



Методические рекомендации по теме

«Работа со списками, выбор случайного элемента. Игра «Предсказание»»

Цель:

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

Задачи:

- практика применения списков при программировании игр в Python;
- понимание списка 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 списков в Python при разработке игр.

Метапредметные: обучающиеся получают возможность владения общепредметными понятиями «список», «индекс», «оформление игр»; 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 начнем знакомство с практикой применения списков при программировании игр).

Проблемная дискуссия по вопросам:

- Какие списки вы подготовили к сегодняшнему заданию?
- Составление каких списков вы бы предложили автоматизировать при помощи специальных приложений?
- Какие игры могли бы быть созданы на основании списков?

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

- Использование списков позволяет упростить работу с большими объемами данных
- Списки можно использовать и при разработке компьютерных игр

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.

Списки и работа с ними.

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

Обсуждением вопросов по просмотренным материалам:

- В чем смысл проекта?

- Какова механика работы проекта?
- Как задействованы списки в игре?
- Можно ли объединять случайный выбор и списки в игровых проектах?

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: «Электронный оракул».

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

«Электронный оракул»: включает *практическое задание 1*, которое выполняется в несколько этапов: подключение модулей random, time и организация диалога; установка временной задержки перед ответом; организация базы ответов и выбор случайного элемента из этой базы; формирование предсказания; сделаем проект многократным.

После выполнения задания ученики получают работающий программный продукт – игровая программа позволяет приветствовать и задавать вопросы, выбирать ответы с определенной задержкой по времени, добавлять пожелание и организовывать повторный цикл диалога вопрос-ответ.

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

**см. сцены 9 – 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

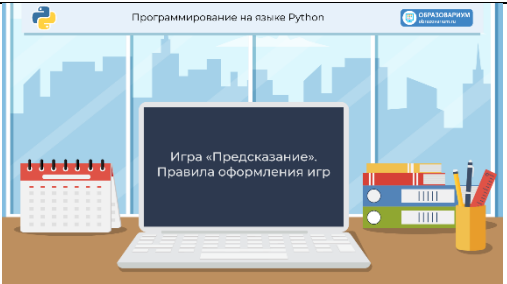
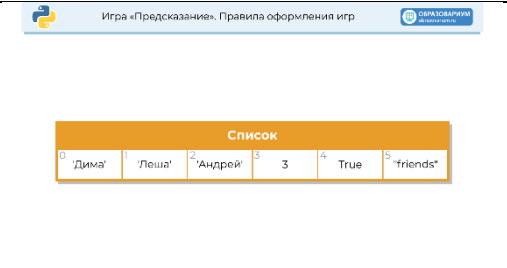
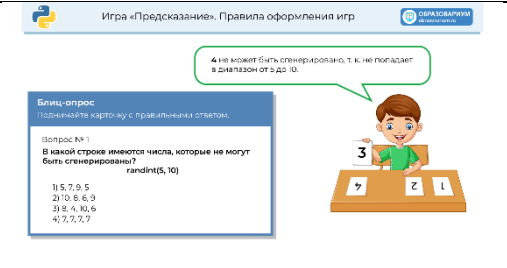
В таблице «Содержание видеоролика» представлены:

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

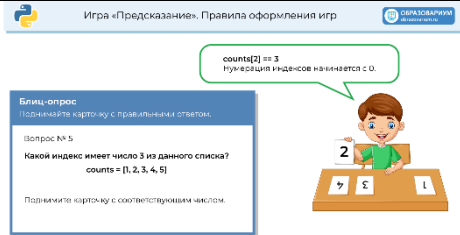
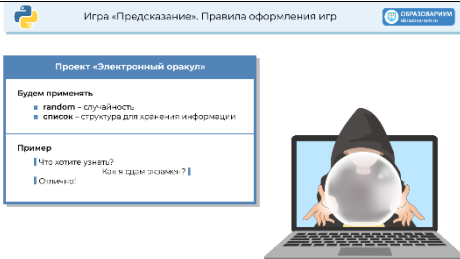
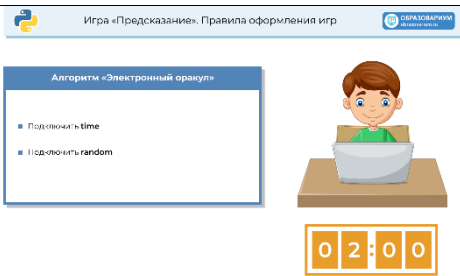
Учитель при подготовке к уроку может ознакомиться с содержанием видеоролика в текстовом формате, при необходимости распечатать фрагменты текста или примеры заданий и задач для использования в работе с учениками. Распечатанные тексты и задания из таблицы также можно применять в качестве раздаточного материала как на уроке, так и для домашних заданий.

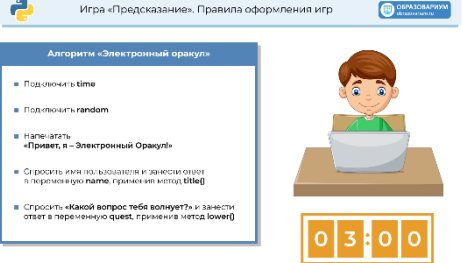
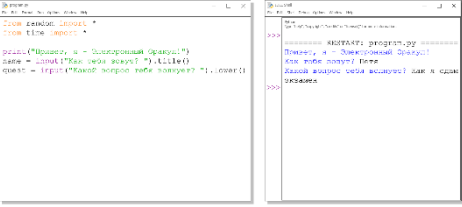
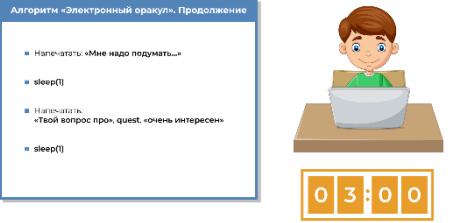
Таблица. Содержание видеоролика

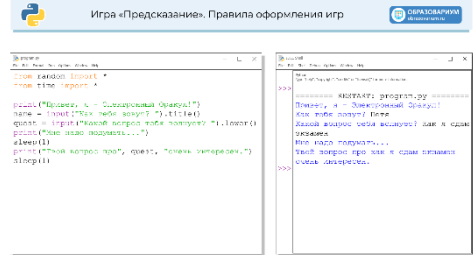
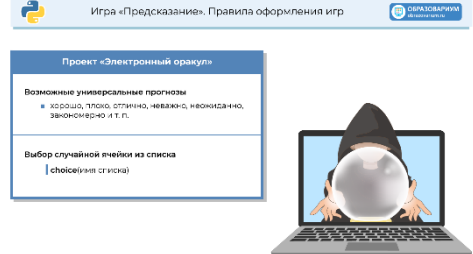
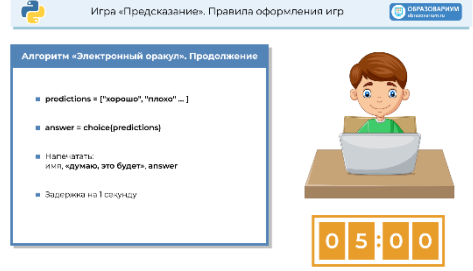
Название блока	Содержание блока и комментарии	Фрагменты из видеоролика	№ сцен
----------------	--------------------------------	--------------------------	--------

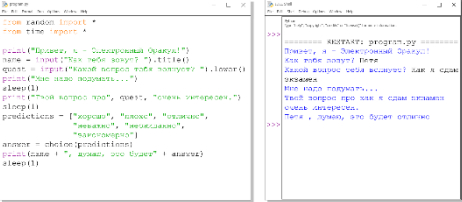
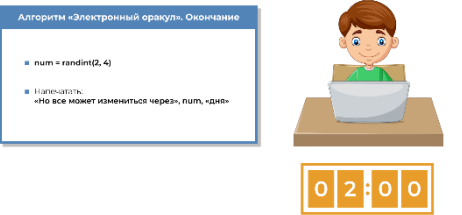
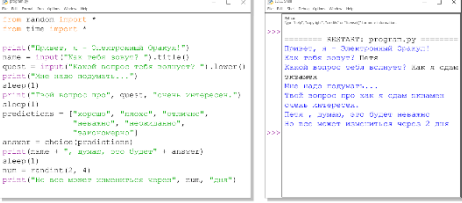
<p>Вводный блок. Мы узнаем</p>	<p>Обозначаем ученикам тему и цели урока.</p> <p>Работа со списками, выбор случайного элемента. Игра «Предсказание»</p>	 <p>Сцена 1</p>	<p>1 2</p>
	<p>На прошлом занятии мы познакомились с таким объектом для хранения информации, как список – изменяемой и индексируемой структурой. В отличие от переменной, в список можно занести много различных данных и получать к ним доступ при помощи индекса. На этом занятии, в рамках игрового проекта, мы продолжим знакомство с этой замечательной структурой.</p>	 <p>Сцена 2</p>	
<p>Блок повторения. Блиц-опрос</p>	<p>Повторение материала предыдущего урока; на столе имеются пронумерованные карточки; после каждого вопроса выбираем ту, номер которой, совпадает с правильным ответом.</p> <p>Первый вопрос. В какой строке имеются числа, которые не могут быть сгенерированы <code>randint(5, 10)</code></p> <ol style="list-style-type: none"> 1) 5, 7, 9, 5 2) 10, 8, 6, 9 3) 8, 4, 10, 6 4) 7, 7, 7, 7 <p>Ответ 3. 4 не может быть сгенерировано, т. к. попадает в диапазон (5, 10).</p>	 <p>Сцена 3</p>	<p>3 4 5 6 7</p>

	<p>Второй вопрос. Какое последнее значение будет у счетчика? Поднимите карточку с номером ответа.</p> <p>for i in range (4, 0)</p> <p>Ответ <u> </u>. <i>Правильный ответ отсутствует – цикл не работает..</i></p>		
	<p>Третий вопрос. Какая команда работает не только со списками?</p> <ol style="list-style-type: none"> 1) append 2) len 3) sort 4) reverse <p>Ответ 2. Команда len может считать длину не только списка, но и других структур.</p>	<p>Сцена 4</p>	
	<p>Четвертый вопрос. В какой строке список составлен правильно?</p> <ol style="list-style-type: none"> 1) counts = 1,2,3,4,5 2) counts = (1,2,3,4,5) 3) counts = [1,2,3,4,5] 4) counts = {1,2,3,4,5} <p>Ответ 3. Верна эта запись: counts = [1,2,3,4,5]. Список создается при помощи квадратных скобок.</p>	<p>Сцена 5</p> <p>Сцена 6</p>	

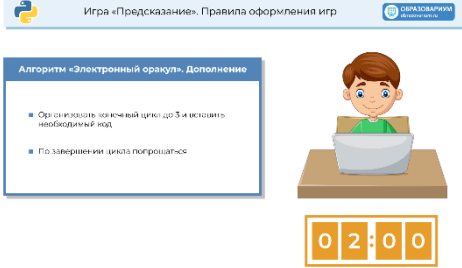
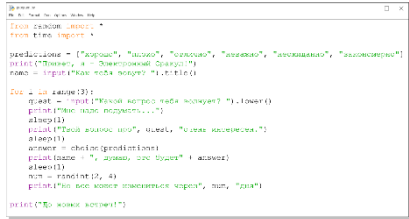
	<p>Пятый вопрос. Какой индекс имеет число 3 из данного списка? <code>counts = [1,2,3,4,5]</code> Поднимите карточку с номером ответа.</p> <p><i>Ответ 2. <code>counts[2] == 3</code>. Нумерация индексов начинается с 0...</i></p>	 <p>Сцена 7</p>	
<p>Теоретический блок.</p> <p>Проект «Электронный оракул»</p>	<p>Наш сегодняшний проект называется «Электронный оракул». Он будет совсем не серьезным с точки зрения прогноза, но позволит нам на практике применить знания по теме random и списки.</p> <ul style="list-style-type: none"> Будем применять random – случайность список – структура для хранения информации <p>Принцип работы его будет следующий: программа запрашивает вопрос и дает на него ответ, который случайным образом выбирается из заранее созданного списка.</p> <ul style="list-style-type: none"> Пример: <ul style="list-style-type: none"> - Что хотите узнать? - Как я сдам экзамен? - Отлично! 	 <p>Сцена 8</p>	8
<p>Блок заданий.</p> <p>Практические задания:</p> <p>Задание 1</p>	<p><i>После окончания дикторского текста запускается таймер на 2 мин.</i></p> <p>Задание 1. Алгоритм «Электронный оракул». Начало</p> <ul style="list-style-type: none"> Подключить random Подключить time <p>random даст нам возможность генерировать случайность, а модуль time* – делать задержку, имитируя размышление программы над вопросом.</p>	 <p>Сцена 9</p>	9 10 11

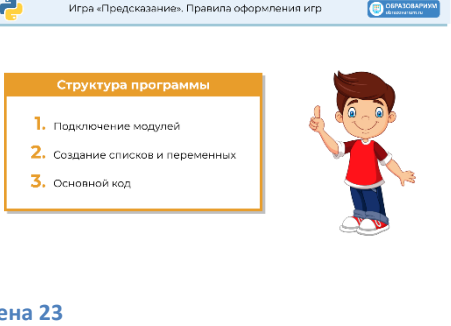
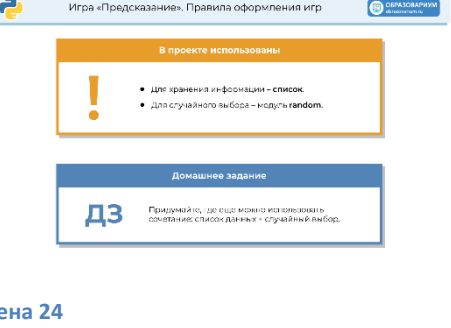
	<p>Далее попросим оракула представиться, спросить наше имя и интересующий вопрос. <i>После окончания дикторского текста запускается таймер на 3 мин.</i></p> <ul style="list-style-type: none"> • Напечатать «Привет, я – Электронный Оракул!» • Спросить имя пользователя и занести ответ в переменную name, применив метод title() • Спросить: «Какой вопрос тебя волнует?» и занести ответ в переменную quest, применив метод lower() 	 <p>Сцена 10</p>	
	<p>Разбор задания 1. Ваш код может выглядеть так.</p> <pre>from random import * from time import * print("Привет, я - Электронный Оракул!") name = input("Как тебя зовут? ").title() quest = input("Какой вопрос тебя волнует? ").lower()</pre> <p><i>В результате сделанного кода у нас получается небольшой диалог между программой и пользователем. Давайте продолжим его.</i></p>	 <p>Сцена 11</p>	
<p>Задание 1. продолжение</p>	<p><i>После окончания дикторского текста запускается таймер на 3 мин.</i></p> <p>Задание 1. Алгоритм «Электронный оракул» (продолжение)</p> <ul style="list-style-type: none"> • Напечатать: «Мне надо подумать...» • sleep(1) • Напечатать: «Твой вопрос про», quest, «очень интересен» • sleep(1) 	 <p>Сцена 12</p>	<p>12 13</p>

	<p>Разбор задания 1. Ваш код может выглядеть так.</p> <p>... дописывается к существующему коду</p> <pre>print("Мне надо подумать...") sleep(1) print("Твой вопрос про", quest, "очень интересен.") sleep(1)</pre> <p><i>Конечно, можно обойтись без поясняющей информации, однако в противном случае программа станет похожа на бездушный автомат.</i></p>	 <p>Сцена 13</p>	
<p>Задание 1.</p> <p>продолжение</p>	<p>Продолжим и организуем сами предсказания.</p> <p>Вначале сформируем перечень возможных универсальных прогнозов: хорошо, плохо, отлично, не важно, неожиданно, закономерно.</p> <ul style="list-style-type: none"> Возможные универсальные прогнозы: хорошо, плохо, отлично, не важно, неожиданно, закономерно и т.п. <p>И используем команду, которая выбирает из списка случайную ячейку.</p> <ul style="list-style-type: none"> Выбор случайной ячейки из списка <pre>choice(имя списка)</pre>	 <p>Сцена 14</p>	<p>14</p> <p>15</p> <p>16</p>
	<p><i>После окончания дикторского текста запускается таймер на 5 мин.</i></p> <p>Задание 1. Алгоритм «Электронный оракул» (продолжение)</p> <ul style="list-style-type: none"> <code>predictions = ["хорошо", "отлично"...]</code> <code>answer = choice(predictions)</code> Напечатать: имя, «думаю это будет», answer Задержка на 1 секунду 	 <p>Сцена 15</p>	

	<p>Разбор задания 1 (продолжение). Ваш код может выглядеть так.</p> <p>... дописывается к существующему коду</p> <pre>predictions = ["хорошо", "отлично", "без неожиданностей", "плохо", "не стабильно", "неважно"] answer = choice(predictions) print(name + ", думаю это будет " + answer) sleep(1)</pre> <p>Если все верно, нам будет выдаваться некий прогноз, на основе тех фраз, которые занесены в наш список.</p>	 <p>Сцена 16</p>	
<p>Задание 1. окончание</p>	<p>После окончания дикторского текста запускается таймер на 2 мин.</p> <p>Задание 1. Алгоритм «Электронный оракул» (окончание)</p> <ul style="list-style-type: none"> num = randint(2, 4) Напечатать: «через», num, «дня все может измениться» <p>Сделаем оракула менее конкретным. Для этого сгенерируем при помощи команды randint целое число в диапазоне от двух до четырех и занесем его в переменную num.</p>	 <p>Сцена 17</p>	<p>17 18</p>
	<p>Разбор задание 1(окончание). Код программы будет выглядеть так:</p> <p>... дописывается к существующему коду</p> <pre>m = randint(2, 4) print("Но все может измениться через", num, "дня")</pre> <p>Мы можем применить в команде randint и другие числа, однако тогда у нас возникнут проблемы со склонением слова «день»</p>	 <p>Сцена 18</p>	

<p>дополнительно</p>	<p>При желании, мы можем сделать наш проект многоразовым: при помощи конечного цикла или цикла с условием.</p> <p>Если делать проект многоразовым – какие строчки не должны попасть в цикл?</p>	<div data-bbox="1489 228 1964 499"><div><div>Игра «Предсказание». Правила оформления игр</div><div><div>Многоразовый проект</div><div>Если делать проект многоразовым, какие строчки кода не нужно оставлять в цикле?</div></div><div>00:15</div></div><div data-bbox="1489 534 1579 563">Сцена 19</div></div>	<p>19 20</p>
	<p>Давайте разберемся.</p> <p>В цикле не нужны первые четыре строчки: там, где подключаются модули, печатается приветственная строка и запрашивается имя пользователя</p>	<div data-bbox="1489 761 1964 1032"><div><div>Игра «Предсказание». Правила оформления игр</div><div><div>Многоразовый проект</div><div>Если делать проект многоразовым, какие строчки кода не нужно оставлять в цикле?</div></div><div></div></div><div data-bbox="1489 1067 1579 1096">Сцена 20</div></div>	

<p>Задание 1.</p> <p>дополнение</p>	<p>Измените код, чтобы можно было задать три вопроса, после чего программа попрощается с нами.</p> <p><i>После окончания дикторского текста запускается таймер на 2 мин.</i></p> <p>Задание 1. Алгоритм «Электронный оракул» (дополнение)</p> <ul style="list-style-type: none"> Организуйте конечный цикл до 3 Вставьте необходимый код По завершении – попрощаться 	 <p>Сцена 21</p>	<p>21 22</p>
	<p>Разбор задание 1(с дополнением). Код программы будет выглядеть так:</p> <pre> from random import * from time import * predictions = ["хорошо", "отлично", "без неожиданностей", "плохо", "не стабильно", "неважно", "как всегда"] print("Привет, я - Электронный Оракул!") name = input("Как тебя зовут? ").title() for i in range(3): quest = input("Какой вопрос тебя волнует? ").lower() print("Мне надо подумать...") sleep(1) print("Твой вопрос про", quest, "очень интересен") sleep(1) answer = choice(predictions) print(name + ", думаю это будет " + answer) sleep(1) num = randint(2, 4) print("Но все может измениться через", num, "дня") print("До новых встреч!") </pre>	 <p>Сцена 22</p>	

Запомни	<p>В дальнейшем, мы всегда будем так поступать – создавать списки и переменные в начале программы, после подключения модулей.</p>		23
<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 в проекте использовали:</p> <ul style="list-style-type: none"> ▪ Для хранения информации – список. ▪ Для случайного выбора – модуль random. <p><i>Преподаватель дает ученикам домашнее задание к следующему занятию (Приложение 2).</i></p>		24

Подумайте и предложите варианты в каких игровых проектах целесообразно использовать связку случайного выбора и списков.

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

Практика

Проект «Жеребьевка команд»

Создайте список из четного количества футбольных команд. Случайным образом составьте пары команд и выведите их на экран.

Проект «Обязанности на день в походе»

Создайте два списка: первый список с именами ребят, которые пошли в поход, второй список обязанностей. Количество элементов в этих списках должно совпадать.

Случайным образом распределите обязанности между ребятами и выведите на экран имя каждого участника похода и его обязанность.

Приложение 3

Краткие организационно-методические рекомендации по организации работы на занятии

«Работа со списками, выбор случайного элемента. Игра «Предсказание»».

В начале занятия необходимо повторить материал по теме «Списки». Что это такое, какими свойствами обладает, синтаксис, методы, способы вывод на экран. Не лишним будет напомнить такой термин как «индекс», который также как и в срезах, начинается с нуля. Также поинтересоваться какие ребята придумали примеры. Самые простые – список продуктов для магазина, список участников соревнований, классный журнал с фамилиями учеников.

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.

На занятии будет создан только один проект. Ввиду большого размера – процесс написания кода разбит на этапы. После каждого этапа рекомендуется проверять код на отсутствие ошибок. Следует иметь в виду, что все фразы проекта носят только рекомендательный характер. Будет хорошо, если ребята самостоятельно придумают им аналоги. При разборе прогнозов можно поставить ролик на паузу и записать на доске варианты фраз, которые можно будет занести в список. Желательно не ограничиваться предложенными вариантами, а дополнить список собственными. Дополнение проекта можно сделать также при помощи цикла с условием, организовав еще один вопрос: «хотите закончить программу?».