

П. В. Кушняров

ИССЛЕДУЕМ, ПРОЕКТИРУЕМ, СОЗДАЁМ

ИНФОРМАТИКА

Образовательные проекты

6 – 9 классы

КОМПЛЕКТЫ ЗАДАНИЙ

Пособие для учащихся учреждений образования, реализующих образовательные программы общего среднего образования, с белорусским и русским языками обучения и воспитания

*Рекомендовано государственным учреждением образования
«Академия образования»*

Минск
2026

ОГЛАВЛЕНИЕ

1. БЕЗОПАСНЫЙ ИНТЕРНЕТ: ПРАВИЛА И СОВЕТЫ ДЛЯ УЧАЩИХСЯ..	3
2. ВИРТУАЛЬНЫЙ ПИТОМЕЦ: ОБУЧАЮЩЕЕ ИНТЕРАКТИВНОЕ ПРИЛОЖЕНИЕ В СРЕДЕ ПРОГРАММИРОВАНИЯ SCRATCH	7
3. ИНТЕРАКТИВНАЯ ТАБЛИЦА МЕНДЕЛЕЕВА В СРЕДЕ SCRATCH.....	15
4. УМНЫЙ ПОМОЩНИК: КАК РАБОТАЕТ РОБОТ-ПЫЛЕСОС	20
5. СОЗДАНИЕ ЦИФРОВОГО ОПРЕДЕЛИТЕЛЯ ЗВУКОВ ЖИВОТНЫХ	27
6. БЕЛАРУСКІ АРНАМЕНТ У ЛІЧБАВАЙ ГРАФІЦЫ	33
7. ИНФОГРАФИКА: ПРЕДСТАВЛЕНИЕ ДАННЫХ В ВИЗУАЛЬНОЙ ФОРМЕ.....	37
8. ЗНАКОМСТВО С 3D-РЕДАКТОРОМ: МОДЕЛИРОВАНИЕ МАТЕМАТИЧЕСКИХ ФИГУР В КОМПАС-3D LT.....	43
9. МОДЕЛИРОВАНИЕ ДВИЖЕНИЯ ТЕЛА ПО ЗАКОНАМ ФИЗИКИ	48
10. ХИМИЯ В ЧИСЛАХ: ИЗУЧЕНИЕ СВОЙСТВ ЭЛЕМЕНТОВ ЧЕРЕЗ ВИЗУАЛЬНОЕ ПРОГРАММИРОВАНИЕ.....	52

БЕЗОПАСНЫЙ ИНТЕРНЕТ: ПРАВИЛА И СОВЕТЫ ДЛЯ УЧАЩИХСЯ

Интернет представляет собой огромную сеть ресурсов, в которой есть полезные сайты для учёбы, игры, видео, социальные сети, а также опасные зоны. Часто учащиеся заходят в сеть, не зная правил безопасной работы, и могут попасть в неприятные ситуации.

Выполняя данный проект, вы изучите основные виды интернет-угроз, проведёте мини-опрос среди одноклассников, научитесь анализировать информацию из различных источников, разработаете советы и рекомендации по безопасной работе в интернете.

Цель проекта:

Разработка интерактивной памятки/презентации «Безопасный интернет» для сверстников.

При выполнении проекта вы:

изучите основные виды интернет-угроз, с которыми могут столкнуться учащиеся;

проведёте мини-опрос среди одноклассников по теме «Как мы используем Интернет»;

научитесь анализировать информацию из различных источников;

разработаете полезные советы и рекомендации;

оформите материалы в виде презентации и памятки.

Технические средства/оборудование:

Компьютеры с выходом в Интернет;

Программное обеспечение: текстовый редактор (MS Word), программа для создания презентаций (PowerPoint);

Ход выполнения проекта

При выполнении проекта важно соблюдать правила безопасного поведения!

Шаг 1. Изучение интернет-угроз

Основные виды угроз: фишинг (обман с целью выманивания паролей), кибербуллинг (травля, оскорбления, угрозы в чатах), вредоносные программы (вирусы), нежелательный контент. Рекомендуется использовать проверенные источники: сайт «Детский правовой сайт».



<https://mir.pravo.by>

Вопросы для самопроверки: Было ли с вами что-то подобное в сети? Знаете ли вы, что такое фишинг и кибербуллинг? Кому вы расскажете, если столкнётесь с проблемой в сети?

Необходимо создать «щит и меч» для каждого ученика: интерактивную памятку и презентацию «Безопасный интернет», которые научат распознавать угрозы и защищаться.

Шаг 2. Распределение ролей

Распределите следующие роли (каждый отвечает за своё направление на всех этапах проекта): исследователь (поиск информации об угрозах), аналитик (обработка результатов опроса), дизайнер (оформление памятки и презентации).

Шаг 3. Поэтапное выполнение проекта

Это самый важный и большой этап. Выполняем его шаг за шагом.

Шаг 3.1: Проведение опроса

Создайте анонимный опрос в Google Forms или на бумаге. Используйте вопросы из учебника (см. в конце). Важно: обещайте одноклассникам, что результаты будут общими и без имён!

Проведите опрос. Вежливо попросите одноклассников уделить 5 минут.

Соберите все ответы. Посчитайте, сколько человек сколько времени проводят в сети? Сколько уже сталкивались с проблемами?

Шаг 3.2: Сбор и анализ информации

Использовать проверенные источники. Составить для каждой угрозы памятку: как выглядит, чего хочет злоумышленник, что делать.

Необходимую информацию можно получить на сайте «Детский правовой сайт» <https://mir.pravo.by/edu/internet-i-rebenok>.



<https://mir.pravo.by/edu/internet-i-rebenok>

Шаг 3.3: Подготовка текстов и материалов

На основе анализа опроса и исследований создайте:

Список из 5-7 обязательных правил. Они должны быть короткими и ясными. Например: «Правило 1: Пароль — как зубная щётка. Не давай никому и меняй раз в три месяца». Алгоритм действий в случае беды. Что делать, если тебя травят? Если выманивают пароль? План: «Стоп. Не отвечай. Сохрани скриншот. Расскажи взрослому».

Список тех, кто всегда поможет: родители, школьный психолог, учитель информатики, телефон доверия.

Шаг 3.4: Создание презентации и памятки

Для памятки:

Возьмите шаблон.

Используйте крупные заголовки, иконки, короткие абзацы. Не делайте «много текста».

Для презентации (6-8 слайдов): на слайде «Результаты опроса» вставить диаграмму, на остальных слайдах — главная мысль и картинка.

Шаг 4. Защита проекта

Подготовить сценарий выступления. Начало (30 сек): «Представьте, что завтра вам в личку пишет незнакомец. Вы знаете, что делать? Мы узнали и сейчас научим вас!» Середина (3-4 мин): Покажите ключевой слайд с результатами опроса. Продемонстрируйте самые важные правила...». Задайте вопрос классу: «Как вы думаете, какой пароль надёжнее: '12345' или 'K0т_В_Тапк@x'?»». Финал (1 мин): Раздайте или покажите на экране вашу памятку. Скажите: «Мы сделали эту шпаргалку для вас».

Говорите громко, чётко, смотрите на одноклассников, а не в экран.

Ожидаемые результаты

В результате проделанной работы вы освоите принципы безопасной работы в интернете, выработаете будущие стратегии поиска необходимой информации, научитесь работать в команде, совместно планировать и выполнять работу, представлять её результаты. Кроме этого, вы разработаете памятку по правилам безопасного поведения в интернете.

Приложение. Твои инструменты для проекта

Пример опросника:

Опрос «Твой цифровой портрет»:

1. Сколько времени ты обычно проводишь в интернете после школы?

Меньше часа (я активный оффлайнер)

1-2 часа (для дел и развлечений)

3-4 часа (уже привычка)

Больше 4 часов (сеть — мой второй дом)

2. Зачем ты чаще всего заходишь в сеть? (можно несколько)

Играть

Смотреть видео (YouTube, TikTok)

Учиться (искать информацию, на «Скайсмарт»)

Общаться (ВКонтакте, Telegram, Discord)

Просто листать ленту

3. Случалось ли, что в интернете тебе было неприятно или страшно?

Нет, никогда

Да, мне писали грубости или угрожали (кибербуллинг)

Да, меня пытались обмануть, выманить данные (фишинг)

Да, я попадал(а) на странные или пугающие сайты

Другое: _____

4. Если случится проблема в сети, к кому ты пойдёшь в первую очередь?

К родителям

К лучшему другу/подруге

К классному руководителю или психологу

Разберусь сам(а)

Ни к кому, постесняюсь

ПЯТЬ СУПЕРПРАВИЛ:

Пароль: Делай длинным (12+ символов) и уникальным для каждого сайта.

Данные: Домашний адрес, номер школы, фото документов — ТОЛЬКО ДЛЯ СЕБЯ.

Незнакомцы: В сети — те же правила, что на улице. Не доверяй, не встречайся.

Скриншот — твой друг: При травле — сохрани доказательства (скрин, ссылку).

Думай перед публикацией: Выложил — не удалишь. Интернет помнит всё!

ВИРТУАЛЬНЫЙ ПИТОМЕЦ: ОБУЧАЮЩЕЕ ИНТЕРАКТИВНОЕ ПРИЛОЖЕНИЕ В СРЕДЕ ПРОГРАММИРОВАНИЯ SCRATCH

Игры с виртуальными питомцами популярны уже давно. В таких играх необходимо кормить, поить, играть и заботиться о персонаже. Данный проект позволяет создать собственное приложение с подобной механикой в среде программирования Scratch.

Выполняя данный проект, вы познакомитесь с основами визуального программирования, научитесь программировать логику поведения персонажа, работать со спрайтами, фонами, переменными и событиями.

Цель проекта: создание игры «Виртуальный питомец» с возможностью кормить, поить, играть и наблюдать за реакциями виртуального персонажа.

При выполнении проекта вы:

познакомитесь с интерфейсом и основными блоками Scratch;
разработаете алгоритм поведения виртуального питомца;
создадите спрайты, фоны, анимации;
реализуете логику взаимодействия (кнопки, счётчики, состояния);
протестируете работу программы и устраните ошибки;
оформите и презентуете готовое приложение.

Технические средства/оборудование:

Персональный компьютер с установленным Scratch 3.0 или доступом к <https://scratch.mit.edu>;
графический планшет/мышь (по желанию);
микрофон и наушники (для записи/воспроизведения звуков — по необходимости).



<https://scratch.mit.edu>

Ход выполнения проекта

При выполнении проекта важно соблюдать правила безопасного поведения!

Шаг 1. Погружение в проблему

Посмотрим на известные игры с питомцами (тамагочи, Talking Tom). Что в них наиболее интересно: милая анимация, смешные реакции, чувство ответственности?

Сформулируем задачу: как сделать так, чтобы на экране жило существо, которое реагирует на ваши действия.

Наша цель — создать в Scratch интерактивное приложение "Виртуальный питомец" с 3-4 действиями (кормить, играть, укладывать спать) и видимыми изменениями его состояния.

Как выполнить (твои шаги):

Заведите «Блокнот геймдизайнера». Это ваш главный рабочий документ, куда вы будете записывать все идеи, решения и расчёты. Ниже приведён пример заполнения такого блокнота.

Пример «Блокнота геймдизайнера» (заполненный образец):

Страница 1. Мой питомец

Имя питомца: Дракоша-сладкоежка.

Вид: маленький фиолетовый дракон с большими глазами.

Характер: любит сладкое, капризный, когда голоден, очень активный, когда сыт.

Страница 2. Что он умеет и чего хочет

Потребности (три шкалы):

Голод (0 — очень голоден, 10 — сыт).

Настроение (0 — грустный, 10 — счастливый).

Энергия (0 — спит, 10 — полон сил).

Страница 3. Как он показывает своё состояние

Когда голоден (Голод меньше 3): уши опущены, живот урчит (звук), глаза грустные.

Когда сыт (Голод больше 7): улыбается, пускает дымок из носа.

Когда настроение плохое (меньше 3): отворачивается от игрока, плачет.

Когда энергия низкая (меньше 3): зевает, медленно моргает, через 10 секунд засыпает.

Страница 4. Действия игрока (кнопки)

Кнопка «Покормить»: увеличивает Голод на 3. Если Голод становится больше 7, то Настроение тоже увеличивается на 1.

Кнопка «Поиграть»: увеличивает Настроение на 2, но уменьшает Энергию на 2.

Кнопка «Уложить спать»: восстанавливает Энергию до 10 (если она была меньше 5) или до максимума.

Страница 5. Что будет, если за питомцем не ухаживать

Каждые 30 секунд Голод уменьшается на 1. Если Голод падает до 0, питомец «болеет» (анимация грусти, кнопки не работают 10 секунд).

Каждые 60 секунд, если Голод меньше 3, Настроение уменьшается на 1.

Если Энергия падает до 0, питомец засыпает прямо во время игры на 15 секунд.

После заполнения блокнота выберите, какой питомец будет у вас. Это не обязательно кот или собака! Это может быть дракоша-сладкоежка, инопланетянин-непоседа, привидение-хохотун или робот-спортсмен.

Шаг 2: Организация деятельности

Объединитесь в команду (3-4 человека). В группе распределите роли: художник (спрайты и фоны), алгоритмист (логика переменных и кнопок), звукорежиссёр (звуковое сопровождение), презентатор (тестирование и защита).

Художник (Графический дизайнер): Рисует (или собирает из библиотеки Scratch) все визуальные элементы: самого питомца в разных состояниях, кнопки, красивый задний план.

Алгоритмист: Продумывает и программирует логику работы приложения.

Звукорежиссёр: Отвечает за звуковое сопровождение. Подбор или запись звуков: "ням-ням" при кормлении, веселую музыку для игры, храп для сна.

Презентатор и тестировщик: Тестирует игру, находит баги, подготавливает презентацию.

Шаг 3. Проектирование алгоритма

Соберите всю команду и ответьте на вопросы:

Что будет, если питомца не кормить? Переменная Голод уменьшается. Если Голод = 0, питомец грустит и вяло двигается.

Что будет, если с ним постоянно играть? Переменная Энергия уменьшается. Если Энергия = 0, питомец засыпает прямо во время игры.

Как параметры влияют друг на друга?

(Например, если Голод низкий, то и Настроение падает).

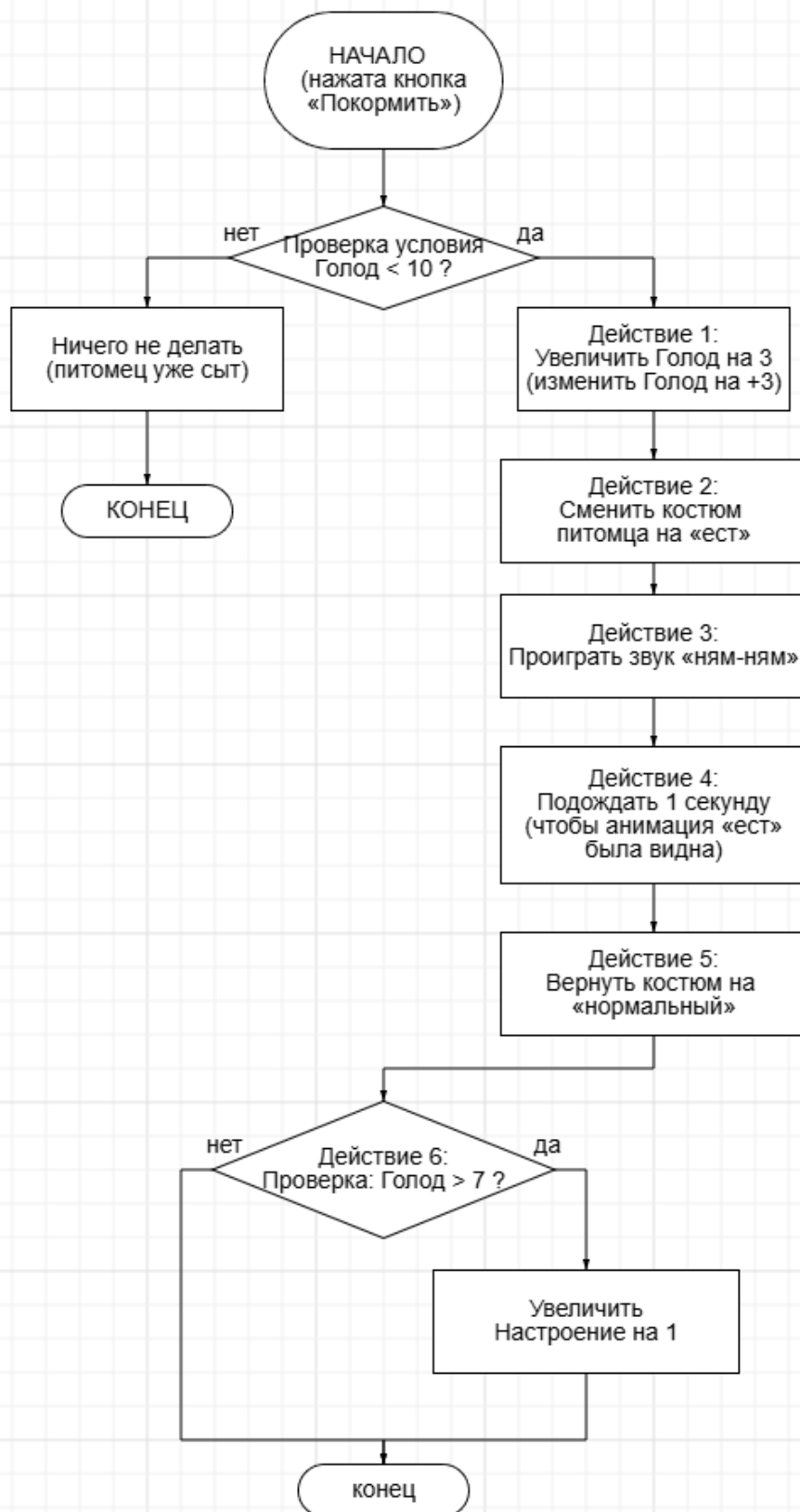


Рис. 1 Блок-схема. Проектирование алгоритма

Проанализируйте блок-схему. По аналогии составьте блок-схемы для кнопок «Поиграть» и «Уложить спать» и для основного цикла уменьшения Голода со временем.

Шаг 4. Реализация в «Scratch»

Графика и интерфейс

Создайте спрайты:

"Питомец": Сделай для него несколько костюмов: "нормальный", "радостный", "грустный", "спит", "ест".

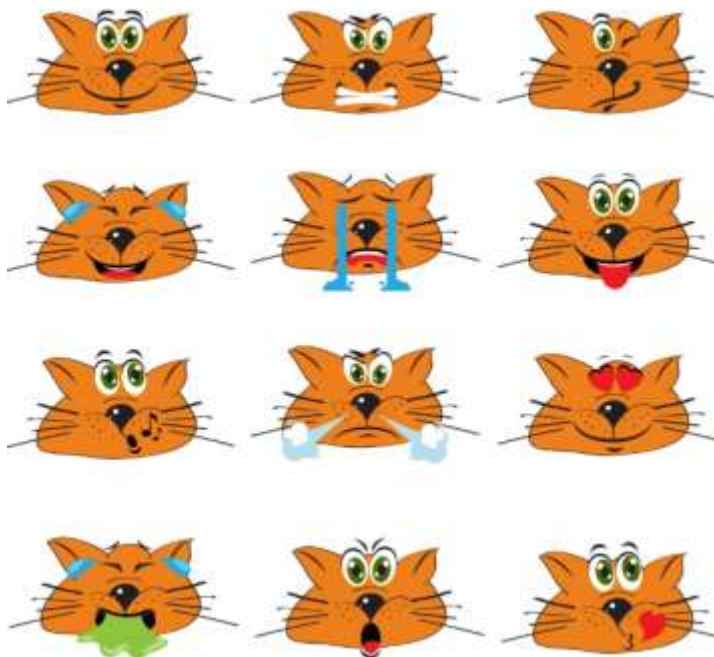


Рис. 2. Пример спрайта состояний питомца

Спрайты-кнопки: "Кормить", "Играть", "Спать". Их можно нарисовать или взять из библиотеки.



Рис. 3. Пример спрайта кнопки «Играть»

Создайте красивый фон. Может быть, уютная комната или фантастический пейзаж.



Рис. 4. Пример спрайта «Комната»

Программирование логики

Создайте переменные. В меню "Переменные" сделай Голод, Настроение, Энергия. Поставь галочки, чтобы они были видны на экране.

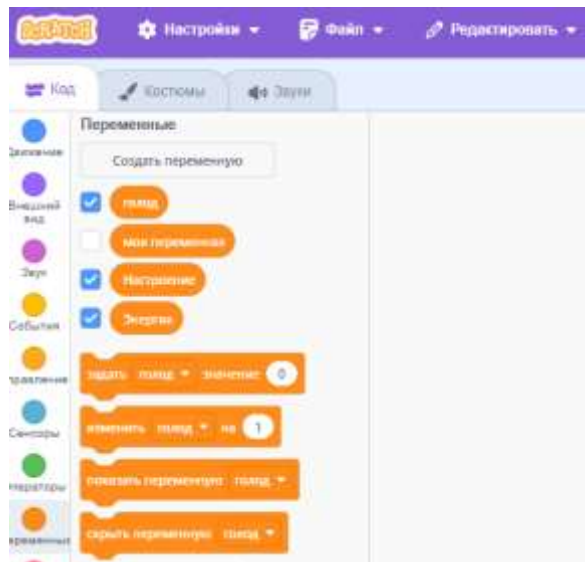


Рис.5. Окно создания переменных в Scratch. Созданы переменные «Голод», «Настроение», «Энергия», напротив каждой стоит галочка.

Скрипт для зелёного флага (инициализация и таймер голода)

Этот скрипт запускается при нажатии на зелёный флаг. Он устанавливает начальные значения переменных и запускает цикл, который каждые 30 секунд уменьшает голод на 1.



Рис. 6. Скрипт инициализации. При запуске проекта переменные получают значение 8, питомец принимает нормальный вид, а затем каждые 30 секунд голод уменьшается на 1.

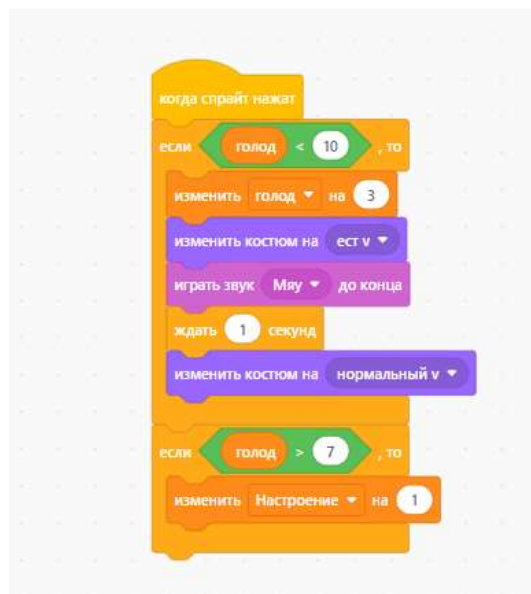


Рис. 7. Скрипт обработки нажатия кнопки «Покормить». Если голод меньше 10, он увеличивается на 3, проигрывается анимация и звук. Если после этого голод стал больше 7, настроение повышается на 1.

Теперь по аналогии добавьте реакцию питомца на низкие показатели. Создайте отдельный скрипт, который постоянно проверяет состояние:

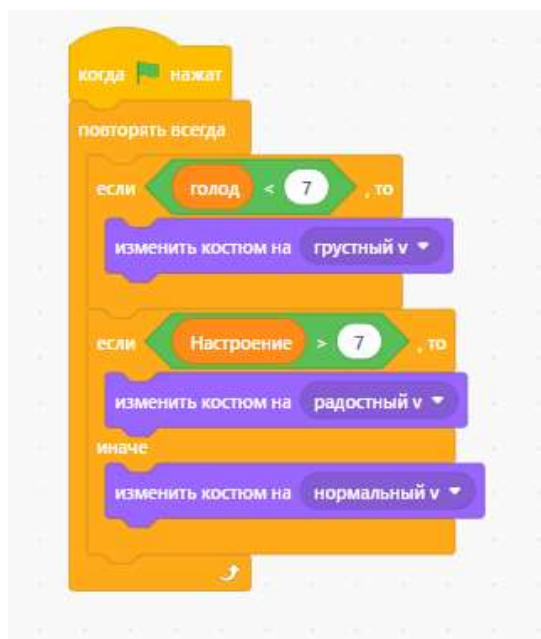


Рис. 8. Скрипт контроля состояния питомца. Если голод падает ниже 7, питомец становится грустным. Если настроение выше 7 — радостным. В остальных случаях — нормальным

По аналогии создаём необходимую механику игры, программируем состояния, кнопки, меняем спрайты настроения.

Проверяем выполнения алгоритмов.

Шаг 5: Подготовка и защита проекта

Создайте презентацию (3-4 слайда в Canva или Power Point):

Слайд 1: Крупное название игры и ваш питомец. Девиз: "Мы создали жизнь в цифре!"

Слайд 2: Механика игры.

Слайд 3: Живая демонстрация. Запустите ваш проект в SCRATCH на большом экране и продемонстрируйте его в живую. покормите питомца, поиграйте с ним.

Слайд 4: Фото/имена команды. И финальный вопрос классу: "А какого питомца создали бы ВЫ?".

Ожидаемые результаты

Выполняя проект, вы изучите основы визуального программирования в среде Scratch, научитесь работать в команде, выдвигать и обсуждать идеи, отстаивать свою точку зрения, представлять результаты совместной работы. В результате у вас получится готовое интерактивное приложение — виртуальный домашний питомец.

ИНТЕРАКТИВНАЯ ТАБЛИЦА МЕНДЕЛЕЕВА В СРЕДЕ SCRATCH

Выполняя данный проект, вы создадите интерактивную цифровую модель таблицы Менделеева, где каждый химический элемент «оживает»: при нажатии на него появляется карточка с информацией — название, символ, атомный номер, интересный факт. Вы не только закрепите знания по химии, но и научитесь структурировать данные в программировании с помощью списков. Результатом станет работающий проект в Scratch, который можно будет продемонстрировать на экране.

Цель проекта:

Создание в среде Scratch интерактивной модели фрагмента таблицы Менделеева (8–10 химических элементов), позволяющей при клике на элемент отображать информацию о нём.

При выполнении проекта вы:

изучите свойства выбранных химических элементов (символ, атомный номер, группа, интересные факты);

освоите структурирование данных в Scratch с помощью списков (базы данных);

создадите спрайты для элементов таблицы и информационной карточки;

запрограммируете алгоритм отображения информации при клике на элемент;

добавите визуальные и звуковые эффекты;

подготовите презентацию проекта.

Технические средства / оборудование / материалы:

Компьютер с доступом в Интернет;

Среда программирования Scratch 3.0 (онлайн-редактор:

<https://scratch.mit.edu>);

Таблица Менделеева (бумажная или цифровая);

Справочные материалы: атомные массы, температуры плавления, интересные факты об элементах;

Бумага, карандаш, линейка (для эскиза расположения элементов);

Программа для презентаций (PowerPoint, LibreOffice Impress).

Ход выполнения проекта

При выполнении проекта важно соблюдать правила безопасного поведения!

Шаг 1: Погружение в проблему. От скучной таблицы к цифровому миру

Обсудим, что сложного в запоминании элементов. Согласитесь, просто смотреть на клетки с буквами — не очень вдохновляет? Посмотрим на крутые цифровые приложения и игры, которые делают учёбу интересной. Как превратить статичную таблицу в интерактивное приложение, где каждый элемент "отвечает" на твой клик? Наша цель — создать в Scratch интерактивную модель таблицы Менделеева, где при клике на элемент появляется карточка с ключевой информацией о нём.

Как выполнить (твои шаги):

Не пытайся охватить все 118 элементов сразу! Выбери 10-15 самых интересных или важных:

Известные (Золото Au, Кислород O, Углерод C)

Опасные (Ртуть Hg, Уран U)

Технологические (Кремний Si для процессоров, Литий Li для батареек)

Придумай, как они будут реагировать при нажатии: "Когда я кликаю на Железо (Fe), оно должно показать карточку, где написано: 'Я — основа стали! Меня много в метеоритах!'".

Шаг 2: Организация деятельности. Распределение ролей в группе

Объединитесь в группы по 3–4 человека. Распределите роли следующим образом.

Химик-исследователь отвечает за сбор и проверку информации об элементах. Он находит точные данные (символы, атомные номера, факты), заполняет итоговую таблицу, следит за достоверностью информации. На всех этапах он проверяет, не появилось ли ошибок в данных.

Программист (Scratch-разработчик) отвечает за реализацию проекта в среде Scratch. Он создаёт списки для хранения данных, программирует спрайты-кнопки, настраивает отображение информационной карточки. На всех этапах он следит за правильностью кода.

Дизайнер-визуализатор отвечает за графическое оформление: создаёт фон таблицы, рисует или подбирает спрайты для элементов, оформляет информационную карточку (цвета, шрифты, расположение). На всех этапах он делает проект наглядным и эстетичным.

Презентатор отвечает за фиксацию результатов. Он делает скриншоты этапов работы (таблица с данными, списки в Scratch, интерфейс), записывает видео работы проекта (по желанию), готовит презентацию и представляет проект на защите.

Важно: все члены группы участвуют в обсуждении и тестировании. Роль определяет лишь степень ответственности за конкретный блок работы.

Создайте "Техническое задание" (ТЗ) проекта. В общем документе укажите:

Список элементов (например, Н, О, С, Fe, Au, Si, Na, Cl, Al, Cu).

Формат карточки: Что в ней будет? (1. Название, 2. Символ, 3. Номер, 4. Интересный факт).

Визуальный стиль: Какие цвета возьмём за основу?

Шаг 3: Реализация проекта

Шаг 3.1: Проектирование интерфейса

Вместе решите, как будет выглядеть ваша таблица в Scratch.

Вариант А (Простой и эффективный): Создаёте один спрайт в виде большой сетки-фона. На нём рисуете или размещаете отдельные спрайты-кнопки для каждого элемента.

Вариант Б (Более продвинутый): Создаёте множество одинаковых спрайтов-"клеточек", которые клонируются и размещаются в нужных позициях, образуя таблицу.

Дизайнер, нарисуйте макет на бумаге: где будет сетка, где — место для всплывающей карточки.

Шаг 3.2: Создание графики

Создайте или найдите в библиотеке Scratch спрайты для элементов. Это могут быть:

- простые цветные квадраты/кружки с символом элемента внутри;
- стилизованные иконки (например, кусочек золота для Au, батарейка для Li).

Создайте спрайт "Информационная карточка". Это окно, которое будет появляться при клике. Оно должно быть чистым и читаемым. Подготовьте красивый фон для сцены (Stage).

Шаг 3.3: Программирование логики

Разберём процесс программирования на примере химического элемента «Водород». Для остальных элементов действия выполняются по аналогии — меняются только названия спрайтов и текст в них.

Что нужно создать перед началом программирования?

Для одного элемента (Водорода) создайте два спрайта:

Спрайт	Назначение	Видимость в начале
«Атом водорода»	Кнопка, по которой щёлкает пользователь	Видимый

Спрайт	Назначение	Видимость в начале
«Графический элемент Водород»	Карточка с информацией о водороде	Невидимый

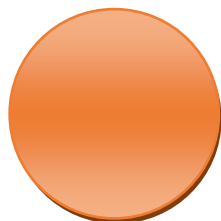


Рис. 1 «Атом водорода»



Рис. 2 «Графический элемент Водород»

Программирование спрайта «Атом водорода»

Теперь напишем скрипт для спрайта «Атом водорода». Логика простая: Пользователь щёлкает по спрайту.

Спрайт «Атом водорода» исчезает (становится невидимым).

Спрайт «Графический элемент Водород» появляется (становится видимым).

Дополнительно: проигрывается звук (по желанию).

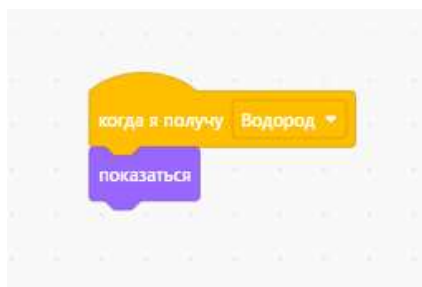
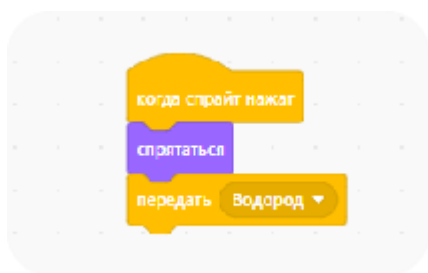


Рис. 3 Скрипты для спрайта «Атом водорода» (при нажатии прячется и отправляет сообщение) и для спрайта «Графический элемент Водород» (получив сообщение, появляется)

Тестирование. Проверить все элементы: кликается ли спрайт, появляется ли карточка, соответствует ли информация элементу. Упростить, если что-то не работает.

После того как вы запрограммировали «Водород», добавьте остальные элементы (Кислород, Углерод, Железо и т.д.) точно по той же схеме. Меняются только:

Названия спрайтов (например, «Атом кислорода», «Графический элемент Кислород»)

Сообщения (например, «показать кислород»)

Содержание информационного спрайта (название, символ, номер, факт)

Шаг 4: Подготовка презентации.

Создайте презентацию (4-5 слайдов):

Слайд 1: Проблема и решение. "Таблица Менделеева казалась скучной. Мы сделали её живой и интерактивной!"

Слайд 2: Наша методология. Покажите скриншот в Scratch и объясните: "Мы создали цифровую базу данных элементов — вот её сердце".

Слайд 3: Живая демонстрация — ГЛАВНОЕ! ЗАПУСТИТЕ ВАШ ПРОЕКТ В SCRATCH НА БОЛЬШОМ ЭКРАНЕ. Продемонстрируйте, как вы кликаете на 3-4 разных элемента и получаете информацию. Сделайте это динамично!

Слайд 4: Чему мы научились. "Мы не только узнали про элементы, но и поняли, как устроены базы данных и интерактивные интерфейсы".

Ожидаемые результаты

Рабочий интерактивный проект в Scratch, представляющий собой фрагмент таблицы Менделеева с функцией просмотра свойств элементов.

Краткая устная презентация проекта.

Понимание связи между структурой данных (списки в программировании) и систематизацией научной информации.

УМНЫЙ ПОМОЩНИК: КАК РАБОТАЕТ РОБОТ-ПЫЛЕСОС

Вы когда-нибудь задумывались, как робот-пылесос самостоятельно объезжая комнату, не врезаясь в стулья и диваны? Секрет не в магии, а в хорошо продуманном алгоритме — пошаговой инструкции, которая управляет движением робота.

В этом проекте вы станете настоящим программистом: вы напишете программу для виртуального робота, который сможет самостоятельно перемещаться по полю, объезжать препятствия и выполнять поставленную задачу.

Цель проекта: создание программы для исполнителя Робот в среде PascalABC.NET, моделирующей алгоритм обхода препятствий и перемещения по заданному маршруту.

При выполнении проекта вы:

- познакомитесь со средой программирования PascalABC.NET;
- освоите базовые команды управления исполнителем Робот;
- научитесь использовать условные операторы (if-else) для принятия решений;
- освоите циклы для повторяющихся действий;
- создадите программу, которая позволит роботу самостоятельно находить выход из лабиринта.

Оборудование и необходимые материалы:

Компьютер с установленной средой PascalABC.NET .

Проектор/интерактивная доска (для демонстрации).

Распечатки с заданиями для исполнителя Робот.

Бумага, карандаш, линейка (для построения алгоритма).

Ход выполнения проекта

При выполнении проекта важно соблюдать правила безопасного поведения!

Любая работа над программой начинается с изучения среды и команд. При выполнении проекта важно соблюдать последовательность шагов и фиксировать промежуточные результаты.

Шаг 1. Погружение в проблему

Изучите «органы чувств» робота: Инфракрасные (ИК) датчики и «бамперы» для обнаружения препятствий вплотную. Лидар (LIDAR): лазерный «радар», который строит точную карту окружения. Гироскоп и одометр: чтобы понимать, насколько и куда он повернулся.

Разберитесь с алгоритмами навигации. Случайное движение. Просто, но неэффективно. Движение по спирали или зигзагу (лавинная уборка) систематично, но можно пропустить углы. Построение карты (SLAM): сначала робот строит карту, потом убирает по ней — самое продвинутое. Создайте совместный документ с выводами. Кратко, с картинками: «Мы выяснили, что для нашей модели, где точная карта не нужна, достаточно алгоритма зигзага с датчиками касания».

Теперь перейдем к программированию и познакомимся со средой PascalABC.NET и исполнителем Робот.

Что такое PascalABC.NET?

PascalABC.NET — это современная среда программирования, которая используется в школах для обучения основам алгоритмизации и программирования. В неё встроены специальные учебные исполнители, в том числе Робот.

Кто такой исполнитель Робот?

Робот — это виртуальный исполнитель, который действует на прямоугольном клеточном поле. Между некоторыми клетками находятся стены. Робот может:

перемещаться по клеткам;

закрашивать клетки;

проверять, есть ли стена впереди, справа, слева, снизу.

Основные команды Робота:

Команда	Что делает
Up;	Перемещает Робота вверх
Down;	Перемещает Робота вниз
Left;	Перемещает Робота влево
Right;	Перемещает Робота вправо
Paint;	Закрашивает текущую клетку

Команды для проверки условий (возвращают True или False):

Команда	Что проверяет
WallFromLeft	Есть ли стена слева от Робота

Команда	Что проверяет
WallFromRight	Есть ли стена справа от Робота
WallFromUp	Есть ли стена сверху от Робота
WallFromDown	Есть ли стена снизу от Робота
FreeFromLeft	Свободно ли слева от Робота
FreeFromRight	Свободно ли справа от Робота
FreeFromUp	Свободно ли сверху от Робота
FreeFromDown	Свободно ли снизу от Робота

Шаг 2. Организация деятельности. Проектирование алгоритма и среды
Объединитесь в группы по 2–3 человека. Распределите роли:

Алгоритмист отвечает за разработку логики движения. Он составляет блок-схему или словесный план того, как робот должен обходить препятствия.

Программист отвечает за написание кода на PascalABC.NET. Он переводит алгоритм в команды для Робота.

Тестировщик отвечает за проверку программы. Он запускает программу на разных полях, находит ошибки, предлагает улучшения.

Важно: все члены группы участвуют в обсуждении и поиске решений.

Шаг 3. Пошаговое выполнение проекта

Шаг 3.1. Первая программа: простое движение.

Научим Робота перемещаться из одной точки в другую.

Шаблон любой программы для Робота:

```
uses Robot;           // подключаем модуль с Роботом
begin Task('c1');     // выбираем задание (поле для Робота)
    .....             // Здесь будут команды для Робота
end.
```

Пример 1. Перемещение на 2 клетки вправо:

```
Begin
  Task('c1');
  Right;
  Right;
end.
```

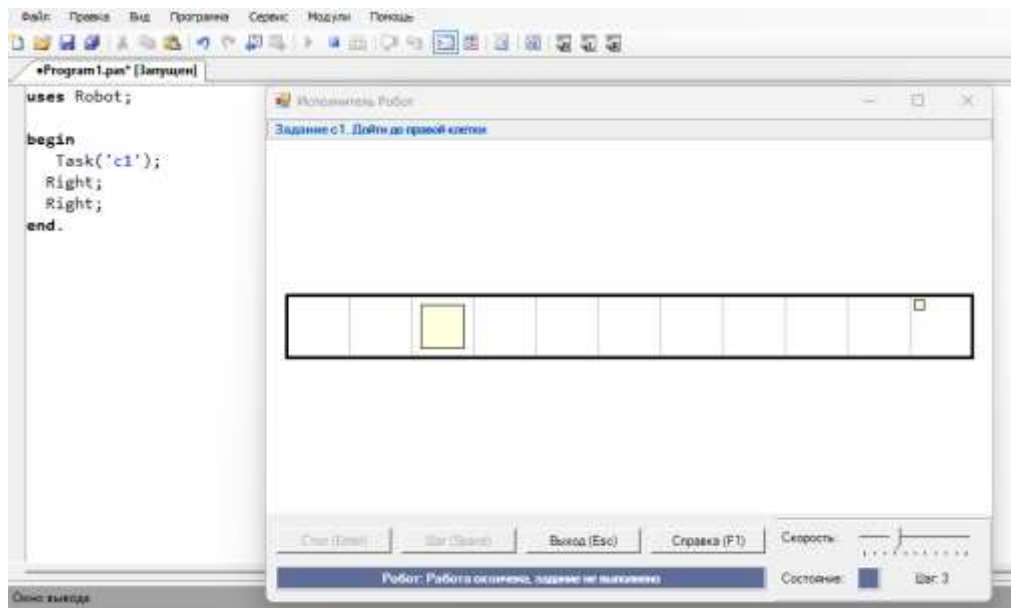


Рис.1. Робот переместился на две клетки вправо.

Шаг 3.2. Использование циклов для повторяющихся действий

Если нужно повторить команду много раз, удобно использовать цикл.

Пример 2. Закрашивание линии из 5 клеток:

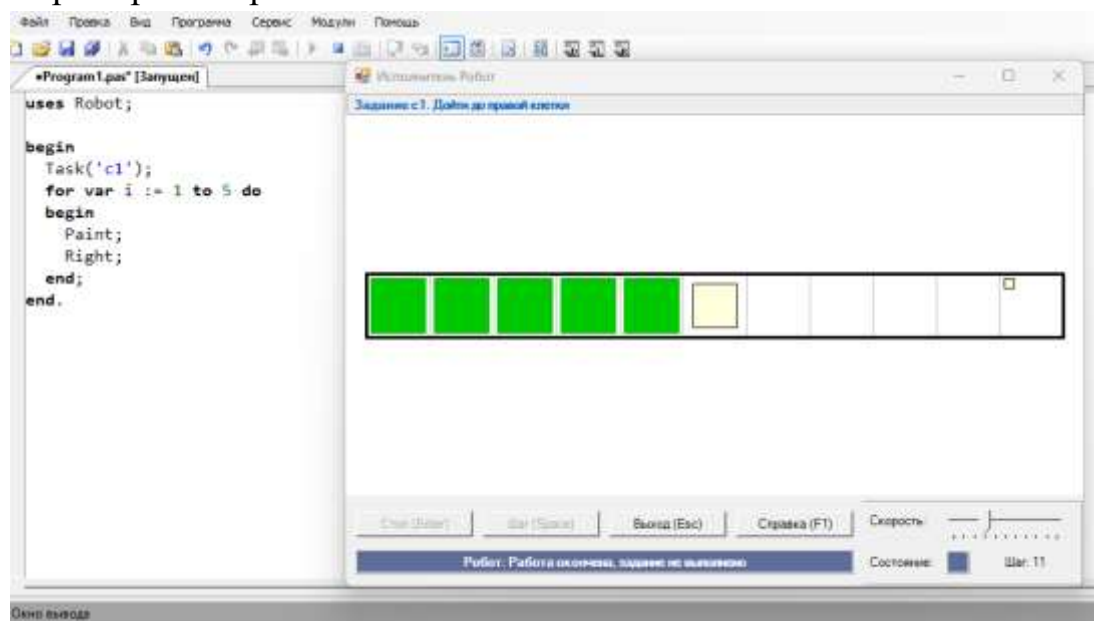


Рис. 2. Робот закрасил 5 клеток подряд с помощью цикла.

Объяснение: `for var i := 1 to 5 do` означает «повторить 5 раз». Внутри цикла Робот сначала закрашивает клетку, затем двигается вправо.

Шаг 3.3. Условный оператор: принимаем решения

Роботу нужно уметь реагировать на препятствия. Для этого используется условный оператор `if`.

Блок-схема условного оператора:

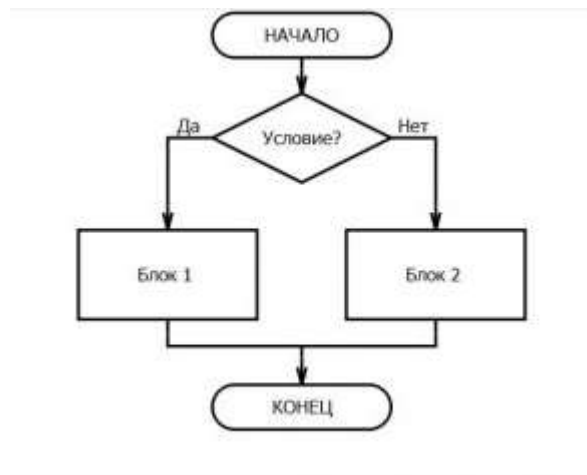


Рис. 3. Блок-схема условного оператора.
Ромб «Условие» → две ветки «ДА» и «НЕТ»

Пример 3. Если справа стена — повернуть налево, иначе — двигаться вправо:

```
uses Robot;  
begin Task('c1');  
  if WallFromRight then  
    Left  
  Else  
    Right;  
end.
```

Программа проверяет, есть ли стена справа, и принимает решение.

Шаг 3.4. Создание алгоритма «Робот-пылесос»

Теперь объединим все знания и создадим программу, которая заставляет Робота обходить препятствия и закрашивать все доступные клетки — «убирать» поле.

Алгоритм движения (стратегия «правой руки»):

Если робот упирается в стену, он поворачивает налево (или направо) и продолжает движение. Этот простой алгоритм позволяет обойти любое препятствие.

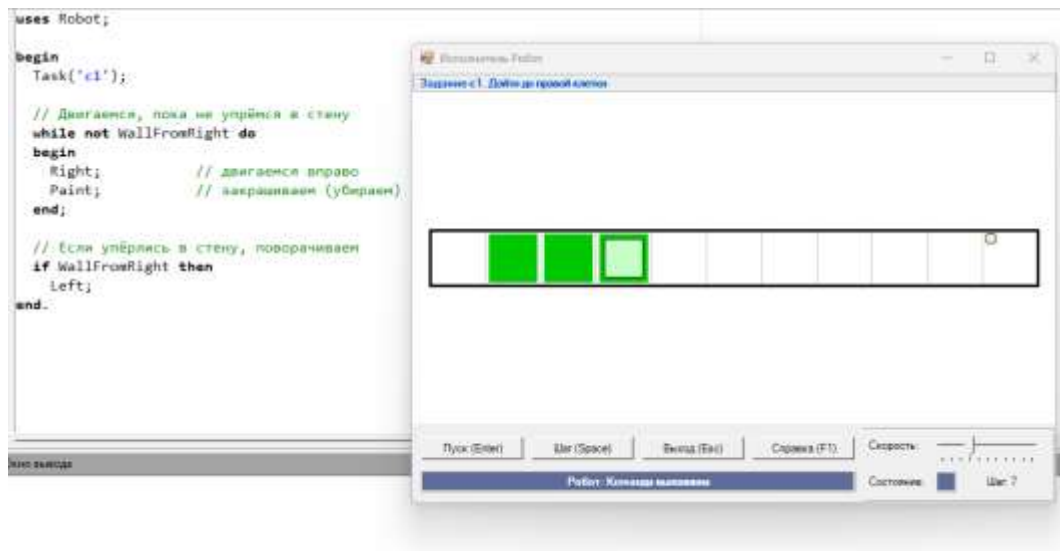


Рис. 4. Программа «Робот-пылесос» (обход препятствий)

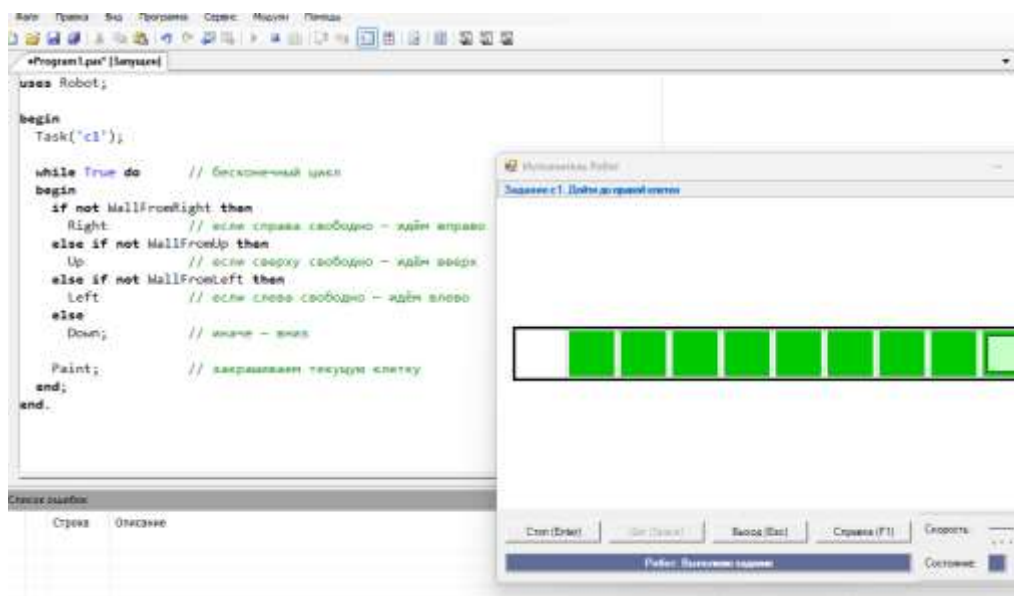


Рис. 5. Улучшенная версия — бесконечное движение с обходом

Шаг 4. Тестирование программы

Запустите симуляцию. Зафиксируйте проблемы: незатронутые углы, заикливание.

Улучшите алгоритм. Обсудите критерии эффективности: скорость покрытия площади, минимизация повторных проходов.

Подготовьте скриншоты/видео для отчёта.

Шаг 5. Подготовка презентации и защита проекта

Создайте презентацию (4–5 слайдов) по следующей структуре:

Слайд 1. Название и команда. «Программирование робота-пылесоса на PascalABC.NET».

Слайд 2. Алгоритм обхода препятствий. Покажите блок-схему.
Расскажите стратегию «правой руки».

Слайд 3. Программный код. Покажите скриншот готовой программы.
Объясните ключевые моменты: uses Robot, циклы, условия.

Слайд 4. Демонстрация. Запустите программу и покажите, как Робот движется по полю.

Слайд 5. Выводы. Что узнали нового? Что было самым сложным? Как можно улучшить программу (добавить счётчик закрашенных клеток)?

Ожидаемые результаты

В результате выполнения проекта вы:

- познакомитесь со средой программирования PascalABC.NET;
- освоите базовые навыки алгоритмизации (циклы, условия);
- научитесь управлять исполнителем Робот;
- создадите программу, моделирующую поведение робота-пылесоса;
- подготовите презентацию и представите проект.

Выполняя данный проект, вы сделаете первый шаг в мир серьёзного программирования. Pascal — это язык, на котором пишут реальные программы, а умение составлять алгоритмы пригодится вам в любом будущем — будь то IT, инженерия или наука.

СОЗДАНИЕ ЦИФРОВОГО ОПРЕДЕЛИТЕЛЯ ЗВУКОВ ЖИВОТНЫХ

Окружающее пространство наполнено многообразием звуковых волн. Современные компьютерные технологии позволяют автоматизировать процесс их анализа и классификации. Данный учебный проект посвящён созданию прототипа системы автоматического распознавания видов животных по их звуковым проявлениям. Работа лежит на междисциплинарной границе биоакустики (раздела биологии, изучающего звуковое общение организмов) и компьютерных наук.

В процессе выполнения проекта вы освоите ключевые этапы обработки аудиосигналов: от первичного сбора и анализа звуковых записей до визуализации аудиоданных в виде спектрограмм, которые отображают частотные характеристики сигнала во времени.

Итогом работы станет функционирующий прототип программы-идентификатора, способный распознавать звуки различных животных. Приобретённые компетенции будут полезны как в области информатики (обработка сигналов, алгоритмизация, основы искусственного интеллекта), так и в биологии, предоставляя инструмент для объективного анализа акустического поведения.

Цель проекта:

Создать прототип программы-определителя звуков животных, которая по загруженному аудиофайлу определяет принадлежность к виду (например: ворона, соловей, лягушка, сверчок, кошка, собака).

При выполнении проекта вы:

изучите цифровое представление звука и основные характеристики аудиофайлов;

познакомитесь с программами для анализа звука (*Audacity*, *Sonic Visualiser*) и библиотеками Python (*librosa*);

соберете небольшую базу аудиофайлов животных (5–6 видов);

выделите признаки звуков: диапазон частот, ритм, длительность;

разработаете упрощённый алгоритм классификации звуков по признакам;

подготовите презентацию с описанием работы программы.

Оборудование и необходимые материалы

Компьютеры с установленными программами для анализа звука: *Audacity*, *Sonic Visualiser* (для спектрограмм), для создания прототипа определителя: *Python* с библиотеками *librosa*, *Tkinter*. Колонки или наушники для прослушивания звуков.

Ход выполнения проекта

При выполнении проекта важно соблюдать правила безопасного поведения!

Шаг 1: Погружение в проблему

Живые организмы существуют в насыщенной звуками среде. Звуковая коммуникация является одним из фундаментальных способов взаимодействия животных. Биоакустика — научная дисциплина, изучающая эти процессы. Каждый вид в процессе эволюции выработал уникальные акустические "подписи" — комбинации частотных, временных и амплитудных характеристик, оптимальные для его экологической ниши и образа жизни.

Необходимо прослушаешь записи разных животных (птица, собака, кошка, лягушка, сверчок). Можно ли заставить компьютер автоматически понимать, кому принадлежит этот звук? Для этого потребуется проанализировать различные звуковые файлы и найти отличия между ними.

Шаг 2: Изучение основ звука

Необходимо из предложенных звуковых файлов выделить основные характеристики. Частота (Гц): определяет высоту звука. Высокие звуки (птицы) имеют высокую частоту, низкие (лягушки) — низкую. Амплитуда: определяет громкость.

Длительность: как долго длится звук. Спектрограмма: это «фотография» звука, где по вертикали — частота, по горизонтали — время, а цветом показана громкость.

Животные издают звуки не просто так. Это их язык: для привлечения партнера, защиты территории или предупреждения об опасности.

Для более продуктивной работы необходимо разделить направления работы:

Биолог: отвечает за изучение видов животных и их повадок.

Звукоинженер: работает с аудиофайлами, строит спектрограммы.

Программист: пишет код алгоритма.

Дизайнер/Докладчик: готовит презентацию и интерфейс.

Необходимо открыть программу Audacity. Затем загрузите любой звуковой файл и найдите функцию «Спектрограмма». Посмотрите, как выглядят знакомые звуки.

Попробуйте определить на спектрограмме, у какого звука частота выше, а у какого ниже. Запишите длительность и ритм нескольких звуков.

Шаг 3: Сбор базы данных

Используя набор аудиофайлов с подписями: ворона.wav, лягушка.wav и т.д. запустите Audacity или аналогичную программу, откройте каждый файл

и постройте его спектрограмму.

Вместе с командой заполните общую таблицу.

Таблица. Пример таблицы для заполнения

Животное	Частотный диапазон (Гц)	Ритм	Длительность
Ворона	400–2000	одиочные крики	0,5–1 с
Соловей	2000–6000	мелодичные трели	1–3 с
Лягушка	100–500	повторяющиеся «ква»	0,5–1 с
Сверчок	2000–4000	быстро повторяющиеся щелчки	0,2–0,5 с
Кошка	300–1000	отдельные мяукания	0,5–1 с
Собака	200–1500	ритмичное гавканье,	0,5–2 с

Шаг 4: Анализ аудио и выделение признаков

Сравните спектрограммы из своей базы. Положите их рядом. Найдите закономерности и запишите их. Например: птица соловей, его звуки на спектрограмме находятся высоко (частоты >2000 Гц) и выглядят как сложные узоры. Лягушки: их звуки находятся низко (100-500 Гц) и выглядят как повторяющиеся «столбики» с паузами. Сверчки: это сплошная «щетка» из очень частых повторяющихся сигналов на высоких частотах (~ 3000 Гц).

Необходимо определить, по каким трём главным признакам будет отличаться звук одного животного от другого. (Подсказка: Частотный диапазон, Ритм, Длительность).

Шаг 5. Написание программы на Python

Шаг 5.1. Как компьютер «видит» звук

Звук в компьютере хранится как последовательность чисел (отсчётов). Каждое число показывает амплитуду звука в определённый момент времени.

Частота звука — это количество колебаний в секунду (измеряется в Герцах).

Для упрощения в нашей программе мы не будем делать точный спектральный анализ. Вместо этого мы зададим диапазоны частот для каждого животного вручную (на основе наблюдений в Audacity). Программа будет просто сравнивать среднюю частоту с этими диапазонами.

Шаг 5.2. Общая структура программы

```
import wave
import struct
import math
def read_wav(filename):
    """Читает WAV-файл и возвращает список отсчётов и частоту дискретизации"""
    wav = wave.open(filename, 'r')
    frames = wav.getnframes()
    rate = wav.getframerate()
    data = wav.readframes(frames)
    wav.close()
    # Преобразуем байты в числа (для моно-файла 16 bit)
    samples = struct.unpack('<' + 'h' * frames, data)
    return samples, rate
def estimate_frequency(samples, rate):
    """Простая оценка средней частоты (через количество переходов через ноль)"""
    zero_crossings = 0
    for i in range(1, len(samples)):
        if samples[i-1] * samples[i] < 0: # смена знака — переход через ноль
            zero_crossings += 1
    # Частота = (переходов за секунду) / 2
    duration = len(samples) / rate
    freq = (zero_crossings / duration) / 2
    return freq
def identify_animal(frequency):
    """Определяет животное по частоте звука"""
    if frequency < 500:
        return "Лягушка (низкочастотное кваканье)"
    elif frequency < 1000:
        return "Собака (среднечастотный лай)"
    elif frequency < 1500:
        return "Ворона (карканье в среднем диапазоне)"
    elif frequency < 2000:
        return "Кошка (мяуканье выше среднего)"
    else:
        return "Соловей (высокочастотные трели)"
# Основная программа
```

```

if __name__ == "__main__":
    filename = input("Введите имя WAV-файла: ")
    try:
        samples, rate = read_wav(filename)
        freq = estimate_frequency(samples, rate)
        print(f"Оценённая частота: {freq:.0f} Гц")
        print(f"Определённое животное: {identify_animal(freq)}")
    except Exception as e:
        print("Ошибка при чтении файла:", e)

```

Шаг 5.3. Пояснение кода

Функция	Что делает
<code>read_wav(filename)</code>	Открывает WAV-файл, читает отсчёты (числа), преобразует их из байтов в список чисел.
<code>estimate_frequency(samples, rate)</code>	Оценивает частоту через подсчёт количества переходов через ноль (Zero Crossing Rate). Этот метод прост и не требует сложных библиотек.
<code>identify_animal(frequency)</code>	Сравнивает полученную частоту с пороговыми значениями и возвращает название животного.

Шаг 5.4. Как подготовить тестовые WAV-файлы

Программа работает только с монофоническими WAV-файлами (16 бит, 22050 Гц или 44100 Гц). Как их создать в Audacity:

Откройте звук в Audacity.

Выберите Дорожка →Stereo в моно.

Выберите Файл → Экспорт → Экспортировать как WAV.

В настройках выберите «WAV (Microsoft) 16 bit».

Шаг 5.5. Тестирование программы

Создайте несколько тестовых аудиофайлов (ворона.wav, соловей.wav, лягушка.wav) и проверьте программу:

Шаг 6. Подготовка презентации и защита проекта

Создайте презентацию из 5 слайдов:

Слайд 1. Название и команда. «Цифровой определитель звуков животных на Python».

Слайд 2. Теоретическая часть. Покажите спектрограммы звуков (из

Audacity), таблицу частотных диапазонов. Объясните, как компьютер анализирует звук.

Слайд 3. Алгоритм и код. Покажите блок-схему и ключевые фрагменты кода (скриншоты из Replit/IDLE).

Слайд 4. Демонстрация. Запустите программу на 2–3 тестовых файлах и покажите результат.

Слайд 5. Выводы и перспективы. Что удалось сделать? Что было самым сложным? Как можно улучшить программу (добавить больше животных, использовать машинное обучение)?

Ожидаемые результаты

В результате выполнения проекта вы изучите характеристики звука, познакомитесь с применением Python для обработки аудиофайлов, создадите действующий прототип программы, имитирующей звуки, издаваемые различными животными.

Приложение. Справочные материалы

QR-код для скачивания Audacity:



<https://www.audacityteam.org>

QR-код для открытия онлайн-среды Replit (Python):



<https://replit.com>

БЕЛАРУСКІ АРНАМЕНТ У ЛІЧБАВАЙ ГРАФІЦЫ

Данный проект — это мост между прошлым и будущим. На основе белорусского орнамента мы создадим современный цифровой продукт: открытку, логотип или плакат. Вы не просто нарисуете картинку, вы оживите наследие с помощью компьютерной графики.

Цель проекта:

Создание цифрового изображения белорусского орнамента (узора) с использованием графического редактора и его оформление для дальнейшего применения (в виде постера, открытки, заставки, логотипа).

При выполнении проекта вы:

изучите значение и символику традиционных белорусских орнаментов;
познакомитесь с возможностями графических редакторов для построения узоров и симметричных изображений;
создадите цифровой макет орнамента или узора, используя инструменты графического редактора;
оформите итоговый продукт (цифровой плакат, открытку или логотип);
подготовите и проведете презентацию с обоснованием выбора символики и способов работы.

Оборудование и необходимые материалы

Компьютеры, проектор/интерактивная доска, графические редакторы (Paint.NET, GIMP, Inkscape), доступ к интернет-ресурсам (сайты о белорусской культуре, онлайн-библиотеки), презентационные программы (PowerPoint, LibreOffice Impress).

Ход выполнения проекта

При выполнении проекта важно соблюдать правила безопасного поведения!

Шаг 1. Погружение в проблему. Расшифровка кода предков

Откроем тайный язык: Мы вместе посмотрим на самые известные белорусские узоры. Вы увидите, что в них нет ничего случайного. Ответьте для себя на вопросы:

Это просто «красивые завитушки» или в них есть глубокий смысл?

Где в современном мире можно встретить такие узоры? (На бренде одежды, сувенирах, в логотипах кафе или фестивалей).

Как с помощью компьютера можно сохранить и передать эту красоту новым поколениям?

Сформулируем задачу: Наша цель — создать цифровой продукт (открытку, логотип, плакат) на основе белорусского орнамента.

Как выполнить (твои шаги):

Заведи «Дневник символиста»: Это будет твоя главная тетрадь для проекта.

Начни исследование: Найди в интернете или книгах 3-5 самых известных белорусских орнаментов. Выпиши в дневник, что означают их основные элементы:

Ромб/Квадрат — символ засеянного поля, плодородия, матери-земли.

Круг/Солнце — символ жизни, света, тепла, гармонии.

Восьмиконечная звезда — символ надежды, путеводности.

«Ёлочка»/«Папараць-кветка» — символ роста, развития, связи с природой.

Крест — символ огня, солнца, защиты.

Придумай идею: Какой продукт тебе хочется создать? Может, открытку ко Дню матери с символом плодородия? Или логотип для школьного экологического кружка с «ёлочкой»?

Результат: В твоём дневнике будут расшифрованные символы и первая идея для будущего продукта.

Шаг 2. Организация деятельности. Создаем творческий альянс

Соберите команду единомышленников (3-4 человека). Важно, чтобы вам нравилась общая тема. Распределите роли:

Историк-аналитик: глубоко изучает символику, ищет аутентичные орнаменты, готовит историческую справку.

Художник-иллюстратор: отвечает за эстетику. Придумывает, как традиционные мотивы превратить в современный дизайн, подбирает цвета.

Цифровой мастер (графический дизайнер): технический специалист, который будет работать в программе (Paint.NET, GIMP, Canva), воплощая эскиз в пиксели и векторы.

Арт-директор/Презентатор: следит за целостностью идеи, готовит презентацию и ярко представляет ваш проект аудитории.

Определите продукт: что именно вы создадите всей командой? Выберите одно:

Праздничная открытка (ко Дню Победы, 8 Марта, Новому году).

Логотип (для вашего класса, школьной спортивной команды, театрального кружка).

Плакат (ко Дню родного языка, «Дню вышиванки»).

Обложка для виртуального дневника или альбома.

Составьте техническое задание (ТЗ):

Напишите простой план:

1) Собрать орнаменты;

- 2) Нарисовать эскиз;
- 3) Сделать в программе;
- 4) Оформить презентацию.

Шаг 3. Поэтапное выполнение проекта. От идеи до пикселя
Это самый большой и творческий этап. Разбиваем его на шаги.

Шаг 3.1: Сбор и анализ информации

Создайте «Мудборд» (доску вдохновения): В отдельном документе или на листе ватмана соберите все найденные изображения орнаментов, их описания, цветовые палитры.

Сделайте выбор: Какой конкретный мотив (ромб, солнце, елочка) или их комбинацию вы возьмете за основу? Запишите в ТЗ, почему именно этот выбор (например: «Выбираем солнце и ромб, чтобы передать идею плодородия и света для открытки ко Дню матери»).

Шаг 3.2: Создание эскиза

Не включайте компьютер! Возьмите карандаш и бумагу.

Набросайте композицию. Где будет главный элемент? Как будет построена симметрия? Где разместится текст (если он нужен)?

Поэкспериментируйте с цветом. Помните о традиционной палитре: красный (жизнь, красота), белый (свет, чистота), черный (земля), зеленый (природа, рост). Раскрасьте эскиз.

Шаг 3.3: Работа в графическом редакторе (3-4 часа)

Выберите ваш «цифровой мольберт»:

Paint.NET / GIMP — отлично подходят для растровой графики (как рисунок кистью).

Inkscape / Canva — хороши для векторной графики (четкие линии и формы, которые можно увеличивать без потери качества).

Освойте ключевые инструменты: геометрические фигуры, кисть/карандаш, заливка, ластик, выделение и отражение (для симметрии!), работа со слоями (каждый элемент на отдельном слое — это магия!).

Переносите эскиз в программу: Начните с фона, потом добавьте основные элементы. Используйте копирование и отражение, чтобы добиться идеальной симметрии.

Работайте с цветом: Используйте палитру или «пипетку», чтобы точно подобрать оттенки.

Шаг 3.4: Финальные штрихи и сохранение

Добавьте текст (если нужно): выберите подходящий, легко читаемый шрифт.

Проверьте композицию: Уберите лишнее. Все ли гармонично?

Сохраните правильно:

Основной файл с слоями (.psd, .xcf, .svg) — чтобы можно было исправить.

Итоговое изображение для показа (.png — если нужен прозрачный фон, .jpeg — для фотокачества).

Этап 4. Защита проекта. Твой вернисаж

Создайте короткую, но эффектную презентацию (3-5 слайдов):

Слайд 1: Название и команда. Красиво представьте ваш проект и себя.

Слайд 2: Идея и символика. Покажите орнамент-прототип и расскажите главную мысль: «Мы взяли символ солнца (круг), потому что он означает жизнь и радость, и добавили ромб — символ плодородия, чтобы создать открытку для самого родного человека».

Слайд 3: Процесс создания. Покажите ваш бумажный эскиз и скриншот работы в программе. Это очень интересно!

Слайд 4: Готовый шедевр! Во весь экран продемонстрируйте вашу открытку, логотип или плакат.

Распределите текст доклада: каждый рассказывает о своей части работы. Говорите с увлечением!

Историк: «Мы узнали, что этот узор раньше вышивали на свадебных рушниках...»

Художник: «Мы решили использовать красно-белую гамму, потому что...»

Дизайнер: «Самым сложным было сделать идеальную симметрию с помощью инструмента...»

Отрепетируйте! Уложите в 5-7 минут. Следите за осанкой и интонациями.

Результат этапа: Яркая презентация и уверенная защита вашего культурного проекта.

Ожидаемые результаты

В результате выполнения проекта вы создадите комплексный культурно-цифровой продукт, объединяющий историческое наследие и современные технологии. Результаты проекта являются наглядными, презентуемыми и имеют практическую применимость.

Итоговым продуктом проекта является не просто «картинка», а осмысленный цифровой артефакт, готовый к использованию в современной коммуникативной среде (социальные сети, школьные мероприятия, сувенирная продукция) и представляющий собой результат синтеза культурологического исследования и компьютерного творчества.

ИНФОГРАФИКА: ПРЕДСТАВЛЕНИЕ ДАННЫХ В ВИЗУАЛЬНОЙ ФОРМЕ

Каждый день на нас обрушивается поток информации: цифры, факты, новости, но есть способ не просто рассказать об этой информации, а показать её так, чтобы она была понятна, интересна и запоминалась с первого взгляда. Этот способ — *инфографика*.

В данном проекте каждый может почувствовать себя настоящим медиа-художником, который превращает скучные цифры в яркую историю. Вы научитесь находить данные, анализировать их и создавать визуальные шедевры, которые смогут увлечь друзей, одноклассников и даже учителей.

Реализация проекта позволяет пройти весь путь создания профессионального продукта: от рождения идеи до её публичной защиты. Вы будете искать информацию как детективы, структурировать её как учёные и оформлять, как дизайнеры.

Цель проекта: научиться собирать, анализировать и визуализировать информацию, создавая инфографику по актуальной теме.

При выполнении проекта вы:

соберете и систематизируете статистические данные из разных источников;

освоите основные правила визуализации информации (выбор цвета, шрифта, структуры);

создадите инфографику по выбранной теме, используя цифровые инструменты;

оформите и защитите итоговый продукт в виде презентации.

Оборудование и необходимые материалы

Компьютеры, проектор/интерактивная доска, офисные пакеты (PowerPoint, LibreOffice Calc/Impress), интернет-ресурсы (онлайн-сервисы для инфографики — Canva, Piktochart и др.).

Ход выполнения проекта

При выполнении проекта важно соблюдать правила безопасного поведения!

Шаг 1: Погружение в проблему (индивидуальная работа)

Сначала разберись, что такое инфографика, выбери тему, от которой «загорятся глаза».

Вопросы для выбора темы:

Что быстрее и проще понять: страницу с таблицей из 100 цифр или один яркий график?

Где в жизни мы сталкиваемся с инфографикой?

(Подсказка: карта метро, инструкция к лекарству, график погоды в приложении, посты в Instagram).

Наша общая миссия — создать собственную инфографику на актуальную тему.

Шаг 2: Организация деятельности

Организовуйтесь в группы из 3-4 человек. Распределите роли.

Примерные роли:

Аналитик/Исследователь. Этот человек обожает копаться в фактах. Он ищет достоверные данные, проверяет источники.

Дизайнер/Визуализатор. Это творческий гений. Он придумывает, как будут выглядеть графики, подбирает цвета, шрифты и иконки. У него хороший вкус.

Оформитель/Технический специалист. Этот человек дружит с технологиями. Он именно тот, кто будет работать в программе (Canva, PowerPoint), собирая все идеи в единый цифровой продукт.

Презентатор/Спикер. Представит ваше творение на защите.

(Важно: роли можно совмещать!)

1. Составьте «Дорожную карту». На листе бумаги или в онлайн-документе напишите план:

Что делать? (Найти статистику, провести опрос, нарисовать эскиз, создать в Canva).

Кто отвечает? (Имя участника).

К какому сроку? (Например, «сбор данных до пятницы»).

Результат этапа: У вас есть команда, распределенные роли и план-график работы.

Этап 3: Сбор и анализ информации. В погоне за цифрами

Цель: найти надежную информацию и превратить ее в готовые к визуализации данные.

Как выполнить (твои шаги):

Отправляйся на «охоту» за данными:

Официальные источники: Сайты Белстата (Национальный статистический комитет).



<https://www.belstat.gov.by>

Социологические опросы. Можно провести свой! Создай короткую анкету в Google Forms и узнай мнение одноклассников.

Научные и образовательные порталы: Проверь, чтобы у информации был автор и источник.

Любую цифру, найденную в интернете, старайся проверить по другому источнику. Занеси все найденные данные в единую таблицу (в Google Таблицах или даже в тетради).

Таблица. Пример таблицы для твоей темы «Соцсети»

Вопрос / Параметр	Данные	Примечание
Самая популярная соцсеть в классе	Instagram (60%)	По результатам опроса 25 человек
Среднее время в соцсетях в день	2,5 часа	
Самая быстрорастущая сеть	TikTok	По данным за 2024 год
Источник	Ссылка на ваш опрос, Ссылка на сайт	

Посмотри на свою таблицу и ответь на следующие вопросы. Что эти цифры значат? Какова главная мысль? Например: «Хотя Instagram лидирует, TikTok стремительно набирает популярность среди подростков».

Результат этапа. У тебя есть таблица с проверенными данными и ключевые выводы, которые ты хочешь донести.

Шаг 4: Создание эскиза инфографики. Рисуем макет

Создать «черновик» или «скелет» твоей будущей инфографики, чтобы не запутаться при цифровом оформлении.

Как выполнить (твои шаги):

1. Возьми лист бумаги и карандаш. Не нужно ничего рисовать идеально!

2. Разметь пространство:

Где будет главный заголовок?

Куда ты поместишь самую важную и яркую диаграмму?

Где будут второстепенные графики или факты в рамочках?

Как ты будешь вести взгляд зрителя: сверху вниз или слева направо?

3. Подбери визуальные элементы:

Круговая диаграмма — идеально, чтобы показать доли (проценты).

Столбчатая диаграмма — чтобы сравнить несколько величин.

Иконки — чтобы быстро донести смысл (телефон для времени в сети, земной шар для глобальной статистики).

Стрелки и линии — чтобы связать разные части.

4. Подумай о цвете: набросай, каким цветом ты выделишь главное, а каким — второстепенное.

Результат этапа. У тебя на руках есть бумажный эскиз — понятный план для следующего этапа.

Шаг 5: Создание инфографики в цифровом виде. Волшебство в действии

Цель: вдохнуть жизнь в твой эскиз, создав яркий и профессиональный цифровой продукт.

1. Выбери свой инструмент:

[Canva.com](https://www.canva.com) (рекомендуется): Огромная библиотека шаблонов, иконок, шрифтов. Очень простой и интуитивно понятный интерфейс.



<https://www.canva.com/templates>

PowerPoint: Ты умеешь в нем работать, в нем тоже можно создавать красивые диаграммы и композиции.

2. Следуй эскизу: Открой программу и начинай переносить свой бумажный макет. Не отклоняйся от плана!

3. Помни о золотых правилах. Правила представлены в виде макета.



Рис. 1. Золотые правила дизайна

Шаг 6: Подготовка презентации и защита проекта

1. Создай короткую презентацию (3-4 слайда):

Слайд 1: Приветствие и проблема. Кто мы и какую проблему/тему исследуем.

Слайд 2: Как мы это делали. Покажи самый интересный момент из твоего исследования (например, скриншот опроса или таблицы). Расскажи, откуда данные.

Слайд 3: Наше открытие и инфографика. Сделай главный вывод и ПРЕДСТАВЬ саму инфографику во всей красе!

2. Распиши речь: Каждый участник команды должен говорить о том, что делал он.

Аналитик: «Мы нашли данные вот здесь... и были удивлены, что...»

Дизайнер: «Мы выбрали такие цвета и графики, потому что...»

Презентатор: «И как вы видите, наша инфографика наглядно показывает...»

Ожидаемые результаты

В результате вы создадите реальный продукт, позволяющий представить огромное количество информации в компактном и привлекательном виде. Вы приобретёте навыки, которые особенно ценятся в современном мире: умение находить нужную информацию, выделять в ней главное. Вы освоите новые инструменты для творчества и создадите с их помощью «говорящее» изображение. Вы научитесь работать в команде, чувствовать себя ее частью, станете увереннее в себе.

ЗНАКОМСТВО С 3D-РЕДАКТОРОМ: МОДЕЛИРОВАНИЕ МАТЕМАТИЧЕСКИХ ФИГУР В КОМПАС-3D LT

Компьютерные игры, мультфильмы, макеты зданий — всё это начинается с создания трёхмерной модели. Вы сможете взять абстрактные формулы из учебника геометрии и превратить их в «осязаемый» цифровой объект. Этот проект — ваш первый шаг в мир 3D-моделирования, где вашим инструментом станет не только программа, но и знания математики.

Цель проекта:

Освоение базовых навыков работы в системе КОМПАС-3D LT через создание моделей математических тел и исследование их свойств.

При выполнении проекта вы:

изучите интерфейс и основные приёмы работы в системе КОМПАС-3D LT;

научитесь создавать и изменять базовые геометрические объекты;

используете знания по математике для расчёта параметров фигур (объём, площадь поверхности);

разработаете собственную 3D-модель, включающую несколько геометрических тел;

оформите результаты в виде презентации и защитите проект.

Технические средства / оборудование / материалы:

Компьютер с установленной программой КОМПАС-3D LT V12 (бесплатная версия для школ);

Проектор / интерактивная доска (для демонстрации);

Офисные программы (PowerPoint, Word);

Тетрадь, карандаш, линейка.

Ход выполнения проекта

При выполнении проекта важно соблюдать правила безопасного поведения!

Шаг 1. Погружение в проблему: включаем воображение

Смотрим на мир глазами 3D-дизайнера: Мы посмотрим, как 3D-графика используется в кино, играх, архитектуре и даже медицине.

Ответь себе на вопросы:

*Какой предмет в этой комнате можно было бы смоделировать?
(Подсказка: всё состоит из простых фигур!)*

Зачем вообще уметь это делать? (Создать модель для печати на 3D-принтере, для своей игры или просто для крутой презентации).

Формулируем главную задачу: Наша цель — создать 3D-модель реального или придуманного объекта, используя знания геометрии.

Выбери объект для моделирования. Начни с простого, состоящего из понятных форм.

Например:

Домик с крышей (куб + треугольная призма).

Шаг 2. Организация деятельности

Организууй команду (3-4 человека). Распределите роли, как в студии разработки:

3D-Моделлер (Дизайнер): отвечает за создание и визуализацию модели.

Инженер-расчётчик (Аналитик): рассчитывает объёмы, площади, проверяет пропорции.

Гейм-дизайнер (Оформитель): готовит презентацию.

Создайте «Дорожную карту проекта»: напишите простой план в общем документе:

Шаг 3. Освоение инструментария: первые шаги в КОМПАС-3D LT

Ваш «цифровой пластилин» — программа КОМПАС-3D LT.

А. Старт и настройка.



<https://kompas.ru/kompas-3d-lt/about>

Запустите КОМПАС-3D LT на своём компьютере.

Нажмите кнопку «Создать новый документ» и выберите тип «Деталь» .
Вы окажетесь в 3D-пространстве.

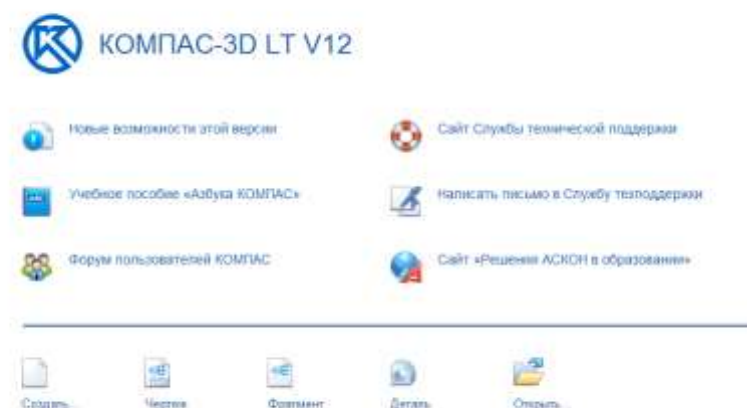


Рис.1. Главное окно КОМПАС-3D LT. Для начала работы выберите тип документа «Деталь»

Б. Базовые операции (Цифровой практикум).

В программе КОМПАС-3D создание 3D-модели всегда начинается с плоского эскиза, который затем «выдавливается» в объём. Процесс неразрывно связан с математикой: вы задаёте точные числа, а программа их вычисляет.

Задание-разминка «Геометрический город»: Создайте сцену, состоящую из куба.

Создание куба:

Сначала нарисуем основание. На панели «Текущее состояние» выберите плоскость (например, «Плоскость ZX»).

Нажмите кнопку «Эскиз». Теперь вы рисуете на выбранной плоскости.

Выберите инструмент «Прямоугольник» на панели «Геометрия» и нарисуйте квадрат. Чтобы задать точные размеры (например, 40 мм), нажмите клавишу Ctrl+Enter. На экране появится таблица, где можно ввести ширину и высоту.

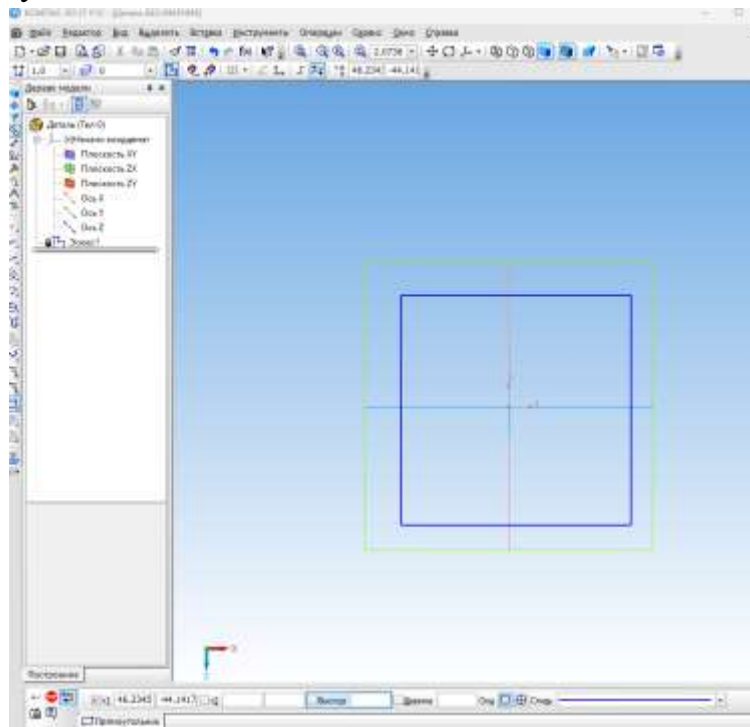


Рис. 2. Рабочее окно КОМПАС-3D LT

После создания эскиза нажмите кнопку «Эскиз» ещё раз, чтобы выйти из режима рисования.

Теперь «выдавим» квадрат в куб. Выберите операцию «Операция выдавливания» на панели «Редактирование детали».

В окне свойств укажите расстояние выдавливания — 40 мм. Нажмите «Создать».

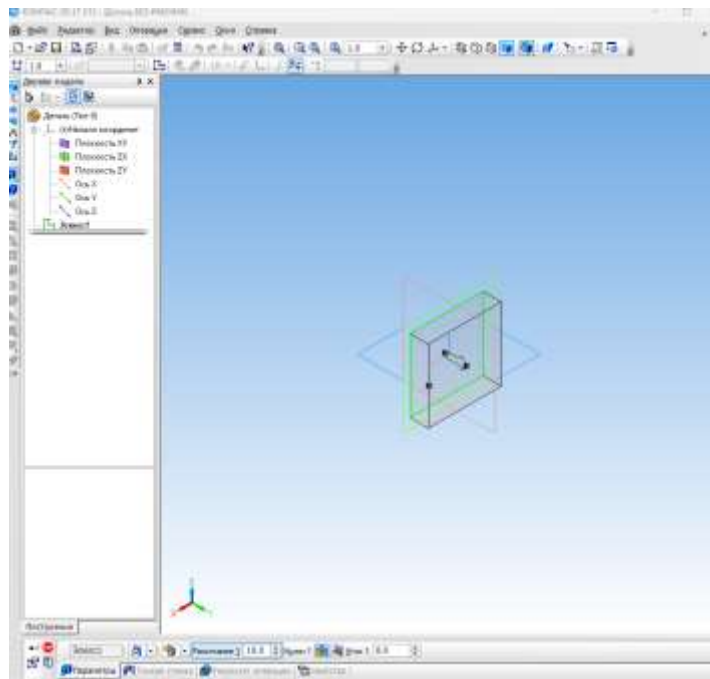


Рис. 3. Рабочее окно КОМПАС-3D LT «Редактирование детали»

Математика: Вы можете сами рассчитать объём вашего дома-куба, зная его сторону ($V = a^3$), чтобы проверить результат программы.

Шаг 4. Построение математической модели: цифры оживают

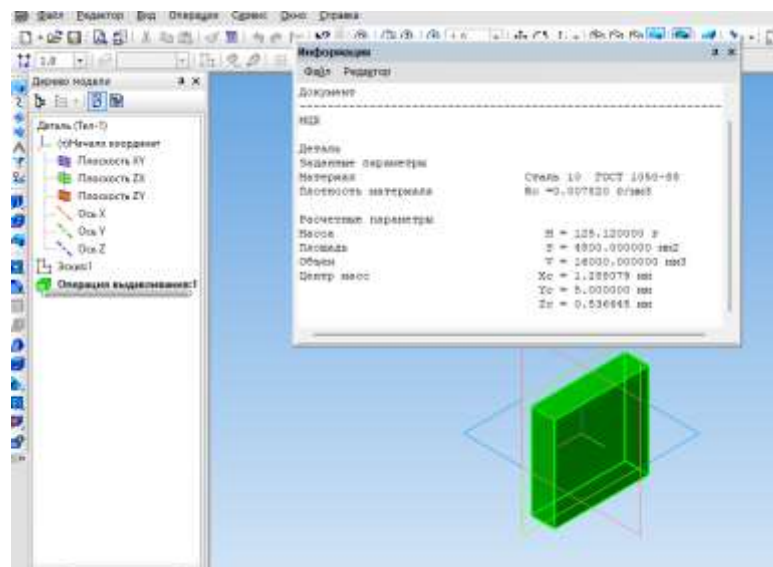
Вернитесь к своему эскизу из Шага 1. Достаньте учебник геометрии и рассчитайте объём и площадь поверхности для каждой фигуры в вашей модели.

Вот формула для расчёта объёма вашей шахматной фигуры:

Для куба (основание): $V = a^3$

Для пирамиды/конуса (шпиль): $V = 1/3 * S(\text{основания}) * h$

Сравните свои расчёты с данными из программы (в КОМПАС-3D есть команда «МЦХ» — масса-центр-характеристики, которая автоматически вычисляет объём вашей модели!). Это ваша математическая проверка модели.



Шаг 5. Презентация результатов проекта

Создайте презентацию (4-5 слайдов):

Слайд 1. Идея. Название проекта. «Мы строим не просто башню, а хранильницу знаний!».

Слайд 2. Магия геометрии. Покажите фотографию реального прототипа (или ваш эскиз) и рядом — его разбор на простые фигуры (куб, цилиндр и т.д.) с подписями формул для расчёта.

Слайд 3. Процесс создания. Коллаж из ваших скриншотов «Было/Стало». Покажите, как из кучи простых фигур родился шедевр. «На начальных этапах это была просто комбинация цилиндров и конуса, а на конечном — полноценная шахматная фигура».

Слайд 4. Вставьте финальный рендер (красивый скриншот) вашей модели.

Ожидаемые результаты

В результате выполнения проекта вы:

освоите первоначальные навыки 3D-моделирования в системе КОМПАС-3D LT;

закрепите свои познания в математике, применив их на практике;

создадите готовую 3D-модель и яркую презентацию для её защиты.

МОДЕЛИРОВАНИЕ ДВИЖЕНИЯ ТЕЛА ПО ЗАКОНАМ ФИЗИКИ

Как компьютерные игры точно рассчитывают полёт стрелы, прыжок персонажа или траекторию мяча? Секрет в том, что игра «решает» те же уравнения, что и вы на уроке физики!

Цель проекта:

Научиться моделировать механическое движение тела по законам физики и визуализировать результаты в виде анимации или интерактивной программы.

При выполнении проекта вы:

углубите знания по физики (скорость, ускорение, масса, сила, сопротивление);

освоите численные методы интегрирования (метод Эйлера как вводный) для решения уравнений движения;

реализуете в программной среде модель движения в нескольких задачах: свободное падение, бросок под углом, движение с сопротивлением;

построите графики зависимости координаты, скорости и ускорения от времени;

сравните аналитические решения с численными результатами и проанализировать погрешности;

подготовите презентацию с пояснениями.

Технические средства/оборудование

Компьютеры с установленными средами: Python (Jupyter/IDLE) + библиотеки matplotlib, numpy, tkinter / или среда Scratch (упрощённый вариант) / или GeoGebra (динамическое моделирование), калькуляторы, тетради.

Ход выполнения проекта

При выполнении проекта важно соблюдать правила безопасного поведения!

Шаг 1. Погружение в проблему: ставим физический эксперимент

Посмотрим на видео, как летит мяч, падает яблоко, движется стрела. Зададимся главным вопросом: «Можно ли предсказать, где окажется мяч через секунду, зная только силу и угол броска?». Создадим команду исследователей: чтобы ответить на такой сложный вопрос, нужны специалисты.

Главный физик (Аналитик): Твой инструмент — учебник и формулы. Ты будешь диктовать законы, по которым живёт наша виртуальная вселенная. Твоя задача — не ошибиться в уравнении.

Волшебник кода (Программист): Ты — переводчик. Ты превратишь слова и формулы физика на язык, понятный компьютеру (Scratch, Python). Без тебя модель останется на бумаге.

Художник данных (Визуализатор): Мир твоей симуляции должен быть красивым и понятным. Ты создашь графики, траектории, подпишешь оси. Ты сделаешь так, чтобы все увидели физику.

Испытатель и оратор (Тестировщик/Презентатор): Ты — связь с реальным миром. Ты проверишь, не «слетает» ли программа, и блестяще представишь наш проект, отвечая на любые вопросы.

Сформулируй конечную цель: Запиши её в общем документе: «Создать программу, которая визуализирует движение брошенного тела и позволяет менять его параметры (скорость, угол)».

Шаг 2. Изучение основ движения: пишем «правила игры»

Веди за собой! Объясни команде простыми словами: Скорость: насколько быстро тело меняет своё место. Ускорение: насколько быстро меняется сама скорость. Самое главное ускорение — $g = 9.8 \text{ м/с}^2$ (сила тяжести, всегда тянет вниз!). Траектория: линия, которую «рисует» движущееся тело. Проведи мысленный эксперимент: представь, что бросаешь мяч. Если бросить его сильнее (больше начальная скорость) — он улетит дальше. Если бросить под углом 45° — он улетит максимально далеко (это твоя гипотеза, которую можно проверить в модели!). Если бы не было воздуха (сопротивления), мяч летел бы по идеальной параболе. Начни думать о формуле. Самое простое движение, которое мы будем моделировать, описывается двумя волшебными строками для каждой координаты (X и Y):

Новая координата X = Старая координата X + (Начальная скорость по X) * время

Новая координата Y = Старая координата Y + (Начальная скорость по Y) * время - (g * время²) / 2

Результат этапа: у всей команды есть общее понимание физических законов, которые лягут в основу программы.

Шаг 3. Сбор исходных данных: выбираем объект и его характеристики

Выбери объект своей симуляции:

Мяч (классика, понятно всем).

Снаряд из катапульты (эпично!).

Бумажный самолётик (с ним можно потом добавить больше сопротивления воздуха).

Задай ему стартовые характеристики (это твои «рычаги управления»):

Начальная высота: с земли (0 м) или с крыши (5 м)?

Начальная скорость: слабо (5 м/с) или сильно (20 м/с)?

Угол броска: почти горизонтально (10°), под оптимальным углом (45°) или почти вертикально (80°)?

Создай «Паспорт объекта»: запиши всё это в таблицу. Эти числа ты позжеставишь в свою программу как начальные значения переменных.

Шаг 4. Создание компьютерной модели движения: оживляем формулы
Выбери свой инструмент:

Scratch (самый наглядный): Используй блоки «изменить x на ...», «изменить y на ...». Силу тяжести сделай блоком «изменить y на -1 » каждый шаг цикла.

Python (более мощный): Используй библиотеку `pygame` или `turtle`. Создай цикл, где на каждом шаге пересчитываешь координаты по формулам из Этапа 2.

Двигайся шаг за шагом:

Сначала сделай простое падение вниз (только начальная скорость по Y).

Потом добавь горизонтальный полёт (только начальная скорость по X).

А теперь объедини их — и получишь бросок под углом! Для этого начальную скорость нужно разложить на V_x и V_y , используя $\sin$ и $\cos$.

Шаг 5: Проверка работы модели

Проведи серию виртуальных экспериментов:

Эксперимент 1: Брось мяч с одной и той же скоростью, но под углами 30° , 45° и 60° . При каком угле он улетел дальше всех? Совпадает ли с теорией (45°)?

Эксперимент 2: Брось мяч вертикально вверх. Вернулся ли он точно в точку старта? Достигает ли он пика на половине общего времени полёта?

Найди «баги» в реальности (или в модели): если мяч в программе летит НЕ так, как подсказывает интуиция или учебник, значит:

Либо «Главный физик» ошибся в формуле.

Либо «Волшебник кода» неверно её запрограммировал.

Либо вы забыли про какой-то фактор (например, силу трения о воздух).

Скорректируйте модель. Идеал недостижим, но приближение к реальности — отличный результат!

Шаг 6. Визуализация результатов: превращаем числа в историю

Научи программу рисовать линию по следу движения мяча. Получится его траектория — та самая парабола.

Создай два отдельных графика: как меняется высота (Y) со временем и как меняется дальность (X). График высоты будет похож на гору: взлёт — пик — падение.

Цветовая динамика: пусть цвет мяча меняется в зависимости от его скорости (быстро = красный, медленно = синий).

Подписи: Добавь на экран цифры: текущая скорость, время полёта, пройденное расстояние.

Шаг 7: Подготовка презентации и защита проекта

Создайте презентацию (3-4 слайда):

Слайд 1: Вопрос. «Мы хотели узнать, можно ли предсказать полёт мяча. И мы научили это делать компьютер!».

Слайд 2: Закон. Покажи главную формулу и объясни её суть (без сложных выводов). Скажи: «Это правило, которое мы вложили в программу».

Слайд 3: Магия. Покажи скриншот или гиф-файл своей работающей симуляции с траекторией. Скажи: «Вот как это правило выглядит в движении».

Слайд 4: Открытие. Покажи результат своего ключевого эксперимента (например, что угол 45° — действительно самый выгодный для дальности).

Распределите текст доклада:

Физик объясняет законы.

Программист рассказывает, как их «оживил».

Художник показывает, какую красоту удалось создать.

Испытатель подводит итоги и держит удар на вопросах.

Не просто рассказывайте, а запускайте симуляцию на большом экране в реальном времени. Меняйте параметры и показывайте, как мгновенно меняется траектория.

Результат этапа: уверенная, эффектная защита, которая доказывает, что вы — команда, способная смоделировать кусочек мира.

Ожидаемые результаты

В результате выполнения проекта вы ближе познакомитесь с законами движения тел. Использование таких инструментов, как Scratch, GeoGebra или Python позволит вам создать модели движения тел, визуализировать траектории, получать количественные характеристики движения. Работая совместно со своими товарищами, вы получите навыки коллективного обсуждения полученных результатов, нахождения оптимального решения поставленной задачи и, в итоге, представите результат выполнения проекта в виде готовой модели, презентации и сообщения.

ХИМИЯ В ЧИСЛАХ: ИЗУЧЕНИЕ СВОЙСТВ ЭЛЕМЕНТОВ ЧЕРЕЗ ВИЗУАЛЬНОЕ ПРОГРАММИРОВАНИЕ

Вы когда-нибудь задумывались, что числа в таблице Менделеева — это не просто скучные цифры? Это целые истории: атомная масса рассказывает о «тяжеловесах» и «невесомых» элементах, температура плавления — о том, кто выдержит адский жар, а кто растает в руках. Но как эти истории увидеть и рассказать другим?

Выполняя данный проект, вы научитесь превращать абстрактные химические данные в яркие, понятные и даже интерактивные визуализации. Вы создадите не просто презентацию, а настоящую цифровую лабораторию в Scratch, где можно исследовать и сравнивать элементы.

Цель проекта:

Создать в среде Scratch интерактивный проект «сравнительную панель», который наглядно показывает различия в свойствах выбранных химических элементов с помощью анимации, графиков или шкал.

При выполнении проекта вы:

выберете группу родственных или контрастных элементов (5-8 штук) и соберёте точные данные об их ключевых свойствах (атомная масса, температура плавления/кипения, плотность);

подготовите данные к визуализации;

проанализируете собранные числа: найдёте максимальные и минимальные значения, чтобы с помощью простой формулы «перевести» реальные величины в размеры спрайтов или координаты в Scratch (например, превратить температуру от -200°C до $+200^{\circ}\text{C}$ в длину столбика от 10 до 300 пикселей);

разработаете и запрограммируете проект в Scratch.

Технические средства/оборудование

Компьютер с доступом к Scratch 3.0;

таблица Менделеева (бумажная или цифровая);

справочные данные: атомная масса, температура плавления/кипения, плотность веществ;

линейка, таблицы, калькулятор (для преобразования и нормировки данных).

Ход выполнения проекта

При выполнении проекта важно соблюдать правила безопасного поведения!

Шаг 1. Погружение в проблему

Смотрим на строки чисел (атомная масса, плотность). Задаём вопросы: Почему у Осмия (Os) плотность $22,59 \text{ г/см}^3$ — это много? А у Лития (Li) всего $0,534 \text{ г/см}^3$ — это как? Как это вообще представить? Где мы встречаем графики и диаграммы? (Спорт: графики скорости, соцсети: диаграммы просмотров). Как они помогают понять информацию?

Наша цель — создать в Scratch интерактивный проект-визуализатор, который превращает числовые свойства 5-10 химических элементов в наглядные, сравнимые графики, диаграммы или анимированные шкалы.

Выберите три самых удивительных, на твой взгляд, свойства из химических объектов и подумай, как их можно показать:

Атомная масса: можно показать шариками разного размера?

Плотность: столбиками разной высоты или цветом (чем плотнее — тем «тяжелее» цвет)?

Температура плавления: шкалой термометра или цветом (от синего = холодный, до красного = очень горячий)?

Шаг 2. Организация деятельности

Организуй команду (3-5 человек). Распределите роли:

Химик-датель (Сборщик данных): Инструменты — таблицы и справочники. Находит числа: атомную массу, плотность, температуру.

Математик-масштабатор (Аналитик): превращает абсолютные значения (например, плотность от 0,5 до 22) в относительные, удобные для Scratch (например, от 10 до 300 пикселей).

Программист-визуализатор (Кодер): оживляет расчёты по математике — создаёт спрайты, которые меняют размер/цвет в зависимости от числа, программирует кнопки и слайды.

Дизайнер-интерфейсер (Художник): создаёт стиль проекта: фон, шрифты, цвета диаграмм.

Шаг 3. Реализация проекта

Сбор и анализ данных.

Создайте таблицу данных в Google Таблицах или на бумаге.

Таблица. Пример таблицы данных в Google Таблицах

Элемент	Символ	Атомная масса (а.е.м.)	Т плавления (°C)	Масштаб: Высота столбика (от 50 до 300)	Масштаб: Длина термометра (от 10 до 200)
Литий	Li	6.94	180.5	?	?
Натрий	Na	22.99	97.8	?	?
...	...	...	...	...	...
Фтор	F	18.99	-219.6	?	?
Хлор	Cl	35.45	-101.5	?	?

Найдите MAX и MIN значение для атомной массы в вашем списке.

Формула для высоты спрайта в Scratch: $\text{Высота} = 50 + ((\text{Масса} - \text{MIN}) / (\text{MAX} - \text{MIN})) * 250$.

Пример: Если масса от 6.94 до 85.47, то для Na (22.99): $\text{Высота} = 50 + ((22.99 - 6.94) / (85.47 - 6.94)) * 250 \approx 50 + 51 \approx 101$ пиксель.

Сделайте выводы: «У элементов 1 группы атомная масса растёт сверху вниз, а температура плавления падает!».

Программирование и визуализация в Scratch.

Подготовьте «сцену»:

Создание титульного слайда

Откройте Scratch и выберите сцену (спрайт «Сцена» или «Stage»).

Перейдите на вкладку «Фоны».

Нажмите кнопку «Выбрать фон» (иконка с картинкой) и выберите любой фон, например «Stars» или «Boardwalk».

Добавьте текст на фон (инструмент «Текст»). Напишите крупно: «Химия в числах».

Добавьте текст поменьше: «Сравнение свойств химических элементов».

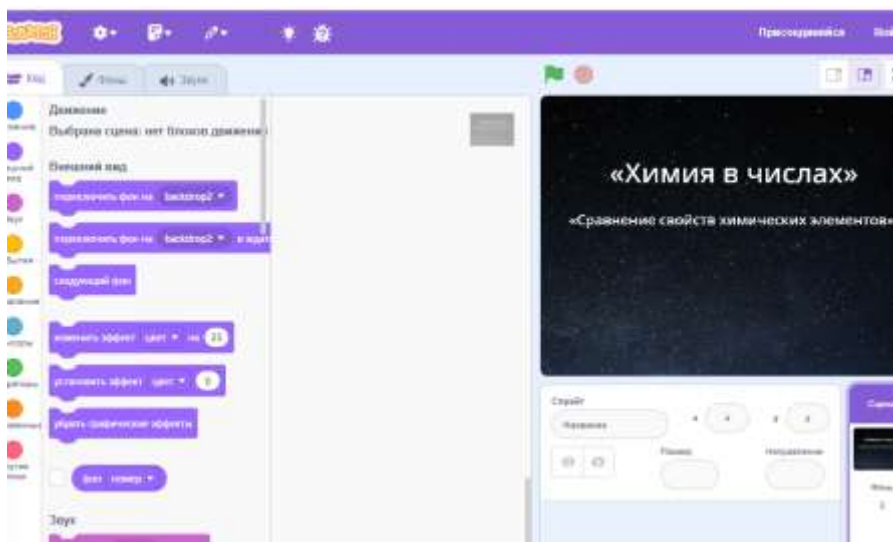


Рис.1. Фон первого слайда

Создание рабочего фона

Нажмите кнопку «Выбрать фон» ещё раз. Выберите или нарисуйте новый фон (например, светло-серый или белый).

На этом фоне мы разместим диаграмму. Для удобства нарисуйте тонкую линию внизу (это будет «ось X» для столбиков) и линию слева (ось Y).

Подпишите на фоне: «Атомная масса».

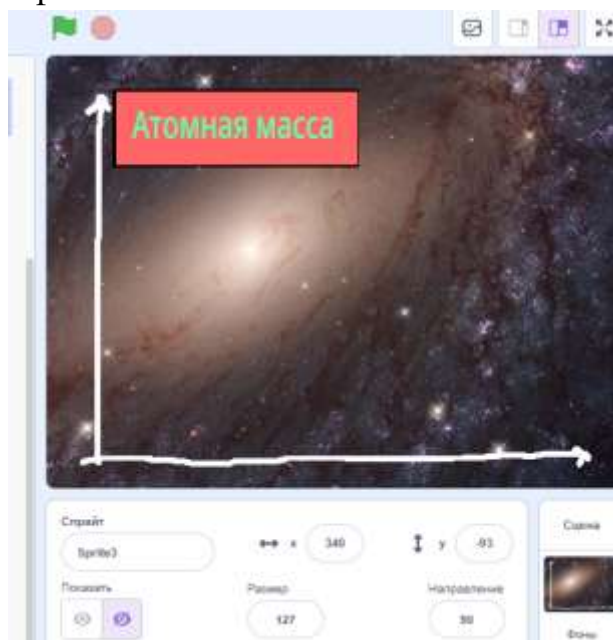


Рис.2. Рабочий фон

Создание списков (базы данных элементов)

Это самый важный шаг. Списки будут хранить информацию о каждом элементе.

Перейдите в категорию «Переменные».

Нажмите «Создать список».

Создайте следующие списки (6 элементов):

Нажимая на знак «+» в каждом списке добавьте значения для выбранных элементов.

1 список «Название» - Литий, Бериллий, Бор, Углерод, Азот, Кислород

2 список «Символы» - Li, Be, B, C, N, O

3 список «Масса» - 6,94. 9,01. 10,81. 12,01. 14,01. 16,00

4 список «Высоты_массы» - 40, 45, 50, 55, 60, 65

5 список «Температура» - 180,5, 1287, 2076, 3550, -210, -218

6 список «Высоты_температуры» - 120, 200, 230, 250, 20, 15



Рис.3. Созданные списки элементов

Формула для расчёта высоты столбика:

Высота = 30 + (Значение_элемента × 2)

Например, для массы Лития (6.94): $30 + 6.94 \times 2 = 30 + 13.88 \approx 44$ (округляем до 45).

Создание «умного» спрайта-столбика

Рисование спрайта «Столбик»

Нажмите «Выбрать спрайт» → «Нарисовать».

Нарисуйте тонкий вытянутый прямоугольник (например, 20 пикселей в ширину, 10 пикселей в высоту — это основа, которая будет растягиваться).

Закрасьте его ярким цветом (например, синим).

Назовите спрайт: «Столбик».

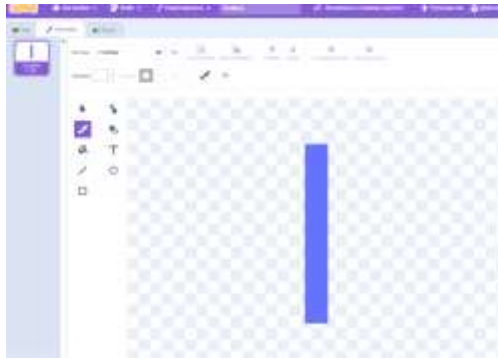


Рис. 4. Спрайт «Столбик на рабочем поле

Программирование создания клонов

Теперь нужно, чтобы при запуске проекта создалось столько клонов столбиков, сколько элементов в списке.

Скрипт для спрайта «Столбик»:

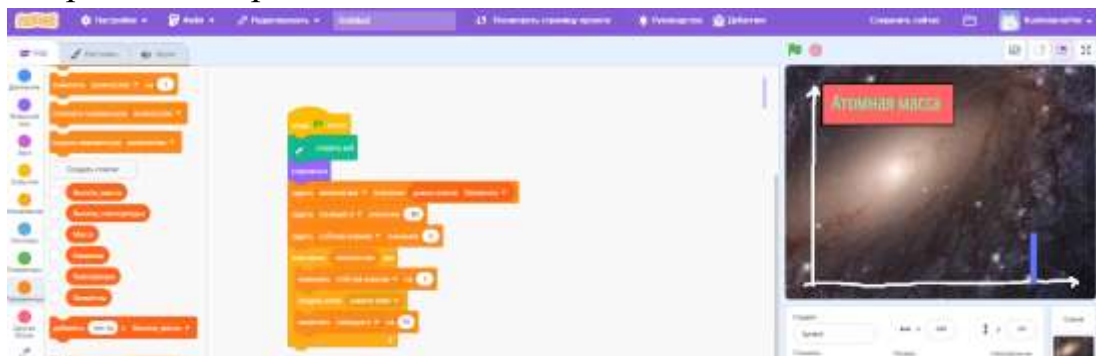


Рис. 5. Скрипт для создания клонов столбика

Программирование поведения клонов

Это самый важный скрипт. Он находится в спрайте «Столбик».

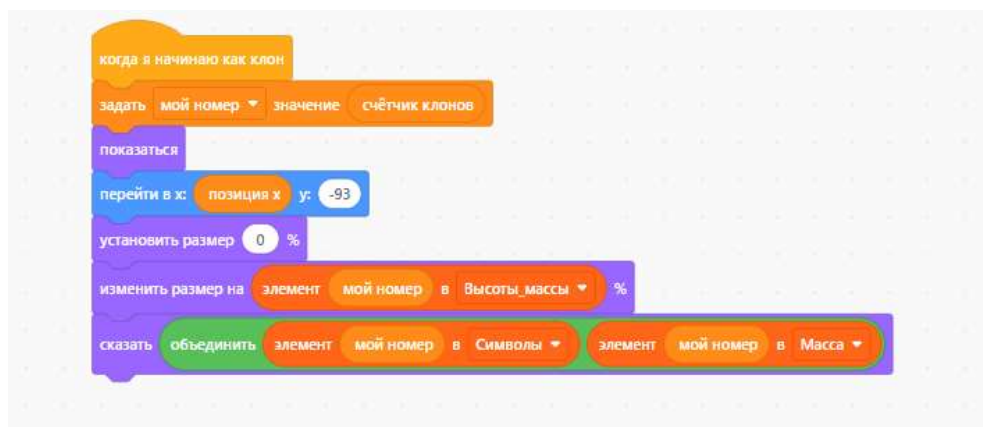


Рис. 6. Скрипт работы клонов

Скрипт создаёт клоны столбиков, располагает их на необходимом расстоянии друг от друга, и подписывает символ элемента и атомную массу. Высота столбиков задаётся программно.



Рис. 7. Результат работы скриптов

Шаг 4: Презентация и защита

Создайте презентацию (4-5 слайдов):

Слайд 1: Вопрос. «Мы устали от скучных цифр в таблице. Мы решили дать химическим свойствам голос и цвет!»

Слайд 2: Методология — ключ к успеху. Покажите вашу таблицу с исходными данными и результатами расчёта масштабов. Скажите: «Чтобы нарисовать эти столбики, нам пришлось превратить реальные величины в пиксели по этой формуле...». Это покажет вашу математическую работу.

Слайд 3: Живая демонстрация

ПЕРЕЙДИТЕ ИЗ ПРЕЗЕНТАЦИИ ПРЯМО В ВАШ SCRATCH-ПРОЕКТ. Запустите его и в реальном времени покажите: «Смотрите, вот атомные массы.

Слайд 4: Команда и перспективы. «Нашу модель можно расширить, добавив все 118 элементов и новые свойства».

Ожидаемые результаты

Выполняя проект, вы получите рабочую модель в Scratch, представляющую собой интерактивную сравнительную визуализацию свойств химических элементов.

Презентацию, раскрывающую этапы работы и демонстрирующую итоговый продукт.

Результатом проекта также будет углубление понимания взаимосвязи между числовыми данными, их математической обработкой и визуальным представлением.