



Методические рекомендации по теме «Массивы и способы работы с ними»

Цель:

- дать представление об использовании массивов в языке Python.

Задачи:

- практика применения массивов в Python;
- соотнесение понятий «список» и «массив» 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 об использовании массивов 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:

- Как бы вы объяснили понятие «массив»?
- Каковы основные признаки «массива»?
- Как можно соотнести «список» и «массив», что у них общего, есть ли различия?
- Как вы думаете, для чего могут использоваться структуры с названием «массив» в программировании?

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

- Массивы и списки практически идентичны.
- Массив содержит элементы только одного типа, а список не имеет таких ограничений

**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.

Словарь для хранения данных справочника.

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

Обсуждением вопросов по просмотренным материалам:

- Что такое список в Python?
- Какие операции мы можем производить со списками?
- Какова основная особенность и отличие классического массива от списка?
- Что такое целочно численный массив?

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 – 10, 22*

5. Блок заданий.

«Работа с массивом».

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

«Работа с массивом»: включает *блок практических заданий*, которые позволят провести прикладные разработки и тестирование основных операций с массивами:

- Создание массива с числами
- Сумма чисел массива
- Среднее арифметическое
- Случайный массив
- Максимальное кратное

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

**см. сцены 11 – 24 (кроме сцен по теории).*

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

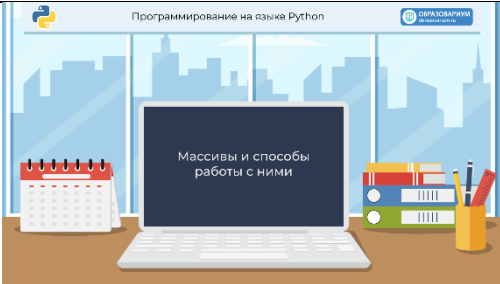
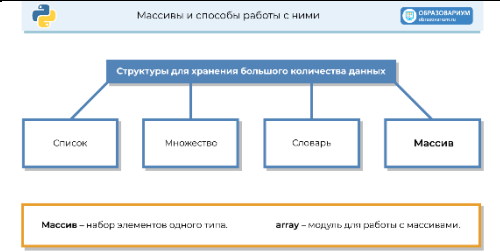
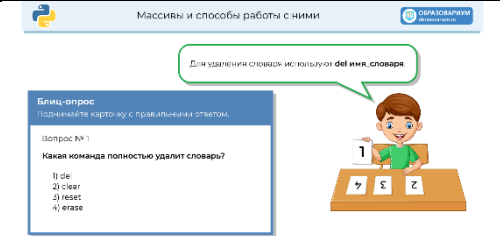
В таблице «Содержание видеоролика» представлены:

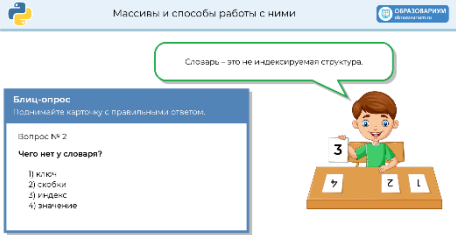
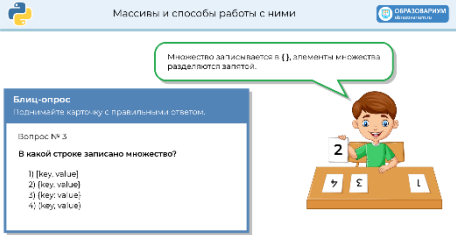
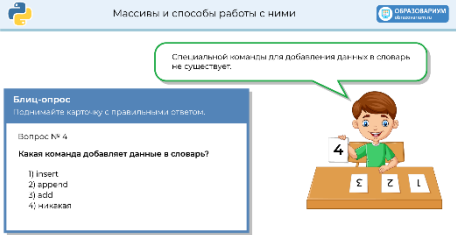
- название блоков видеоролика (тайминг);
- краткое описание содержания в каждом блоке;
- фрагменты из видеоролика, относящиеся к соответствующему блоку;
- номера сцен в каждом блоке.

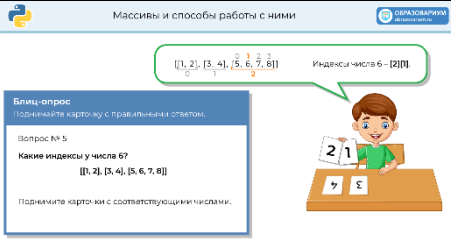
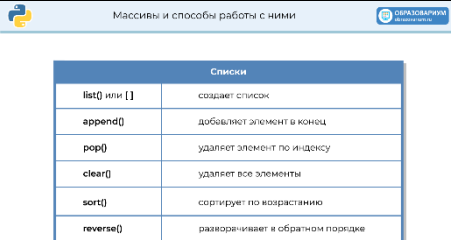
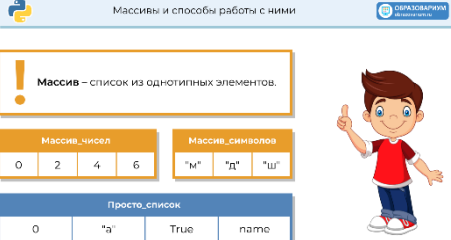
Учитель при подготовке к уроку может ознакомиться с содержанием видеоролика в текстовом формате, при необходимости распечатать фрагменты текста или примеры заданий и задач для использования в работе с учениками. Распечатанные тексты и задания из таблицы также можно применять в качестве раздаточного материала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

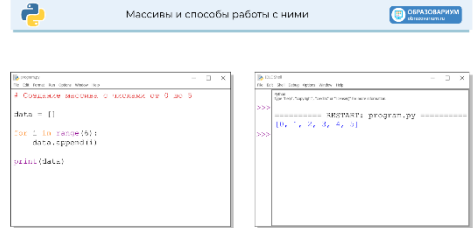
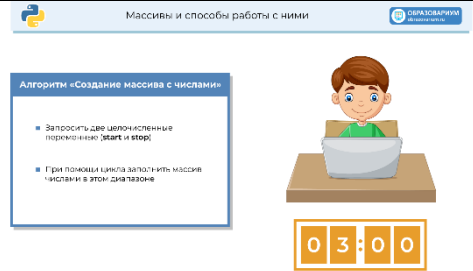
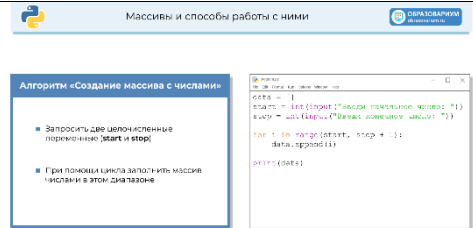
Таблица. Содержание видеоролика

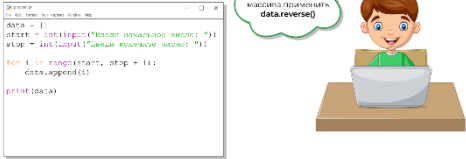
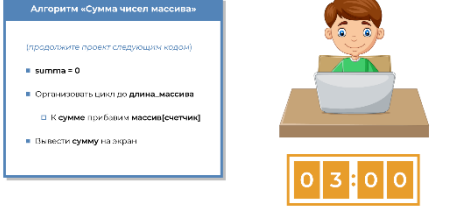
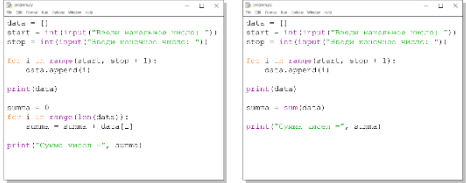
Название блока	Содержание блока и комментарии	Фрагменты из видеоролика	№ сцен
----------------	--------------------------------	--------------------------	--------

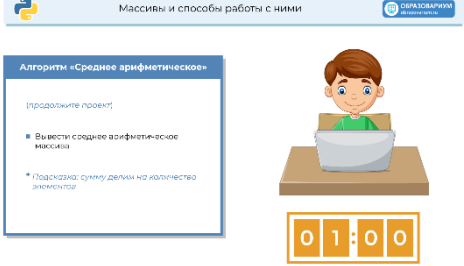
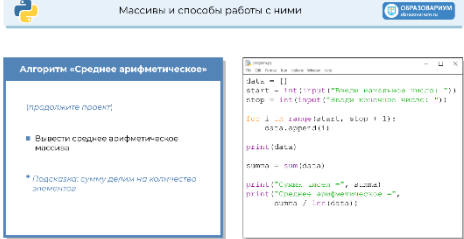
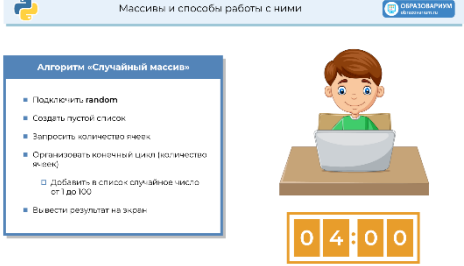
<p>Вводный блок. Мы узнаем</p>	<p><i>Обозначаем ученикам тему и цели урока.</i></p> <p>Массивы и способы работы с ними</p>	 <p>Сцена 1</p>	<p>1 2</p>
	<p>Начинаем знакомство с еще одной структурой для хранения данных – массивы. В базовом наборе Pythona это фактически те же самые списки. Разница лишь в том, что классический массив должен содержать элементы только одного типа, а в списке нет никаких ограничений. Также существует возможность создать целый класс массивов благодаря дополнительному модулю array, который может компактно представлять массив значений базовых типов: символы, целые числа, числа с плавающей запятой.</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 команда полностью удалит словарь?</p> <ol style="list-style-type: none"> 1) del 2) clear 3) reset 4) erase <p><i>Ответ 1. Для удаления словаря используют del имя_словаря.</i></p>	 <p>Сцена 3</p>	<p>3 4 5 6 7</p>

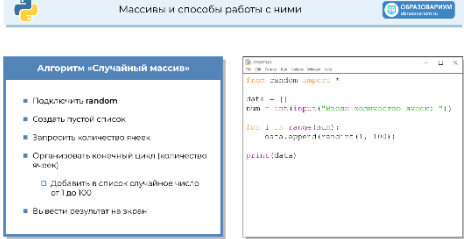
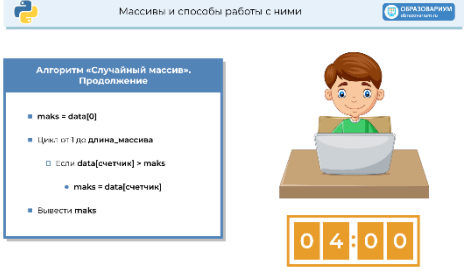
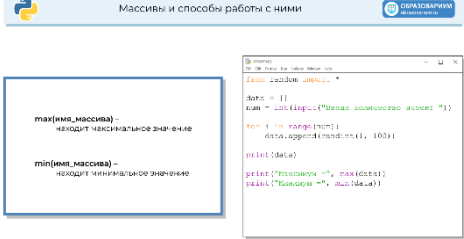
	<p>Второй вопрос. Чего нет у словаря?</p> <ol style="list-style-type: none"> 1) ключ 2) скобки 3) индекс 4) значение <p><i>Ответ 3. Словарь – это не индексируемая структура.</i></p>	 <p>Сцена 4</p>	
	<p>Третий вопрос. В какой строке записано множество?</p> <ol style="list-style-type: none"> 1) [key, value] 2) {key, value} 3) {key: value} 4) (key, value) <p><i>Ответ 2. Множество записывается в {}, элементы множества разделяются запятой.</i></p>	 <p>Сцена 5</p>	
	<p>Четвертый вопрос. Какая команда добавляет данные в словарь?</p> <ol style="list-style-type: none"> 1) insert 2) append 3) add 4) никакая <p><i>Ответ 4. Специальной команды для добавления данных в словарь не существует.</i></p>	 <p>Сцена 6</p>	

	<p>Пятый вопрос. Поднимите карточки - какие индексы у числа «6»? [[1, 2], [3, 4], [5, 6, 7, 8]]</p> <p>Поднимите карточки с соответствующими числами.</p> <p><i>Ответ 2 и 1. Индексы числа 6 – [2][1]</i></p> <p>[[1, 2], [3, 4], [5, 6, 7, 8]]</p> <p style="text-align: center;"> 0 1 2 3 </p>	 <p>Сцена 7</p>	
<p>Теоретический блок.</p> <p>Понятие массива</p>	<p>Вспомним то, что мы уже знаем про списки: это набор элементов, которые имеют общее имя, и каждый элемент имеет свой индекс — порядковый номер.</p> <p>Список создается при помощи квадратных скобок или команды лист.</p> <p>В список можно добавлять и удалять элементы, производить сортировку.</p> <p>Причем добавлять можно как в конец списка, так и в любую позицию.</p> <p>А удалять ячейку можно указав индекс или значение в ней.</p>	 <p>Сцена 8</p>	<p>8 9 10</p>
	<p>Классический массив в Python – это список, который составлен из однотипных элементов. Например, только из чисел или только из символов.</p> <p>Соответственно, если список является массивом, то с у нас появляется гораздо больше возможностей для работы с ним.</p>	 <p>Сцена 9</p>	

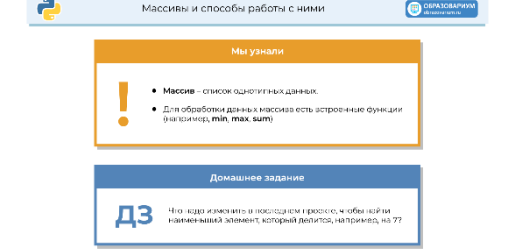
	<p>Мы будем работать с целочисленными массивами и первое с чего начнем – это создание и заполнение массива программным способом.</p> <p>Один из самых удобных способов – через цикл.</p> <p>Предположим, нам нужен массив с числами от нуля до пяти включительно. Создаем пустой список, организуем цикл на один больше, чем пять, и добавляем при помощи команды append в наш массив переменную счетчик. Проверим работу нашей программы.</p>	 <p>Сцена 10</p>	
<p>Блок заданий.</p> <p>Практические задания:</p> <p>Задание 1</p> <p>Задание 2</p>	<p><i>После окончания дикторского текста запускается таймер на 3 мин.</i></p> <p>Задание 1. Алгоритм «Создание массива с числами»</p> <ul style="list-style-type: none"> ■ Запросить две целочисленные переменные (start и stop) ■ При помощи цикла заполнить массив числами в этом диапазоне 	 <p>Сцена 11</p>	<p>11</p> <p>12</p> <p>13</p> <p>14</p> <p>15</p>
	<p>Разбор задания 1. Ваш код может выглядеть так.</p> <pre>data = [] start = int(input("Введи начальное число: ")) stop = int(input("Введи конечное число: ")) for i in range(start, stop + 1): data.append(i) print(data)</pre>	 <p>Сцена 12</p>	

	<p><i>Дополнительный вопрос:</i> а как сформировать массив из убывающих чисел?</p> <ul style="list-style-type: none"> – Как сформировать массив из убывающих чисел? – После создания массива применить data.reverse() 	<div data-bbox="1503 231 1966 459"> <p>Массивы и способы работы с ними</p>  <pre>data = [] start = int(input("Введите начальное число: ")) stop = int(input("Введите конечное число: ")) for i in range(start, stop + 1): data.append(i) print(data) data.reverse() print(data)</pre> </div> <p>Сцена 13</p>
	<p><i>После окончания дикторского текста запускается таймер на 3 мин.</i></p> <p>Задание 2. Алгоритм «Сумма чисел массива» <i>(продолжите проект следующим кодом)</i></p> <p>summa = 0 Цикл до (длина массива) К сумме прибавим массив [счетчик] Вывод суммы на экран</p>	<div data-bbox="1503 630 1966 901"> <p>Массивы и способы работы с ними</p>  </div> <p>Сцена 14</p>
	<p>Разбор задания 2. Продолжение Ваш код может выглядеть так. ... дописывается к существующему коду</p> <pre>summa = 0 for i in range(len(data)): summa = summa + data[i] print("Сумма чисел =", summa)</pre> <p><i>Вместо только что написанных строчек кода, мы можем воспользоваться встроенной функцией summa, которая нам сама все посчитает.</i></p>	<div data-bbox="1503 984 1966 1228"> <p>Массивы и способы работы с ними</p>  </div> <p>Сцена 15</p>

<p>Блок заданий. Практические задания: Задание 3 Задание 4</p>	<p>После окончания дикторского текста запускается таймер на 1 мин.</p> <p>Задание 3. Алгоритм «Среднее арифметическое» (продолжите проект)</p> <p>Вывести среднее арифметическое массива</p> <p>* Подсказка: сумму делим на количество элементов</p>	 <p>Массивы и способы работы с ними</p> <p>Алгоритм «Среднее арифметическое»</p> <p>(продолжите проект)</p> <ul style="list-style-type: none"> Вывести среднее арифметическое массива <p>* Подсказка: сумму делим на количество элементов</p> <p>01:00</p> <p>Сцена 16</p>	<p>16 17 18 19</p>
	<p>Разбор задание 3. Продолжение. Код программы будет выглядеть так: ... дописывается к существующему коду</p> <pre>print("Среднее арифметическое =", summa / len(data))</pre>	 <p>Массивы и способы работы с ними</p> <p>Алгоритм «Среднее арифметическое»</p> <p>(продолжите проект)</p> <ul style="list-style-type: none"> Вывести среднее арифметическое массива <p>* Подсказка: сумму делим на количество элементов</p> <pre>data = [] start = int(input("Введите количество точек: ")) stop = int(input("Введите количество точек: ")) for i in range(start, stop + 1): data.append(i) summa = sum(data) print("Среднее арифметическое =", summa / len(data))</pre> <p>Сцена 17</p>	
	<p>После окончания дикторского текста запускается таймер на 4 мин.</p> <p>Задание 4. Алгоритм «Случайный массив»</p> <ul style="list-style-type: none"> Подключить random Создать пустой список Запросить количество ячеек Организовать конечный цикл (количество ячеек) <ul style="list-style-type: none"> Добавить в список случайное число от 1 до 100 Вывести массив на экран 	 <p>Массивы и способы работы с ними</p> <p>Алгоритм «Случайный массив»</p> <ul style="list-style-type: none"> Подключить random Создать пустой список Запросить количество ячеек Организовать конечный цикл (количество ячеек) <ul style="list-style-type: none"> Добавить в список случайное число от 1 до 100 Вывести результат на экран <p>04:00</p> <p>Сцена 18</p>	

	<p>Разбор задание 4. Код программы будет выглядеть так:</p> <pre> from random import * data = [] num = int(input("Введи количество ячеек: ")) for i in range(num): data.append(randint(1, 100)) print(data) </pre>	 <p>Сцена 19</p>	
<p>Блок заданий. Практические задания: Задание 5</p>	<p><i>После окончания дикторского текста запускается таймер на 4 мин.</i></p> <p>Задание 5. Алгоритм «Случайный массив». Продолжение</p> <ul style="list-style-type: none"> ■ maks = data[0] ■ Цикл от 1 до длина_массива <ul style="list-style-type: none"> ○ Если data[счетчик] > maks <ul style="list-style-type: none"> • maks = data[счетчик] ■ Вывести maks 	 <p>Сцена 20</p>	<p>20 21</p>
	<p>Разбор задание 5. Код программы будет выглядеть так:</p> <p><i>... дописывается к существующему коду</i></p> <pre> maks = data[0] for i in range(1, len(data)): if data[i] > maks: maks = data[i] print("Максимум =", maks) </pre> <p>*Есть функция, которая сама находит в массиве максимум. Она так и называется – max. Все строки этого задания можно заменить на одну.</p> <p>Было бы странно, если б не было и ее антипода – минимума (min).</p>	 <p>Сцена 21</p>	

<p>Теоретический блок</p>	<p>Аналог этой задачи вам точно встретится на ОГЭ.</p> <p>Задача «Максимальное кратное»</p> <p>Дан числовой массив.</p> <p>Найдите в нем максимальное число, которое делится нацело на 5.</p> <p>Для решения задачи вернемся к нашему массиву случайных чисел.</p>	<div> <div>Массивы и способы работы с ними</div> <div> <div> <div>Задача «Максимальное кратное»</div> <div> <p>Дан числовой массив. Найдите в нем максимальное число, которое делится нацело на 5.</p> </div> </div> <div> <pre>from random import * data = [] max = int(input("Введите количество элементов: ")) for i in range(max): data.append(randomint(1, 100)) print(data)</pre> </div> </div> </div> <div>Сцена 22</div>	<p>22</p>
<p>Блок заданий.</p> <p>Практические задания:</p> <p>Задание 6</p>	<p>После окончания дикторского текста запускается таймер на 5 мин.</p> <p>Задание 6. Алгоритм «Максимальное кратное» (продолжите проект)</p> <ul style="list-style-type: none"> count = min(data) Конечный цикл до длина_массива <ul style="list-style-type: none"> if data[i] > count and data[i] % 5 == 0: <ul style="list-style-type: none"> count = data[i] Если count нацело НЕ делится на 5 <ul style="list-style-type: none"> Вывести «Таких чисел нет» Иначе <ul style="list-style-type: none"> Вывести «Это число =» и count 	<div> <div>Массивы и способы работы с ними</div> <div> <div> <div>Алгоритм «Максимальное кратное»</div> <div> <p>(продолжите проект)</p> <ul style="list-style-type: none"> count = min(data) Конечный цикл до длина_массива <ul style="list-style-type: none"> if data[i] > count and data[i] % 5 == 0: <ul style="list-style-type: none"> count = data[i] Если count нацело не делится на 5 <ul style="list-style-type: none"> Вывести «Таких чисел нет» Иначе <ul style="list-style-type: none"> Вывести «Это число =» и count </div> </div> <div> </div> </div> </div> <div>Сцена 23</div>	<p>23 24</p>
	<p>Разбор задание 6. Код программы будет выглядеть так:</p> <p>... дописывается к существующему коду</p> <pre>count = min(data) for i in range(num): if data[i] > count and data[i] % 5 == 0: count = data[i] if count % 5 != 0: print("Таких чисел нет") else: print("Это число =", count)</pre>	<div> <div>Массивы и способы работы с ними</div> <div> <div> <div>Алгоритм «Максимальное кратное»</div> <div> <p>(продолжите проект)</p> <ul style="list-style-type: none"> count = min(data) Конечный цикл до длина_массива <ul style="list-style-type: none"> if data[i] > count and data[i] % 5 == 0: <ul style="list-style-type: none"> count = data[i] Если count нацело не делится на 5 <ul style="list-style-type: none"> Вывести «Таких чисел нет» Иначе <ul style="list-style-type: none"> Вывести «Это число =» и count </div> </div> <div> <pre>data = [] num = int(input("Введите количество элементов: ")) for i in range(num): data.append(randomint(1, 100)) print(data) count = min(data) for i in range(num): if data[i] > count and data[i] % 5 == 0: count = data[i] if count % 5 != 0: print("Таких чисел нет") else: print("Это число =", count)</pre> </div> </div> </div> <div>Сцена 24</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>В ходе проекта:</p> <ul style="list-style-type: none"> ▪ Использовали словарь для хранения и обработки данных. ▪ Убедились, что словарь удобно использовать, когда <ul style="list-style-type: none"> ○ нумерация ячеек невозможна, ○ необходима высокая скорость доступа к данным. <p><i>Преподаватель дает ученикам домашнее задание к следующему занятию (Приложение 2).</i></p>	 <p>Сцена 25</p>	<p>25</p>
--	---	---	-----------

Приложение 2

Домашнее задание

Что надо изменить в завершающем проекте занятия, чтобы найти наименьший элемент, который делится, например, на семь?

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

Практика

Проект «Максимум»

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

Запросите число, на которое будем делить элементы массива.

Найдите максимальное значение в массиве, которое делится на заданное пользователем число. Если чисел, кратных заданному пользователем нет, выведите соответствующее сообщение.

Проект «Объединение массивов»

Запросите у пользователя количество ячеек в массиве и сгенерируйте два массива случайных чисел (data1 и data2).

Сформируйте третий массив так, чтобы значение в каждой его ячейке являлось максимальным значением из ячеек массивов data1 и data2 с тем же индексом.

Выведите все три массива на экран.

Приложение 3

Краткие организационно-методические рекомендации по организации работы на занятии

«Массивы и способы работы с ними».

В начале занятия желательно повторить материал по теме «списки», поскольку массивы, с которыми мы будем работать это не класс **array**, а просто однотипные списки. Соответственно методы работы будут одинаковые.

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.

В проектах отсутствуют генераторы списков, так как им будет посвящена отдельная тема. Поэтому заполнение всех массивов осуществляется через цикл.

Код одного из проектов (34.4) будет использован дважды для экономии времени. Обратите на это внимание.

Условия последней задачи взяты из ОГЭ для 9-х классов, поэтому ее желательно разобрать более подробно на будущее.

В дополнении будет предложен альтернативный вариант решения при условии, что массив можно изменять, меняя ячейки местами.