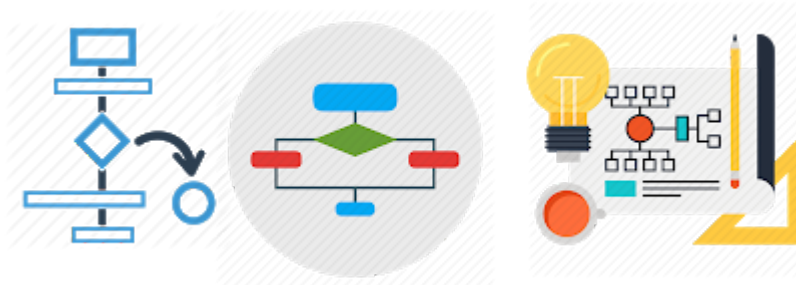


При розв'язуванні задач часто використовуються **алгоритми з розгалуженням**, які передбачають виконання певних дій залежно від істинності деякого висловлювання, що є умовою виконання певних команд.

Наприклад, алгоритм переходу дороги по пішохідному переходу, який регулюється світлофором: якщо горить зелене світло, слід переходити дорогу, в іншому разі — слід зупинитися перед пішохідним переходом.

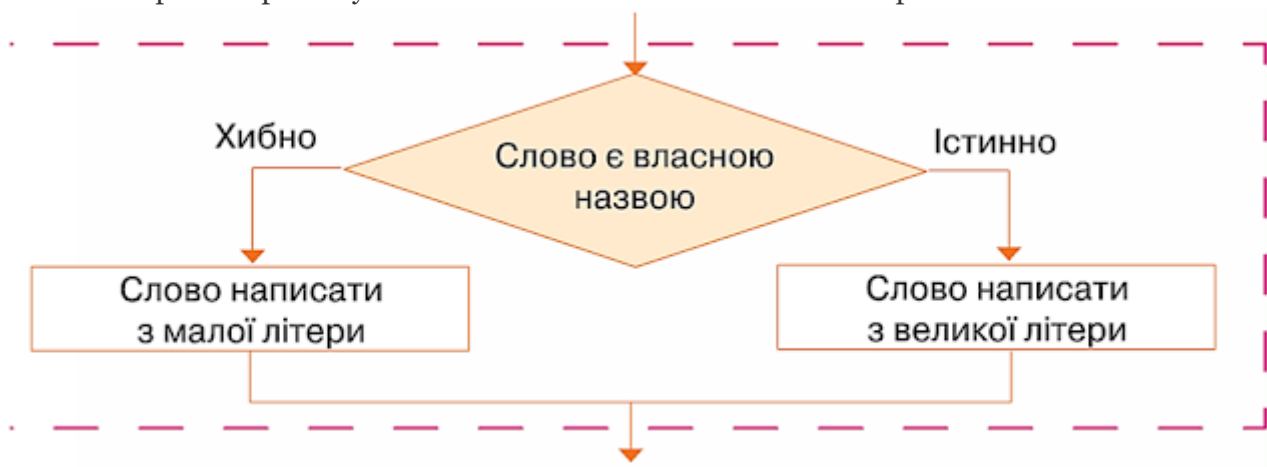
Алгоритмом із розгалуженням можна вважати алгоритм здійснення дзвінка з мобільного телефону: якщо є кошти на рахунку і мережа доступна, то ти набираєш номер адресата, інакше виклик не буде здійснено. При побудові таких алгоритмів використовують **алгоритмічну структуру розгалуження**.

Алгоритмічна структура, що дає змогу виконавцеві алгоритму вибрати сценарій подальших дій залежно від істинності певного висловлювання, називається **розгалуженням**.



Розрізняють дві форми структури розгалуження: **повну** та **неповну**. Структура розгалуження **повної форми** схожа на умовне висловлювання «**Якщо — то — інакше**», у якому після «то» та «інакше» записують не висловлювання, а команди, які необхідно виконати залежно від істинності висловлювання, записаного в умові. Її можна подати графічно.

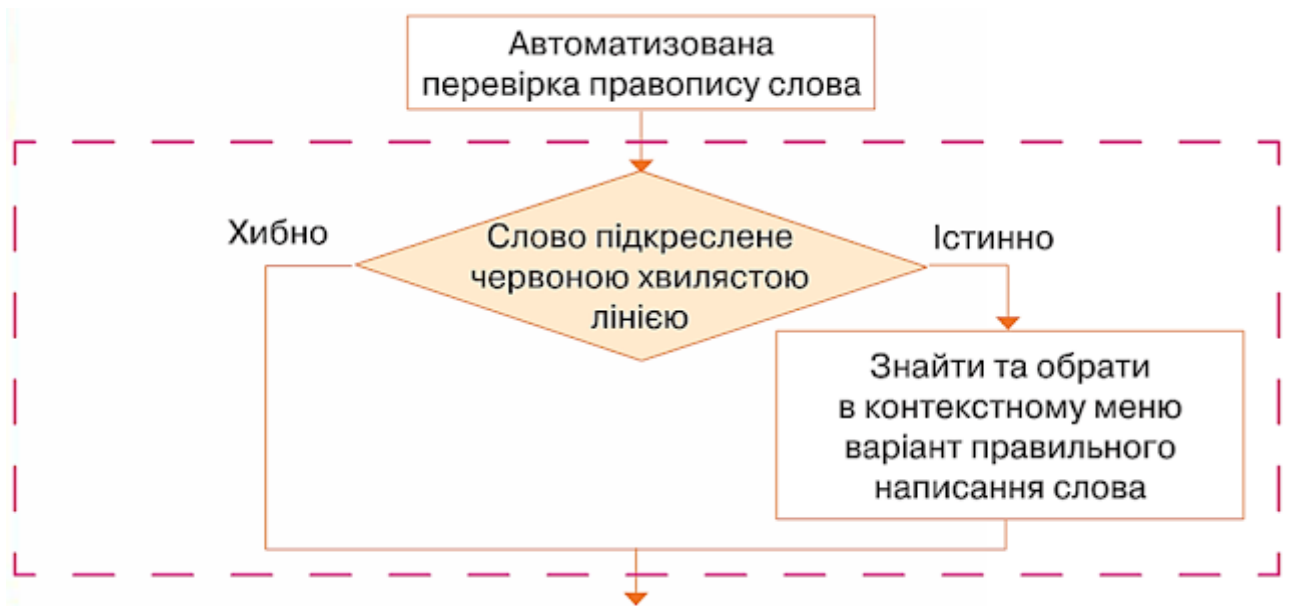
Алгоритм із розгалуженням написання слів із великої літери.



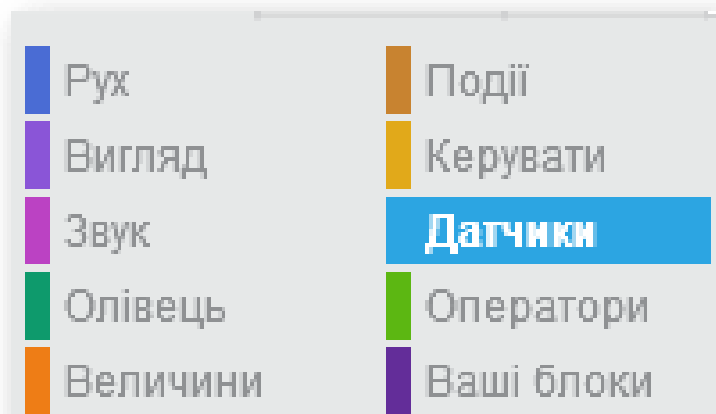
Скорочену форму розгалуження використовують тоді, коли деяку послідовність команд слід виконати за умови істинності висловлювання.

Структура розгалуження **неповної форми** схожа на умовне висловлювання «**Якщо — то**», у якому після «то» записують не висловлювання, а послідовність команд, які необхідно виконати, коли висловлювання, записане в умові, є істинним. Її можна подати графічно.

Наприклад, на малюнку подано графічну форму алгоритму застосування засобів автоматизованої перевірки правопису тексту, що складається зі слів, у середовищі текстового процесора .



Для опису алгоритмів із розгалуженням у середовищі **Скретч** можна використовувати блоки з групи Датчики. Ці блоки можуть бути складовими відповідних команд, що передбачають виконання певних дій після перевірки висловлювання на істинність чи хибність.



Середовище Скретч	Висловлювання
мишку натиснуто?	натиснуто ліву кнопку миші
клавішу пропуск натиснуто?	натиснуто вказану клавішу
доторкається ?	об'єкт торкається вказаного об'єкта, межі, або вказівника миші (обирають із списку)
доторкається кольору ?	об'єкт торкається іншого об'єкта чи фону вказаного кольору (обирається на сцені піпеткою)

Середовище Скретч	Висловлювання
	об'єкт першого кольору торкається об'єкта або фону другого кольору
	комп'ютерний мікрофон виявляє гучність звуку більше, ніж 30 (у межах від 1 до 100)

Складені умови, які використовують сполучники **І**, **АБО**, **НЕ**, у середовищі Скретч можна описати за допомогою блоків:



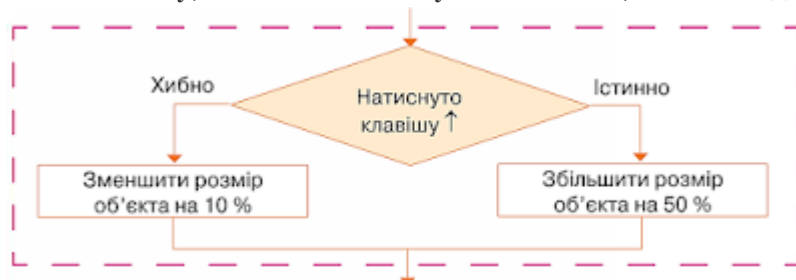
з групи **Оператори**. Наприклад, в алгоритмі в середовищі Скретч, виконавцем якого є Об'єкт 1, умову «якщо Об'єкт 1 доторкається до Об'єкта 2 або доторкається межі вікна», можна подати в такому вигляді:



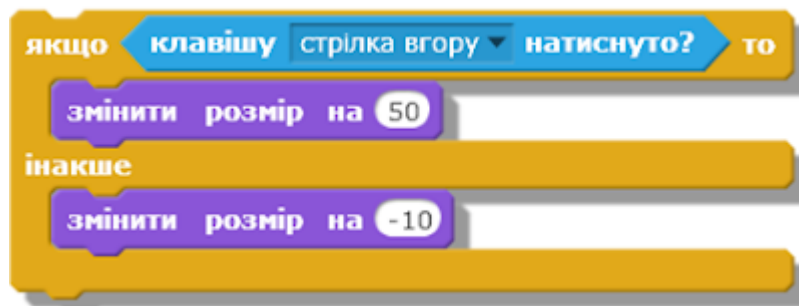
Базову структуру розгалуження повної форми в середовищі Скретч можна подати командою:



Наприклад, алгоритм, за яким при натисненні клавіші на клавіатурі розмір об'єкта збільшується наполовину, інакше — зменшується на 10%, можна подати графічно.



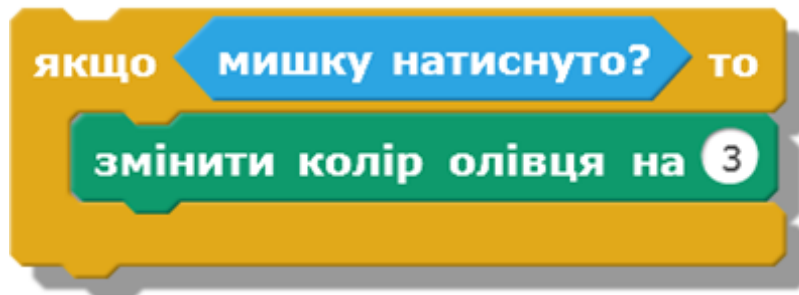
Або фрагментом програми в середовищі Скретч.



Для реалізації розгалуження **неповної** форми в середовищі **Скретч** використовують команду:



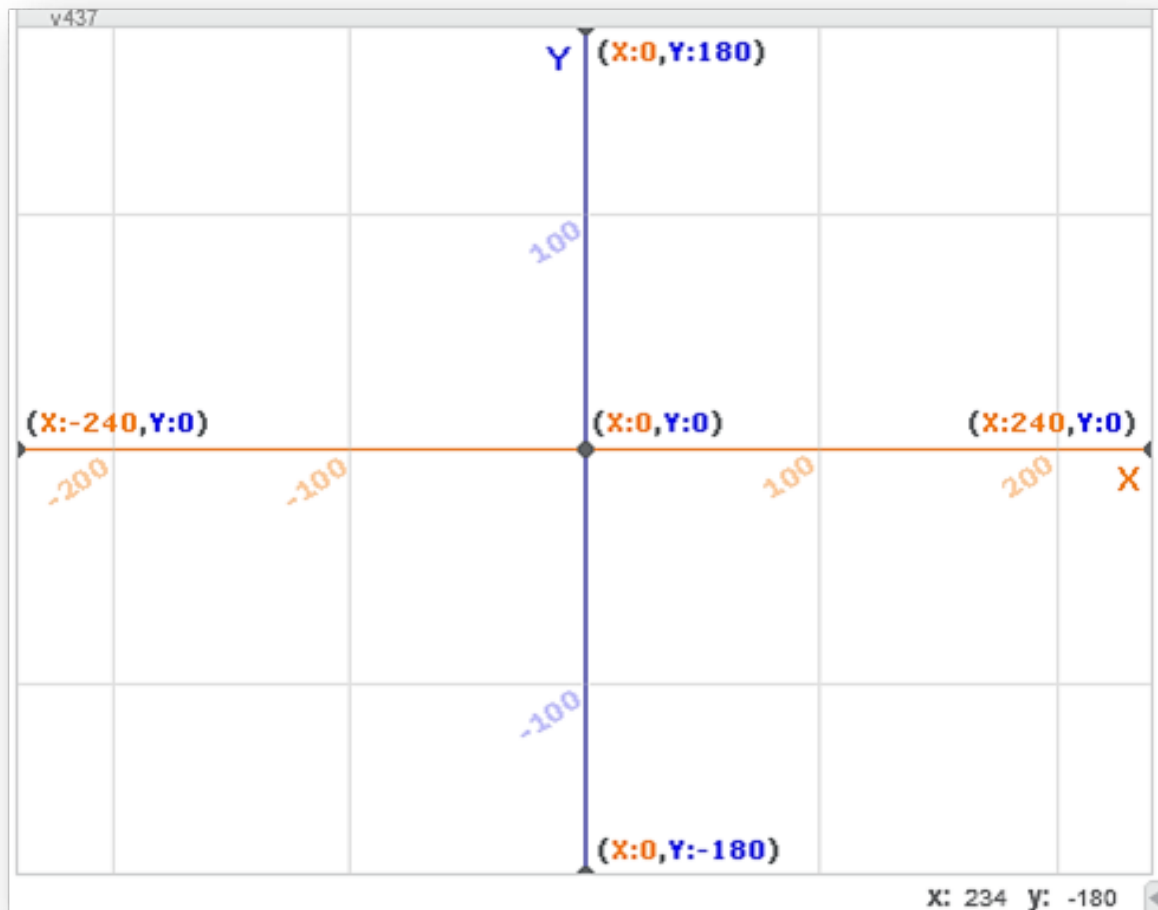
Наприклад, фрагмент алгоритму для середовища **Скретч** для малювання різнокольорової ламаної, за яким колір ліній буде змінюватися після натиснення кнопки миші, для середовища **Скретч** можна описати:



Реалізація алгоритмів з розгалуженням у середовищі Скретч

При складанні алгоритмів із розгалуженням у середовищі **Скретч** для того, щоб, наприклад, управляти рухом виконавців на сцені чи визначати відстань до вказаного об'єкта, можна використовувати числові величини, значення яких можуть змінюватись або задаватись.

Оскільки сцену середовища **Скретч** можна вважати координатною площиною.



Для того, щоб під час виконання програми визначити положення деякого об'єкта (координати **x** та **y**), можна скористатись відповідними величинами:

значення x	Значення величини, яка є координатою x (у межах від -240 до 240)
значення y	Значення величини, яка є координатою y (у межах від -180 до 180)

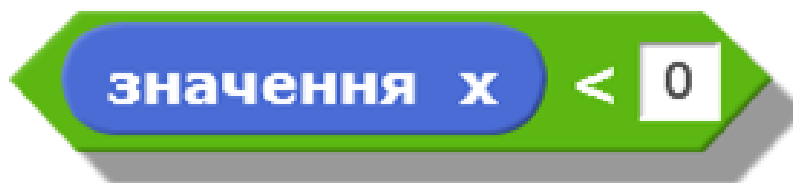
Крім величин, що визначають місце розташування об'єкта на сцені, при створенні проектів у середовищі **Скретч** використовують й інші величини.

відстань до	вказує на відстань до обраного об'єкта чи вказівника миші
розмір	розмір об'єкта у відсотках до початкового
гучність	величина гучності звуку

Такі команди передають у проект числові дані — значення відповідних величин.

Числові величини можна використовувати як при формулюванні умов розгалуження, так і наслідків — відповідних дій виконавців алгоритму.

Наприклад, при використанні числових величин в умові



можна визначити, чи знаходиться об'єкт у «лівій половині сцени», оскільки центр сцени має координати (0,0). При перевірці істинності зазначеної умови визначається поточне значення координати x — позиції розміщення об'єкта на сцені, та порівнюється із вказаним в умові — у даному випадку 0.

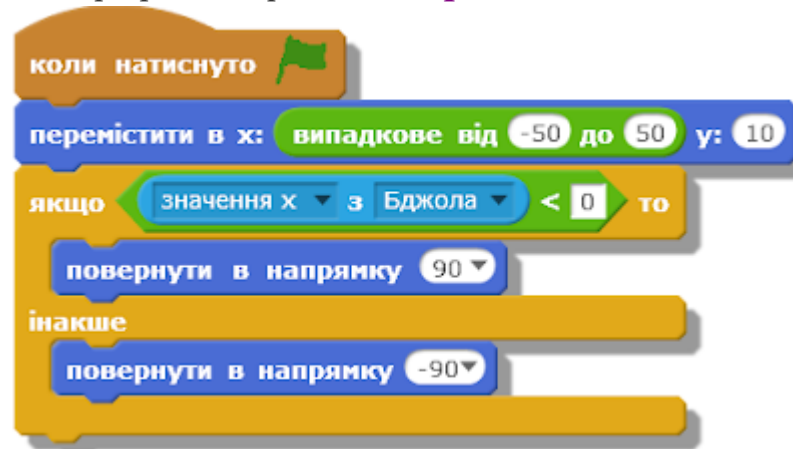
У середовищі **Скретч** можна задати випадкове значення параметрів деяких команд у процесі виконання програми. З випадковими величинами ти вже ознайомився в курсі математики. Наприклад, при підкиданні грального кубика цілком випадково може випасти число від 1 до 6.

У середовищі **Скретч** випадкову величину отримують із певного числового діапазону, вказавши її найменше та найбільше з можливих значень. Наприклад, команда задає деяке число, значення якого більше або дорівнює числу -50 та менше або дорівнює числу 50 .



Розглянемо, наприклад, проект Бджола, у якому бджола переміщується ліворуч або праворуч у точку, значення координати x якої задається випадково. Після чого повертається праворуч, якщо отримане число додатне, а якщо навпаки — ліворуч.

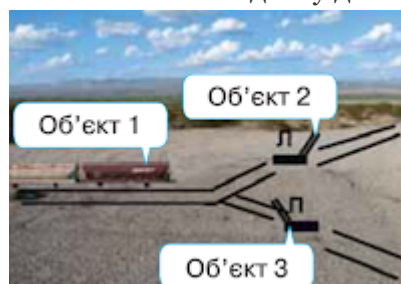
Фрагмент програми в середовищі **Скретч** матиме вигляд:



У середовищі **Скретч** можна створити проект, у якому декілька виконавців виконують свою програму незалежно одне від одного. Наприклад, можна створити проект Дитячий майданчик, де дівчинка стрибатиме через скакалку, а хлопчик підкидатиме м'яч, і при цьому вони не будуть взаємодіяти.

Але можна створити і проект, у якому буде передбачено взаємодію кількох об'єктів.

Розглянемо засоби, які дають змогу виконати пов'язані дії кількох об'єктів: коли виконання події з одним об'єктом викликає деяку дію іншого об'єкта.



Для цього у програмі має з'являтися повідомлення про цю подію. Наприклад, нехай у проекті Шлагбаум беруть участь такі три об'єкти:

- Потяг, що рухається залізничним полотном із регульованим шлагбаумом,
- Шлагбаум Л,
- Шлагбаум П.

У проекті передбачено дві події:

Коли натиснути на Шлагбаум Л, потяг рухатиметься ліворуч, а коли на Шлагбаум П — праворуч. Лівий шлагбаум передає повідомлення потягу — рухатися ліворуч, а правий — рухатися праворуч. Для побудови програми зазначеного руху потягу в проекті Скретч можна скористатися спеціальною командою, яка передає появу відповідного повідомлення.



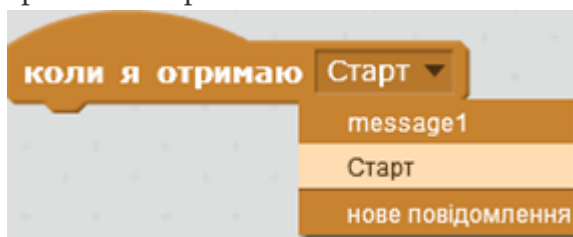
Команда



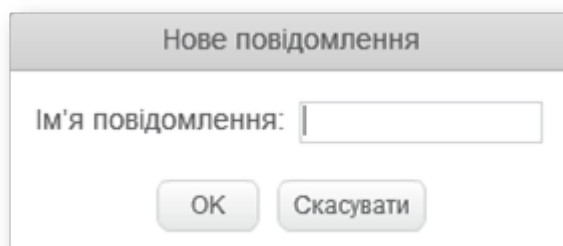
дає змогу після отримання повідомлення розпочати виконання команд, що слідують за нею. Для створення нового повідомлення або вибору з уже існуючих натискають



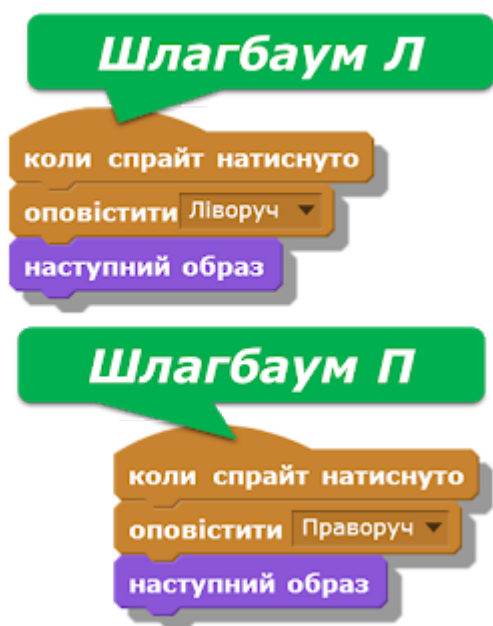
Отримують нове повідомлення
Або обирають раніше створене



Нове повідомлення можна створити у вікні



Програми для взаємодії об'єктів:



У середовищі **Скретч** можна створювати складні проекти, наприклад, мультиплікації за мотивами відомих казок або власних історій, розробляти комп'ютерні ігри чи тренажери, моделювати процеси, що відбуваються у природі, або взаємодію частин механізмів.

Перш ніж створювати проект у середовищі **Скретч**, у тому числі і складний, слід:

- визначити об'єкти проекту та їх зображення;
- виділити події проекту та дібрати відповідні команди для їх реалізації;
- задати метод розв'язування завдання, розділити завдання на під-завдання, у яких виділити базові алгоритмічні структури;
- створити алгоритм і подати його словесно чи у вигляді схеми.

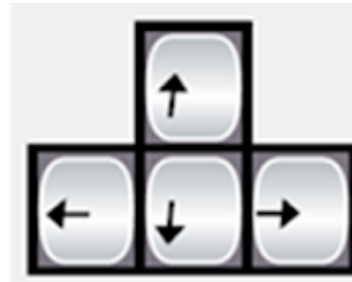
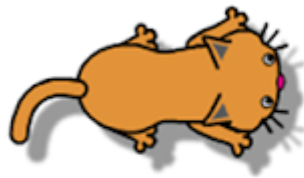
Далі алгоритм слід описати мовою середовища виконання алгоритмів, перевірити його правильність і зберегти разом з усіма об'єктами у файлі проекту.

В алгоритмах, які описують складні проекти, можуть використовуватися базові структури алгоритмів:

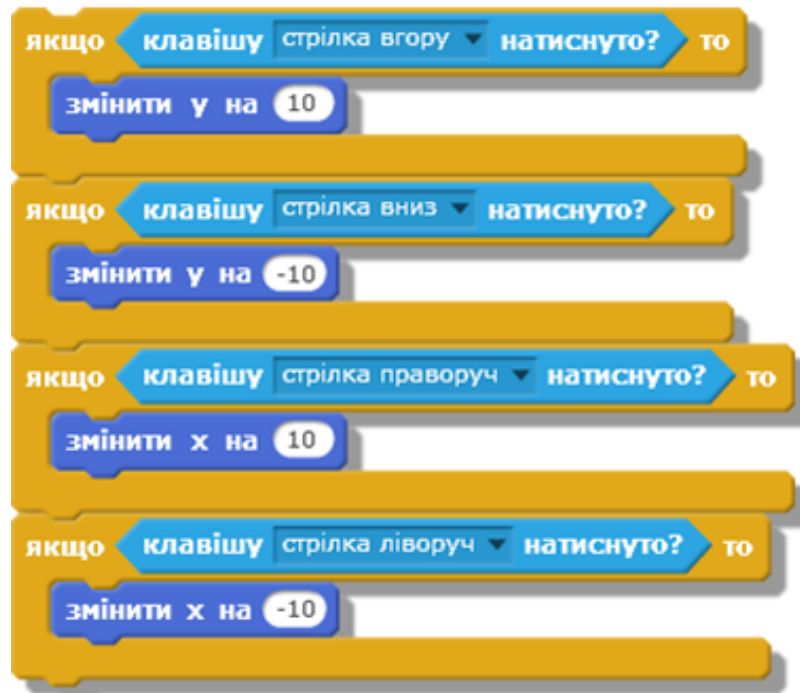


а тож різні їх комбінації.

Розглянемо приклад. Нехай напрямок руху виконавця **Кіт**, який за замовчуванням переміщується вперед, можна змінити за допомогою відповідних **клавіш клавіатури**.

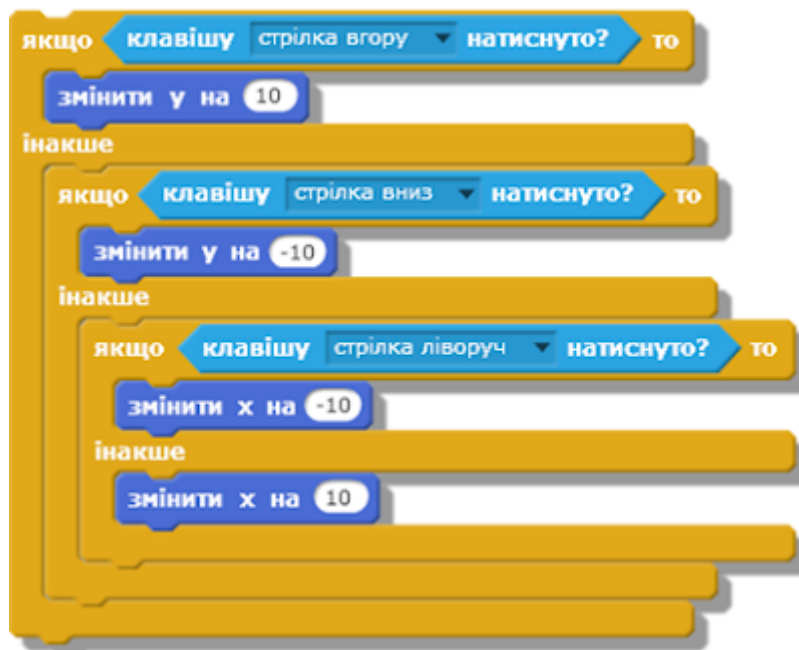


Зазначені дії можна реалізувати за допомогою таких фрагментів двох програм у середовищі **Скретч**.



Приклад 1 демонструє послідовне використання чотирьох команд неповного розгалуження. Рух виконавця в певному напрямі буде здійснено тільки за умови натиснення відповідної клавіші.

Зазначені дії можна реалізувати за допомогою таких фрагментів двох програм у середовищі **Скретч**.

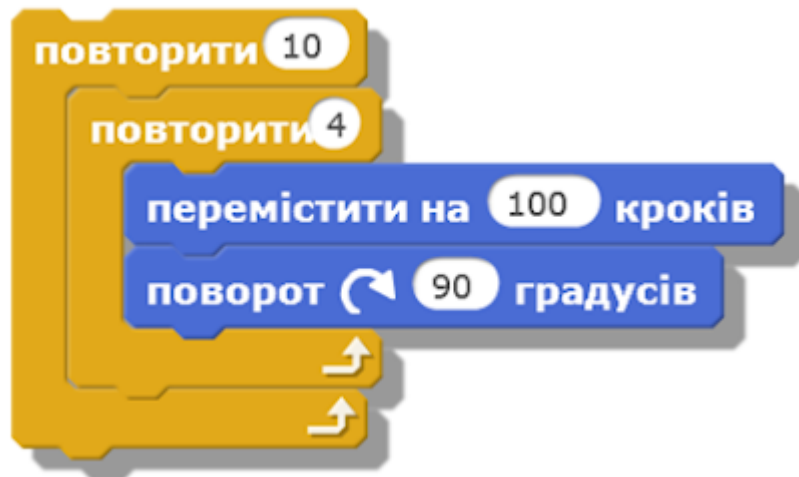


У прикладі 2

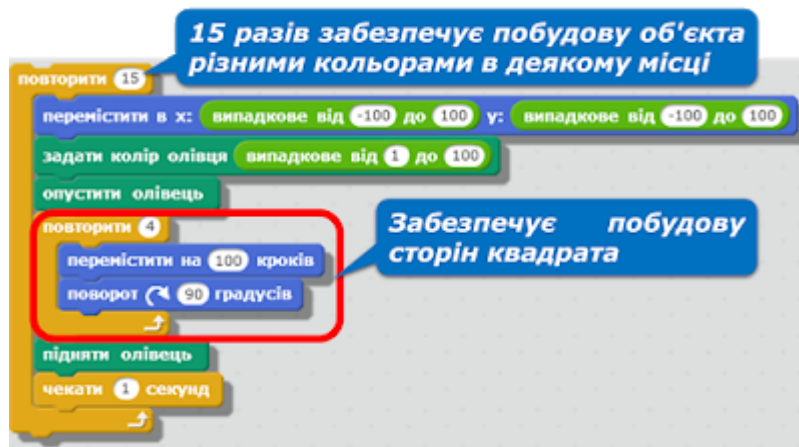
використано три команди розгалуження повної форми, у яких тричі перевіряється, яка клавіша натиснута. Рух буде здійснюватися без натиснення клавіші **Стрілка вправо**.

При розв'язуванні задач часто необхідно деякі дії повторити кілька разів. Один із способів написання такої програми — включити повторення в набір команд, що повторюються всередині іншого циклу.

Така структура, що складається з циклу в циклі, називається **вкладеними циклами**.



Наприклад, щоб у проекті Вітраж, реалізованому в середовищі **Скретч**, отримати зображення з 15 різнокольорових квадратів, використовують два цикли.



У програмі, що містить вкладені цикли, обов'язково спочатку виконується «внутрішній» цикл, а потім — «зовнішній».

IV. Формування практичних умінь і навичок

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм. (Інструктаж з правил техніки безпеки)

Підготовка до практичної роботи

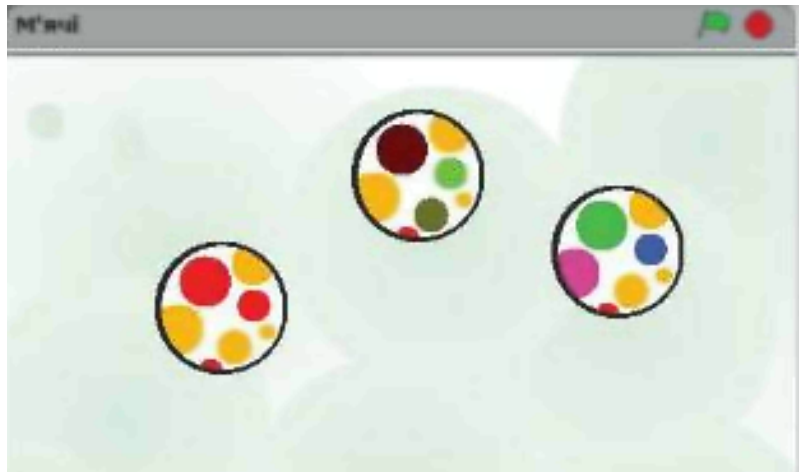
(виконати одне завдання на вибір)

Завдання 1. Історія.

На різних уроках для перевірки знань учнів використовують тести. Для цього їм пропонують виконати завдання на папері або за допомогою комп'ютера. Склади програму для підтримки комп'ютерного тестування в середовищі *Скретч*. Сплануй і створи проект *Тестер* – комп'ютерний тест, що складається із трьох тестових запитань з історії. Передбач, що на кожне із запитань тесту є тільки одна правильна відповідь із чотирьох запропонованих. Якщо користувач обирає правильну відповідь, він отримує наступне запитання та чотири варіанти відповіді. Тестування завершується, якщо користувач дав правильні відповіді на всі три запитання або якщо користувач дав першу неправильну відповідь на тестове запитання. Придумай самостійно запитання та варіанти відповідей, або скористайся підручником історії.

Завдання 2. М'ячі.

Під час проведення концертів часто використовують світлові ефекти. Склади проект, у якому три кольорові кульки будуть змінювати свій колір та опускатись зверху-донизу з різною швидкістю. Якщо на клавіатурі натискати клавішу 1, то перша кулька буде підніматись вгору, а якщо 2 чи 3 – то відповідна кулька також буде підніматись. Якщо кульки торкаються нижньої межі сцени – вони зникають.



Завдання 3. Гра.

У середовищі **Скретч** сплануй та створи гру, у якій за допомогою дощечки чорного кольору (як на малюнку), яку можна рухати вліво-вправо стрілками на клавіатурі, відбивають м'яч. М'яч у довільному напрямку відбивається як від стін, так і від дощечки. Якщо м'яч торкається червоної лінії внизу – гра припиняється. На початку гри можна обрати два рівні – легкий, коли дощечка широка та м'яч рухається не швидко, або складний – розмір дощечки зменшується, а швидкість руху м'яча збільшується.

