



Методические рекомендации по теме «Игра «Быки и коровы». Формирование данных»

Цель:

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

Задачи:

- практика применения множеств при программировании игр в Python;
- понимание множеств 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 возможностей множеств в Python при разработке игр.

Метапредметные: обучающиеся получают возможность владения общепредметными понятиями «список», «множество»; 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: 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.

2. Вводный блок.

Тема.

Играем в Быки и Коровы:

Преподаватель объясняет правила игры «Быки и Коровы» и предлагает детям сыграть между собой в эту игру.

Правила игры:

Один игрок загадывает комбинацию из 4 неповторяющихся цифр. Другой игрок старается угадать комбинацию.

Угадывающий предлагает свои варианты. Загадывающий после каждого варианта сообщает результат попытки. Если

угадывающий угадал цифру, которая есть в комбинации – это К (корова), если и цифру, и ее место в комбинации, то это Б (бык). Опираясь на эти данные угадывающий, постепенно расшифровывает комбинацию.

**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.

Игра «Быки и коровы»

Новый материал излагается в сопровождении видеоролика, рекомендуется разместить на доске или флип-чарте изображения объектов, сопровождающих материалы по теме.

Обсуждением вопросов по просмотренным материалам:

- В чем смысл проекта?
- Какова механика работы проекта?
- Что мы сделали на определенном этапе работы над проектом?
- Как задействованы списки, множества, случайный выбор, переменные и другие понятия программирования в игре?
- Для чего нужно создавать множество, а потом переводить его в список?
- Что такое команды, именуемые генераторы?

- Как вводить данные в список, использовать команду – разделитель информации?

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 8-12, 15, 18, 21-23*

5. Блок заданий.

Проект: «Быки и коровы».

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

«Быки и коровы»: включает *практическое задание 1* которое выполняется в несколько этапов: представление четырехзначного числа в виде четырех однозначных, которые являются ячейками списка; организовать ввод четырех чисел и перемешивание данных; организация проверки – введено ровно 4 числа, введены разные числа.

После выполнения задания ученики получают работающий программный продукт – программа вида викторина, позволяющая проверить знание столиц разных стран с выводом количества правильных ответов.

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

**см. сцены 13 – 25 (кроме сцен по теории).*

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 26*

Приложение 1

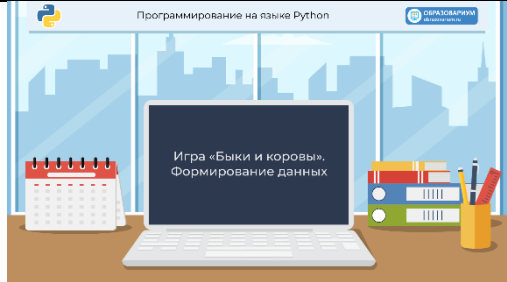
Сценарный план видеоролика

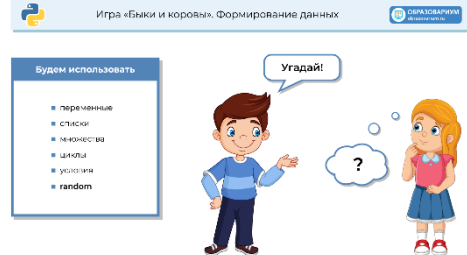
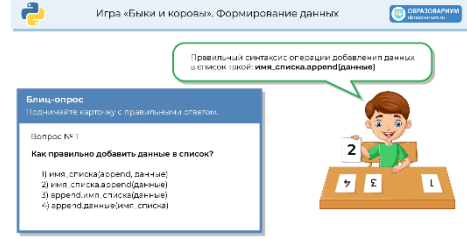
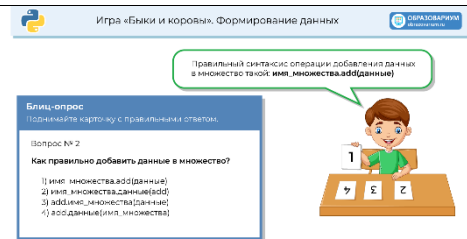
В таблице «Содержание видеоролика» представлены:

- название блоков видеоролика (тайминг);
- краткое описание содержания в каждом блоке;
- фрагменты из видеоролика, относящиеся к соответствующему блоку;
- номера сцен в каждом блоке.

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 «Быки и коровы». Формирование данных	 Сцена 1	1 2

	<p>Приступаем к созданию игрового проекта, в котором потребуются применить все имеющиеся знания: переменные, списки, множества, циклы, условия и рандом.</p> <p>Это будет «Быки и коровы» – логическая игра, в ходе которой за некоторое количество попыток один из игроков должен определить, что загадал другой игрок.</p>	 <p>Игра «Быки и коровы». Формирование данных</p> <p>Будем использовать</p> <ul style="list-style-type: none"> ■ переменные ■ списки ■ множества ■ циклы ■ условия ■ random <p>Угадай!</p> <p>Сцена 2</p>	
Блок повторения.	<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. Как правильно добавить данные в список?</p> <ol style="list-style-type: none"> 1) имя_списка(append, данные) 2) имя_списка.append(данные) 3) append.имя_списка(данные) 4) append.данные(имя_списка) <p><i>Ответ 2. Правильный синтаксис операции добавления данных в список такой: имя списка.append(данные)</i></p>	 <p>Игра «Быки и коровы». Формирование данных</p> <p>Правильный синтаксис операции добавления данных в список такой: <code>имя_списка.append(данные)</code></p> <p>Блиц-опрос Поднимайте карту, у с правильным ответом.</p> <p>Вопрос № 1</p> <p>Как правильно добавить данные в список?</p> <ol style="list-style-type: none"> 1) имя_списка(append, данные) 2) имя_списка.append(данные) 3) append.имя_списка(данные) 4) append.данные(имя_списка) <p>Сцена 3</p>	3 4 5 6 7
Блиц-опрос	<p>Второй вопрос. Как правильно добавить данные в множество?</p> <ol style="list-style-type: none"> 1) имя_множества.add(данные) 2) имя_множества.данные(add) 3) add.имя_множества(данные) 4) add.данные(имя_множества) <p><i>Ответ 1. Правильный синтаксис операции добавления данных в множество такой: имя множества.add(данные)</i></p>	 <p>Игра «Быки и коровы». Формирование данных</p> <p>Правильный синтаксис операции добавления данных в множество такой: <code>имя_множества.add(данные)</code></p> <p>Блиц-опрос Поднимайте карту, у с правильным ответом.</p> <p>Вопрос № 2</p> <p>Как правильно добавить данные в множество?</p> <ol style="list-style-type: none"> 1) имя_множества.add(данные) 2) имя_множества.данные(add) 3) add.имя_множества(данные) 4) add.данные(имя_множества) <p>Сцена 4</p>	

	<p>Третий вопрос. Какое утверждение является ложным?</p> <ol style="list-style-type: none"> 1) множество – изменяемая структура 2) множество – индексируемая структура 3) ячейки множества можно сосчитать 4) множество использует фигурные скобки <p><i>Ответ 2. Множество не имеет индексов.</i></p>	<p>Игра «Быки и коровы». Формирование данных</p> <p>Множество не имеет индексов.</p> <p>Блиц-опрос Поднимите карточку с правильным ответом.</p> <p>Вопрос № 3 Какое утверждение является ложным?</p> <ol style="list-style-type: none"> 1) множество – изменяемая структура 2) множество – индексируемая структура 3) ячейки множества можно сосчитать 4) множество использует фигурные скобки <p>2</p> <p>4 5 1</p> <p>Сцена 5</p>	
	<p>Четвертый вопрос. Какая последовательность чисел не может быть множеством?</p> <ol style="list-style-type: none"> 1) 1 2 3 4 5 2) 5 4 3 2 1 3) 1 2 3 4 1 4) 1 5 3 4 2 <p><i>Ответ 3. 1 2 3 4 1 не может быть множеством, т. к. множество не может содержать повторяющиеся элементы.</i></p>	<p>Игра «Быки и коровы». Формирование данных</p> <p>Множество не может содержать повторяющиеся элементы, поэтому 1 2 3 4 1 – не множество.</p> <p>Блиц-опрос Поднимите карточку с правильным ответом.</p> <p>Вопрос № 4 Какая последовательность чисел не может быть множеством?</p> <ol style="list-style-type: none"> 1) 1 2 3 4 5 2) 5 4 3 2 1 3) 1 2 3 4 1 4) 1 5 3 4 2 <p>3</p> <p>4 2 1</p> <p>Сцена 6</p>	
	<p>Пятый вопрос. Сколько основных ячеек в данном сложном списке?</p> <p>[[1, 2], [3], [4, 5, 6, 7, 8], []]</p> <p>Поднимите карточку с соответствующим числом.</p> <p><i>Ответ 4. В списке 4 основных ячейки.</i></p> <p>[[1, 2], [3], [4, 5, 6, 7, 8], []]</p> <p>0 1 2 3</p>	<p>Игра «Быки и коровы». Формирование данных</p> <p>[[1, 2], [3], [4, 5, 6, 7, 8], []] 0 1 2 3 В списке 4 основных ячейки.</p> <p>Блиц-опрос Поднимите карточку с правильным ответом.</p> <p>Вопрос № 5 Сколько основных ячеек в данном сложном списке?</p> <p>[[1, 2], [3], [4, 5, 6, 7, 8], []]</p> <p>Поднимите карточку с соответствующим числом.</p> <p>4</p> <p>5 2 1</p> <p>Сцена 7</p>	

Теоретический блок.

Игра «Быки и коровы»

Начнем создавать код для логической игры «Быки и коровы». Это игра для двоих, поэтому мы в качестве соперника будем использовать компьютер. Классический вариант игры называется «4 неповторяемых числа». Компьютер загадает четырехзначное число из неповторяющихся цифр. Задача игрока отгадать это число, пользуясь подсказками программы. Количество попыток не ограничено. Каждая попытка — это один вариант загаданного числа. Если в нем присутствует цифра из загаданного числа – то это корова, а если эта цифра еще и на той же позиции, то это бык. Разумеется, компьютер не сообщает какая именно цифра является коровой или быком. Угадать можно только анализируя свои ходы и ответы на них.

Рассмотрим пример игры. Компьютер загадал число 3509

Мы первым ходом предполагаем число 1234

Компьютер сообщает, что это одна корова.

Наша следующая попытка – число 9876.

Ответ компьютера - одна корова

Третья попытка – 3450

Ответ – две коровы и один бык

И так далее, пока не отгадаем все число

Мы сделаем первую часть проекта – сначала научим компьютер генерировать число. Потом создадим код, который будет принимать наши запросы.

А анализ попытки сделаем на следующем уроке.

Общий алгоритм

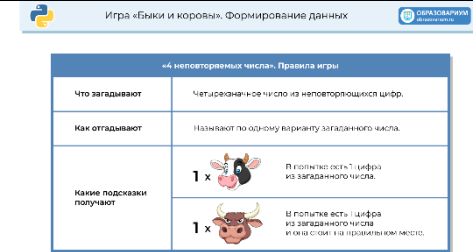
Компьютер генерирует число

Цикл

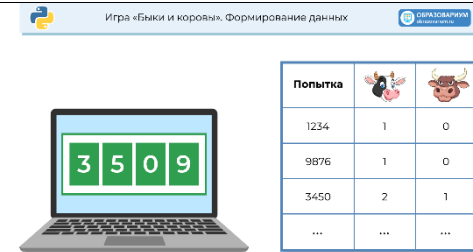
Наша попытка

Анализ и ответ компьютера

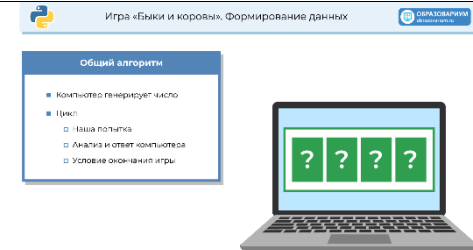
Условие окончания игры



Сцена 8

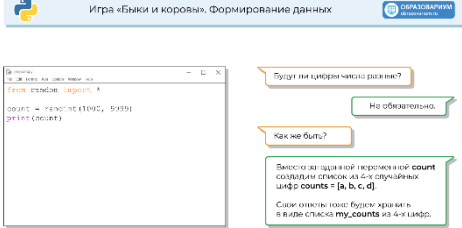
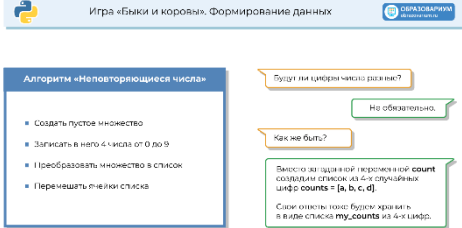
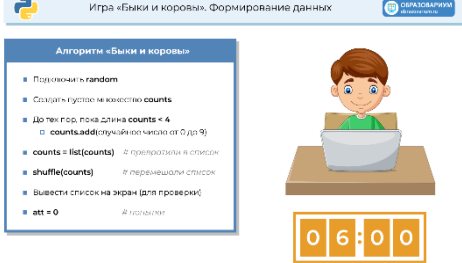


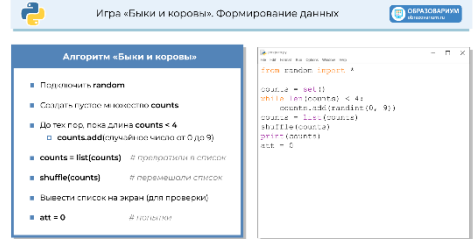
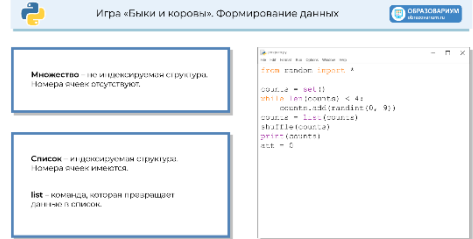
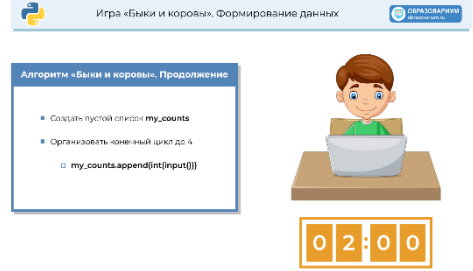
Сцена 9

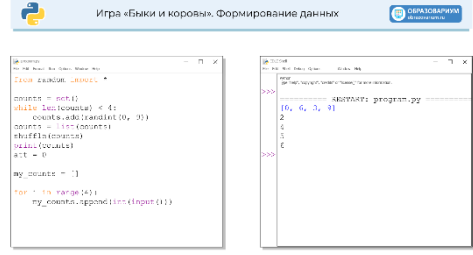
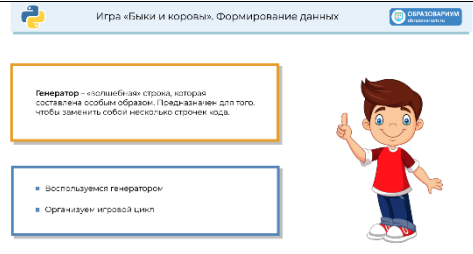
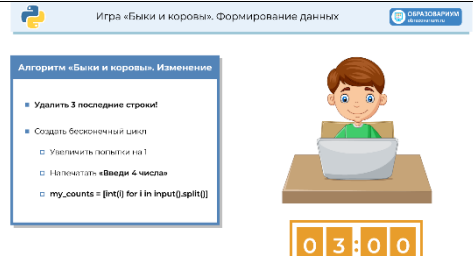


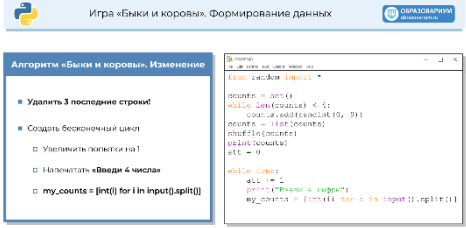
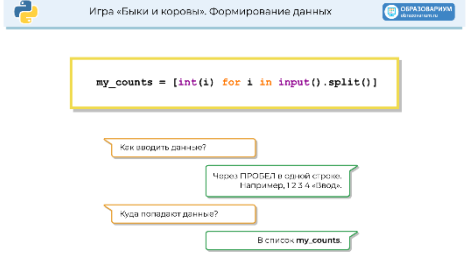
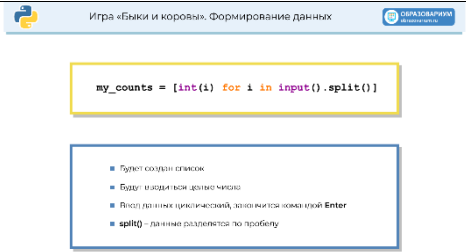
Сцена 10


8
9
10
11
12

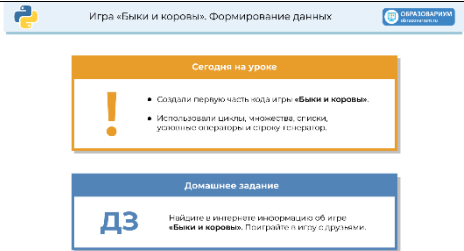
	<p>Нет никакой проблемы, чтобы компьютер сгенерировал четырехзначное число. (подключить random, указать в настройках метода randint диапазон 1000-9999.</p> <p>Но будут ли все цифры этого числа разными? Не факт! Может – да, а может – нет. Как же быть?</p> <pre>from random import * count = randint(1000, 9999) print(count)</pre>	 <p>Игра «Быки и коровы». Формирование данных</p> <p>Будут ли цифры числа разными? На обязательно. Как же быть? Вместо загадки про перемешивание count создадим список из 4-х случайных info counts = [a, b, c, d]. Сможете тогда будем хранить в виде списка my_counts из 4-х цифр.</p> <p>Сцена 11</p>	
	<p>Пойдем на хитрость. Представим, что компьютер загадал не четырехзначное число, а четыре однозначных, которые являются ячейками списка.</p> <p>Свой ответ мы тоже будем вводить в виде списка из четырех цифр.</p> <p>И поскольку у списков есть индексы – мы легко сможем проверять ячейки на совпадение.</p> <p>Алгоритм «Неповторяющиеся числа»</p> <ul style="list-style-type: none"> ■ Создать пустое множество ■ Записать в него 4 числа от 0 до 9 ■ Преобразовать множество в список ■ Перемешать ячейки списка 	 <p>Игра «Быки и коровы». Формирование данных</p> <p>Алгоритм «Неповторяющиеся числа»</p> <ul style="list-style-type: none"> ■ Создать пустое множество ■ Записать в него 4 числа от 0 до 9 ■ Преобразовать множество в список ■ Перемешать ячейки списка <p>Будут ли цифры числа разными? На обязательно. Как же быть? Вместо загадки про перемешивание count создадим список из 4-х случайных info counts = [a, b, c, d]. Сможете тогда будем хранить в виде списка my_counts из 4-х цифр.</p> <p>Сцена 12</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 «Быки и коровы»</p> <ul style="list-style-type: none"> ■ Подключить random ■ Создать пустое множество counts ■ Цикл работать пока длина counts < 4 ■ counts.add(случайное число от 0 до 9) ■ counts = list(counts) # превратили в список ■ shuffle(counts) # перемешали список ■ Вывести список на экран (для проверки) ■ att = 0 # попытки 	 <p>Игра «Быки и коровы». Формирование данных</p> <p>Алгоритм «Быки и коровы»</p> <ul style="list-style-type: none"> ■ Подключить random ■ Создать пустое множество counts ■ До тех пор, пока длина counts < 4 <ul style="list-style-type: none"> counts.add(случайное число от 0 до 9) ■ counts = list(counts) # превратили в список ■ shuffle(counts) # перемешали список ■ Вывести список на экран (для проверки) ■ att = 0 # попытки <p>Будут ли цифры числа разными? На обязательно. Как же быть? Вместо загадки про перемешивание count создадим список из 4-х случайных info counts = [a, b, c, d]. Сможете тогда будем хранить в виде списка my_counts из 4-х цифр.</p> <p>Сцена 13</p>	<p>13 14</p>

	<p>Разбор задания 1. Ваш код может выглядеть так.</p> <pre> from random import * counts = set() while len(counts) < 4: counts.add(randint(0, 9)) counts = list(counts) shuffle(counts) print(counts) att = 0 </pre>	 <p>Сцена 14</p>	
Теоретический блок.	<p>Для чего нам понадобилось сначала создавать множество, а потом переводить его в список?</p> <p>Множество, как вы должны помнить, является не индексируемой структурой.</p> <p>То есть мы не смогли бы понять, какая ячейка в нем первая, какая вторая и т.д.</p> <p>А вот список даст нам такую возможность.</p>	 <p>Сцена 15</p>	15
Блок заданий. Практические задания: Задание 1	<p><i>После окончания дикторского текста запускается таймер на 2 мин.</i></p> <p>Задание 1. Алгоритм «Быки и коровы». Продолжение</p> <ul style="list-style-type: none"> Создадим пустой список my_counts Конечный цикл до 4 <ul style="list-style-type: none"> my_counts.append(int(input())) 	 <p>Сцена 16</p>	16 17

	<p>Разбор задания 1. Ваш код может выглядеть так. <i>... дописывается к существующему коду</i></p> <pre>my_counts = [] for i in range(4): my_counts.append(int(input()))</pre> <p><i>Однако данный код не очень удобен в использовании. Приходится 4 раза нажимать на ввод, и введенные числа располагаются не строку, а в столбик.</i> <i>Нет ли другого решения?</i></p>	 <p>Игра «Быки и коровы». Формирование данных</p> <p>Сцена 17</p>	
Теоретический блок.	<p>Есть, и очень необычное. В Python имеются команды, именуемые генераторами. Это специальным образом составленная строка, которая заменяет собой несколько строчек кода. Мы воспользуемся генератором, который мало того, что заменит три строки кода, но и сделает ввод данных более удобным.</p> <p>Кроме того, давайте сразу организуем игровой цикл, в котором у нас будет происходить ввод и анализ данных</p>	 <p>Игра «Быки и коровы». Формирование данных</p> <p>Генератор – «кованная» строка, которая составлена особым образом. Предназначен для того, чтобы заменить собой несколько строчек кода.</p> <ul style="list-style-type: none"> ■ Воспользуемся генератором ■ Организуем игровой цикл <p>Сцена 18</p>	18
Блок заданий. Практические задания: Задание 1	<p><i>После окончания дикторского текста запускается таймер на 3 мин.</i></p> <p>Задание 1. Алгоритм «Быки и коровы». Изменение</p> <ul style="list-style-type: none"> ■ Удалить 3 последние строки! ■ Создать бесконечный цикл <ul style="list-style-type: none"> ○ Увеличить попытки на 1 ○ Напечатать «Введи 4 числа» ○ <code>my_counts = [int(i) for i in input().split()]</code> 	 <p>Игра «Быки и коровы». Формирование данных</p> <p>Алгоритм «Быки и коровы». Изменение</p> <ul style="list-style-type: none"> ■ Удалить 3 последние строки! ■ Создать бесконечный цикл <ul style="list-style-type: none"> ○ Увеличить попытки на 1 ○ Напечатать «Введи 4 числа» ○ <code>my_counts = [int(i) for i in input().split()]</code> <p>Сцена 19</p>	19 20

	<p>Разбор задание 1. Код программы будет выглядеть так:</p> <pre> from random import * counts = set() while len(counts) < 4: counts.add(randint(0, 9)) counts = list(counts) shuffle(counts) print(counts) att = 0 while True: att += 1 print("Введи 4 цифры") </pre>	 <p>Сцена 20</p>	
Теоретический блок.	<p>Прежде всего, давайте поймем, как нам теперь вводить данные. Очень просто: одной строкой через ПРОБЕЛ.</p> <p>А куда попадают данные? В список my_counts, который создался автоматически</p> <pre>my_counts = [int(i) for i in input().split()]</pre>	 <p>Сцена 21</p>	21 22 23
	<p>Давайте повнимательней разберем эту строку. Внешние квадратные скобки сообщают нам, что в результате будет создан список.</p> <p>Наличие команды int свидетельствует, что это будут целые числа. for i in input означают, что циклический ввод данных закончится командой enter.</p> <p>А split – это команда разделитель информации, которая по умолчанию использует для этого ПРОБЕЛ.</p>	 <p>Сцена 22</p>	

	<p>В итоге мы получаем список, состоящий из определенного количества целых чисел. И наша следующая задача - проверить, чтобы было введено именно четыре (а не больше или меньше) числа, и чтобы эти числа были разные.</p> <p>Для этого мы воспользуемся оператором условия, а также представим, что наш список вдруг стал множеством.</p> <p>Нужно проверить</p> <ul style="list-style-type: none">• введено ровно 4 числа.• введены разные числа	<div><div>Игра «Быки и коровы». Формирование данных</div><div><pre>my_counts = [int(i) for i in input().split()]</pre></div><div><table><tr><td>Получаем</td><td>Список из целых чисел</td></tr><tr><td>Нужно проверить</td><td><ul style="list-style-type: none">• введено ровно 4 числа• введены разные числа</td></tr></table></div></div> <div>Сцена 23</div>	Получаем	Список из целых чисел	Нужно проверить	<ul style="list-style-type: none">• введено ровно 4 числа• введены разные числа	
Получаем	Список из целых чисел						
Нужно проверить	<ul style="list-style-type: none">• введено ровно 4 числа• введены разные числа						
<p>Блок заданий.</p> <p>Практические задания:</p> <p>Задание 1</p> <p>Окончание</p>	<p><i>После окончания дикторского текста запускается таймер на 3 мин.</i></p> <p>Задание 1. Алгоритм «Быки и коровы». Окончание</p> <p><i>*(Внутри цикла)</i></p> <ul style="list-style-type: none">o Если длина (set(my_counts)) != 4<ul style="list-style-type: none">▪ Напечатать «Не корректный ввод»o Иначе:<ul style="list-style-type: none">▪ bulls = 0▪ cows = 0o # продолжение кода начнется здесь	<div><div>Игра «Быки и коровы». Формирование данных</div><div><div>Алгоритм «Быки и коровы». Окончание</div><div><pre>[внутри цикла] □ Если длина (set(my_counts)) != 4 • Напечатать «Некорректный ввод» □ Иначе: • bulls = 0 • cows = 0 • # продолжение кода начнется здесь</pre></div></div><div></div></div> <div>Сцена 24</div>	24 25				
	<p>Разбор задание 1 (продолжение). Код программы будет выглядеть так:</p> <pre>if len(set(my_counts)) != 4: print("Некорректный ввод") else: bulls = 0 cows = 0 # продолжение кода начнется здесь</pre>	<div><div>Игра «Быки и коровы». Формирование данных</div><div><div>Алгоритм «Быки и коровы». Окончание</div><div><pre>[внутри цикла] □ Если длина (set(my_counts)) != 4 • Напечатать «Некорректный ввод» □ Иначе: • bulls = 0 • cows = 0 • # продолжение кода начнется здесь</pre></div></div><div><pre>my_counts = set() while len(my_counts) < 4: counts, no_transfer = 0, 0 counts = set(my_counts) shuffle(counts) print(counts) ste = 0 while True: str = '' print(f'Введите 4 цифры') my_counts = [int(i) for i in input().split()] if len(my_counts) != 4: print("Некорректный ввод") else: bulls = 0 cows = 0 # продолжение кода начнется здесь</pre></div></div> <div>Сцена 25</div>					

<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>Мы узнали:</p> <ul style="list-style-type: none"> ▪ Создали первую часть кода игры «Быки и коровы» ▪ Использовали циклы, множества, списки, условные операторы и строку-генератор. <p><i>Преподаватель дает ученикам домашнее задание к следующему занятию (Приложение 2).</i></p>	 <p>Сцена 26</p>	<p>26</p>
--	---	---	-----------

Приложение 2

Домашнее задание

Поиграйте в игру «Быки и Коровы» с друзьями, родителями, чтобы лучше понять механику и стратегию игры.

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

Практика

Проект «Словарик программиста. Пропуски»

Цель проекта: сделать игру, в которой нужно будет отгадать зашифрованное слово.

Создайте список терминов, которые вы освоили на этом курсе. Случайным образом выберите одно слово из этого списка. Выведите на экран буквы слова, пропуская все четные. На месте четных букв поставьте знак подчеркивания.

Попросите пользователя отгадать зашифрованное слово. Продолжайте до тех пор, пока слово не будет отгадано или пользователь не сдастся.

Вы можете усовершенствовать проект, сделав его многоразовым: чтобы пользователю поочередно предлагалось угадать несколько слов из списка.

Приложение 3

Краткие организационно-методические рекомендации по организации работы на занятии

«Игра «Быки и коровы». Формирование данных».

В начале занятия обходимо повторить материал по теме «Множества». Лучше это делать в сравнении со списками. Для наглядности можно использовать доску, на которой в виде таблицы отмечать общие и различные методы свойства.

Возвращаясь к домашнему заданию, можно предоставить ребятам данные по теории вероятности о выигрыше в данную лотерею. Для лотереи «5 из 36» вероятности, всегда следующие:

☐

угадать два числа — 1 : 8

☐

угадать три числа — 1 : 81

☐

угадать четыре числа — 1 : 2 432

☐

угадать пять чисел — 1 : 376 992

Если позволит время – можно вспомнить код с прошлого урока, в котором создавалось множество из чисел. Данный код нам пригодится.

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 объяснения правил игры, рекомендую поставить ролик на паузу, и сыграть в нее с учениками. Упор желательно сделать на то, что здесь мы, при отгадывании, имеем дело с разрядами числа, но запрограммировать подобный код будет проблематично. Поэтому четырехзначное число заменено на список из четырех чисел. Тема «генераторы» в курсе будет раскрыта позднее. Поэтому пока можно ограничиться информацией из ролика, но если останется время и уровень любознательности ребят выше среднего, то можно привести им еще другие примеры генераторов (самый простой — это сплит-строка). Названия списков и переменных носит рекомендательный характер. Рекомендую на доске написать блок схему проекта, чтобы понимать, какой раздел в данный момент выполняется. Не забудьте сохранить и запомнить название проекта. Следующий урок будет продолжением его.