

«Создание тематической викторины в среде Scratch».

Кушей Ирина Зеноновна.
Педагог дополнительного
образования.
Центр Ювента.

Цель: создание благоприятных условий для формирования интереса учащихся к программированию и развитию их творческих способностей через создание тематической викторины в среде визуального программирования Scratch.

Задачи:

- сформировать навыки разработки готового программного продукта в среде Scratch;
- развивать интеллектуальные, познавательные и творческие способности, используя возможности программирования на Scratch;
- способствовать развитию логического, критического, системного, алгоритмического и творческого мышления;
- развить любознательность, наблюдательность, память;
- воспитать потребность в самообразовании и творческой реализации.

Оснащение: персональные компьютеры, проектор, программное обеспечение Scratch 2.0., раздаточный материал (ТЗ, инструкционная карта).

План мастер-класса (продолжительность: 40 минут).

1. Организационный этап (1 минута).
2. Мотивационно-целевой этап (9 минут).
3. Основной этап (25 минут).
4. Заключительный этап (5 минут).

Конспект мастер-класса.

Организационный этап. Приветствие, проверка готовности группы к работе.

Мотивационно-целевой этап. Беседа с демонстрацией на экране. Тема мастер-класса и ТЗ.

Приложение Scratch представляет собой среду для создания программ и анимации. Мы с вами разберемся с принципами работы в Scratch, поймем как составлять программы. Вам откроются большие просторы для самостоятельного творчества и исследования.

Давайте создадим викторину-тренажер, проверяющую знание таблицы умножения. Она будет построена в виде диалога программы и пользователя. В ходе выполнения заданий викторины будет вестись подсчет правильных и неправильных ответов (*демонстрация тренажера*).

Перед тем как приступить к созданию викторины необходимо разработать техническое задание.

Обратите внимание, на экран (демонстрация готового ТГ викторины-тренажера, проверяющего знание таблицы умножения, подробный разбор каждого этапа ТЗ).

Техническое задание.

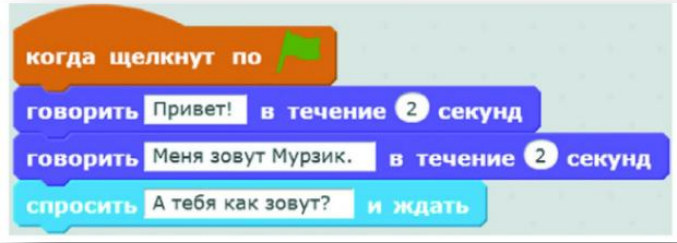
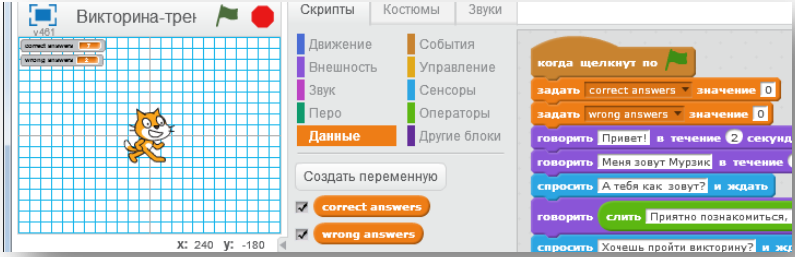
№	Этап ТЗ	Содержание
1	Название проекта	Тренажер «Таблица умножения»
2	Идея проекта	Тренажер построен в виде диалога программы и пользователя.
3	Сценарий проекта	Спрайт задает вопросы, а пользователь отвечает на них. Происходит подсчет верных и неверных ответов. Выводится результат.

4	Среда	Фон для сцены библиотечный.
5	Спрайты	Мурзик из библиотеки. 
6	Интерфейс	Приложение запускается по зелёному флажку. Программа реагирует на ответы пользователя.

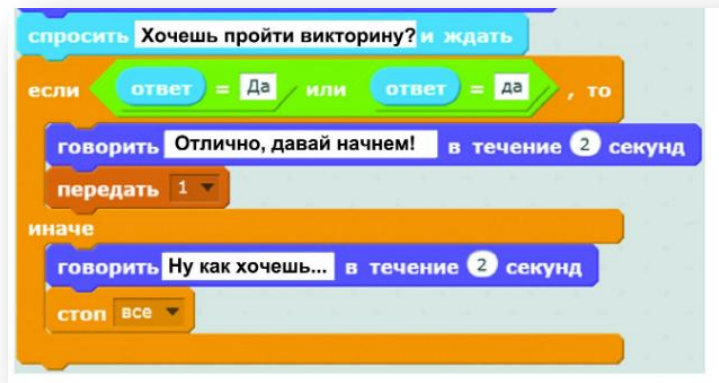
Основной этап. Теперь мы можем приступить к выполнению задания.

Педагог с помощью проектора поэтапно демонстрирует шаги, проговаривая действия (создание сцены, выбор и добавление спрайта, написание скриптов). Дети выполняют все действия вместе с педагогом.

Инструкционная карта

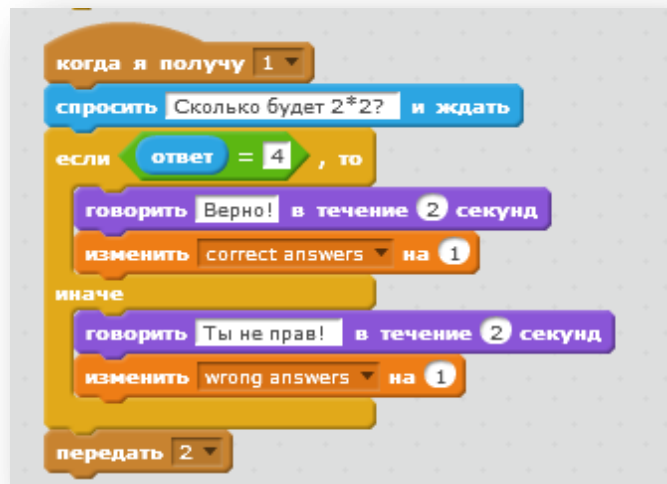
1 Сначала Мурзик здоровается с пользователем, называет свое имя, спрашивает имя пользователя и после ввода пользователем имени обращается к нему по имени. Мурзик должен сказать: «Привет!», «Меня зовут Мурзик!», «А тебя как зовут?».	
2 Мурзик — вежливый кот, он хочет говорить: «Приятно познакомиться» и называть нас по имени. Добавим следующую сложную конструкцию: Как мы помним, в ответе содержится введенное с клавиатуры имя.	
3 Создадим переменные, которые назовем: correct answers (количество правильных ответов) и wrong answers (количество неправильных ответов).	
4 Продолжим диалог. Пусть Мурзик спросит пользователя, хочет ли он пройти викторину. Если пользователь ответит «Да», Мурзик должен сказать: «Отлично, давай начнем!» В противном случае Мурзик должен сказать: «Ну, как	

хочешь...» и программа должна остановиться.



Проверьте, правильно ли работает программа!

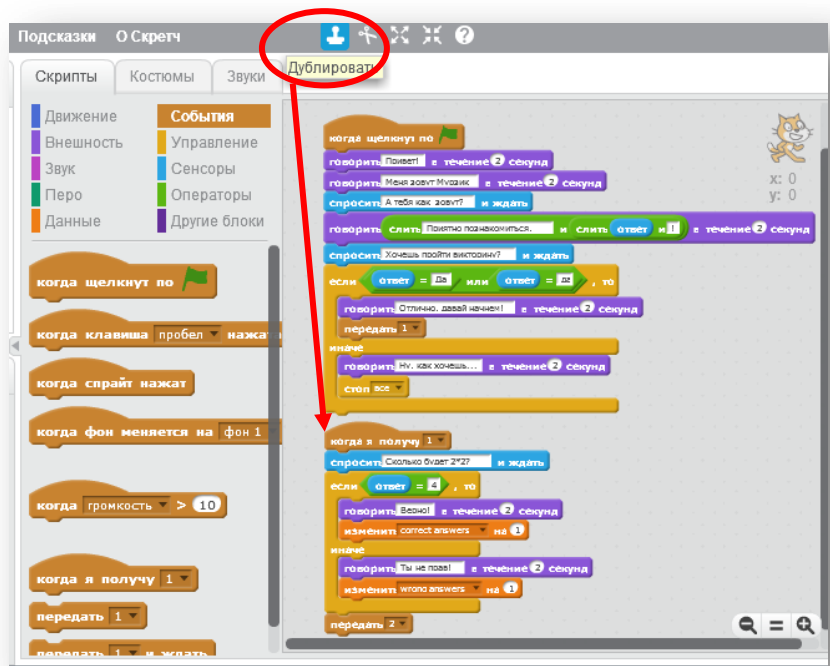
5 Предыдущий скрипт закончился передачей сообщения 1. Значит, нужно его получить, используя команду **Когда я получу 1**. Задаем вопрос. Если пользователь отвечает верно, спрайт должен подтвердить, что ответ верный и увеличить на 1 переменную: **correct answers**, отвечающую за количество правильных ответов. Если же ответ неверный, Мурзик должен сказать, что ответ неверный, и увеличить на 1 переменную **wrong answers**.



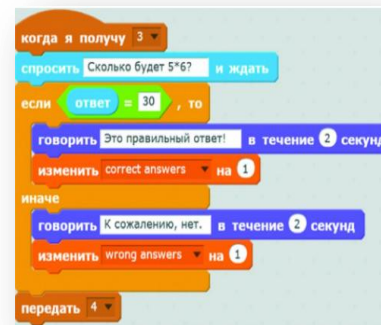
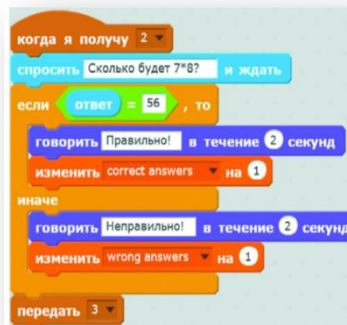
6 Аналогично задаем все следующие вопросы. Для того чтобы вручную не прописывать каждый раз все команды, используем маленькую хитрость. В верхней части окна программы есть кнопка **Дублировать**.

Нажмем ее, а потом нажмем на **верхнюю команду** из тех, которые нужно дублировать. Все они скопируются.

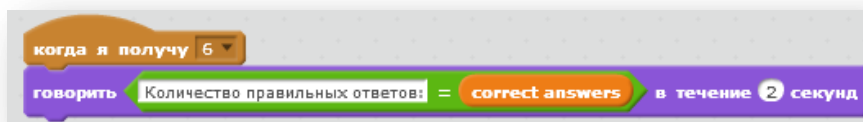
Останется только подправить вопросы и ответы.



7 Оставшиеся скрипты должны выглядеть следующим образом:



8 После этого Мурзик должен вывести количество правильных ответов.



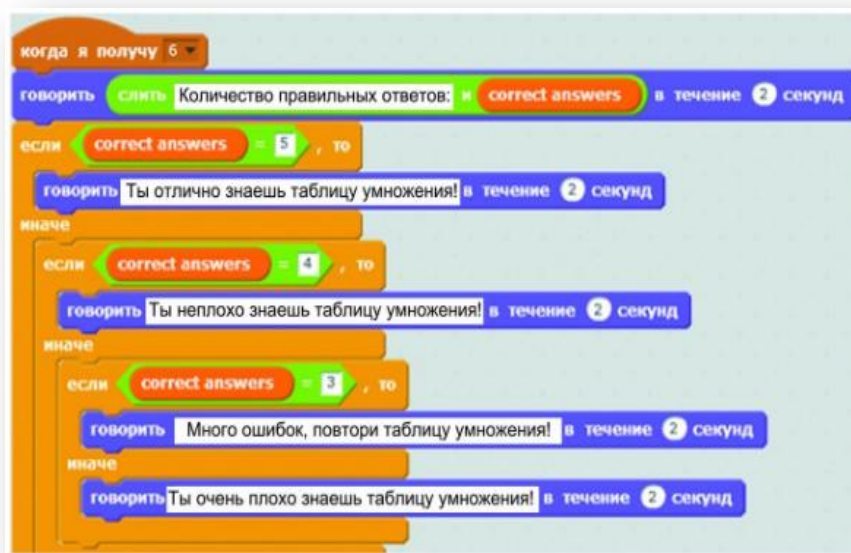
9 В заключение Мурзик должен оценить знания пользователя, то есть сказать, насколько хорошо тот знает таблицу умножения. Для этого допишем последний скрипт.

Правильных ответов 5 – Мурзик говорит: «Ты отлично знаешь таблицу умножения!».

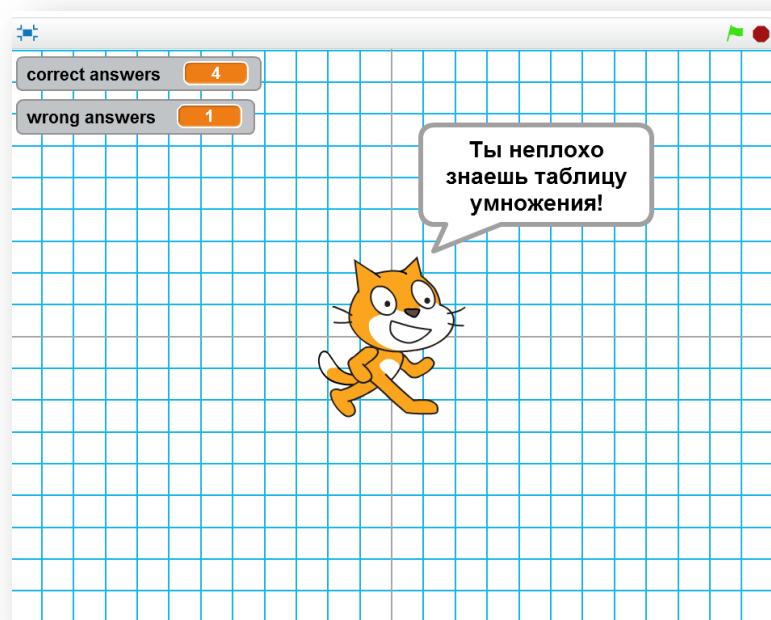
Правильных ответов 4 – Мурзик говорит: «Ты неплохо знаешь таблицу умножения!».

Правильных ответов 3 – Мурзик говорит: «Много ошибок, повтори таблицу умножения!».

Иначе: «Ты очень плохо знаешь таблицу умножения!».



Сделайте интерфейс программы более дружелюбным: можно заменить спрайт или фон либо сделать так, чтобы спрайт двигался, и так далее. Измените программу так, чтобы на каждое действие выдавалось не 5, а 10 примеров.



Викторина по математике готова!

Заключительный этап.

Способ, который мы только что использовали для написания тренажера, удобен для создания викторины, например, по биологии, географии и другим предметам. А вот для создания математического тренажера совсем не обязательно каждый раз сравнивать ответ пользователя с правильным ответом, Scratch сам умеет считать!

Создайте свою собственную викторину. Вопросы викторины могут относиться к любой из тем, изучаемых в рамках учебных предметов «Математика», «История», «География», «Биология», «Литература» и т.д. Подобное задание предлагалось учащимся 5-6 классов на Scratch-Олимпиаде.

Вопросы заключения:

- Что интересного было сегодня в нашем путешествии для вас? –
- Что вызвало затруднения и оказалось самым трудным? –
- Какие проекты вам показались более сложными или интересными? –
- Что лежит в основе создания проектов в Скретч? –
- Сложно ли научиться программированию? –

Литература.

1. Голиков Д. В. Scratch для юных программистов. — СПб.: БХВ-Петербург, 2017г.
2. Голиков Д. В. 40 проектов на Scratch для юных программистов. — СПб.: БХВ-Петербург, 2018. — 192 с.
3. Патаракин Е. Д. Учимся готовить в среде Скретч (Учебно-методическое пособие). М: Интуит.ру, 2008. — 73 с.
4. Рындак В.Г., Дженжер В.О., Денисова Л.В. Проектная деятельность школьника в среде программирования Scratch. Учебно-методическое пособие. Оренбург: Оренб. гос. ин-т. менеджмента, 2009.