



Методические рекомендации по теме «Типовые алгоритмы обработки числовых массивов»

Цель:

- дать представление о прикладном применении массивов в языке Python.

Задачи:

- практика применения массивов в Python;
- изучение различных операций при работе с массивами в Python;
- 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 о прикладном применении и основных операциях с массивами в Python.

Метапредметные: обучающиеся получают возможность владения общепредметными понятиями «массив», «список», «класс массивов»; 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.

2. Вводный блок.

Тема.

Преподаватель при необходимости останавливая трансляцию, комментируя дополнительно тему занятия.

Проблемная дискуссия по вопросам:

- Что такое индексация списка?
- Как работает индексация списка при помощи цикла?
- Какие операции при работе с массивами вы можете назвать?
- Как вы думаете, какие еще операции с массивами и данными массива могут быть полезны? Для каких прикладных задач?

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

- Список операций, которые, по мнению учеников, могут быть полезны при работе с массивами
- Прикладные задачи, которые могут быть решены с использованием массивов

**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.

Работа с массивом.

Новый материал в занятии не выделяется в отдельный теоретический блок. Занятие по своему формату является практикумом и направлено на решение прикладных программных задач. Новый материал изучается в теоретических вставках при постановке и разборе практических заданий.

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.

«Числовой массив».

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

«Числовой массив»: включает *блок практических заданий*, которые позволят провести прикладные разработки типовых задач для работы с массивами:

- Создание массива на 10 ячеек
- Четные и нечетные индексы
- Четные числа
- Уникальные элементы
- Число больше предыдущего
- Генератор и массив

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

**см. сцены 9 – 24.*

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 25*

Приложение 1

Сценарный план видеоролика

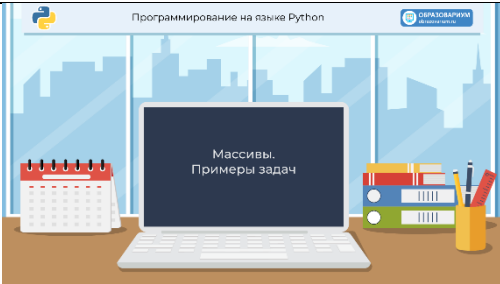
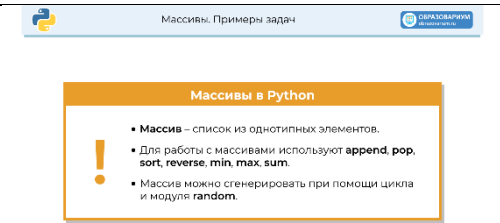
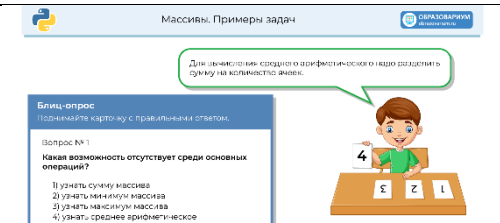
В таблице «Содержание видеоролика» представлены:

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

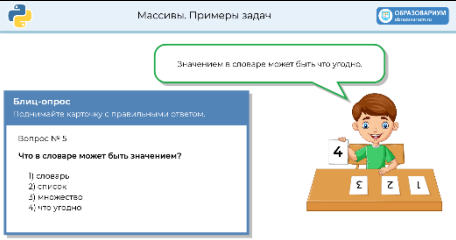

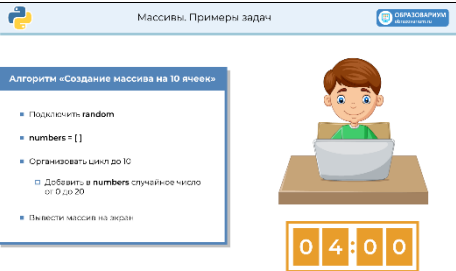
Учитель при подготовке к уроку может ознакомиться с содержанием видеоролика в текстовом формате, при необходимости распечатать фрагменты текста или примеры заданий и задач для использования в работе с учениками. Распечатанные тексты и задания из таблицы также можно применять в качестве раздаточного материала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

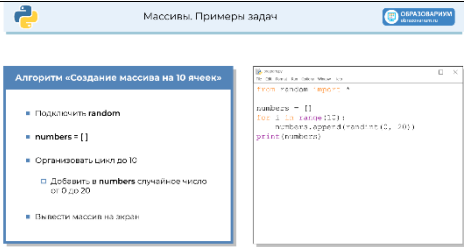
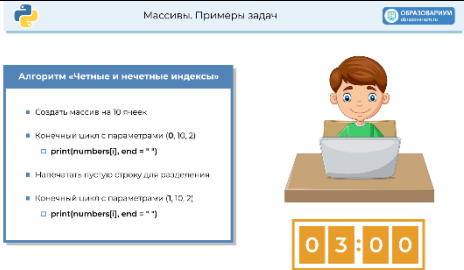
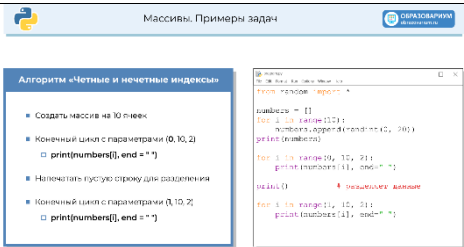
Таблица. Содержание видеоролика


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

<p>Вводный блок. Мы узнаем</p>	<p><i>Обозначаем ученикам тему и цели урока.</i></p> <p>Типовые алгоритмы обработки числовых массивов</p>	 <p>Сцена 1</p>	<p>1 2</p>
	<p>На прошлом уроке мы выяснили, что в Python массивом называется список, составленный из однотипных элементов. Например: из целых чисел, из дробных чисел, из символов. Основные команды для работы с массивом нам известны: append, pop, sort, reverse, min, max, sum. Также мы научились генерировать массивы при помощи цикла и random. Эти знания нам пригодятся для решения различных задач, связанных с анализом большого количества данных.</p>	 <p>Сцена 2</p>	
<p>Блок повторения. Блиц-опрос</p>	<p><i>Повторение материала предыдущего урока; на столе имеются пронумерованные карточки; после каждого вопроса выбираем ту, номер которой, совпадает с правильным ответом.</i></p> <p>Первый вопрос. Какая возможность отсутствует среди основных операций?</p> <ol style="list-style-type: none"> 1) Узнать сумму массива 2) Узнать минимум массива 3) Узнать максимум массива 4) Узнать среднее арифметическое <p><i>Ответ 4. Для вычисления среднего арифметического надо разделить сумму на количество ячеек.</i></p>	 <p>Сцена 3</p>	<p>3 4 5 6 7</p>

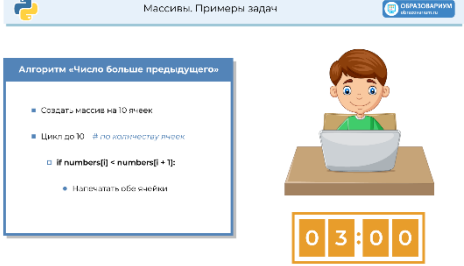
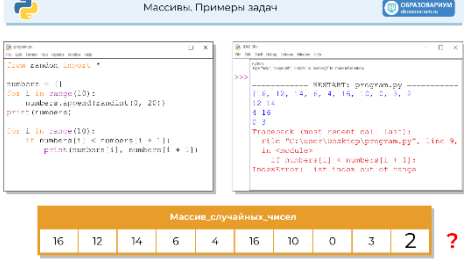
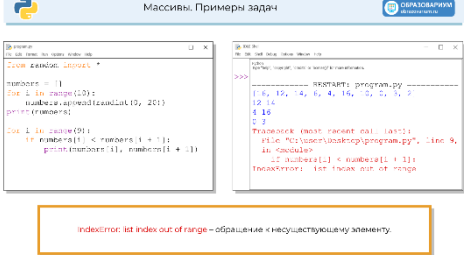
	<p>Второй вопрос. Куда добавляет данные команда append?</p> <ol style="list-style-type: none"> 1) В начало массива 2) В середину массива 3) В конец массива 4) В случайное место <p><i>Ответ 3. <code>имя_массива.append(данные)</code> добавляет данные в конец массива..</i></p>	<p>Массивы. Примеры задач</p> <p>имя_массива.append(данные) добавляет данные в конец массива.</p> <p>Блиц-опрос Подберите карту(ю) с правильным ответом.</p> <p>Вопрос № 2 Куда добавляет данные команда <code>append</code>?</p> <ol style="list-style-type: none"> 1) в начало массива 2) в середину массива 3) в конец массива 4) в случайное место <p>3</p>	
	<p>Третий вопрос. У какой команды неправильное описание?</p> <p><code>pop</code> – Удаляет один элемент <code>clear</code> – Удаляет все элементы <code>sort</code> – Сортирует по возрастанию <code>reverse</code> – Сортирует по убыванию</p> <p><i>Ответ 4. <code>reverse</code> разворачивает последовательность, т. е. возвращает элементы в обратном порядке, но не сортирует.</i></p>	<p>Массивы. Примеры задач</p> <p><code>reverse</code> возвращает элементы в обратном порядке, но не сортирует.</p> <p>Блиц-опрос Подберите карту(ю) с правильным ответом.</p> <p>Вопрос № 3 У какой команды неправильное описание?</p> <ol style="list-style-type: none"> 1) <code>pop</code> – удаляет один элемент 2) <code>clear</code> – удаляет все элементы 3) <code>sort</code> – сортирует по возрастанию 4) <code>reverse</code> – сортирует по убыванию <p>4</p>	<p>Сцена 4</p>
	<p>Четвертый вопрос. Чем похожи множества и массивы?</p> <ol style="list-style-type: none"> 1) индексом 2) именем 3) типами данных 4) скобками <p><i>Ответ 2. Правила формирования имени для множеств и массивов одинаковы.</i></p>	<p>Массивы. Примеры задач</p> <p>Правила формирования имени для множеств и массивов одинаковы.</p> <p>Блиц-опрос Подберите карту(ю) с правильным ответом.</p> <p>Вопрос № 4 Чем похожи множества и массивы?</p> <ol style="list-style-type: none"> 1) индексом 2) именем 3) типами данных 4) скобками <p>2</p>	<p>Сцена 5</p> <p>Сцена 6</p>

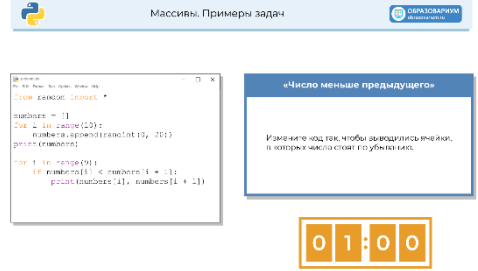
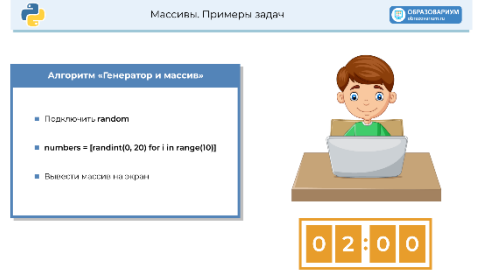
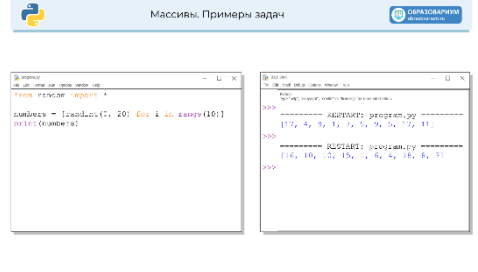
	<p>Пятый вопрос. Что в словаре может быть значением?</p> <ol style="list-style-type: none"> 1) Словарь 2) Список 3) Множество 4) Что угодно. <p><i>Ответ 4. Значением в словаре может быть что угодно.</i></p>	 <p>Сцена 7</p>	
<p>Теоретический блок.</p> <p>Создание массива</p>	<p>Для работы нам понадобится массив из случайных чисел. Поэтому сначала вспомним как его создавать, а потом будем использовать данный код во всех сегодняшних проектах.</p> <p>Договоримся, что у нас будет массив из десяти ячеек в диапазоне от нуля до двадцати.</p> <p style="text-align: center;">Создание массива случайных чисел</p> <p>Будем использовать во всех проектах занятия.</p> <p>Размер массива – 10 ячеек.</p> <p>Элементы массива – целые числа от 0 до 20.</p>	 <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 4 мин.</i></p> <p>Задание 1. Алгоритм «Создание массива на 10 ячеек»</p> <ul style="list-style-type: none"> ■ Подключить random ■ <code>numbers = []</code> ■ Цикл до 10 <ul style="list-style-type: none"> ○ Добавить в <code>numbers</code> случайное число от 0 до 20 ■ Вывести массив на экран 	 <p>Сцена 9</p>	9 10 11 12

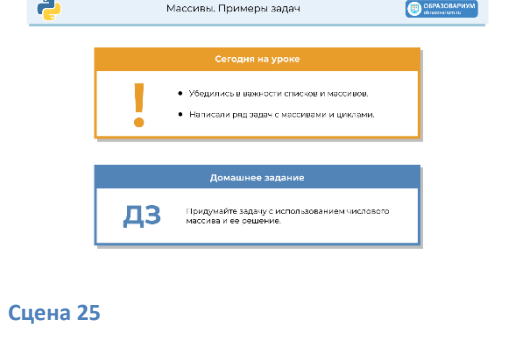
	<p>Разбор задания 1. Ваш код может выглядеть так.</p> <pre>from random import * numbers = [] for i in range(10): numbers.append(randint(0, 20)) print(numbers)</pre>	 <p>Массивы. Примеры задач</p> <p>Алгоритм «Создание массива на 10 ячеек»</p> <ul style="list-style-type: none"> Подключить random numbers = [] Организовать цикл до 10 <ul style="list-style-type: none"> Добавить в numbers случайное число от 0 до 20 Вывести массив на экран <pre>numbers = [] for i in range(10): numbers.append(randint(0, 20)) print(numbers)</pre> <p>Сцена 10</p>	
	<p><i>После окончания дикторского текста запускается таймер на 3 мин.</i></p> <p>Задание 1. Алгоритм «Четные и нечетные индексы»</p> <ul style="list-style-type: none"> Создать массив на 10 ячеек Цикл (0, 10, 2) <ul style="list-style-type: none"> print(numbers[i], end = " ") Напечатать пустую строку для разделения Цикл (1, 10, 2) <ul style="list-style-type: none"> print(numbers[i], end = " ") 	 <p>Массивы. Примеры задач</p> <p>Алгоритм «Четные и нечетные индексы»</p> <ul style="list-style-type: none"> Создать массив на 10 ячеек Конечный цикл с параметрами (0, 10, 2) <ul style="list-style-type: none"> print(numbers[i], end = " ") Напечатать пустую строку для разделения Конечный цикл с параметрами (1, 10, 2) <ul style="list-style-type: none"> print(numbers[i], end = " ") <p>03:00</p> <p>Сцена 11</p>	
	<p>Разбор задания 1. Ваш код может выглядеть так.</p> <p><i>... дописывается к существующему коду</i></p> <pre>for i in range(0, 10, 2): print(numbers[i], end=" ") print() # разделяет данные for i in range(1, 10, 2): print(numbers[i], end=" ")</pre>	 <p>Массивы. Примеры задач</p> <p>Алгоритм «Четные и нечетные индексы»</p> <ul style="list-style-type: none"> Создать массив на 10 ячеек Конечный цикл с параметрами (0, 10, 2) <ul style="list-style-type: none"> print(numbers[i], end = " ") Напечатать пустую строку для разделения Конечный цикл с параметрами (1, 10, 2) <ul style="list-style-type: none"> print(numbers[i], end = " ") <pre>numbers = [] for i in range(10): numbers.append(randint(0, 20)) print(numbers) for i in range(0, 10, 2): print(numbers[i], end=" ") print() for i in range(1, 10, 2): print(numbers[i], end=" ")</pre> <p>Сцена 12</p>	

<p>Блок заданий. Практические задания: Задание 2</p>	<p>После окончания дикторского текста запускается таймер на 3 мин.</p> <p>Задание 2. Алгоритм «Четные числа»</p> <ul style="list-style-type: none"> Создать массив на 10 ячеек Цикл на 10 # по количеству ячеек <ul style="list-style-type: none"> Если ячейка с индексом счетчик $\% 2 == 0$ <ul style="list-style-type: none"> Напечатать ячейку 	<div> <div>Массивы. Примеры задач</div> <div> <div>Алгоритм «Четные числа»</div> <ul style="list-style-type: none"> Создать массив на 10 ячеек Цикл до 10 # по количеству ячеек <ul style="list-style-type: none"> Если ячейка с индексом счетчик $\% 2$ равна 0 <ul style="list-style-type: none"> Напечатать ячейку </div> <div>  <div>03:00</div> </div> </div> <p>Сцена 13</p>	<p>13 14</p>
	<p>Разбор задание 2. Продолжение. Код программы будет выглядеть так: ... <i>дописывается к существующему коду</i></p> <pre>for i in range(10): if numbers[i] % 2 == 0: print(numbers[i], end=" ")</pre>	<div> <div>Массивы. Примеры задач</div> <div> <div>Алгоритм «Четные числа»</div> <ul style="list-style-type: none"> Создать массив на 10 ячеек Цикл до 10 # по количеству ячеек <ul style="list-style-type: none"> Если ячейка с индексом счетчик $\% 2$ равна 0 <ul style="list-style-type: none"> Напечатать ячейку </div> <div> <pre> numbers = [] for i in range(10): numbers.append(random.randint(0, 20)) print(numbers) for i in range(10): if numbers[i] % 2 == 0: print(numbers[i], end=" ") </pre> </div> </div> <p>Сцена 14</p>	
<p>Теоретический блок</p>	<p>Следующий проект будет посвящен уникальным элементам массива. Да, мы знаем, что при помощи множества, можно убрать из массива повторяющиеся элементы. Но как быть, если мы узнать, какие из них не повторялись?</p> <p>В этом нам поможет метод count. Он покажет сколько раз искомый элемент встречается в массиве.</p> <p>имя_массива.count(элемент) == сколько раз элемент встречается в массиве</p>	<div> <div>Массивы. Примеры задач</div> <div> <div>Массив случайных чисел</div> <div>16 10 11 12 7 11 16 6 7 14</div> </div> <div> <div>Количество вхождений элемента</div> <div>имя_массива.count(элемент) == сколько раз элемент встречается в массиве</div> </div> </div> <p>Сцена 15</p>	<p>15</p>

<div>Блок заданий.</div> <div>Практические задания:</div> <div>Задание 3</div>	<div>После окончания дикторского текста запускается таймер на 4 мин.</div> <div>Задание 3. Алгоритм «Уникальные элементы»</div> <div><div><div>■ Создать массив на 10 ячеек</div><div>■ Цикл на 10 # по количеству ячеек<div><div>○ if numbers.count(numbers[i]) == 1:</div><div><div>• вывести ячейку на печать</div></div></div></div></div></div>	<div><div><div>Массивы. Примеры задач</div><div><div>Алгоритм «Уникальные элементы»</div><div><div>■ Создать массив на 10 ячеек</div><div>■ Цикл до 10 # по количеству ячеек<div><div>□ if numbers.count(numbers[i]) == 1:</div><div><div>• Вывести ячейку на печать</div></div></div></div></div><div><div>03:00</div></div></div></div><div>Сцена 16</div></div>	<div>16</div> <div>17</div>
	<div>Разбор задание 3. Код программы будет выглядеть так: ... дописывается к существующему коду for i in range(10): if numbers.count(numbers[i]) == 1: print(numbers[i], end=" ")</div>	<div><div><div>Массивы. Примеры задач</div><div><div>Алгоритм «Уникальные элементы»</div><div><div>■ Создать массив на 10 ячеек</div><div>■ Цикл до 10 # по количеству ячеек<div><div>□ if numbers.count(numbers[i]) == 1:</div><div><div>• Вывести ячейку на печать</div></div></div></div></div><div><pre>from random import randint n = 10 numbers = [] for i in range(n): numbers.append(randint(1, 20)) print(numbers) for i in range(n): if numbers.count(numbers[i]) == 1: print(numbers[i], end=" ")</pre></div></div></div><div>Сцена 17</div></div>	<div>18</div>
<div>Теоретический блок</div>	<div>В завершении этого блока, найдем ячейки, в которых числа стоят по возрастанию (но не обязательно по порядку).</div> <div>Нам снова понадобится цикл, в котором мы будем сравнивать содержимое ячеек с текущим индексом со следующей ячейкой (у которой индекс на 1 больше)</div>	<div><div><div>Массивы. Примеры задач</div><div><div>Массив_случайных_чисел</div><div><div>16101112711166714</div></div><div><div>Следующие больше предыдущего</div><div><div><div>10111116</div><div>111267</div><div>711714</div></div></div></div></div></div><div>Сцена 18</div></div>	

<p>Блок заданий. Практические задания: Задание 4</p>	<p><i>После окончания дикторского текста запускается таймер на 3 мин.</i></p> <p>Задание 4. Алгоритм «Число больше предыдущего»</p> <ul style="list-style-type: none"> ■ Создать массив на 10 ячеек ■ Цикл на 10 # по количеству ячеек <ul style="list-style-type: none"> ○ if numbers[i] < numbers[i + 1]: <ul style="list-style-type: none"> ● Напечатать обе ячейки 	 <p>Сцена 19</p>	<p>19 20</p>
	<p>Разбор задание 4. Код программы будет выглядеть так: ... <i>дописывается к существующему коду</i></p> <pre>for i in range(10): if numbers[i] < numbers[i + 1]: print(numbers[i], numbers[i + 1])</pre> <p><i>Запускаем программу...</i> <i>И получаем сообщение об ошибке индекса. Что произошло?</i></p>	 <p>Сцена 20</p>	
<p>Теоретический блок</p>	<p>Программа сравнивала ячейку № 0 с ячейкой № 1. Потом первую со второй и так далее. Пока не дошла до последней ячейки, № 9 и ... Ее сравнивать не с чем, но цикл то продолжается! Как избавиться от этой ошибки? Необходимо сделать цикл не до 10, а до 9. Чтобы последняя ячейка уже ни с чем дальше не сравнивалась. Запомните эту ошибку. Она встречается достаточно часто, если мы в цикле пытаемся найти несуществующую ячейку.</p>	 <p>Сцена 21</p>	<p>21</p>

<p>Блок заданий. Практические задания: Задание 5</p>	<p><i>После окончания дикторского текста запускается таймер на 1 мин.</i></p> <p>Задание 5. Алгоритм «Числа меньше предыдущего»</p> <p>Измените код так, чтобы выводились ячейки, в которых числа стоят по убыванию.</p> <p><i>Как вывести числа, которые меньше предыдущего, я думаю вы уже догадались. Изменить знак меньше на больше.</i></p> <p><i>Сделайте это самостоятельно</i></p>	 <p>Массивы. Примеры задач</p> <p>«Число меньше предыдущего»</p> <p>Измените код так, чтобы выводились ячейки, в которых числа стоят по убыванию.</p> <p>01:00</p> <p>Сцена 22</p>	<p>22 23 24</p>
	<p><i>После окончания дикторского текста запускается таймер на 2 мин.</i></p> <p>Задание 6. Алгоритм «Генератор и массив»</p> <ul style="list-style-type: none"> Подключить random numbers = [randint(0, 20) for i in range(10)] Вывести массив на экран 	 <p>Массивы. Примеры задач</p> <p>Алгоритм «Генератор и массив»</p> <ul style="list-style-type: none"> Подключить random numbers = [randint(0, 20) for i in range(10)] Вывести массив на экран <p>02:00</p> <p>Сцена 23</p>	
	<p>Разбор задание 6. Код программы будет выглядеть так:</p> <pre>from random import * numbers = [randint(0, 20) for i in range(10)] print(numbers)</pre>	 <p>Массивы. Примеры задач</p> <pre>from random import * numbers = [randint(0, 20) for i in range(10)] print(numbers)</pre> <p>02:00</p> <p>Сцена 24</p>	

<p>Блок завершения занятия. Рефлексия. Сообщение домашнего задания</p>	<p><i>Завершаем демонстрацией ролика и кратким обобщением материалов занятия.</i></p> <p>Подведем итоги.</p> <p>Мы узнали:</p> <ul style="list-style-type: none"> ▪ Убедились в важности списков и массивов. ▪ Написали ряд задач с массивами и циклами. <p><i>Преподаватель дает ученикам домашнее задание к следующему занятию (Приложение 2).</i></p>		<p>25</p>
--	---	---	-----------

Приложение 2

Домашнее задание

Придумать, свое собственное задание на тему «а что еще можно найти в числовом массиве» и алгоритм его решения.
Задание можно выполнить на компьютере и представить результат и код в виде файла или снимка экрана, или распечатки.

Практика

Проект «Смена соседей»

Запросите у пользователя количество ячеек в массиве и сгенерируйте массив случайных чисел. Выведите массив на экран.

Поменяйте местами значения соседних ячеек: нулевую с первой, вторую с третьей и т. д. Если количество элементов нечетное, то последний элемент остается на своем месте.

Выведите на экран получившийся массив.

Проект «Больше соседей»

Запросите у пользователя количество ячеек в массиве и сгенерируйте массив случайных чисел. Выведите массив на экран.

Выведите на экран те значения массива, которые больше значений соседних значений (и слева, и справа). Первый и последний элементы не учитываются, так как у них недостаточно соседей.

Приложение 3

Краткие организационно-методические рекомендации по организации работы на занятии

«Типовые алгоритмы обработки числовых массивов».

В начале занятия традиционно повторить материал по теме «списки» и методы работы с ними. Также можно лишний раз затронуть индексацию при помощи цикла, поскольку все задания урока будут связаны именно с этим инструментом.

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.

Во всех трех проектах используется единый код, создающий случайным образом числовой массив. Параметры массива и **random** можно настраивать и самостоятельно.

Проект «четные и нечетные индексы» основан на операции деление без целой части.

Проект «количество вхождений» использует новый материал в виде метода count, который выдает число вхождений элемента в массив. Отчасти по этой причине во всех проектах, во избежание путаницы, отсутствует имя count как у переменной, так и у списка.

В проекте «больше предыдущего» обратите внимание на индексацию ячеек. Будет отлично, если ребята до написания кода, сами обратят внимание на невозможность сравнения последней ячейки. Если же нет – то можно остановить ролик и дать им возможность расшифровать ошибку и сделать свои предположения о природе ее появления.