



Методические рекомендации по теме

«Рандомизация выбора числа»

Цель:

- дать представление о понятии случайный выбор и генерации случайных чисел.

Задачи:

- практика применения случайного выбора при программировании в Python;
- понимание случайного выбора в программировании;
- 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 функции случайного выбора и генерации случайных целых и дробных чисел в Python.

Метапредметные: обучающиеся получают возможность владения общепредметными понятиями «случайный выбор», «случайное число», «генерация случайных чисел»; информационно-логическими умениями; умениями самостоятельно планировать пути достижения целей; владения умениями принятия решений и осуществления

осознанного выбора; повышения уровня ИКТ – компетентности и расширение кругозора в области информатики и программирования; знакомство с профессиональной деятельностью программиста в рамках ранней профориентации; развитие интеллектуальных способностей, а также логического и критического мышления.

Материалы к занятию

Приложение 1: Сценарный план видеоролика

Приложение 2: Домашнее задание и практика

Приложение 3: Краткие организационно-методические рекомендации по организации работы на занятии

Ход проведения урока

1. Организационный момент.

Мотивация на учебную деятельность.

Приветствие учащихся, сообщение темы и целей занятия (мы узнаем, об интересном виде искусства «псевдографике» – рисовании и создании анимации с помощью символов, научимся автоматизировать процесс создания «шедевров» в технике «псевдографики»).

Проблемная дискуссия по вопросам:

- Что такое случайность, случайный выбор?
- Приведите примеры случайного выбора (игральная кость (кубик), лотерея, бросание монетки?)
- Для каких целей можно использовать случайный выбор в программировании?

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

- случайный выбор характеризуется тем, что он не зависит от самого выбирающего
- в программировании случайный выбор часто используют для создания постоянно меняющихся параметров, например в компьютерных играх.

Преподаватель называет ученикам тему и цели урока.

2. Вводный блок.

Тема.

Преподаватель при необходимости останавливая трансляцию, комментируя дополнительно тему занятия.

**см. сцены 1 – 2 (здесь и далее приводится Таблица «Содержание видеоролика». Приложение 1).*

3. Блок повторения.

Блиц-опрос.

Преподаватель предлагает ученикам ответить на **5 вопросов** по предыдущей теме; задания выполняются в сопровождении видеоролика с использованием таймера; ученики выполняют задания, голосуют, обсуждают результаты. Процедура голосования определяется инструкцией **в сцене 3**; учитель должен убедиться, что всем понятна процедура голосования.

Преподаватель может поставить ролик на паузу и обсудить результаты голосования; объяснить правильный ответ руководствуясь материалами предыдущего занятия

**см. сцены 3 – 7*

4. Теоретический блок.

Рандомизация выбора числа.

Новый материал излагается в сопровождении видеоролика.

Обсуждение вопросов по просмотренным материалам:

- Для чего служит модуль **random**?
- Какие новые команды мы узнали, как они работают?
- Что такое аргумент?
- Как соотносятся первый и второй аргумент в команде **randint**?

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*

5. Блок заданий.

Проекты: «Таблица умножения», «Градусник».

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

«Таблица умножения»: включает *практическое задание 1* для учеников с последующим разбором. Задание представляет собой небольшой программный проект для вывода случайных множителей с использованием команды генерации случайных чисел модуля **random**.

После выполнения задания ученики получают работающий программный продукт – программа выводит случайные множители и вычисляет их произведение, затем запрашивает у пользователя ответ и проверяет его на правильность.

«Градусник»: включает *практическое задание 2* для учеников с последующим разбором. Задание представляет собой небольшой программный проект для вывода случайного числа с округлением до 1 знака после точки, а затем сравнивает с указанными величинами.

После выполнения задания ученики получают работающую программу для организации вывода случайного значения температуры в указанных границах, сравнивает его с указанными параметрами температуры и выводит экспертное заключение (переохлаждение, здоров, болен).

Блок включает теоретическую вставку «команда **uniform** модуля **random**, функция **round** для округления».

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

**см. сцены 11 – 19*

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.

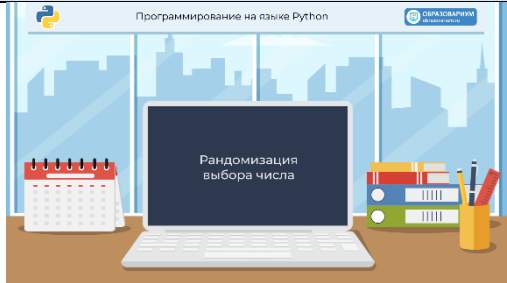

Преподаватель дает ученикам домашнее задание к следующему занятию (*Приложение 2*).

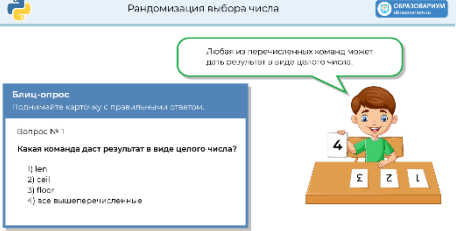
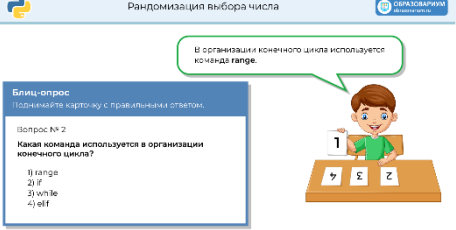
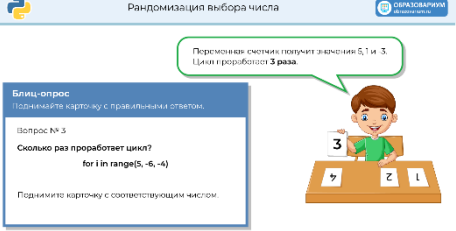
**см. сцена 20*

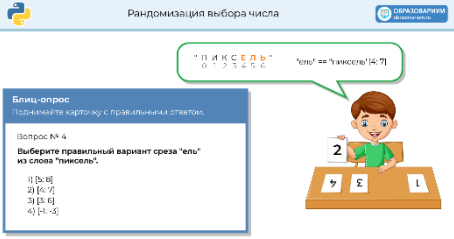
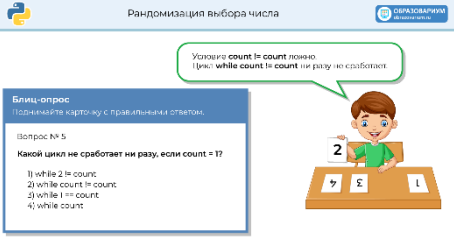
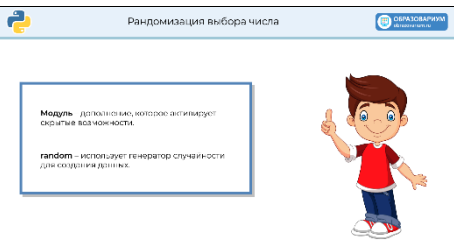
Сценарный план видеоролика


В таблице «Содержание видеоролика» представлен краткий текст из видеоролика, примеры заданий и задач, которые будут демонстрироваться на экране. Учитель при подготовке к уроку может ознакомиться с содержанием видеоролика в текстовом формате, при необходимости распечатать фрагменты текста или примеры заданий и задач для использования в работе с учениками. Распечатанные тексты и задания из таблицы также можно применять в качестве раздаточного материала как на уроке, так и для домашних заданий.

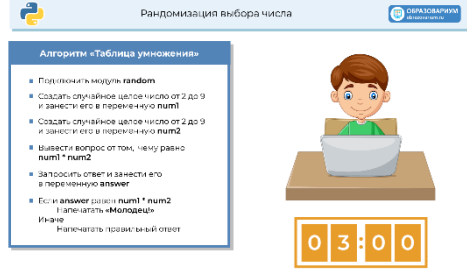
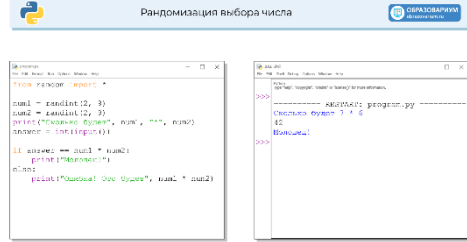
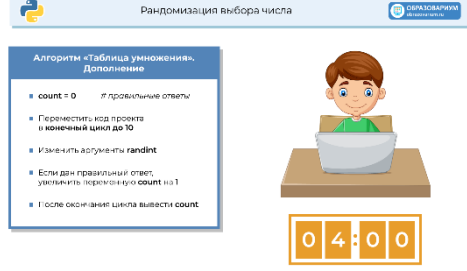
Таблица. Содержание видеоролика

Название блока	Содержание блока и комментарии	Фрагменты из видеоролика	№ сцен
Вводный блок. Мы узнаем	Обозначаем ученикам тему и цели урока. Рандомизация выбора числа.	 Сцена 1	1 2
	На этом занятии начнем знакомиться с таким понятием как случайный выбор. «Генерация случайных чисел слишком важна, чтобы оставлять ее на волю случая» - сказал однажды математик Роберт Кавью. Попробуем учиться управлять этим очень любопытным и важным процессом.	 Сцена 2	

<p>Блок повторения.</p> <p>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 даст результат в виде целого числа?</p> <ol style="list-style-type: none"> 1) len 2) ceil 3) floor 4) все вышеперечисленные <p><i>Ответ 4. Любая из перечисленных команд может дать результат в виде целого числа.</i></p>	 <p>Сцена 3</p>	<p>3</p> <p>4</p> <p>5</p> <p>6</p> <p>7</p>
	<p>Второй вопрос. Какая команда используется в организации конечного цикла?</p> <ol style="list-style-type: none"> 1) range 2) if 3) while 4) elif <p><i>Ответ 1. В организации конечного цикла используется команда range.</i></p>	 <p>Сцена 4</p>	
	<p>Третий вопрос. Сколько раз проработает цикл. Поднимите карточку с номером ответа.</p> <p>for i in range (5, -6, -4)</p> <p><i>Ответ 3. Переменная счетчик получит значения 5, 1 и -3. Цикл проработает 3 раза.</i></p>	 <p>Сцена 5</p>	

	<p>Четвертый вопрос. Выберите правильный вариант среза "ель" из слова "пиксель".</p> <ol style="list-style-type: none"> 1) [5: 8] 2) [4: 7] 3) [3: 6] 4) [-1: -3] <p><i>Ответ 2. "ель" = "пиксель"[4: 7]</i></p>	 <p>Сцена 6</p>	
	<p>Пятый вопрос. Какой цикл не сработает ни разу, если count = 1?</p> <ol style="list-style-type: none"> 1) while 2 != count 2) while count != count 3) while 1 == count 4) while count <p><i>Ответ 2. Условие count != count ложно. Цикл while count != count ни разу не сработает..</i></p>	 <p>Сцена 7</p>	
<p>Теоретический блок.</p> <p>Рандомизация выбора числа</p>	<p>Известно, что для использования дополнительных возможностей, необходимо подключить расширение, которое называется модуль. Очередной модуль, с которым мы познакомимся, называется random. Он представляет набор команд для генерации случайных букв, чисел, элементов последовательности.</p> <p>Модуль – дополнение, которое активирует скрытые возможности. random – использует генератор случайности для создания данных.</p>	 <p>Сцена 8</p>	<p>8 9 10</p>

	<p>Перечислим несколько команд, с которыми нам предстоит познакомиться. Некоторые команды модуля random</p> <p>randint (A, B)</p> <p>– создает целое число от A до B</p> <p>uniform (A, B)</p> <p>– создает дробное число от A до B</p> <p>shuffle (набор данных)</p> <p>– перемешивает набор данных</p> <p>choice (набор данных)</p> <p>– выбирает из данных случайный элемент</p>	<div><div>Рандомизация выбора числа</div><div><table><tr><th colspan="2">Некоторые команды модуля random</th></tr><tr><td>randint(A, B)</td><td>Создает целое число от A до B.</td></tr><tr><td>uniform(A, B)</td><td>Создает дробное число от A до B.</td></tr><tr><td>shuffle(набор данных)</td><td>Перемешивает набор данных.</td></tr><tr><td>choice(набор данных)</td><td>Выбирает случайный элемент из набора данных.</td></tr></table></div></div> <p>Сцена 9</p>	Некоторые команды модуля random		randint(A, B)	Создает целое число от A до B.	uniform(A, B)	Создает дробное число от A до B.	shuffle(набор данных)	Перемешивает набор данных.	choice(набор данных)	Выбирает случайный элемент из набора данных.
Некоторые команды модуля random												
randint(A, B)	Создает целое число от A до B.											
uniform(A, B)	Создает дробное число от A до B.											
shuffle(набор данных)	Перемешивает набор данных.											
choice(набор данных)	Выбирает случайный элемент из набора данных.											
	<p>Рассмотрим это на примере. Подключим модуль random и присвоим при помощи команды randint, переменной num случайное число от 0 до 100, выведем его на экран, в результате получим какое-то число не меньше нуля и не больше ста.</p> <pre>from random import * num = randint(0, 100) print(num)</pre> <p>Числа, которые мы ввели в скобках, называются по-научному аргументы или параметры. Первый аргумент обязательно должен быть меньше второго!</p>	<div><div>Рандомизация выбора числа</div><div><div><pre>from random import * num = randint(0, 100) print(num)</pre></div><div><pre>from random import * num = randint(0, 100) print(num)</pre></div></div><div>randint(аргумент 1, аргумент 2) аргумент 1 < аргумент 2</div></div> <p>Сцена 10</p>										
<p>Блок заданий.</p> <p>Практические задания:</p> <p>Задание 1</p> <p>Задание 2</p> <p>Задание 3</p> <p>Задание 4</p>	<p><i>После окончания дикторского текста запускается таймер на 4 мин.</i></p> <p>Задание 1. Алгоритм «Таблица умножения»</p> <ul style="list-style-type: none">• Подключить модуль random• Создать случайное целое число от 2 до 9 и занести его в переменную num1• Создать случайное целое число от 2 до 9 и занести его в переменную num2• Напечатать: чему равно num1 * num2• Запросить ответ и занести его в переменную answer	<div><div>Рандомизация выбора числа</div><div><div>Алгоритм «Таблица умножения»</div><ul style="list-style-type: none">• Подключить модуль random• Создать случайное целое число от 2 до 9 и занести его в переменную num1• Создать случайное целое число от 2 до 9 и занести его в переменную num2• Вывести вопрос от том, чему равно num1 * num2• Запросить ответ и занести его в переменную answer</div><div><div>04:00</div></div></div> <p>Сцена 11</p>	<p>11</p> <p>12</p> <p>13</p> <p>14</p> <p>15</p> <p>16</p> <p>17</p> <p>18</p> <p>19</p>									

	<p>После окончания дикторского текста запускается таймер на 3 мин.</p> <p>Продолжим и организуем проверку.</p> <p>Задание 1. Алгоритм «Таблица умножения» (продолжение)</p> <ul style="list-style-type: none"> Если answer равен num1 * num2 Напечатать «молодец» Иначе Напечатать правильный ответ 	 <p>Сцена 12</p>	
	<p>Разбор задания 1. Ваш код может выглядеть так.</p> <pre>from random import * num1 = randint(2, 9) num2 = randint(2, 9) print("Сколько будет", num1, "*", num2) answer = int(input()) if answer == num1 * num2: print("Правильно!") else: print("Ошибка! Это будет", num1 * num2)</pre> <p>Какие недостатки у этого проекта? Он одноразовый и примеры слишком легкие.</p>	 <p>Сцена 13</p>	
	<p>После окончания дикторского текста запускается таймер на 4 мин.</p> <p>Продолжим и организуем проверку.</p> <p>Задание 1. Алгоритм «Таблица умножения» (дополнение)</p> <ul style="list-style-type: none"> count = 0 # правильные ответы Переместите код проекта в конечный цикл до 10 Измените аргументы randint Если дан правильный ответ – измените переменную count на 1 После окончания цикла – вывод count 	 <p>Сцена 14</p>	

Разбор задания 1 (дополнение). Код программы может выглядеть так.

```
from random import *
count = 0
for i in range(10):
    num1 = randint(10, 15)
    num2 = randint(2, 9)
    print("Сколько будет", num1, "*", num2)
    answer = int(input())
    if answer == num1 * num2:
        print("Правильно!")
        count = count + 1
    else:
        print("Ошибка! Это будет", num1 * num2)
print(count, "правильных ответов")
```

Ещё немного теории:

Теперь познакомимся с командой **uniform**.
Она также генерирует случайное число, но не целое, а дробное.

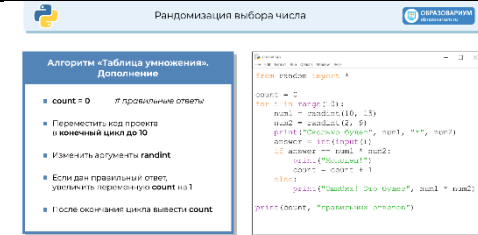
Для изменения формата числа, часто используют функцию **round**, которая умеет округлять числа.

После окончания дикторского текста запускается таймер на 3 мин.

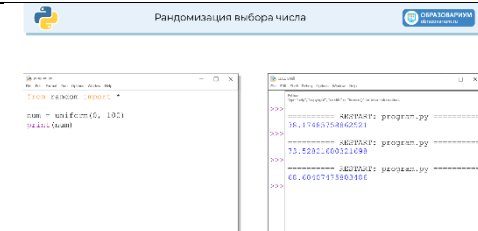
Задание 2. Алгоритм «Градусник»

- Подключить модуль **random**
- Создать случайное дробное число от 35 до 41 и занести его в переменную **temp**
- Округлить **temp** до 1 знака после точки
- Напечатать: температура равна **temp**

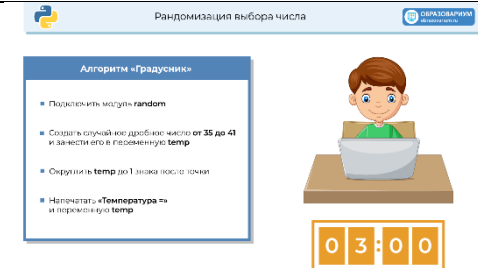
Программа сгенерирует дробное число в диапазоне от 35 до 41 и занесет его в переменную.



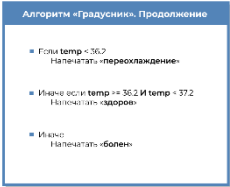

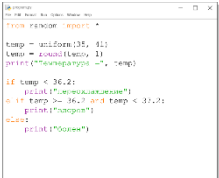

Сцена 15



Сцена 16



Сцена 17

	<p><i>После окончания дикторского текста запускается таймер на 3 мин.</i></p> <p>Задание 2. Алгоритм «Градусник» (продолжение)</p> <ul style="list-style-type: none"> Если <code>temp < 36.2</code> Напечатать «переохлаждение» Иначе если <code>temp > 36.1 И temp < 37.2</code> Напечатать: «здоров» Иначе Напечатать: «болен» <p><i>В продолжении попробуем поставить диагноз с помощью операторов условия.</i></p>	<div data-bbox="1496 228 1960 502">   </div> <p>Сцена 18</p> <div data-bbox="1496 635 1960 882">   </div> <p>Сцена 19</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"> ▪ модуль random использует генератор случайности для создания данных; ▪ randint (A, B) – создает целое число от A до B; ▪ uniform (A, B) – создает дробное число от A до B. <p><i>Преподаватель дает ученикам домашнее задание к следующему занятию (Приложение 2).</i></p>	<div data-bbox="1512 231 1960 486"> </div>	<p>20</p>
--	--	--	-----------

Приложение 2

Домашнее задание

Вспомните и составьте список игр (10-12 названий) как реальных, так и компьютерных в которых есть случайный выбор.

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

Практика

Проект «Случайные покупки»

Проект представляет собой симулятор похода в магазин.

Задайте начальное значение: количество денег в «кошельке».

Сгенерируйте цену товара в диапазоне на ваш выбор и выведите ее на экран. Спросите у пользователя, хочет ли он совершить покупку?

Если пользователь хочет купить товар и у него достаточно денег, симитируйте покупку: уменьшите количество денег в «кошельке» и выведите сообщение о покупке и остатке средств. Если же средств недостаточно, выведите соответствующие сообщение.

Повторяйте генерацию цены товара и дальнейшие действия.

Придумайте и запрограммируйте возможность для пользователя завершить покупки.

По желанию дополните проект подсчетом количества покупок и потраченной суммы.

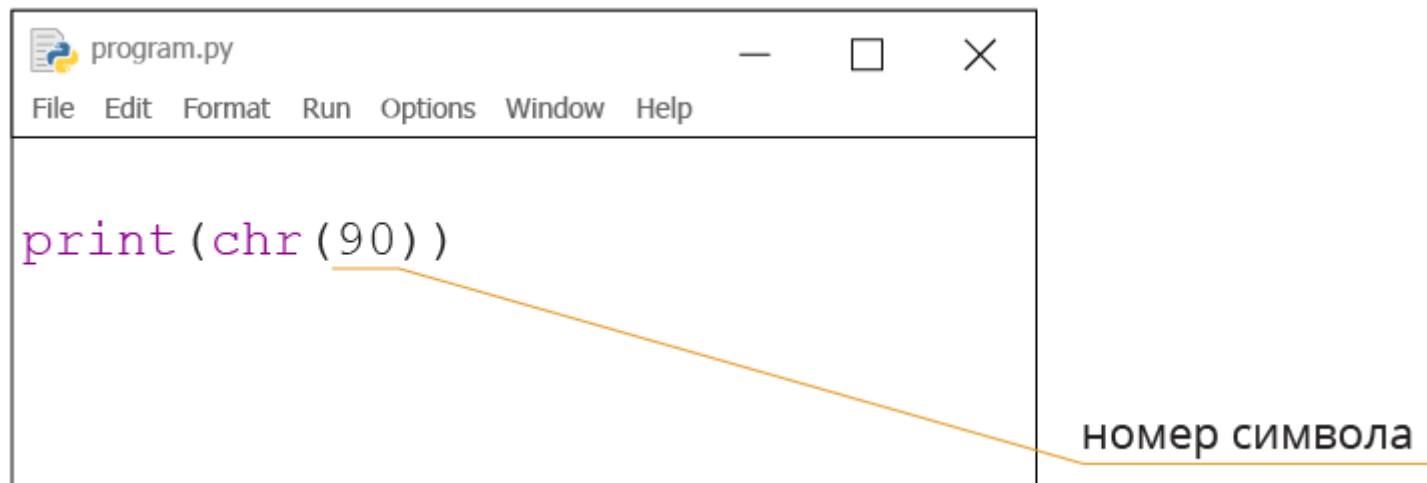
Проект «Генератор пароля»

Запросите длину пароля.

Сгенерируйте пароль из символов с номерами в диапазоне от 48 до 122 (цифры, некоторые символы и латинские буквы).

Выведите пароль на экран.

Чтобы получить символ по его номеру используйте метод **chr()**.



Такой код выдаст символ **Z** (символ с номером 90).

Краткие организационно-методические рекомендации по организации работы на занятии

«Рандомизация выбора числа».

В начале занятия можно посмотреть творчество ребят, посвященного доработки проекта в стиле псевдографики и, если есть интересные идеи – желательно их продемонстрировать.

Перед началом просмотра ролика поинтересуйтесь у ребят, что они знают о случайности. Можно привести пример игральных костей, как генератора случайных чисел. Или самый простой вариант – подбрасывание монетки. В этом примере мы получаем два случая: 0 или 1.

Также можно вспомнить информацию о том, что такое модули, для чего нужны и с какими мы уже работали (**math, time**).

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. Создание каждого проекта разбито на этапы, чтобы было удобнее контролировать успеваемость ребят.

Если **в проекте «Таблица умножения»** на основной код ребята потратят больше времени чем запланировано, то дополнение в виде конечного цикла и подсчета очков делать не нужно.

Если наоборот, проект не вызовет сложности, то можете ввести еще переменную, отвечающую за подсчет неправильных ответов (с последующим выводом этой информации).

В проекте «Градусник» можно также при наличии времени сделать больше условий с более подробными «диагнозами»: легкое недомогания, тяжелое состояние.

Обязательно обратите внимание на использование логических переменных (прежде всего **and**) для корректности условий.