p>	
	<p><i>После окончания дикторского текста запускается таймер на 3 мин.</i></p> <p><b>Алгоритм «Угадай число». Часть 3.</b></p> <ul style="list-style-type: none"> <li>Если ответ равен count <p>Написать: «верно»</p> <p>Завершить цикл</p> </li> <li>Иначе если ответ &lt; загаданного <p>Написать: «мое число больше»</p> </li> <li>Иначе <p>Написать: «мое число меньше»</p> </li> </ul>	 <p>Сцена 14</p>	

**Разбор Часть 3.** Код добавленных строк программы может выглядеть так.

```
elif count > answer:
    print("Мое число больше")
else:
    print("Мое число меньше")
```

Компьютер дает нам подсказки, на которые мы можем ориентироваться.

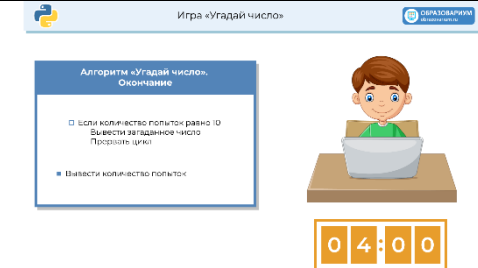


Сцена 15

После окончания дикторского текста запускается таймер на 3 мин.

#### Алгоритм «Угадай число». Часть 4.

- Если попытки равно 10  
Напечатать загаданное число  
Прервать цикл
- Напечатать количество попыток

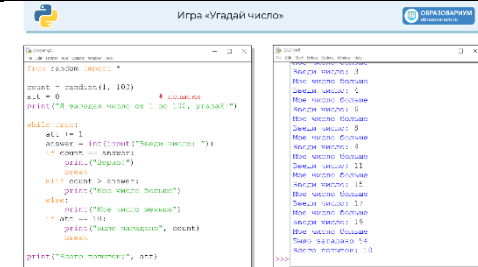


Сцена 16

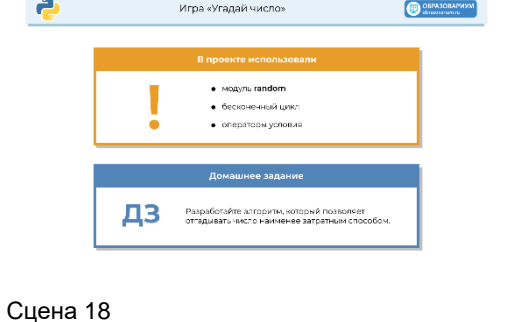
**Разбор Часть 4.** Код добавленных строк программы может выглядеть так.

```
if att == 10:
    print("Было загадано", count)
    break
print("Всего попыток:", att)
```

Проект готов, можем соревноваться – кто потратит меньше всех попыток на отгадывание числа! Возможно, победит тот, кто случайно угадает с первого раза. Но это игра и кому-то может повезти. Если же и десяти попыток вам не хватает – значит надо пересмотреть свою нынешнюю стратегию.



Сцена 17

Блок завершения занятия. <b>Рефлексия.</b> <b>Сообщение домашнего задания</b>	<p><i>Завершаем демонстрацией ролика и кратким обобщением материалов занятия.</i></p> <p><b>Подведем итоги.</b></p> <p>Для проекта понадобились</p> <ul style="list-style-type: none"> <li>▪ модуль <b>random</b>;</li> <li>▪ бесконечный цикл;</li> <li>▪ операторы условия.</li> </ul> <p><i>Преподаватель дает ученикам домашнее задание к следующему занятию (Приложение 2).</i></p>	 <p>Сцена 18</p>	18
---	--	---	----

## Приложение 2

### Домашнее задание

Разработайте алгоритм, который позволяет отгадывать число наименее затратным способом.

В качестве подсказки – максимальное количество попыток не должно превышать восьми.

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

### Практика

Проект «Игра в кости»

Цель проекта: реализовать имитацию игры в кости.

Сгенерируйте два случайных числа, обозначающих сколько выпало у пользователя на кубиках. Выведите соответствующее сообщение и подсчитайте очки пользователя.

Сделайте задержку времени. Затем повторите действия для «броска кубиков компьютером».

Введите очки пользователя и компьютера. Определите, кто выиграл и выведите соответствующее сообщение.

Проект «Орел и решка»

Цель проекта: реализовать имитацию подбрасывания монеты.

Запросите количество «бросков» монеты. Для каждого броска случайным образом определите, выпал «орел» или «решка». Выведите соответствующие сообщения.

Подсчитайте и выведите количество бросков монеты (можно добавить и в процентах) с каждым из результатов «орел» и «решка».

### Приложение 3

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

«Игра «Угадай число»».

*В начале занятия* опросить ребят, что им удалось вспомнить по теме «случайность в играх». Привести наиболее известные примеры: карты, домино, лото, нарды, камень-ножницы-бумага, морской бой...

Повторите методы **random** и **uniform**: принцип работы, параметры и синтаксис. Можно попросить привести несколько примеров генерации чисел. Например: какие числа буду сгенерированы, если **randint** (2,4).

*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 просмотром теоретической части*, можно поставить ролик на паузу и сыграть в эту игру с кем-то из учеников: учитель записывает загаданное число, а ученик отгадывает, ориентируясь на подсказки. Роль счетчика можно отвести другому ученику. Таким образом ребята лучше поймут игровую механику проекта.

Ввиду определенной сложности, проект создается по частям. После каждой части рекомендуется проверять код на отсутствие ошибок и работоспособность.

**Игровой параметры** (диапазон от 1 до 100 и количество попыток = 10) достаточно условны. Вы можете изменить их на свое усмотрение.

**Если останется время** – можете еще раз сыграть между собой, но усложнив правила: загадывается не одно, а два числа. Отгадывать тоже придется 2 одновременно. Записями и калькулятором пользоваться нельзя.

**Пример.** Загаданы 7 и 16 (от 1 до 20);

5 и 10? - больше / больше

10 и 15? – меньше/ больше...