



4.1. АЛГОРИТМИ. СЕРЕДОВИЩЕ СКЛАДАННЯ ТА ВИКОНАННЯ АЛГОРИТМІВ SCRATCH

1. Що таке команда? Що таке система команд виконавця?
2. Якими способами можна подавати команди? Наведіть приклади.
3. Що таке алгоритм?

КОМАНДИ І ВИКОНАВЦІ

Ви вже знаєте, що команда – це вказівка виконавцеві виконати певні дії.


Якщо вчитель математики скаже учневі: «Знайди суму чисел 22 і 12 та повідом її мені», то учень виконає операцію додавання і повідомить учителю число 34.

Якщо мати попросить доньку: «Підмети підлогу в кухні», то донька візьме віник і підмете підлогу в кухні.

Якщо суддя футбольного матчу під час гри дасть свисток, то гра зупиниться (мал. 4.1).



Мал. 4.1. Футбольний матч

Якщо користувач вибере кнопку **Закрити**  вікна програми, то вікно програми закриється.

У наведених прикладах відповідно *учень, донька, футболісти, комп'ютер* були виконавцями команд.

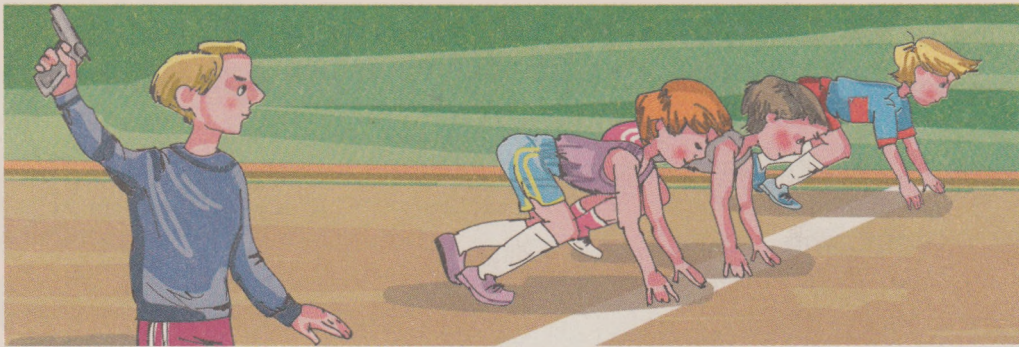
Команди виконавцям можуть подаватися різними способами:

а) спонукальними реченнями, наприклад:

- Закрий вікно.
- Розфарбуйте прямокутник.
- Підійдіть до мене;

б) звуковими сигналами, наприклад третій дзвінок у театрі, постріл стартового пістолета на спортивних змаганнях (мал. 4.2);

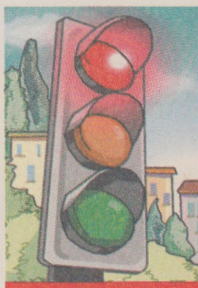
→ Розділ 4



Мал. 4.2. Спортивні змагання

в) світловими сигналами, наприклад сигнали світлофора (мал. 4.3);

г) жестами, наприклад команди регулювальника на вулиці водіям і пішоходам (мал. 4.4);



Мал. 4.3. Світлофор



Мал. 4.4. Регулювальник на перехресті

д) вибором команди меню, або вибором кнопки у вікні, або подвійним клацанням на значку та ін.

У результаті виконання команд відбуваються певні події, зокрема можуть змінюватися значення властивостей об'єктів, створюватися нові об'єкти, вилучатися існуючі об'єкти, відбуватися взаємодія об'єктів.

Працюючи з комп'ютером, ви використовували команди, що містяться в меню вікон програм, у контекстних меню об'єктів, у Головному меню. Після виконання кожної з команд відбуваються події з певними комп'ютерними об'єктами: вікнами, значками, програмами, рисунками, слайдами, файлами тощо.

Для кожного виконавця є команди, які він може виконати, і команди, які він виконати не може.

Наприклад, виконавець *учень* може виконати команди:

- Подивись на дошку
- Відкрий підручник



- Запиши в зошиті розв'язання задачі
- Послухай відповідь Тетянки,

але не може виконати команду **Стрибни у висоту на 10 м.**

Виконавець *дресирований собака* може виконати команди, яких навчив його хазяїн (мал. 4.5):

- Сидіти
- Поруч
- Служити,

але не може виконати команду **Знайти добуток чисел 125 і 183.**



Мал. 4.5. Дресирований собака

Виконавець *комп'ютер* може виконати команди, наприклад:

- Запустити програму
- Зберегти файл з малюнком на диску,

але не може виконати команду **Прибрати в кімнаті.**

У той самий час команду **Прибрати в кімнаті** може виконати інший виконавець, наприклад *людина* або спеціально навчений *робот*.

Команди, які може виконати виконавець, складають **систему команд цього виконавця.**

Звертаємо вашу увагу! Кожний виконавець може виконати тільки ті команди, які входять до його системи команд.

Кожний виконавець виконує команди, які входять до його системи команд, у певному **середовищі**. Так, у розглянутих вище прикладах для виконавця *учень* середовищем виконання команд є школа, для виконавців *футболісти* – футбольне поле, для виконавця *донька* – кухня тощо. Учень і донька можуть виконувати команди і в іншому середовищі. А футболісти і регулювальник – лише у визначених середовищах.

АЛГОРИТМИ

Ви вже знаєте, що послідовність команд виконавцю, що визначає, які дії і в якому порядку потрібно виконати, щоб досягти поставленої мети, називають **алгоритмом**.

Так, наприклад, алгоритмом приготування яєчні є така послідовність команд:

1. Поставити пательню на плиту.
2. Покласти на пательню шматочок вершкового масла.
3. Увімкнути конфорку.
4. Чекати, поки масло на пательні розтане.
5. Розбити по черзі два яйця і вилити їх вміст на пательню.
6. Посолити.
7. Чекати, поки загусне білок.
8. Вимкнути конфорку.

Складемо алгоритм для розв'язування такої задачі.



Мал. 4.6

Задача. Є повна посудина рідини місткістю 8 літрів і дві порожні посудини місткістю 5 літрів та 3 літри (мал. 4.6). Потрібно одержати в одній із цих посудин 1 літр рідини.

Розробляючи алгоритм, потрібно перш за все визначити, для якого виконавця він буде призначений, і використовувати в алгоритмі тільки

ти ті команди, які входять до системи команд цього виконавця.

Розглянемо виконавця, який має таку систему команд:

- 1) Перелити вміст указаної посудини в іншу вказану посудину.
- 2) Наповнити вказану посудину рідиною з іншої вказаної посудини.
- 3) Вивести повідомлення.

Для виконавця з наведеною системою команд алгоритм розв'язування цієї задачі буде такий:

1. Наповнити 3-літрову посудину рідиною з 8-літрової.
2. Перелити вміст 3-літрової посудини в 5-літрову.
3. Наповнити 3-літрову посудину рідиною з 8-літрової.
4. Наповнити 5-літрову посудину рідиною з 3-літрової.
5. Вивести повідомлення: «1 літр рідини отримано у 3-літровій посудині».

Комп'ютер як виконавець також виконує різноманітні алгоритми. Алгоритм, призначений для виконання комп'ютером, називають **програмою**.



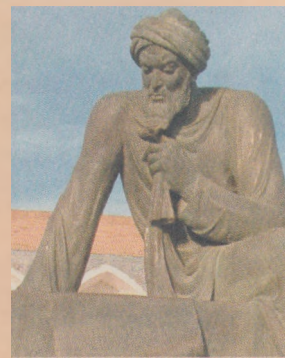
Ви вже ознайомилися з виконанням комп'ютером таких програм, як графічний редактор, редактор презентацій, середовище складання і виконання алгоритмів **Scratch** і багатьох інших.

Сучасні комп'ютерні програми досить складні. Часто на носіях вони зберігаються в багатьох файлах. Інколи такі програми називають **проектами**. Розробляють і створюють такі складні проекти великі групи спеціалістів.



Цікаві факти з історії

Слово *алгоритм* походить від імені видатного вченого середньовічного Сходу **Мухаммеда бен-Муса аль-Хорезмі** (783–850 рр.) (мал. 4.7), який у своїх наукових працях сформулював правила виконання чотирьох основних арифметичних дій: додавання, віднімання, множення та ділення. Європейські вчені ознайомилися з його працями завдяки перекладам їх на латину. У ході перекладу ім'я автора було подано як **Algorithmus**. Звідси й пішло слово *алгоритм*. А правила виконання арифметичних дій, які він розробив, вважають першими алгоритмами.



Мал. 4.7. Аль-Хорезмі

СЕРЕДОВИЩЕ СКЛАДАННЯ ТА ВИКОНАННЯ АЛГОРИТМІВ SCRATCH

У 5-му класі ви продовжите вивчати середовище складання і виконання алгоритмів **Scratch** (англ. *scratching* – шкрябання або спеціальні рухи руками діджеїв вінілових платівок для змішування музичних тем).

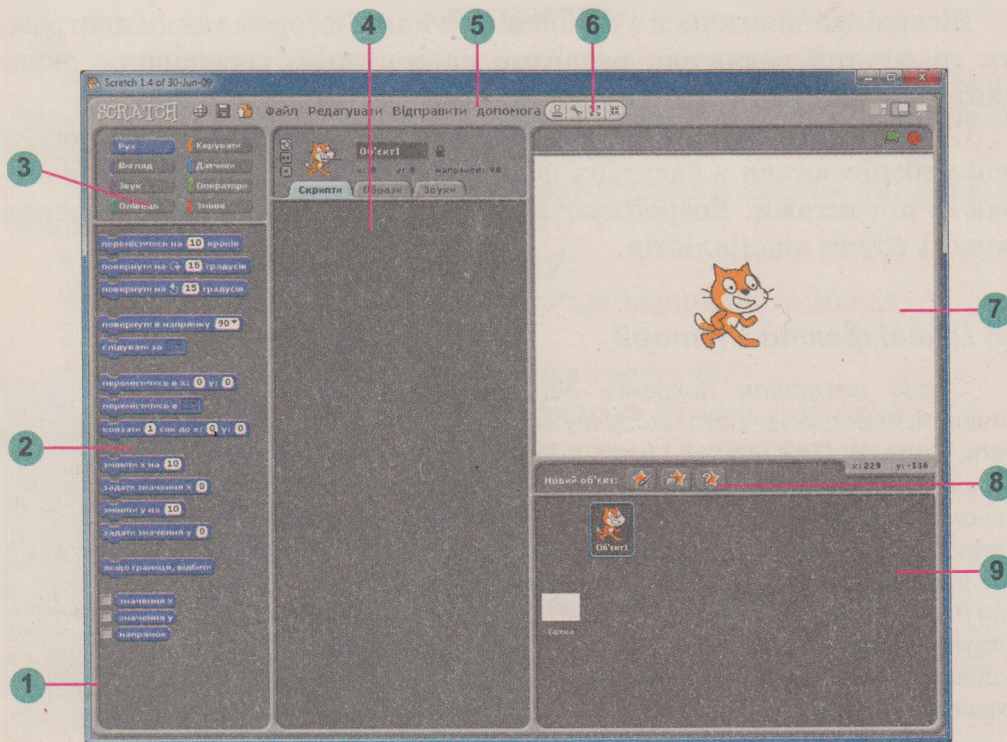
Ви вже вмієте складати і виконувати в цьому середовищі алгоритми для різних виконавців. Виконуючи складені алгоритми, ці виконавці можуть рухатися, малювати, говорити, змінювати зовнішній вигляд, виконувати обчислення та ін.

Нагадаємо, що для запуску **Scratch** можна використати **Головне меню** (*Пуск* ⇒ *Усі програми* ⇒ *Scratch* ⇒ *Scratch*) або значок



на **Робочому столі**. Після цього відкривається головне вікно програми, яке розділено на три частини: ліву, центральну і праву (мал. 4.8).

У лівій частині вікна розташовано блоки з командами виконавцям. Ця частина вікна називається **Палітра блоків** (мал. 4.8, № 1).



- | | |
|------------------------------|--------------------------------------|
| 1. Палітра блоків | 6. Панель інструментів |
| 2. Блоки з командами | 7. Сцена |
| 3. Кнопки груп блоків команд | 8. Кнопки додавання нових виконавців |
| 4. Область скриптів | 9. Список виконавців |
| 5. Рядок меню | |

Мал. 4.8. Вікно програми Scratch

Для зручності всі ці команди розподілено на 8 груп, імена яких відображаються на кнопках: **Рух**, **Вигляд**, **Звук**, **Олівець** та інші (мал. 4.8, № 3) у верхній частині Палітри блоків. Якщо вибрати одну із цих кнопок, вона зафарбовується в колір, який визначено для цієї групи команд, а в нижній частині Палітри блоків (мал. 4.8, № 2) відкриється список команд цієї групи. На малюнку 4.8 відображено команди, що входять до групи **Рух**.

Сценарій – детально розроблений план проведення певних заходів, здійснення яких-небудь дій.

У центральній частині вікна розташована **Область скриптів** (англ. *script* – сценарій) (мал. 4.8, № 4). Вона призначена для складання алгоритмів для виконавців. Блоки з командами перетягуються в **Область скриптів** з Палітри блоків.



У правій частині вікна розташована **Сцена** (мал. 4.8, № 7). На **Сцені** демонструється виконання команд виконавцями. Після запуску програми **Scratch** за замовчуванням у центрі **Сцени** з'являється виконавець **Рудий кіт**. Це початкове положення виконавця можна змінити, перетягнувши його в інше місце **Сцени**. Можна розміщувати на **Сцені** й інших виконавців. Розміри **Сцени** – 480 на 360 кроків виконавця.

Нагадаємо деякі відомі вам команди виконавцям та результати їх виконання (табл. 4.1).


Таблиця 4.1

Команди та результати їх виконання

Команда	Результат виконання команди
Команди групи Рух	
	Виконавець переміщується вперед на 10 кроків
	Виконавець повертається на 15 градусів за годинниковою стрілкою
	Виконавець повертається на 15 градусів проти годинникової стрілки
	Виконавець повертається у вибраному напрямку (90) вправо (-90) вліво (0) вгору (180) вниз
Команда групи Вигляд	
	Біла виконавця з'являється напис «Привіт!», який зникає через 2 секунди
Команди групи Олівець	
	Зі Сцени зникають всі лінії, намальовані раніше
	Виконавець піднімає олівець (Після виконання цієї команди виконавець не залишає слід під час свого переміщення)
	Виконавець опускає олівець (Після виконання цієї команди виконавець залишає слід під час свого переміщення)

Розділ 4

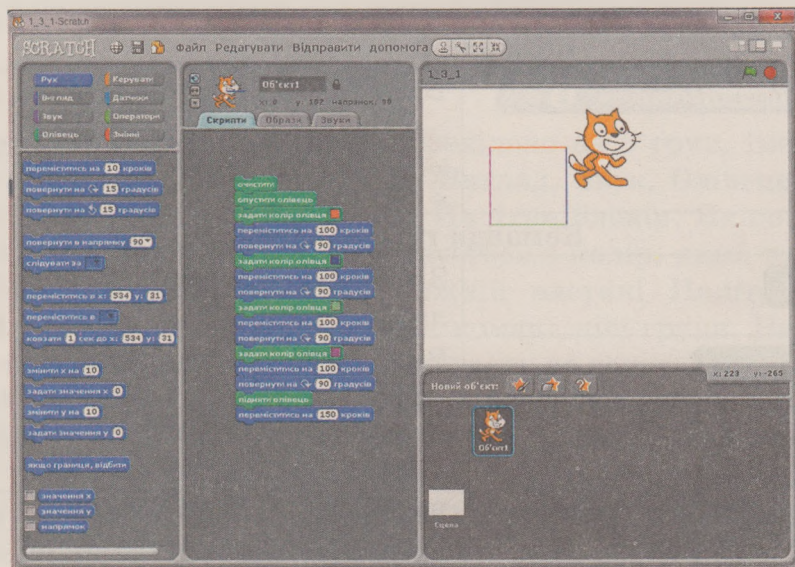
Продовження таблиці 4.1

Команда	Результат виконання команди
	<p>Установлюється колір олівця, яким Виконавець залишає слід під час свого переміщення.</p> <p>Для вибору кольору потрібно:</p> <ol style="list-style-type: none"> 1. Вибрати поле у блоці цієї команди. 2. Вибрати потрібний колір на палітрі, що відкрилася.

Деякі блоки з командами мають одне, два або три поля, у яких містяться або можна ввести певні значення. Щоб змінити ці значення, потрібно:

1. Вибрати відповідне поле блока.
2. Увести з клавіатури потрібне значення або вибрати його зі списку.
3. Натиснути клавішу **Enter** або вибрати будь-яку точку поза цим блоком.

Наведемо як приклад алгоритм (мал. 4.9), виконавши який, Рудий кіт намалює різними кольорами квадрат зі стороною завдовжки 100 кроків.




Мал. 4.9. Алгоритм малювання різнокольорового квадрата зі стороною завдовжки 100 кроків і результат його виконання

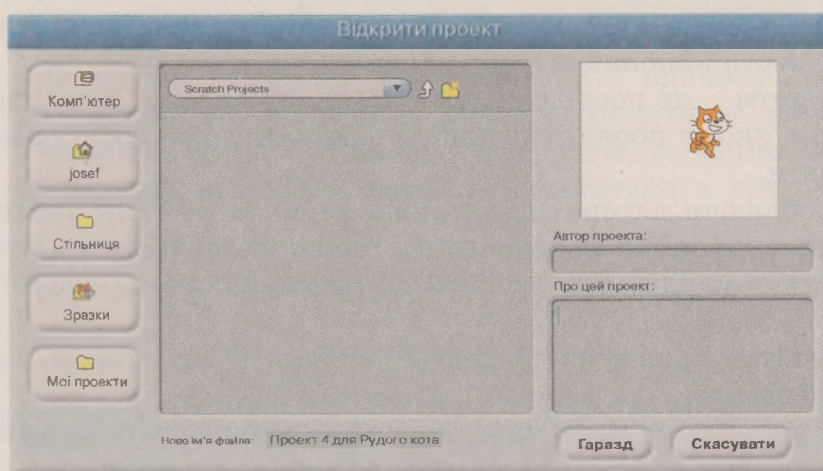


Нагадаємо: щоб виконати у **Scratch** складений алгоритм, слід вибрати будь-яку команду цього алгоритму в **Області скриптів**.

ЗБЕРЕЖЕННЯ ПРОЕКТІВ


Ви вже знаєте, що проект можна зберегти на носії для подальшого використання. Це можна зробити, вибравши кнопку  **Зберегти цей проект** у **Рядку меню** вікна або виконавши **Файл** ⇒ **Зберегти**.

Якщо проект зберігається вперше, то відкривається вікно **Зберегти Проект** (мал. 4.10), у якому потрібно вибрати папку для збереження проекту і ввести ім'я файлу з проектом у поле **Нове ім'я файлу**. Доцільно зберігати проекти або в папці **Мої проекти**, або на тому носії і в тій папці, яку вам укаже вчитель.



Мал. 4.10. Вікно **Зберегти Проект**

Зробити поточною папку **Мої проекти** можна, вибравши кнопку **Мої проекти** на панелі в лівій частині вікна **Зберегти Проект** (мал. 4.10). Вибір кнопки **Стільниця** робить поточною папку **Робочий стіл**. А вибрати потрібний носій даних і папку на ньому можна, відкривши список доступних носіїв даних, вибравши кнопку **Комп'ютер**.

У цьому самому вікні можна за потреби створити нову папку, вибравши кнопку .

У цьому самому вікні можна в поле **Автор проекту** ввести своє прізвище, а в поле **Про цей проект** – короткий опис проекту.

Після введення даних у поля цього вікна потрібно вибрати кнопку **Гаразд**.

Розділ 4

Якщо вищевказаними діями проект зберігається не вперше, то жодних вікон не відкривається і він автоматично зберігається в тій самій папці і під тим самим іменем, що були вказані під час першого збереження проекту.

Якщо потрібно зберегти проект в іншій папці і/або під іншим іменем, то потрібно виконати **Файл** ⇒ **Зберегти як**. Після цього у вікні **Зберегти проект** потрібно вибрати іншу папку і/або вказати інше ім'я файлу проекту.

! Найважливіше в цьому пункті

Команда — це вказівка виконавцеві виконати певні дії.

Команда може подаватися спонукальними реченнями, звуковими і світловими сигналами, жестами, вибором команд меню, кнопок та ін.


Команди, які може виконати виконавець, складають **систему команд цього виконавця**.

Алгоритм — це послідовність команд виконавцю, що визначає, які дії і в якому порядку потрібно виконати, щоб досягти поставленої мети.

Розробляючи алгоритм, потрібно перш за все визначити, для якого виконавця він призначений, і використовувати в алгоритмі тільки ті команди, які входять до системи команд цього виконавця.

Алгоритм, призначений для виконання комп'ютером, називають **програмою**.


Щоб виконати в **Scratch** складений алгоритм, потрібно вибрати будь-яку команду цього алгоритму в **Області скриптів**.

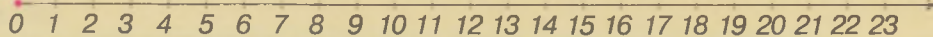
Проект можна зберегти на носії даних для подальшого використання: вибравши кнопку  **Зберегти цей проект**, виконавши **Файл** ⇒ **Зберегти** або **Файл** ⇒ **Зберегти як**.

? Дайте відповіді на запитання


- 1°. Що називають командою?
- 2°. Як може подаватися команда?
- 3°. Що називають системою команд виконавця?
- 4°. Що називають алгоритмом?
- 5°. Що називають програмою?
- 6°. Як скласти алгоритм для виконавця в середовищі **Scratch**?
- 7°. Як виконати алгоритм для виконавця в середовищі **Scratch**?
- 8°. Як зберегти на носії створений проект?

**Виконайте завдання**

- 1°. Складіть алгоритм приготування чаю.
-  2°. Складіть алгоритм приготування вашої улюбленої страви.
- 3°. Виконайте алгоритм:
 1. Знайти суму чисел 1 і 3.
 2. Додати до одержаної суми число 5.
 3. Додати до одержаної суми число 7.
 4. Додати до одержаної суми число 9.
 5. Додати до одержаної суми число 11.
 6. Повідомити результат.
- 4*. Знайдіть і запишіть у зошит по одному алгоритму в підручниках математики і української мови.
- 5°. Є координатний промінь з позначеними на ньому числом нуль і натуральними числами. На цьому промені мешкає виконавець **Коник**, який уміє переміщуватися по ньому, виконуючи команди: 1) стрибни на 3 одиниці праворуч; 2) стрибни на 2 одиниці ліворуч. Початкове положення **Коника** – точка 0. Складіть алгоритм, за яким **Коник** за найменшу кількість стрибків опиниться в точці: а) 12; б) 7.





Мал. 4.11. Координатний промінь

- 6°. Є повна посудина місткістю 8 літрів та дві порожні посудини місткістю 3 літри і 5 літрів. Складіть алгоритм одержання в одній з посудин 2 літрів рідини для виконавця, систему команд якого описано в даному пункті.
- 7*. Потрібно приготувати суп з концентрату. У нас є піскові годинники на 3 хвилини і 8 хвилин. Складіть алгоритм відліку часу для приготування супу, якщо його треба готувати рівно: а) 5 хвилин; б) 7 хвилин; в) 10 хвилин.
- 8°. Візнику потрібно перевезти в човні через річку вовка, козу та капусту. У човні, крім візника, вміщується або тільки вовк, або тільки коза, або тільки капуста. На березі не можна залишати козу з вовком або козу з капустою. Складіть алгоритм перевезення. (Ця старовинна задача вперше трапляється в математичних рукописах VIII ст.).
-  9°. Двом студентам потрібно переправитися з одного берега річки на інший. Вони побачили двох хлопчиків на маленькому човні. У ньому можуть переправлятися або один



Розділ 4

студент, або один чи двоє хлопчиків. Складіть алгоритм переправлення студентів. (Після переправлення студентів човен повинен залишитися у хлопчиків).

- 10*. Потрібно підсмажити 3 скибки хліба на пательні, що вміщує одночасно тільки 2 такі скибки. На смаження однієї сторони витрачається 2 хвилини. Складіть систему команд для виконавця, який би зміг досягти поставленої мети. Складіть алгоритм для цього виконавця. Який час витрачається під час виконання вашого алгоритму? Складіть алгоритм, на виконання якого витрачається 6 хвилин.
-  11°. Складіть проект для Рудого kota, щоб він перейшов у правий верхній кут Сцени і привітався. Збережіть створений проект у вашій папці у файлі з іменем завдання 4.1.11.
- 12°. Складіть проект для Рудого kota, щоб він обійшов по краях Сцену і в кожному її куті повідомив, де він знаходиться. Збережіть створений проект у вашій папці у файлі з іменем завдання 4.1.12.
-  13°. Складіть проект для Рудого kota, щоб він намалював на Сцені два прямокутники різних розмірів і різних кольорів. Збережіть створений проект у вашій папці у файлі з іменем завдання 4.1.13.



4.2. СПОСОБИ ПОДАННЯ АЛГОРИТМІВ. РЕДАГУВАННЯ І ВІДКРИВАННЯ ПРОЕКТІВ У СЕРЕДОВИЩІ SCRATCH



1. Що таке алгоритм?
2. Якими способами можуть подаватися команди виконавцям? Наведіть приклади.
3. Для чого використовуються команди **Зберегти** і **Зберегти як**? Яка відмінність в їх виконанні?

СПОСОБИ ПОДАННЯ АЛГОРИТМІВ

Ви вже знаєте, що команди виконавцям можуть подаватися різними способами: спонукальними реченнями, звуковими або світловими сигналами, жестами, вибором команди меню або кнопки вікна та ін.

Відповідно до цього алгоритми також можуть бути подані різними способами: