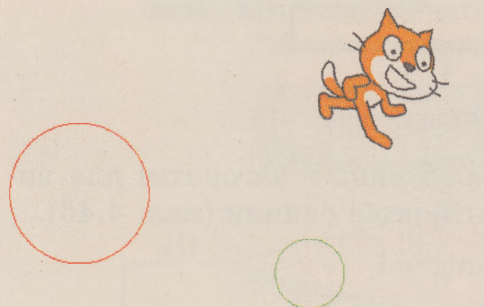
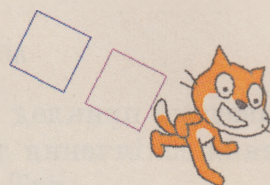




3. Збережіть складений алгоритм у вашій папці у файлі з іменем **практична 4_1**.
4. Складіть алгоритм, щоб отримати зображення, подане на малюнку 4.43.
5. Збережіть складений алгоритм у вашій папці у файлі з іменем **практична 4_2**.
6. Закрийте вікно програми **Scratch**.



Мал. 4.42



Мал. 4.43



4.5. АЛГОРИТМИ З РОЗГАЛУЖЕННЯМ



1. Який алгоритм називають лінійним?
2. Який фрагмент алгоритму називають циклом?
3. Який фрагмент алгоритму називають розгалуженням?

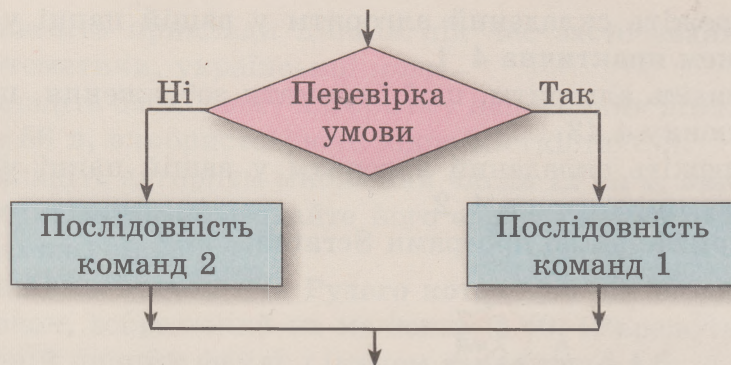
РОЗГАЛУЖЕННЯ В АЛГОРИТМІ

Ви вже знаєте, що в алгоритмі може бути фрагмент, який містить команду перевірки умови, і залежно від результату виконання цієї команди (**Так** чи **Ні**) будуть виконуватись або одна послідовність команд, або інша.

Такий фрагмент в алгоритмі називають **повним розгалуженням**.

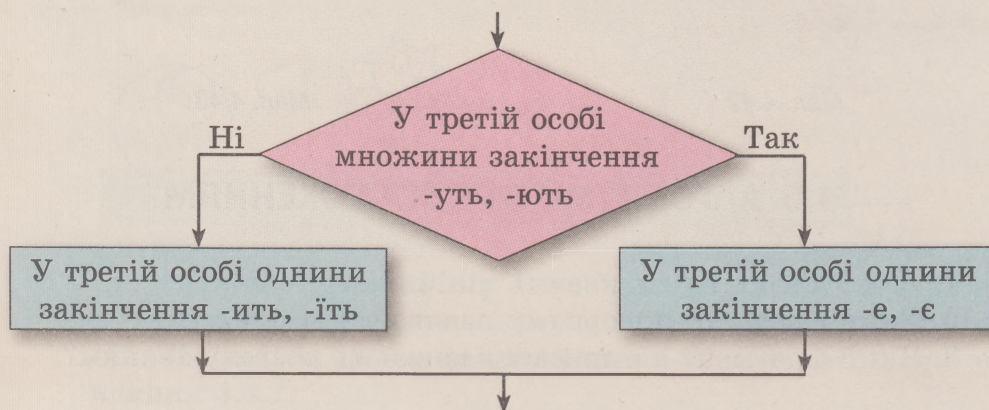
Блок-схему повного розгалуження зображено на малюнку 4.44.

Виконання повного розгалуження відбувається так: виконавець виконує **команду перевірки умови**; якщо результат виконання цієї команди **Так**, то виконавець виконує **послідовність команд 1**, після чого переходить до виконання першої команди наступного фрагмента алгоритму; якщо результат виконання цієї команди **Ні**, то виконавець виконує **послідовність команд 2**, після чого також переходить до виконання першої команди наступного фрагмента алгоритму.



Мал. 4.44. Повне розгалуження

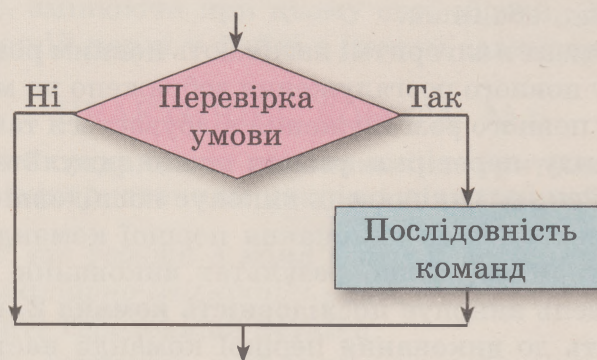
Наведемо приклад повного розгалуження в алгоритмі для визначення закінчення дієслова у третій особі однини (мал. 4.45).



Мал. 4.45. Повне розгалуження

В алгоритмах використовується також і **неповне розгалуження**.

Блок-схема неповного розгалуження має такий вигляд (мал. 4.46):

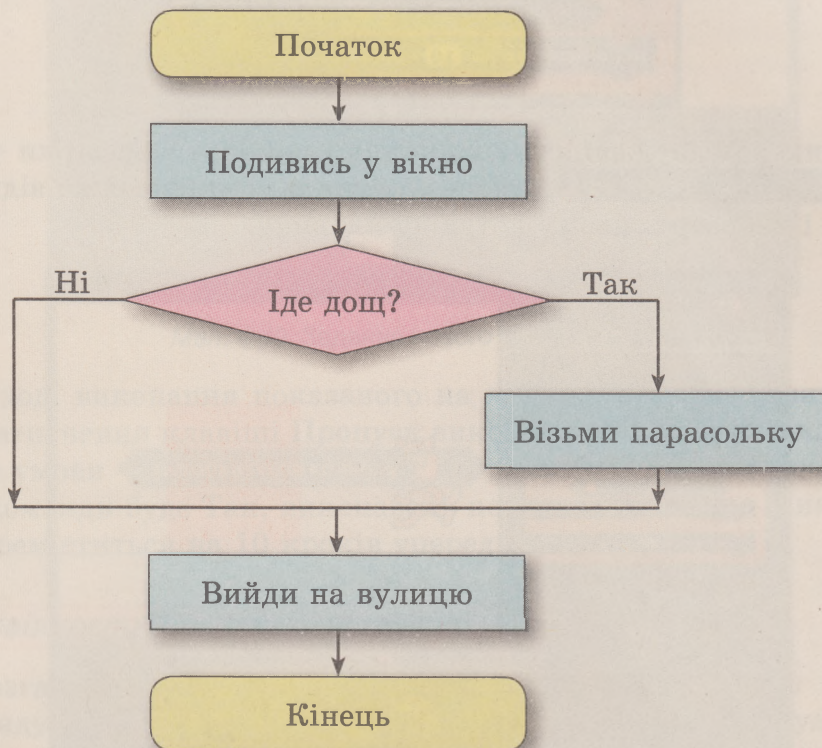


Мал. 4.46. Неповне розгалуження



Виконання неповного розгалуження відрізняється від виконання повного розгалуження тим, що якщо результат виконання команди перевірки умови **Ні**, то виконавець одразу переходить до виконання першої команди наступного фрагмента алгоритму.

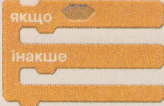

Наведемо приклад алгоритму з неповним розгалуженням (мал. 4.47):



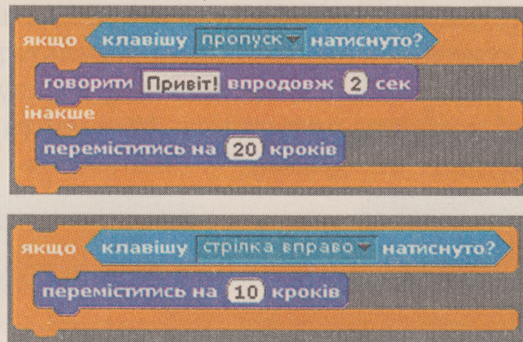
Мал. 4.47. Алгоритм з неповним розгалуженням

Алгоритм, що містить розгалуження, називають **алгоритмом з розгалуженням**.

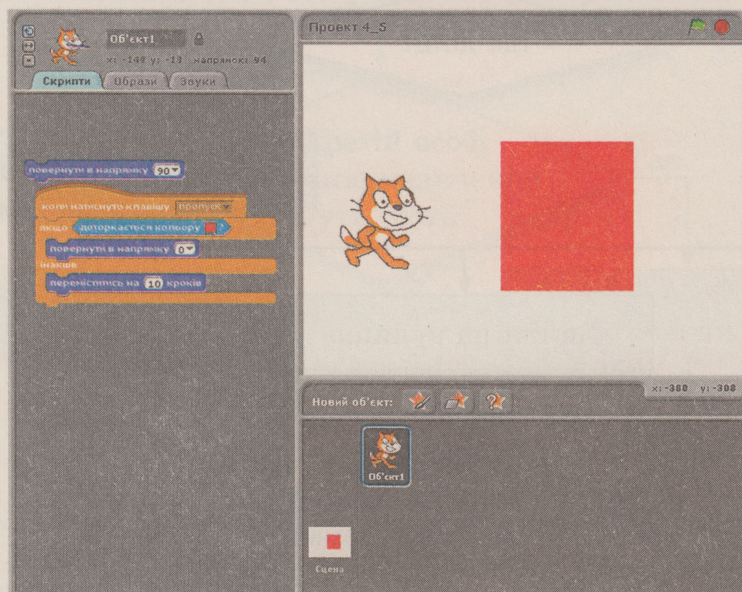
РОЗГАЛУЖЕННЯ В SCRATCH

Ви вже знаєте, що в **Scratch** можна використати команду  для організації повного розгалуження і  для організації неповного розгалуження.

Наведемо приклади таких команд:



У цих розгалуженнях використано команду перевірки умови **якщо клавішу пропуск натиснуто?** з групи Датчики, у якій можна відкрити список і вибрати в ньому потрібну клавішу.

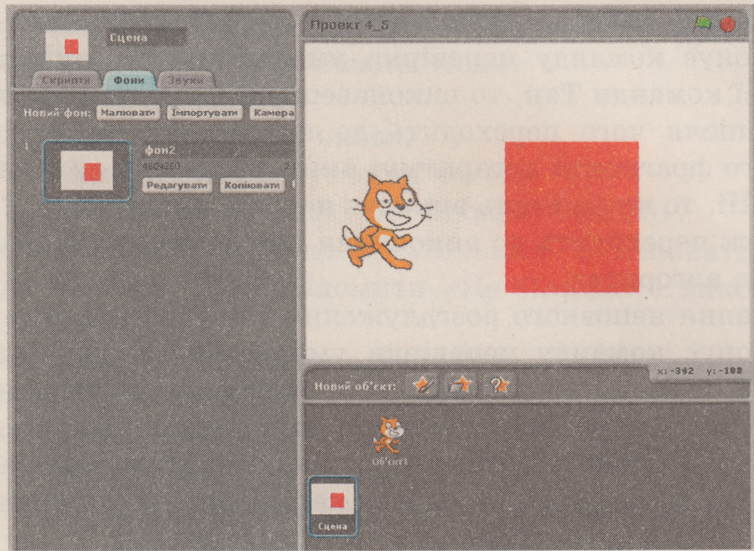


Мал. 4.48. Проект з розгалуженням

На малюнку 4.48 подано приклад проекту з використанням розгалуження. У цьому проекті використано команду перевірки умови **якщо доторкається кольору [red]?** з групи Датчики.

Щоб намалювати червоний прямокутник на Сцені, потрібно:

1. Вибрати Сцену в нижній правій частині вікна середовища.
2. Відкрити вкладку Фони.
3. Вибрати кнопку Малювати.
4. У вікні вбудованого графічного редактора намалювати червоний прямокутник.
5. Вибрати кнопку Гарзд.



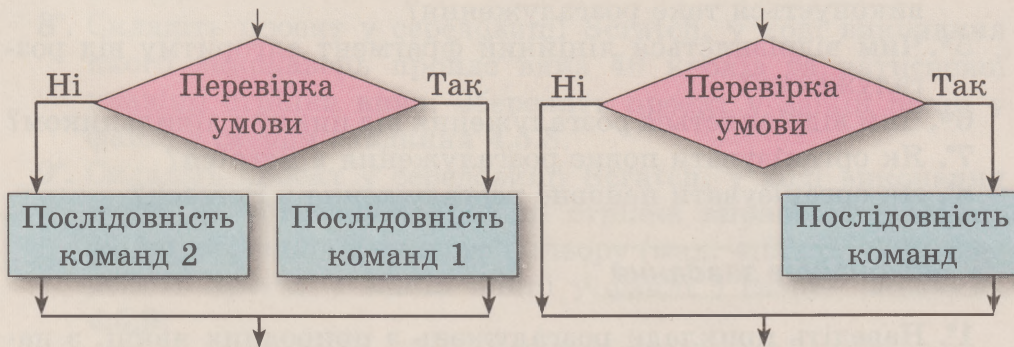
Мал. 4.49. Задання нового фону на Сцені

У ході виконання показано на малюнку 4.49 алгоритму під час натиснення клавіші **Пропуск** виконуватиметься команда перевірки умови **доторкається кольору [кількість] ?**. Якщо результат виконання цієї команди буде **Так**, виконавець повернеться вгору, а якщо **Ні**, то переміститься на 10 кроків уперед.

Найважливіше в цьому пункті

Розгалуженням називають фрагмент алгоритму, що містить команду перевірки умови і послідовності команд, які будуть виконуватися або не виконуватися залежно від результату виконання команди перевірки умови.

В алгоритмах використовують розгалуження двох видів: **повне розгалуження** (мал. 4.50) і **неповне розгалуження** (мал. 4.51).



Мал. 4.50. Повне розгалуження

Мал. 4.51. Неповне розгалуження



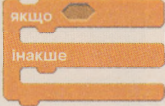
Розділ 4


Виконання повного розгалуження відбувається так: виконавець виконує команду перевірки умови: якщо результат виконання цієї команди **Так**, то виконавець виконує **послідовність команд 1**, після чого переходить до виконання першої команди наступного фрагмента алгоритму; якщо результат виконання цієї команди **Ні**, то виконавець виконує **послідовність команд 2**, після чого також переходить до виконання першої команди наступного фрагмента алгоритму.

Виконання неповного розгалуження відбувається так: виконавець виконує команду перевірки умови: якщо результат виконання цієї команди **Так**, то виконавець виконує **послідовність команд 1**, після чого переходить до виконання першої команди наступного фрагмента алгоритму; якщо результат виконання цієї команди **Ні**, то виконавець одразу переходить до виконання першої команди наступного фрагмента алгоритму.

Алгоритм, який містить розгалуження, називають **алгоритмом з розгалуженнями**.

У **Scratch** для організації повного розгалуження використовую-

ється команда , а для організації неповного розгалу-

ження використовується команда .



Дайте відповіді на запитання

- 1°. Який фрагмент алгоритму називають розгалуженням?
- 2°. Які два види розгалуження використовують в алгоритмах?
- 3°. Який вигляд має блок-схема повного розгалуження? Як виконується таке розгалуження?
- 4°. Який вигляд має блок-схема неповного розгалуження? Як виконується таке розгалуження?
- 5*. Чим відрізняється лінійний фрагмент алгоритму від розгалуження?
- 6*. Чим відрізняється розгалуження від циклу з лічильником?
- 7°. Як організувати повне розгалуження в **Scratch**?
- 8°. Як організувати неповне розгалуження в **Scratch**?



Виконайте завдання

- 1°. Наведіть приклади розгалужень з природних явищ, з навколишнього життя, з навчальної діяльності.



2*. Наведіть приклади розгалужень під час застосування правил математики, української мови.

3°. Виконайте алгоритм:

1. Задумати два цілих числа.
2. Обчислити суму задуманих чисел.
3. Порівняти «Знайдена сума більша, ніж 30».
4. Якщо **Так**, повідомити «Більша» і виконати команду 5, якщо **Ні**, повідомити «Не більша» і виконати команду 5.
5. Закінчити виконання алгоритму.



4°. Виконайте алгоритм:

1. Задумати два цілих числа.
2. Обчислити добуток задуманих чисел.
3. Порівняти «Знайдений добуток дорівнює 100».
4. Якщо **Так**, повідомити «Дорівнює» і виконати команду 5, якщо **Ні**, повідомити «Не дорівнює» і виконати команду 5.
5. Закінчити виконання алгоритму.

5°. Складіть блок-схему алгоритму із завдання № 3. Виконайте цей алгоритм для двох різних пар цілих чисел. Підберіть ці пари так, щоб кожного разу виконання алгоритму відбувалося по-різному.



6°. Складіть блок-схему алгоритму із завдання № 4. Виконайте цей алгоритм для двох різних пар цілих чисел. Підберіть ці пари так, щоб кожного разу виконання алгоритму відбувалося по-різному.



7*. Є дев'ять однакових на вигляд монет. Відомо, що серед них є одна фальшива і вона важча за справжню. Складіть блок-схему алгоритму виявлення фальшивої монети найменшою кількістю зважувань на шалькових терезах без важків.

8°. Складіть проект у середовищі **Scratch**, у ході виконання якого виконавець пройде вниз 40 кроків за натисненої клавіші **стрілка вниз**. Збережіть проект у вашій папці у файлі з іменем завдання 4.5.8.

9°. Складіть проект у середовищі **Scratch**, у ході виконання якого за натисненої клавіші **стрілка вправо** виконавець обійде перешкоду синього кольору (мал. 4.52, с. 170). Збережіть проект у вашій папці у файлі з іменем завдання 4.5.9.