

Розділ 3. Алгоритми та програми

Прочитавши цей розділ, ви дізнаєтеся про:

- ▶ програмні об'єкти (спрайти) та їх властивості в середовищі **Scratch**
- ▶ створення програмних об'єктів (спрайтів) у середовищі **Scratch**
- ▶ дії над спрайтами та програмні події в середовищі **Scratch**
- ▶ програмне опрацювання події в середовищі **Scratch**
- ▶ вкладені повторення і розгалуження
- ▶ створення проектів у середовищі **Scratch** з використанням вкладених повторень і розгалужень

Виконавши запропоновані в розділі вправи, практичні завдання, ви навчитеся:

- додавати об'єкти до програмного проекту
- змінювати значення властивостей об'єктів, у тому числі програмно
- програмувати опрацювання подій
- застосовувати вкладені алгоритмічні структури повторення та розгалуження
- розкласти задачу на підзадачі та розв'язувати їх



3.1. ПРОГРАМНІ ОБ'ЄКТИ ТА ПРОГРАМНЕ ОПРАЦЮВАННЯ ПОДІЇ В SCRATCH

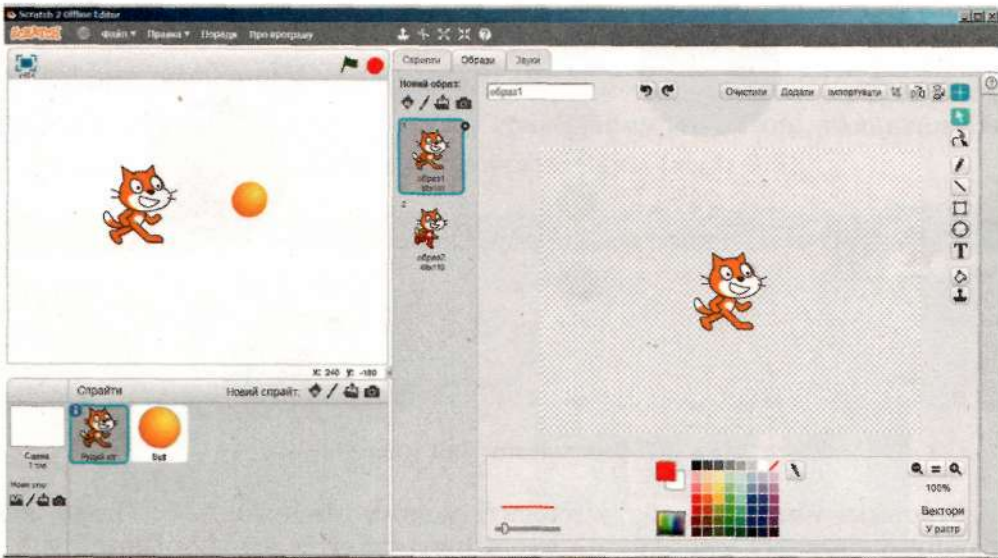
1. Які об'єкти в середовищі Scratch ви знаєте?
2. Які властивості об'єктів середовища Scratch ви знаєте?
3. Які дії можна виконувати над об'єктами в середовищі Scratch?

ПРОГРАМНІ ОБ'ЄКТИ ТА ЇХ ВЛАСТИВОСТІ В SCRATCH


У 5-му класі ви створювали проекти для виконавців у середовищі складання та виконання алгоритмів Scratch. Ви використовували різних виконавців, кожен з них мав один або кілька образів (костюмів). Нагадаємо, що і виконавці, і їх образи є **програмними об'єктами** середовища Scratch.

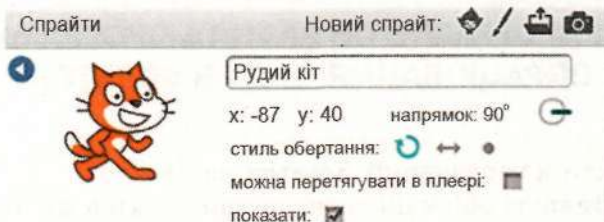
У Scratch виконавців також називають **спрайтами** (англ. *sprite* – уявний герой).

Усі спрайти мають свої властивості: ім'я, положення на Сцені, розміри, напрямок, у якому вони будуть рухатися, колір костюма та інші (мал. 3.1). Кожна із цих властивостей має своє значення.



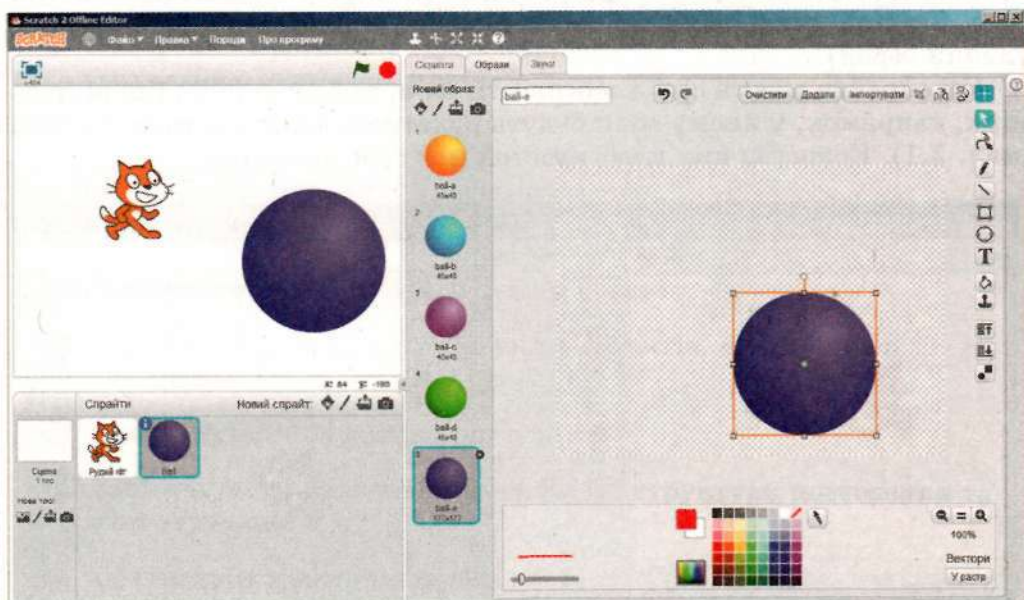
Мал. 3.1. Виконавці та їх образи

Так, на малюнку 3.1 у лівій нижній частині вікна в розділі **Спрайти** ви бачите ескізи двох виконавців: **Рудий кіт** і **Ball** (англ. *ball* – м'яч). Поточним (вибраним) є виконавець **Рудий кіт**. Переглянути значення властивостей цього об'єкта можна в розділі **Інформація**, який відкриється вибором кнопки  в лівому верхньому куті ескізу вибраного виконавця (мал. 3.2).







Мал. 3.2. Інформація про об'єкт Рудий кіт

На малюнку 3.3 ви бачите вибраного в розділі **Спрайти** виконавця **Ball** та його образи на вкладці **Образи** в центральній частині вікна. Значення деяких властивостей об'єкта та його образів ви можете бачити на цьому малюнку, інші можна переглянути в розділі **Інформація** для цього виконавця.



Мал. 3.3. Виконавець Ball і його образи

Створити новий спрайт у проєкті можна кількома способами, вибравши відповідну кнопку на панелі інструментів розділу **Спрайти**:

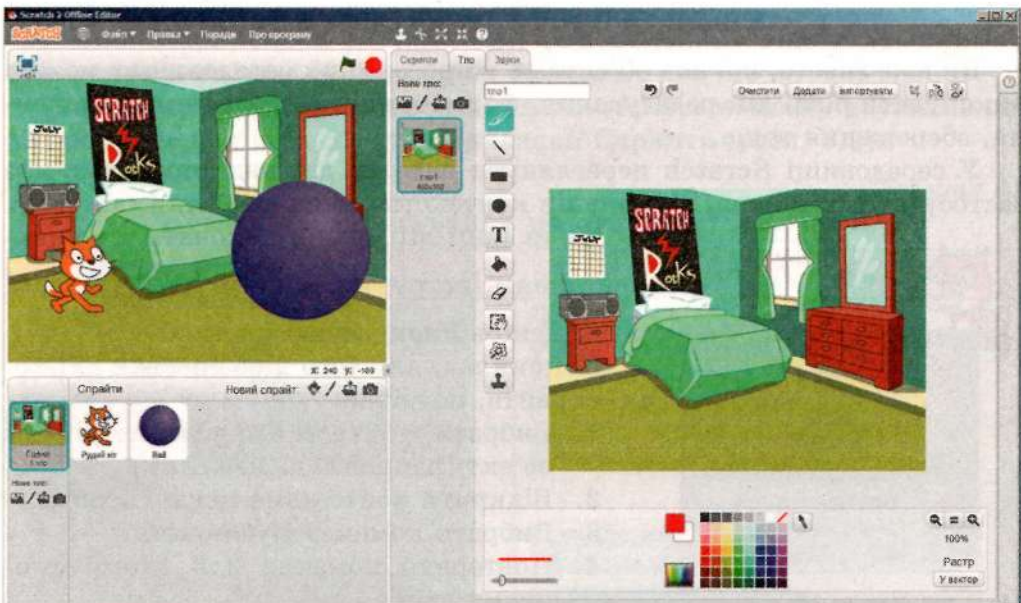
-  – обрати готовий об'єкт з бібліотеки спрайтів (мал. 3.4);
-  – намалювати в графічному редакторі, вбудованому в середовище Scratch;
-  – вставити з файла;
-  – сфотографувати камерою, підключеною до комп'ютера.



Мал. 3.4. Бібліотека спрайтів у Scratch

Після цього новий об'єкт з'являється на вкладці **Спрайти** і його можна використовувати у проєкті.

Програмним об'єктом у середовищі **Scratch** є також і **Сцена**. Сцена має такі властивості: розмір (480 на 360 кроків виконавця) і тло. Якщо вибрати цей об'єкт, то на вкладці **Тло** можна змінити зображення на тлі **Сцени**, використавши кнопки на панелі інструментів : обрати з бібліотеки, намалювати у графічному редакторі, завантажити з будь-якого файлу з графічним зображенням (мал. 3.5).



Мал. 3.5. Сцена та її тло

Значення деяких властивостей спрайтів можна змінити в розділі **Інформація**, наприклад: ім'я об'єкта, напрямок руху, стиль обертання; інших – у вбудованому графічному редакторі у правій частині вікна (початкове положення на **Сцені**, колір, розміри тощо).

Змінити значення деяких властивостей спрайтів можна і під час виконання проекту. Наприклад, через кілька кроків руху змінити колір об'єкта або його розміри, змінити положення об'єкта або взагалі сховати під час виконання деякої умови, змінити тло **Сцени** під час переходу на новий кадр сценарію тощо.

Такі команди розміщено в різних групах вкладки **Скрипти**, як показано в таблиці 3.1.

Таблиця 3.1

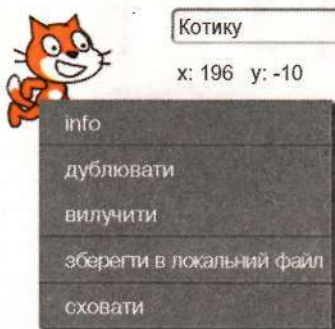
Команди змінення значень

Група Рух	Група Вигляд
тип обертання ліворуч-праворуч	показати
змінити x на 10	сховати
задати значення x 0	змінити образ на costume2
якщо на межі, відбити	наступний образ
	змінити тло на desert


ПРОГРАМНІ ДІЇ ТА ПОДІЇ З ОБ'ЄКТАМИ В SCRATCH

Ви вже знаєте, що над об'єктами у програмних середовищах можна виконувати різні дії: редагування, форматування, створення, видалення, збереження тощо.

У середовищі **Scratch** переглянути перелік дій над спрайтами або застосувати їх можна, відкривши контекстне меню об'єкта (мал. 3.6).



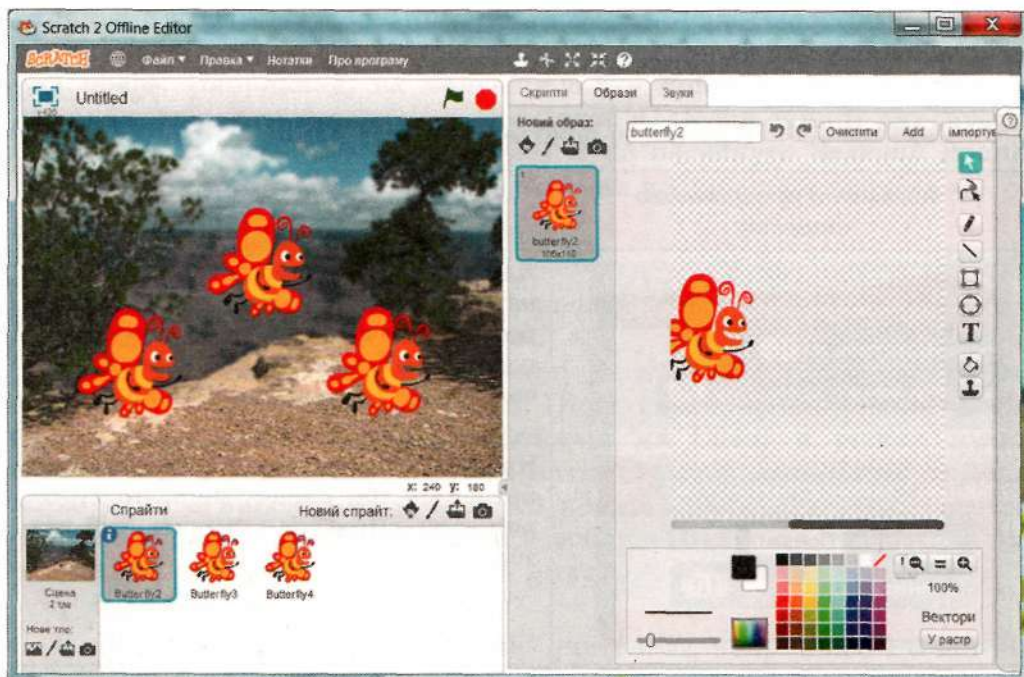
Мал. 3.6. Контекстне меню ескізу спрайта

Ці та інші дії можна виконати і з використанням інструментів , які розміщено в **Рядку меню** програми.

Наприклад, для того щоб зробити дві копії спрайта, потрібно:

1. Вибрати в області **Спрайти** потрібний об'єкт (наприклад, *метелик*).
2. Відкрити контекстне меню спрайта.
3. Вибрати команду **Дублювати**.
4. Повторити команди 2–3 цього алгоритму двічі.

Результат представлено на малюнку 3.7.



Мал. 3.7. Дублювання спрайтів

Під час виконання програмного проекту **Scratch** зі спрайтами можуть бути пов'язані події, опрацювання яких приведе до змінення значень властивостей об'єктів. Наприклад, після настання події «натискання на клавіатурі клавіші стрілка вправо» виконавець повертається в напрямку вправо та робить 10 кроків (реакція на подію). Або після настання події «сцена змінила свій фон на синій» відбувається реакція – спрайт змінює свій колір на зелений.

Команди, які призначено для відстежування моменту настання події, розміщено в різних групах вкладки **Скрипти** (табл. 3.2).

Таблиця 3.2



Команди відстежування моменту настання події

Група Події	Вкладка Датчики
<p>коли натиснуто </p> <p>коли натиснуто клавішу <input type="text" value="пропуск"/></p> <p>коли спрайт натиснуто</p> <p>коли тло зміниться на <input type="text" value="desert"/></p>	<p>доторкається <input type="text" value="Межа"/> ?</p> <p>доторкається кольору <input type="text" value=""/> ?</p> <p>колір <input type="text" value=""/> торкається <input type="text" value=""/> ?</p> <p>клавішу <input type="text" value="пропуск"/> натиснуто?</p> <p>мишку натиснуто?</p>

Приклади фрагментів таких програмних проектів та їх опис наведено в таблиці 3.3.

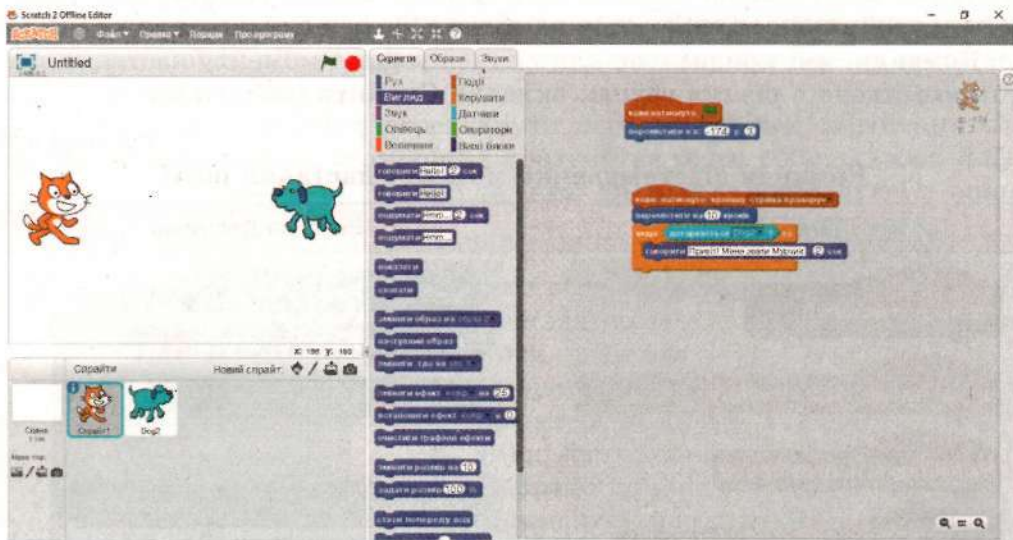
Таблиця 3.3

Приклади фрагментів та їх опис

Фрагмент програмного проекту	Опис дій
	<p>Якщо під час виконання проекту буде натиснуто клавішу пропуск, спрайт переміститься по Сцені з поточного місця розташування на 10 кроків у напрямку свого руху та змінить свій колір</p>
	<p>Якщо під час виконання проекту буде натиснуто ліву кнопку миші, то спрайт вимовить «Привіт!» і змінить свій розмір</p>

Як бачимо у цих фрагментах, події (натискання клавіші **пропуск** або кнопки миші) приводять до змінення значень властивостей спрайта – зміна положення, розміру, кольору.

В одному проекті для кожного виконавця може бути створено кілька скриптів, кожний з яких описує дії об'єкта на настання тієї чи іншої події. Кожен скрипт починається з команди, блок якої має особливу форму – до нього зверху не можна приєднати іншу команду. Усі ці



Мал. 3.8. Кілька скриптів для одного спрайта



скрипти не будуть пов'язані між собою, але будуть розміщуватися в одній **Робочій області** та будуть виконуватися після настання відповідних подій. Наприклад, на малюнку 3.8 у **Робочій області** наведено два скрипти для виконавця з ім'ям *Спрайт1*. Один скрипт виконується, коли під час виконання проекту буде натиснуто *Зелений прапорець*, інший – коли користувач натисне на клавіатурі клавішу →.

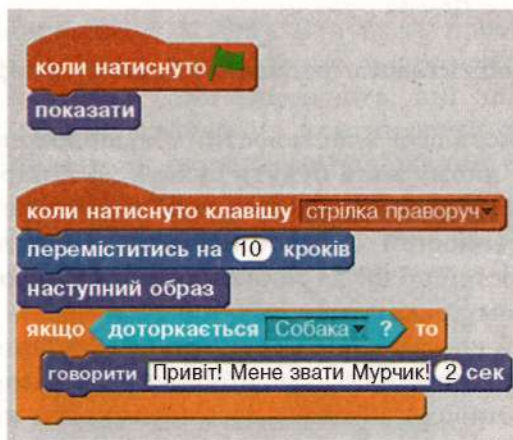


Працюємо з комп'ютером

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

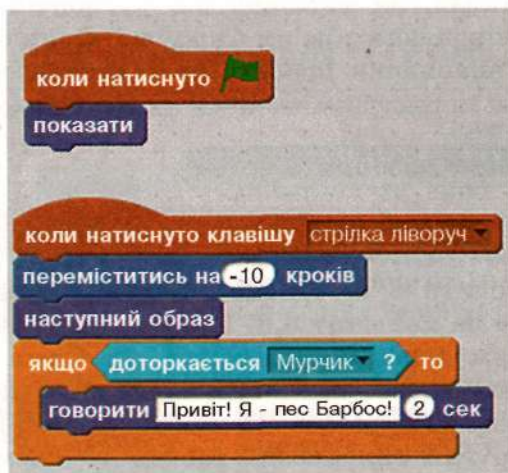
Завдання. Створити в середовищі **Scratch** проект за сценарієм, який представлено на малюнку 3.8. Після вибору користувачем кнопки із зображенням зеленого прапорця на **Сцені** в різних кутах з'являються два виконавці. Після натискання на клавіатурі **стрілки вліво** один з виконавців робить 10 кроків уліво, після натискання **стрілки вправо** другий виконавець робить 10 кроків управо. Якщо один з них торкнувся іншого – він вітається та називає себе.

1. Запустіть **Scratch**.
2. Відкрийте проект, указаний учителем, наприклад **Розділ 3\ Пункт 3.1\вправа 3.1.sb2**.
3. Опишіть, які об'єкти використано у проекті, які образи вони мають.
4. Виберіть у розділі **Спрайти** об'єкт *Рудий кіт*. Перегляньте інформацію про цей спрайт, надайте йому нове ім'я – *Мурчик*, установіть його початкове положення в лівому нижньому куті **Сцени**, приховайте його зображення.
5. Створіть у **Робочій області** для спрайта *Мурчик* скрипт за наведеним зразком:



6. Виберіть об'єкт *Собака*. Перегляньте інформацію про цей спрайт, надайте йому нове ім'я – *Барбос*, установіть його початкове положення – у правому нижньому куті **Сцени**, приховайте його зображення.

7. Створіть у **Робочій області** для спрайта *Барбос* скрипт за наведеним зразком:



- Виберіть область **Сцена** та вкладку **Тло**. Оберіть тло **Сцени** з бібліотеки – графічне зображення *party*.
- Переіменуйте об'єкт фонового зображення на ім'я *Свято*. Для цього в лівому верхньому куті **Робочої області** введіть у текстове поле вказаний текст.
- Виконайте проект. За якої події розпочинається виконання проекту? За яких подій відбувається переміщення об'єктів по **Сцені**? За яких подій виконавці вимовляють текст?
- Збережіть проект з тим самим іменем у вашій папці.

Найважливіше в цьому пункті

Програмними об'єктами середовища **Scratch** є **спрайти** (виконавці) та **Сцена**.

Усі спрайти мають свої властивості: ім'я, положення на **Сцені**, розміри, напрямок, у якому вони будуть рухатися, колір костюма та інші. **Сцена** як об'єкт середовища **Scratch** має властивості **Тло** та розмір. Кожна із цих властивостей має своє значення. Переглянути та змінити значення властивостей об'єктів можна в розділі **Інформація** або у вбудованому графічному редакторі.

Створити новий спрайт або вибрати тло для сцени можна кількома способами: обрати готовий об'єкт із бібліотеки, намалювати у графічному редакторі, уставити з файла, сфотографувати камерою.

У середовищі **Scratch** вибрати дії зі спрайтами (вилучити, дублювати, сховати, збільшити, зменшити, переглянути інформацію) можна в контекстному меню об'єкта або в **Рядку меню** програми.

Під час виконання програмного проекту **Scratch** для спрайтів може бути визначено деякі події, опрацювання яких приведе до змінення зна-



чень властивостей об'єктів. Команди, які можна застосувати для таких цілей, розміщено у групах Подія, Керування, Датчики вкладки Скрипти.

Дайте відповіді на запитання

- 1°. Що таке спрайти в середовищі Scratch?
- 2°. Які властивості мають програмні об'єкти в середовищі Scratch? Як їх можна переглянути?
- 3°. Якими способами можна змінити значення властивостей об'єктів у середовищі Scratch?
- 4°. Які властивості має об'єкт **Сцена**? Якими способами можна змінити тло **Сцени**?
- 5°. Якими способами можна створити новий спрайт у середовищі Scratch?
- 6°. Які дії можна виконувати над об'єктами в середовищі Scratch? Які засоби для цього можна використати?
- 7*. Як можна змінювати значення властивостей об'єктів під час виконання проекту в Scratch?
- 8*. Які програмні події існують у середовищі Scratch?
- 9°. Які команди можна використати для опрацювання подій у середовищі Scratch? Де вони розміщені?
- 10°. Чи можна в середовищі Scratch скласти кілька скриптів для одного об'єкта; різні скрипти для різних об'єктів; однакові скрипти для різних об'єктів?

Виконайте завдання

1. Відкрийте проект, указаний учителем (наприклад, Розділ 3\ Пункт 3.1\завдання 3.1.1.sb2). Запустіть проект на виконання. Опишіть, які об'єкти використано у проекті; які значення їх властивостей; які дії вони виконують; які події опрацьовуються у проекті. Змініть значення властивостей спрайта 1: образ – *soccerball* з **Бібліотеки**, початкове положення – *біля ноги Рудого kota*. Змініть значення властивостей **Сцени**: тло – *goal1*, віддзеркалити зліва направо. Змініть скрипт на такий:



Виконайте проект. Збережіть проект у вашій папці з тим самим іменем.

- 2°. Відкрийте проект, указаний учителем (наприклад, Розділ 3\ Пункт 3.1\завдання 3.1.2.sb2). Запустіть проект на виконання.

Опишіть, які об'єкти використано у проекті, які значення їх властивостей; які дії вони виконують; які події опрацьовуються у проекті. Перегляньте відеоролик, указаний учителем (наприклад, **Розділ 3\Пункт 3.1\завдання 3.1.2.avi**). Внесіть відповідні зміни у проект, щоб він дублював дії відеоролика. Збережіть проект у вашій папці з тим самим іменем.

3*. Перегляньте відеоролик, указаний учителем (наприклад, **Розділ 3\Пункт 3.1\завдання 3.1.3.avi**). Створіть у **Scratch** проект, який би дублював дії відеоролика. Збережіть проект у вашій папці у файлі з іменем завдання 3.1.3.



4*. Перегляньте відеоролик, указаний учителем (наприклад, **Розділ 3\Пункт 3.1\завдання 3.1.4.avi**). Створіть у **Scratch** проект, який би дублював дії відеоролика. Збережіть проект у вашій папці у файлі з іменем завдання 3.1.4.



3.2. ВКЛАДЕНІ ЦИКЛИ



1. Який фрагмент алгоритму називають циклом? Що таке тіло циклу?
2. Який загальний вигляд команди циклу з лічильником? Як вона виконується?
3. Який загальний вигляд команди циклу з передумовою? Як вона виконується?

ВКЛАДЕНІ ЦИКЛИ

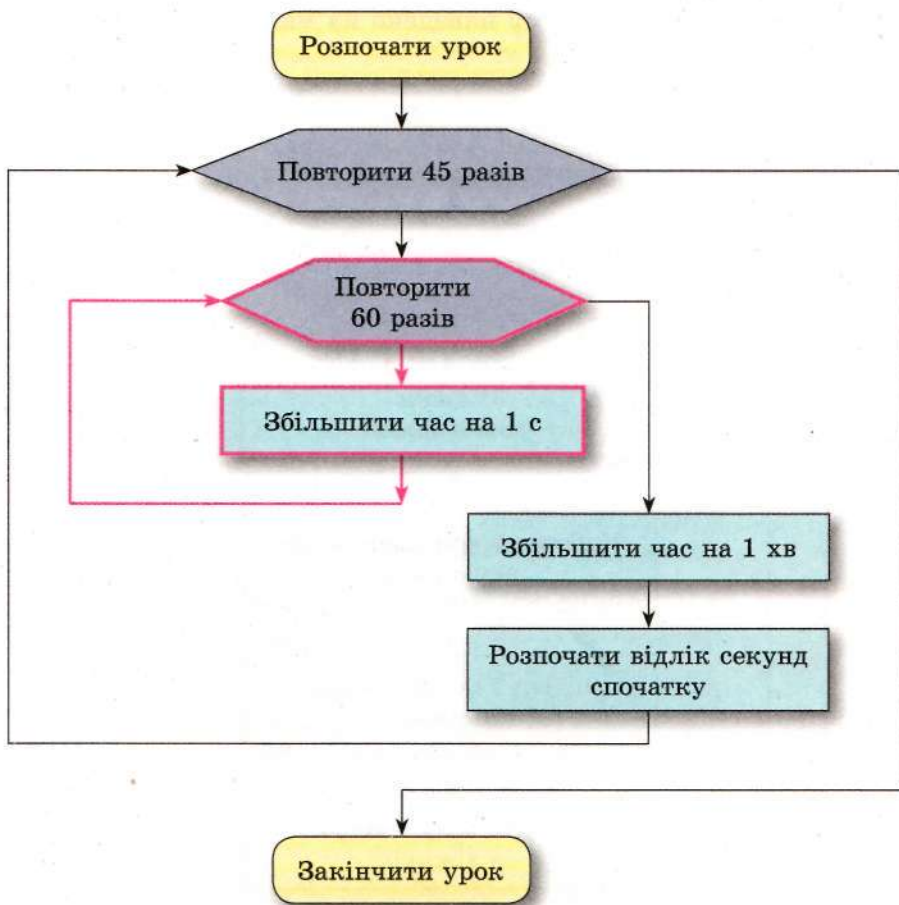
Нагадаємо, що цикл – це фрагмент алгоритму, команди якого можуть виконуватися більше ніж один раз. Ті команди, які можуть виконуватися більше ніж один раз, утворюють **тіло циклу**.

Якщо серед команд тіла циклу є інші цикли, то такий фрагмент алгоритму називають **вкладеним циклом**. Цикл, який міститься в тілі іншого циклу, називають **внутрішнім**. А цикл, у тілі якого розміщено інший цикл, називають **зовнішнім**.

Наведемо приклад вкладених циклів.

Урок триває 45 хвилин. Коли розпочинається урок, розпочинається перша хвилина і розпочинається відлік секунд цієї першої хвилини. Ви знаєте, що в кожній хвилині 60 секунд. Коли проходять 60 секунд першої хвилини, розпочинається друга хвилина і знову розпочинається відлік секунд від 1 до 60. І так повторюється 45 разів (хвилин). Блок-схему цих вкладених циклів подано на малюнку 3.9.

На наведеній на малюнку 3.9 блок-схемі внутрішній цикл **Повторити 60 разів** зображено рожевим кольором. Тіло цього циклу складається з однієї команди **Збільшити час на 1 с**. Цей цикл міститься в тілі іншого циклу – циклу **Повторити 45 разів**, який є зовнішнім.



Мал. 3.9. Приклад вкладених циклів

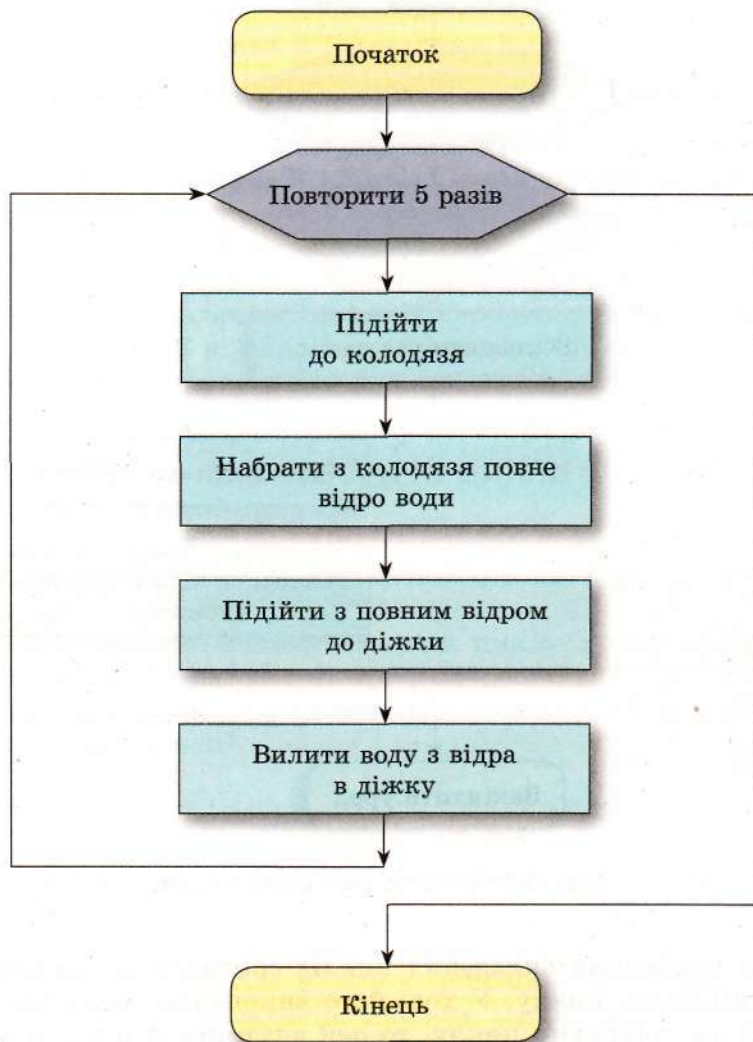
Під час виконання вкладених циклів спочатку починається виконання зовнішнього циклу. У ході його виконання, коли настає черга виконання внутрішнього циклу, то цей внутрішній цикл виконується повністю, після чого продовжується виконання зовнішнього циклу. І так відбувається під час кожного виконання тіла зовнішнього циклу.

Ще одним прикладом вкладених циклів може бути змінення місяців протягом кількох років поспіль. Розпочинається перший рік і протягом нього змінюються місяці від січня до грудня, після чого розпочинається другий рік і знову протягом нього змінюються місяці від січня до грудня. І так далі, поки не закінчатся вибрані кілька років.

ВКЛАДЕНІ ЦИКЛИ З ЛІЧИЛЬНИКОМ

У 5-му класі ви розглядали алгоритм розв’язування задачі наповнення водою порожньої діжки місткістю 50 л, беручи воду з колодязя і використовуючи відро місткістю 10 л.

Блок-схему цього алгоритму наведено на малюнку 3.10.

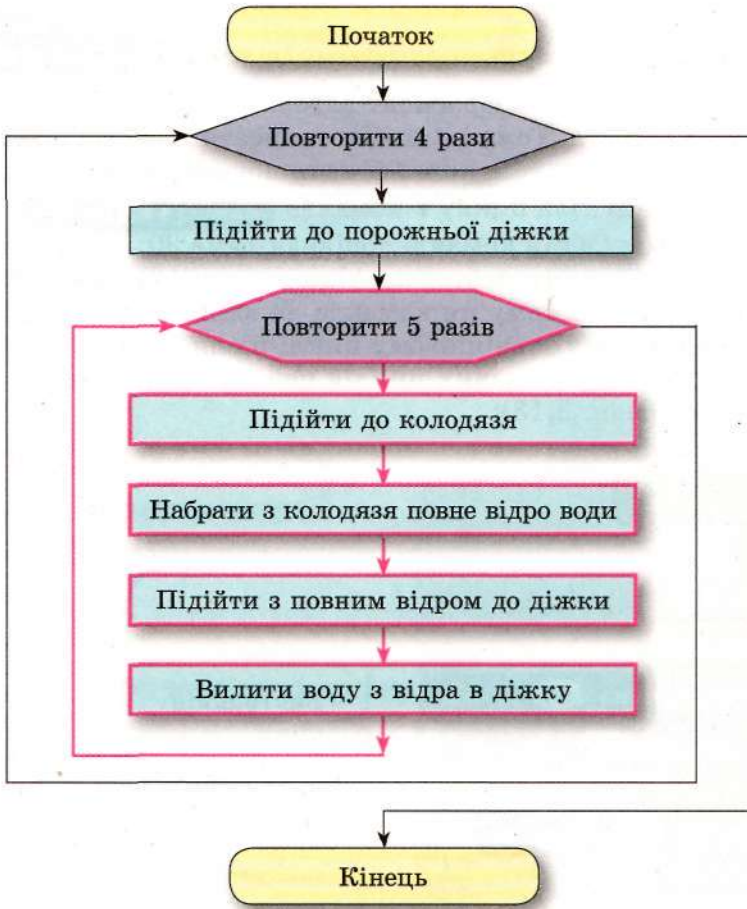


Мал. 3.10. Блок-схема алгоритму наповнення водою порожньої діжки місткістю 50 л, використовуючи відро місткістю 10 л

Якщо необхідно наповнити водою 4 такі діжки, то потрібно 4 рази виконати команди алгоритму, наведеного вище. Алгоритм розв'язування цієї задачі наведено на малюнку 3.11.

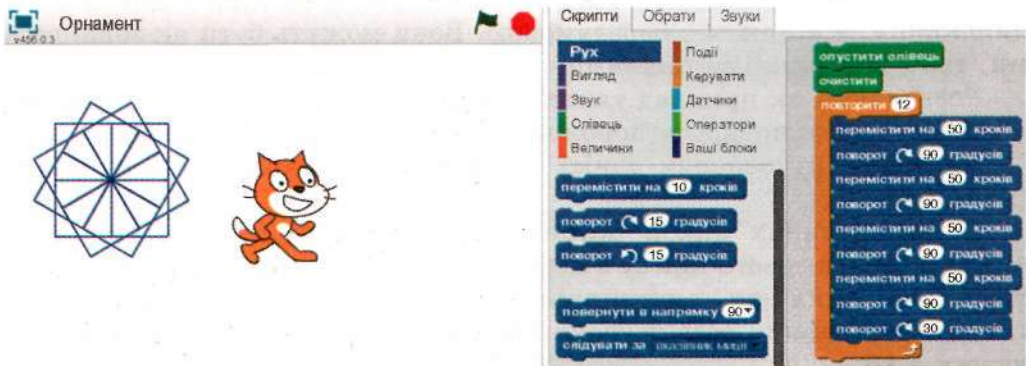
У цьому алгоритмі зовнішній цикл призначено для наповнення водою 4 порожніх діжок. Тіло цього циклу виконується 4 рази. А внутрішній цикл призначено для наповнення водою однієї порожньої діжки. Команди тіла цього циклу повторюються 5 разів.

Вкладені цикли з лічильниками можна використовувати і в **Scratch**.



Мал. 3.11. Блок-схема алгоритму наповнення водою 4 порожніх діжок місткістю 50 л кожна, використовуючи відро місткістю 10 л

Ви вже знаєте, що для малювання орнаменту з 12 квадратів виконавець **Рудий кіт** може виконати, наприклад, такий алгоритм із циклом з лічильником (мал. 3.12):



Мал. 3.12. Алгоритм малювання орнаменту з 12 квадратів

перемістити на 50 кроків
поворот 90 градусів

Серед команд тіла циклу цього алгоритму дві команди

повторюються 4 рази. Цей фрагмент алгоритму призначено для малювання одного квадрата. Тому їх можна записати в цьому циклі як внутрішній цикл, команди тіла якого будуть повторюватися 4 рази. Крім цих команд, у кінці тіла циклу є команда **поворот 30 градусів**. Нагадаємо, що кут повороту в цій команді дорівнює саме 30° , тому що цей кут, помножений на кількість повторень тіла циклу (12), дає 360° , що відповідає повному колу. Тому ця команда не ввійде у внутрішній цикл. Вона буде розміщена в зовнішньому циклі після внутрішнього циклу.

З використанням вкладених циклів цей алгоритм можна записати значно коротше (мал. 3.13):



Мал. 3.13. Алгоритм малювання орнаменту з використанням вкладених циклів з лічильником

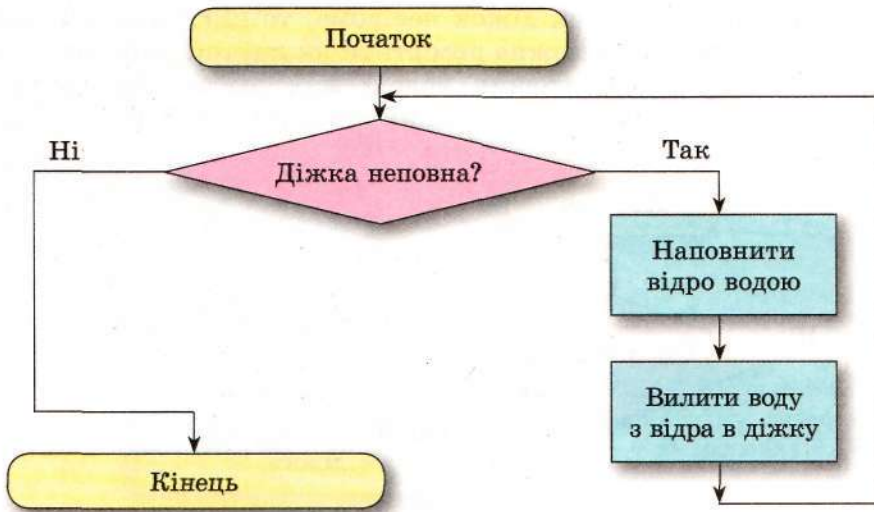
ЦИКЛИ З ПЕРЕДУМОВОЮ У ВКЛАДЕНИХ ЦИКЛАХ

У вкладених циклах можна використовувати не тільки цикли з лічильником, а й цикли з передумовою. Вони можуть бути як зовнішніми, так і внутрішніми циклами.

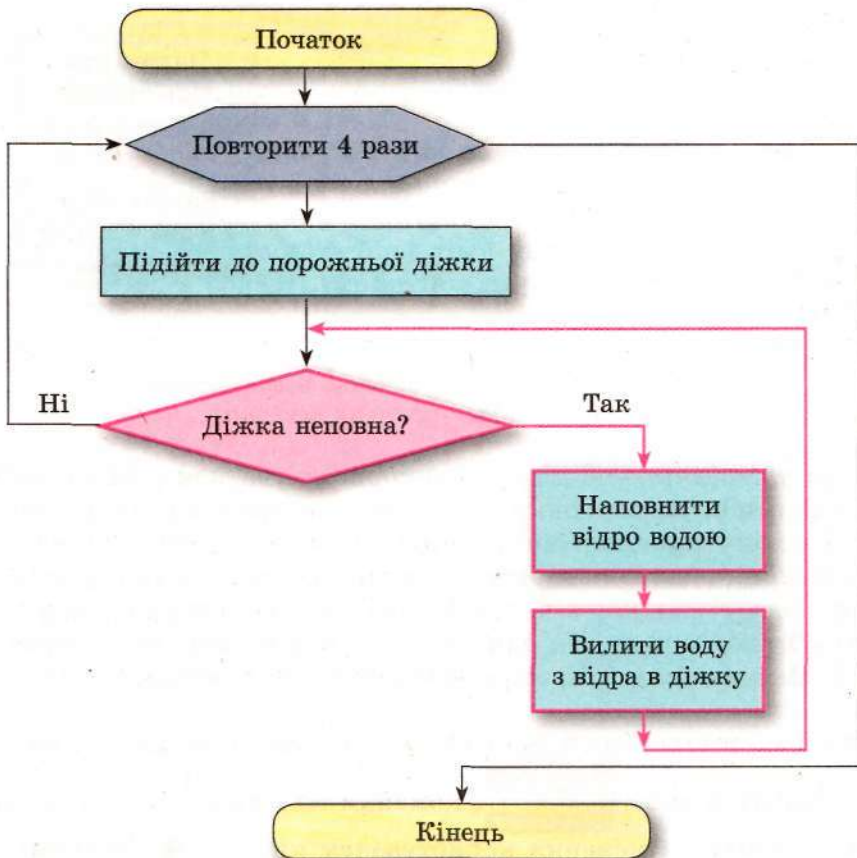
Розглянемо як приклад таких вкладених циклів алгоритм для наповнення водою чотирьох порожніх діжок невідомої місткості.

Алгоритм наповнення однієї такої діжки ви знаєте з 5-го класу (мал. 3.14).

Оскільки таких діжок потрібно наповнити 4, то слід виконати 4 рази команди цього циклу з передумовою. А для цього можна помістити цей цикл як внутрішній цикл до циклу з лічильником. І тоді алгоритм для розв'язування цієї задачі матиме вигляд як на малюнку 3.15.

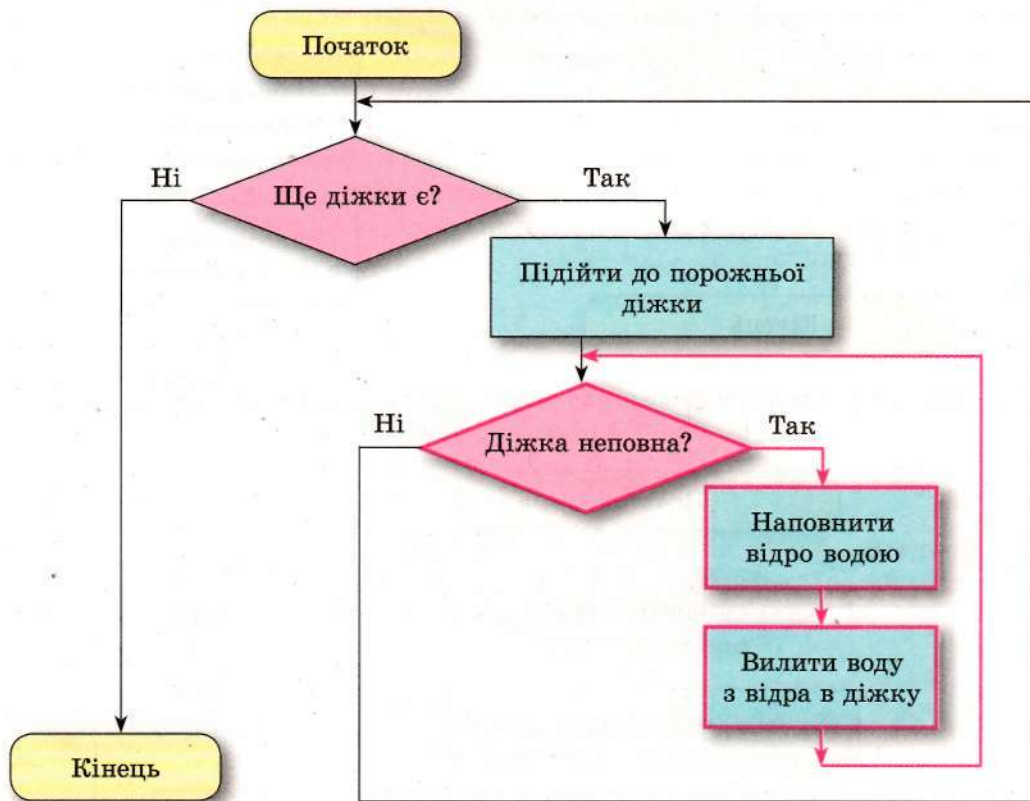


Мал. 3.14. Алгоритм наповнення водою однієї діжки невідомої місткості



Мал. 3.15. Алгоритм наповнення водою чотирьох діжок невідомої місткості

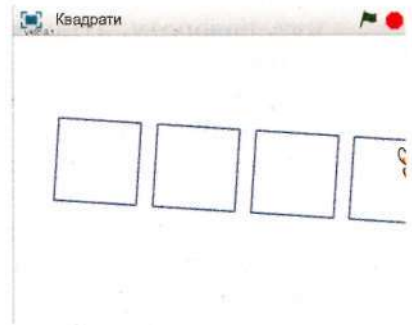
Якщо ж кількість таких діжок невідома, то цикл для наповнення водою однієї такої діжки можна помістити як внутрішній цикл у цикл з передумовою для наповнення водою всіх діжок. Алгоритм для розв'язування цієї задачі матиме такий вигляд (мал. 3.16):



Мал. 3.16. Алгоритм наповнення водою діжок, місткість і кількість яких невідомі

Цикли з передумовою можуть використовуватися у вкладених циклах і в **Scratch**. На малюнку 3.17 наведено приклад такого використання. У цьому прикладі внутрішній цикл є циклом з лічильником для малювання одного квадрата з довжиною сторони 100 кроків, а зовнішній цикл є циклом з передумовою для малювання квадратів на відстані 20 кроків один від одного, поки виконавець не доторкається до межі. На малюнку 3.18 наведено результат виконання цього алгоритму.

Зазначимо також, що в **Scratch** у вкладених циклах можна також використовувати цикл  **завжди**, команди тіла якого будуть виконуватися до моменту натиснення користувачем кнопки  **Зупинити все**, що закінчує виконання всього проекту.



Мал. 3.17. Вкладені цикли з використанням циклів з передумовою в Scratch

Мал. 3.18. Результат виконання алгоритму, наведеного на малюнку 3.17

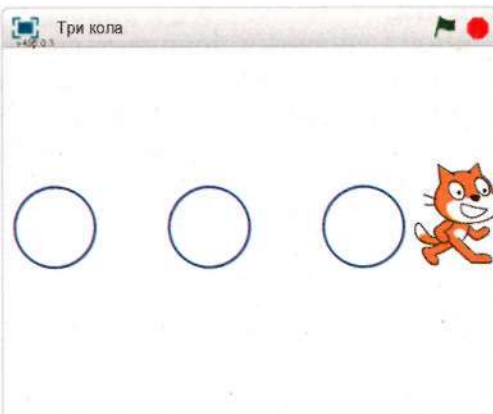
Працюємо з комп'ютером

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

Задача. Скласти проект для малювання трьох кіл, розміщених як це наведено на малюнку 3.19.

Щоб виконавець намалював три кола, потрібно відомий вам з 5-го класу цикл для малювання кола повторити 3 рази.

1. Відкрийте середовище Scratch.
2. Розмістіть **Рудого kota** біля лівої межі **Сцени**.
3. Розмістіть в області скриптів алгоритм, наведений на малюнку 3.20.
4. Збережіть проект у вашій папці у файлі з іменем **вправа 3.2**.
5. Виконайте проект.



Мал. 3.19. Три кола



Мал. 3.20. Алгоритм малювання трьох кіл

6. Змініть у внутрішньому циклі кількість повторень і відповідно кут повороту. (Нагадуємо, що добуток міри кута повороту на кількість повторень тіла циклу має дорівнювати 360).
7. Збережіть нову версію проекту.
8. Виконайте нову версію проекту.
9. Порівняйте результат виконання останньої версії проекту з результатами виконання попередньої версії.
10. Змініть кількість кроків переміщення у внутрішньому циклі для малювання кола та кількість кроків переміщення у зовнішньому циклі.
11. Збережіть нову версію проекту.
12. Виконайте нову версію проекту.
13. Порівняйте результат виконання останньої версії проекту з результатами виконання попередніх версій.
14. Змініть алгоритм так, щоб кожне наступне коло було іншого кольору.
15. Збережіть нову версію проекту.
16. Виконайте нову версію проекту.
17. Порівняйте результат виконання останньої версії проекту з результатами виконання попередніх версій.
18. Змініть зовнішній цикл на цикл **повторити поки не доторкається вказівник миші**.
19. Збережіть нову версію проекту.
20. Виконайте нову версію проекту.
21. Закрийте середовище Scratch.

Найважливіше в цьому пункті

Якщо серед команд тіла циклу є інші цикли, то такі фрагменти алгоритмів називають **вкладеними циклами**. Цикл, який міститься в тілі іншого циклу, називають **внутрішнім**. А цикл, у тілі якого розміщено інший цикл, називають **зовнішнім**.





Під час виконання вкладених циклів спочатку починається виконання зовнішнього циклу. У ході його виконання, коли настає черга виконання внутрішнього циклу, то цей внутрішній цикл виконується повністю, після чого продовжується виконання зовнішнього циклу. І так відбувається під час кожного виконання тіла зовнішнього циклу.

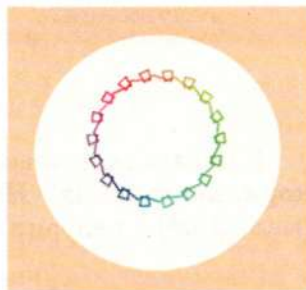
Дайте відповіді на запитання

- 1°. Який фрагмент алгоритму називають вкладеним циклом?
- 2°. Який цикл називають зовнішнім?
- 3°. Який цикл називають внутрішнім?
- 4°. Поясніть, як виконуються вкладені цикли.



Виконайте завдання

- 1°. Складіть блок-схему змінення місяців протягом чотирьох років, використовуючи вкладені цикли.
- 2°. Складіть проект для малювання трьох квадратів зі сторонами 50 кроків, які розміщені один від одного на відстані 30 кроків. Збережіть проект у вашій папці у файлі з іменем **завдання 3.2.2**.
-  3°. Складіть проект для малювання п'яти квадратів зі сторонами 40 кроків різного кольору, які розміщені один від одного на відстані 10 кроків. Збережіть проект у вашій папці у файлі з іменем **завдання 3.2.3**.
- 4°. Складіть проект для малювання трьох прямокутників зі сторонами 60 кроків і 30 кроків. Усі сторони одного прямокутника мають бути одного кольору, а сторони різних прямокутників – різного кольору. Збережіть проект у вашій папці у файлі з іменем **завдання 3.2.4**.
- 5°. Складіть проект для малювання трьох однакових орнаментів, кожен з яких складається з 10 квадратів. Збережіть проект у вашій папці у файлі з іменем **завдання 3.2.5**.
-  6°. Складіть проект для малювання п'яти однакових орнаментів. Вид орнаменту та їх взаємне розміщення встановіть самостійно. Збережіть проект у вашій папці у файлі з іменем **завдання 3.2.6**.
-  7°. Складіть проект для малювання різнокольорових кіл, розміщених на відстані 20 кроків одне від одного, поки виконавець не доторкнеться до межі **Сцени**. Збережіть проект у вашій папці у файлі з іменем **завдання 3.2.7**.
- 8*. Складіть проект для малювання орнаменту, наведеного на малюнку 3.21. Збережіть проект у вашій папці у файлі з іменем **завдання 3.2.8**.
-  9*. Складіть проект для малювання кілець на прапорі Олімпійських ігор. Збережіть проект у вашій папці у файлі з іменем **завдання 3.2.9**.



Мал. 3.21. Орнамент

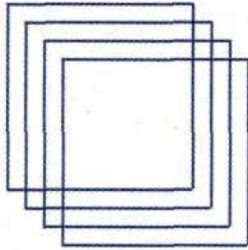
ПРАКТИЧНА РОБОТА № 5

«Проекти з вкладеними циклами»

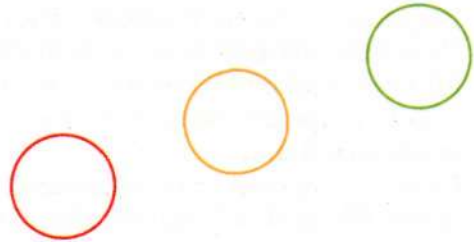
Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

1. Відкрийте середовище **Scratch**.
2. Складіть проект для малювання чотирьох квадратів, зображених на малюнку 3.22.

3. Збережіть проект у вашій папці у файлі з іменем **практична 5_1**.
4. Складіть проект для малювання кіл різного кольору (мал. 3.23), поки виконавець не доторкнеться до межі **Сцени**.
5. Збережіть проект у вашій папці у файлі з іменем **практична 5_2**.
6. Закрийте середовище **Scratch**.



Мал. 3.22. Чотири квадрати



Мал. 3.23. Кола різних кольорів

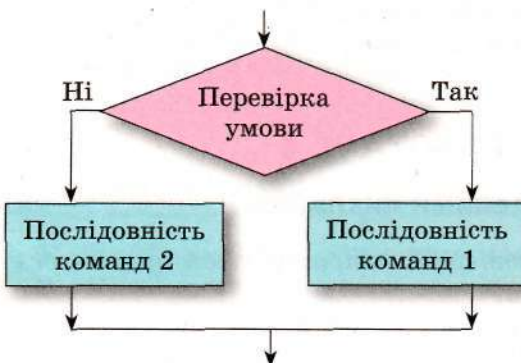
3.3. ВКЛАДЕНІ РОЗГАЛУЖЕННЯ



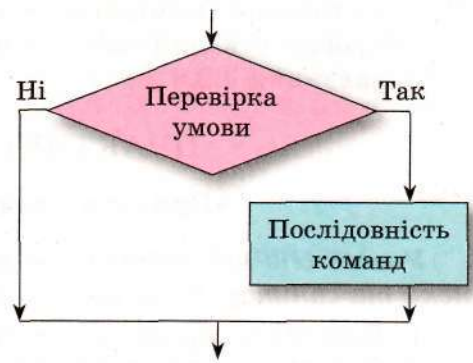
1. Який фрагмент алгоритму називають повним розгалуженням?
2. Який фрагмент алгоритму називають неповним розгалуженням?
3. Під час розв'язування яких завдань ви використовували розгалуження?

ВКЛАДЕНІ РОЗГАЛУЖЕННЯ

У попередніх класах ви вже ознайомилися з розгалуженнями та використовували їх. Нагадаємо, що розгалуження може бути повним (мал. 3.24) і неповним (мал. 3.25).



Мал. 3.24. Повне розгалуження



Мал. 3.25. Неповне розгалуження

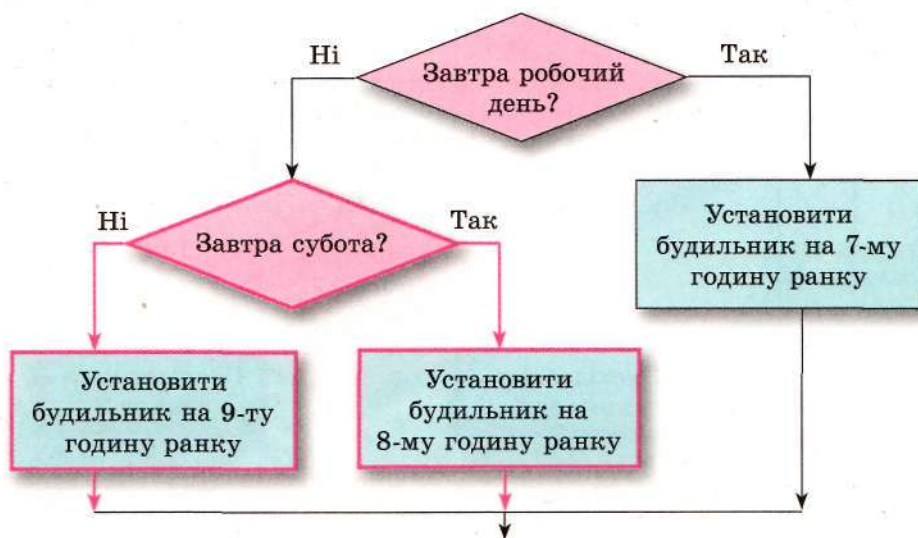


Ви розглядали алгоритми, у яких було кілька розгалужень, що виконувалися по черзі, одне за одним. У них кожне наступне розгалуження виконувалося тоді, коли виконання попереднього розгалуження вже закінчилося.

Але бувають й інші випадки, інші життєві ситуації. У них наступне розгалуження виконується, коли виконання попереднього розгалуження ще не закінчено.

Наприклад, вам потрібно встановити будильник на завтра. Якщо завтра робочий день, то ви повинні встати о 7-й годині ранку, щоб іти до школи. Якщо завтра субота, то ви повинні встати о 8-й годині ранку, щоб їхати на заняття гуртка. Якщо завтра неділя, то ви встаете о 9-й годині ранку.

Блок-схема алгоритму встановлення будильника має такий вигляд (мал. 3.26):



Мал. 3.26. Алгоритм встановлення будильника

У наведеному на малюнку 3.26 алгоритмі друге розгалуження з умовою *Завтра субота?* міститься всередині першого розгалуження з умовою *Завтра робочий день?*.

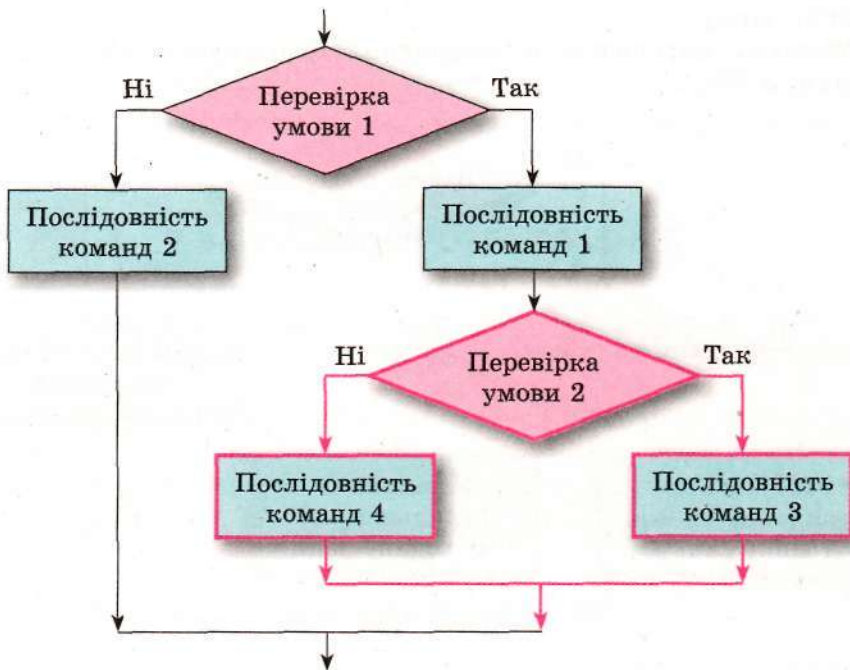
Такий фрагмент алгоритму називають **вкладеним розгалуженням**.

Вкладені розгалуження – це фрагмент алгоритму, у якому одне розгалуження міститься всередині іншого розгалуження.

Розглянемо виконання наведеного на малюнку 3.26 фрагмента алгоритму. Спочатку перевіряється умова *Завтра робочий день?*. Якщо результат перевірки цієї умови **Так**, то виконується команда *Установити будильник на 7-му годину ранку* і на цьому виконання всього цього фрагмента алгоритму закінчується. Якщо результат перевірки умови *Завтра робочий день?* – **Ні**, то перевіряється умова *Завтра субота?*. Якщо результат перевірки цієї умови **Так**, то виконується команда

Установити будильник на 8-му годину ранку і на цьому виконання всього цього фрагмента алгоритму закінчується, а якщо результат перевірки цієї умови **Ні**, то виконується команда *Установити будильник на 9-ту годину ранку* і виконання всього цього фрагмента алгоритму закінчується.

У наведеному на малюнку 3.26 фрагменті алгоритму внутрішнє розгалуження виконується, якщо результат перевірки умови зовнішнього розгалуження **Ні**. Аналогічно можливі такі вкладені розгалуження, у яких внутрішнє розгалуження виконується, якщо результат перевірки умови зовнішнього розгалуження **Так**. Приклад такого фрагмента алгоритму наведено на малюнку 3.27.



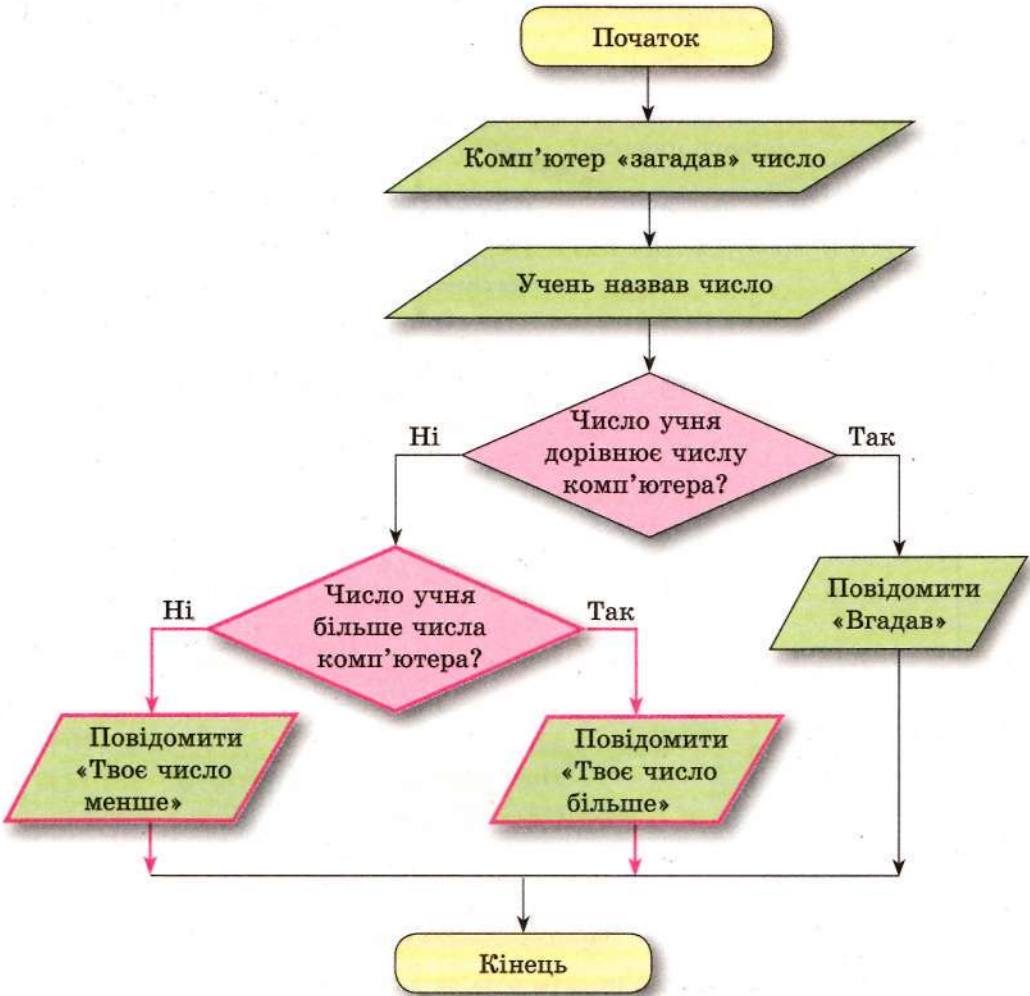
Мал. 3.27. Вкладені розгалуження

Розглянемо виконання наведеного на малюнку 3.27 фрагмента алгоритму. Спочатку перевіряється *умова 1*. Якщо результат перевірки цієї умови **Ні**, то виконується *Послідовність команд 2* і виконання всього цього фрагмента алгоритму закінчується. Якщо результат перевірки *умови 1* – **Так**, то виконується *Послідовність команд 1* і потім перевіряється *умова 2*. Якщо результат перевірки *умови 2* – **Так**, то виконується *Послідовність команд 3* і виконання всього цього фрагмента алгоритму закінчується. А якщо результат перевірки *умови 2* – **Ні**, то виконується *Послідовність команд 4* і виконання всього цього фрагмента алгоритму закінчується.

Наведемо приклад алгоритму з вкладеними розгалуженнями для розв’язування нижченаведеної задачі-гри.



Задача-гра 1. Комп'ютер «загадав» натуральне число від 1 до 20. Учень намагається його вгадати. А комп'ютер відповідає, чи вгадав учень загадане число, чи учень назвав число, більше від загаданого, чи учень назвав число, менше від загаданого (мал. 3.28).

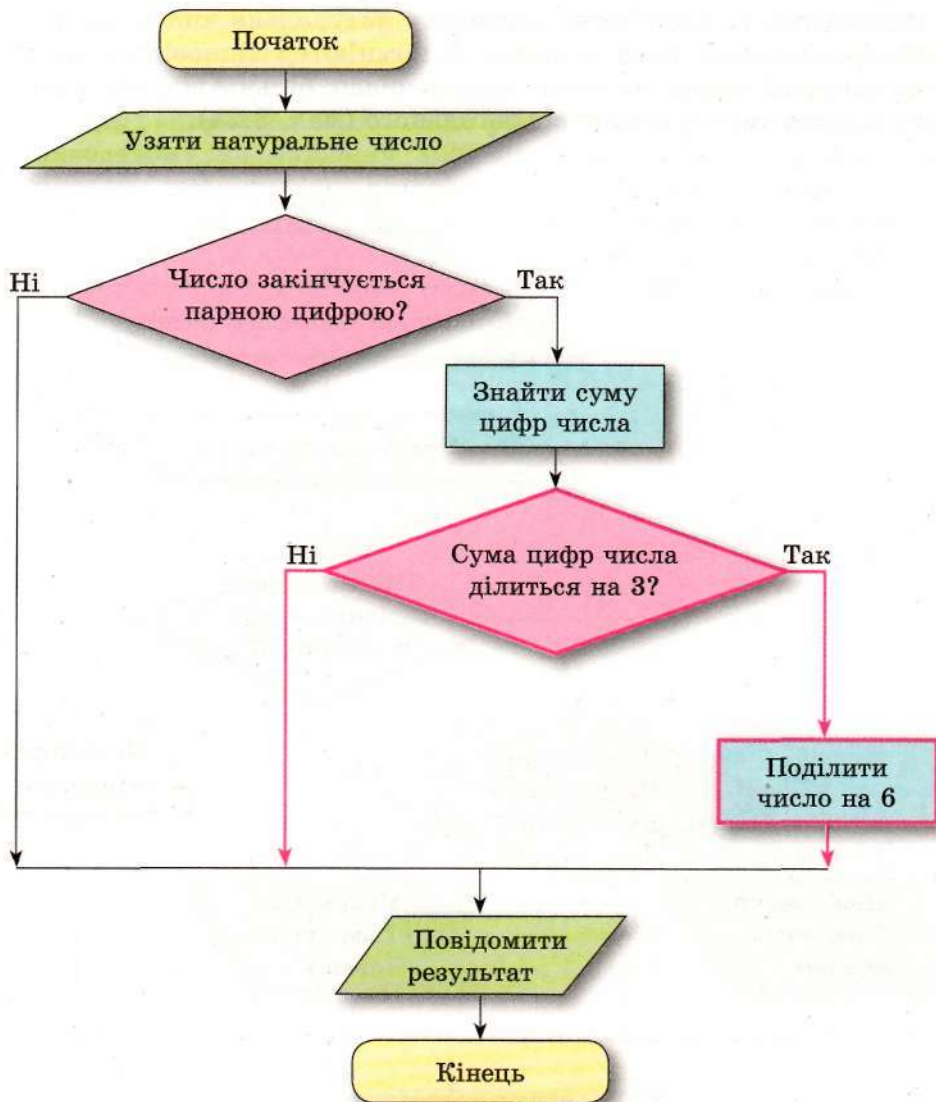


Мал. 3.28. Блок-схема алгоритму задачі-гри 1

У наведених вище прикладах і зовнішнє, і внутрішнє розгалуження є повним. Але серед вкладених розгалужень можуть бути і неповні розгалуження. Наведемо приклад алгоритму з такими вкладеними розгалуженнями.

Задача 2. Нехай задано натуральне число. Якщо воно ділиться на 6, то потрібно його розділити на 6.

Щоб число ділилося на 6, воно повинно ділитися на 2 і на 3. А ознаки подільності на 2 і на 3 ви знаєте. Тоді маємо такий алгоритм (мал. 3.29).



Мал. 3.29. Алгоритм розв'язування задачі 2

ВКЛАДЕНІ РОЗГАЛУЖЕННЯ В SCRATCH

Вкладені розгалуження можна використовувати і в **Scratch**.

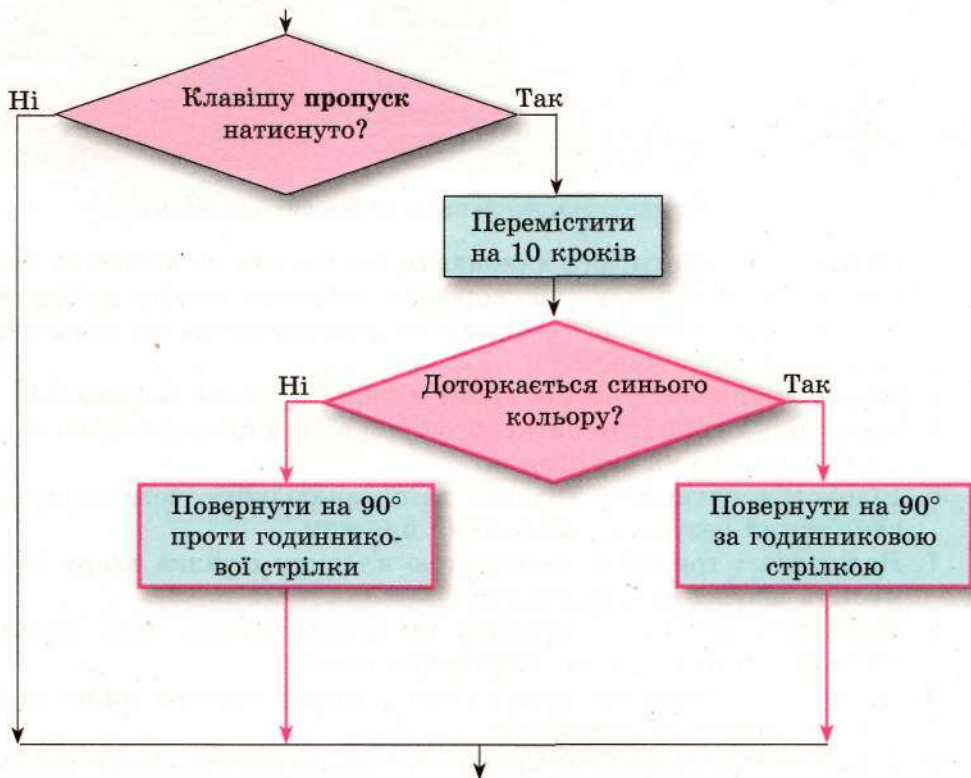
Наведемо приклад фрагмента проекту в **Scratch** з використанням вкладених розгалужень (мал. 3.30).

У наведеному фрагменті **якщо клавішу пропуск натиснуто**, виконавець переміщується на 10 кроків уперед. Якщо після цього переміщення він доторкається синього кольору, то він повертається на 90° за годинниковою стрілкою, а інакше (не доторкається синього кольору) він повертається на 90° проти годинникової стрілки.



Мал. 3.30. Приклад вкладеного розгалуження в Scratch

Блок-схема цього фрагмента має такий вигляд (мал. 3.31):



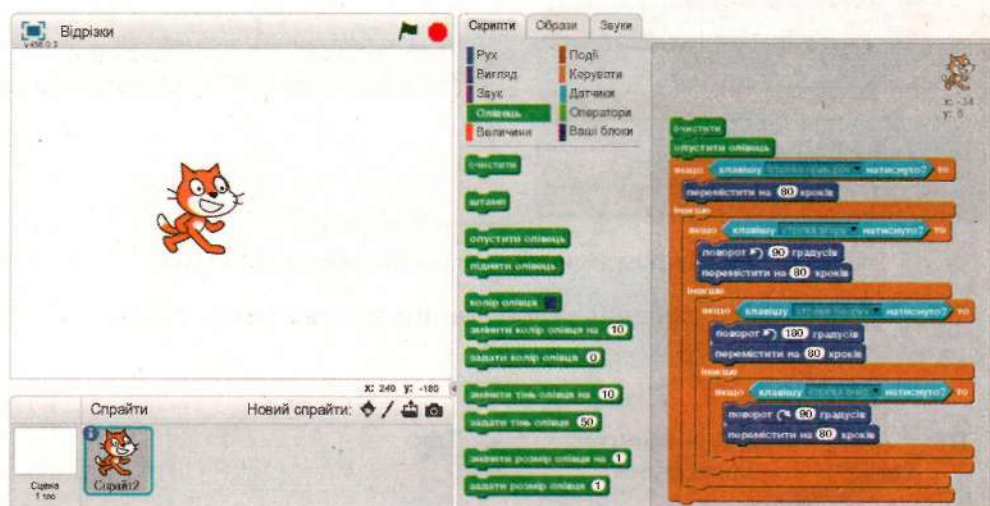
Мал. 3.31. Блок-схема фрагмента з вкладеними розгалуженнями в Scratch

Працюємо з комп'ютером

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

Задача. Скласти проект для малювання відрізка завдовжки 80 кроків у відповідному напрямку від положення виконавця залежно від того, яку клавішу-стрілку натиснуто.

1. Відкрийте середовище **Scratch**.
2. Розмістіть **Рудого kota** на **Сцені** як це показано на малюнку 3.32.



Мал. 3.32. Проект з використанням вкладених розгалужень

3. Складіть алгоритм для виконавця відповідно до малюнка 3.32. (*Зверніть увагу:* для прискорення введення можна дублювати дві команди з першого вкладеного розгалуження та редагувати копії потрібним чином.)
4. Збережіть проект у вашій папці у файлі з іменем **вправа 3.3**.
5. Натисніть і тримайте натиснутою клавішу **стрілка вправо** та запустіть проект на виконання.
6. Відпустіть натиснуту клавішу та проаналізуйте, чому отримали саме такий результат виконання проекту.
7. Натисніть і тримайте натиснутою клавішу **стрілка вгору** та запустіть проект на виконання.
8. Відпустіть натиснуту клавішу та проаналізуйте, чому отримали саме такий результат виконання проекту.
9. Натисніть і тримайте натиснутою клавішу **стрілка вліво** та запустіть проект на виконання.
10. Відпустіть натиснуту клавішу та проаналізуйте, чому отримали саме такий результат виконання проекту.
11. Натисніть і тримайте натиснутою клавішу **стрілка вниз** та запустіть проект на виконання.
12. Відпустіть натиснуту клавішу та проаналізуйте, чому отримали саме такий результат виконання проекту.
13. Натисніть і тримайте натиснутою будь-яку іншу клавішу та запустіть проект на виконання.
14. Відпустіть натиснуту клавішу та проаналізуйте, чому отримали саме такий результат виконання проекту.
15. Закрийте середовище **Scratch**.



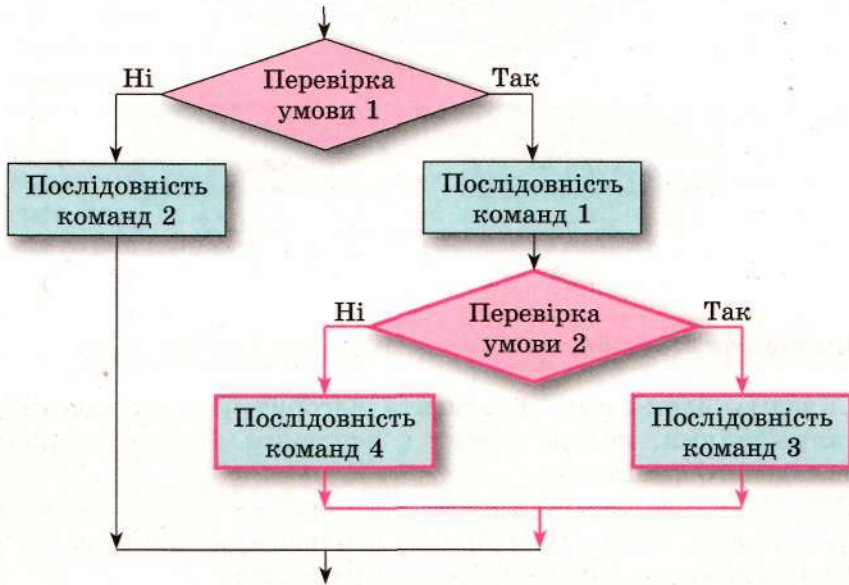
Найважливіше в цьому пункті

Вкладені розгалуження – це фрагмент алгоритму, у якому одне розгалуження міститься всередині іншого розгалуження.

І зовнішні, і внутрішні розгалуження можуть бути як повними, так і неповними.

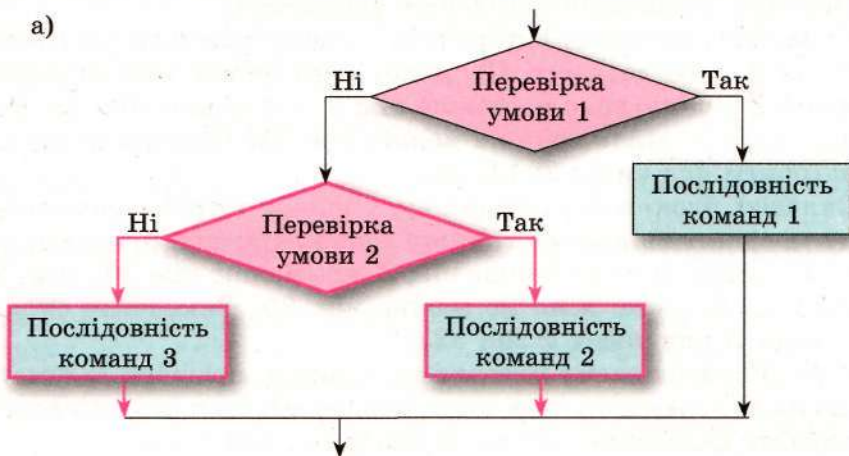
Дайте відповіді на запитання

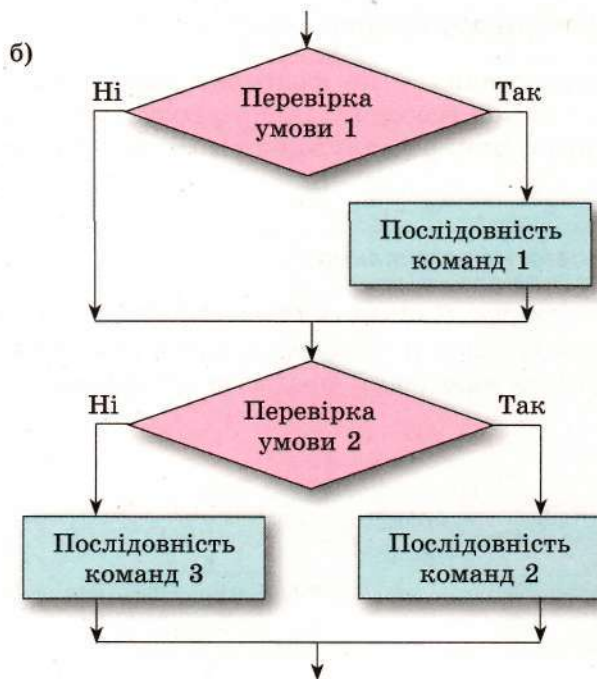
- 1°. Які розгалуження називають вкладеними?
- 2°. Поясніть, як виконується таке вкладене розгалуження?
- 3°. Як виконується наведений фрагмент алгоритму?






4°. Чим відрізняється виконання наведених фрагментів алгоритмів?

а)






Виконайте завдання

- 1°. Складіть блок-схему фрагмента алгоритму з вкладеними розгалуженнями, кожне з яких є неповним. Поясніть виконання складеного фрагмента.
- 2°. Складіть блок-схему фрагмента алгоритму з вкладеними розгалуженнями, з яких внутрішнє є неповним, а зовнішнє – повним. Поясніть виконання складеного фрагмента.
-  3°. Складіть блок-схему фрагмента алгоритму з вкладеними розгалуженнями, з яких зовнішнє є неповним, а внутрішнє – повним. Поясніть виконання складеного фрагмента.
- 4°. Складіть блок-схему алгоритму, у якому потрібно задумати число та додати до нього 10, якщо воно менше ніж 3; додати до нього 20, якщо воно не менше ніж 3, але менше ніж 25; відняти від нього 1, якщо воно не менше ніж 25. Виконайте складений алгоритм для чисел 2; 14; 30.
-  5°. Складіть блок-схему алгоритму, у якому потрібно задумати число та відняти від нього 1, якщо воно менше ніж 8; додати до нього 15, якщо воно не менше ніж 8, але менше ніж 30; помножити його на 5, якщо воно не менше ніж 30. Виконайте складений алгоритм для чисел 4; 20; 52.
-  6°. Складіть блок-схему алгоритму, у якому потрібно задумати два числа та визначити, яке з них більше або чи є вони рівними. Виконайте складений алгоритм для трьох пар чисел.



- 7°. Складіть проект, у якому виконавець намалює відрізок завдовжки 50 кроків синім кольором, якщо натиснуто клавішу **стрілка вправо**, або намалює відрізок завдовжки 100 кроків червоним кольором, якщо натиснуто клавішу **стрілка вниз**. Збережіть проект у вашій папці у файлі з іменем завдання 3.3.7.
- 8°. Складіть проект, у якому виконавець змінить свій образ, якщо натиснуто клавішу **стрілка вліво**, збільшить свій розмір на 20, якщо натиснуто клавішу **стрілка вправо**, або зіграє звук на барабані, якщо натиснуто клавішу **пропуск**. Збережіть проект у вашій папці у файлі з іменем завдання 3.3.8.
- 9°. Складіть проект, у якому зміниться тло, якщо натиснуто клавішу **пропуск**, виконавець змінить образ, якщо натиснуто клавішу **стрілка вгору**, прозвучить нота ля, якщо натиснуто клавішу **стрілка вправо**. Збережіть проект у вашій папці у файл з іменем завдання 3.3.9.
-  10°. Складіть проект, у якому виконавець зменшить свій образ на 10, якщо до виконавця **доторкається вказівник миші**, переміститься вправо на 100, якщо натиснуто клавішу **стрілка вправо**, або зіграє ноту до, якщо натиснуто клавішу **пропуск**. Збережіть проект у вашій папці у файлі з іменем завдання 3.3.10.

3.4. ВКЛАДЕНІ ЦИКЛИ ТА РОЗГАЛУЖЕННЯ



1. Що таке вкладені цикли?
2. Що таке вкладені розгалуження?
3. Які команди можна використати для створення проектів з розгалуженнями та циклами в Scratch?

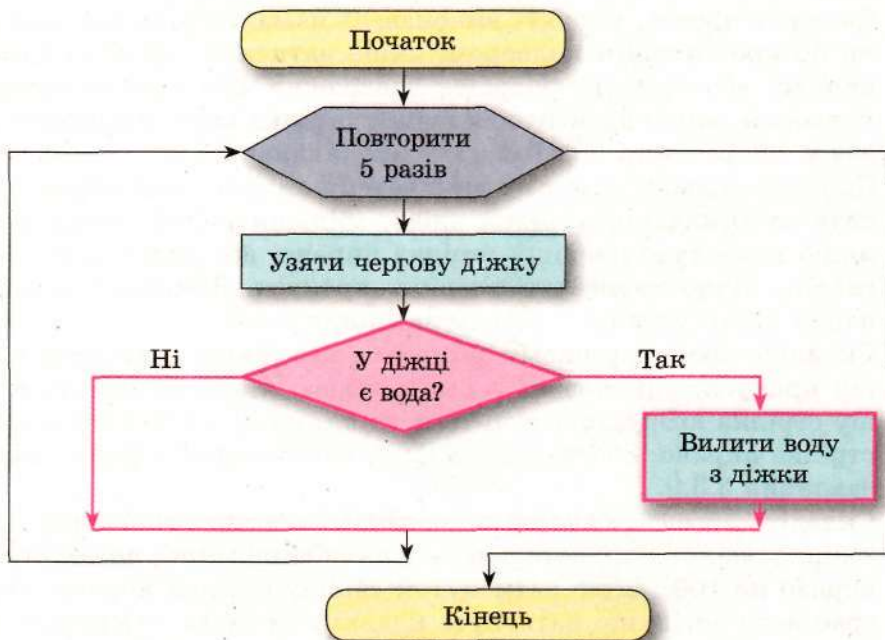
Ви вже ознайомилися з вкладеними циклами та вкладеними розгалуженнями і використовували їх для складання різноманітних алгоритмів. В алгоритмах можна також використовувати цикл у розгалуженні та розгалуження в циклі.

РОЗГАЛУЖЕННЯ, ВКЛАДЕНЕ В ЦИКЛ

Розглянемо приклади алгоритмів з використанням розгалуження в циклі.

Задача 1. Є 5 діжок, у яких може бути вода. Вилити воду з усіх таких діжок.

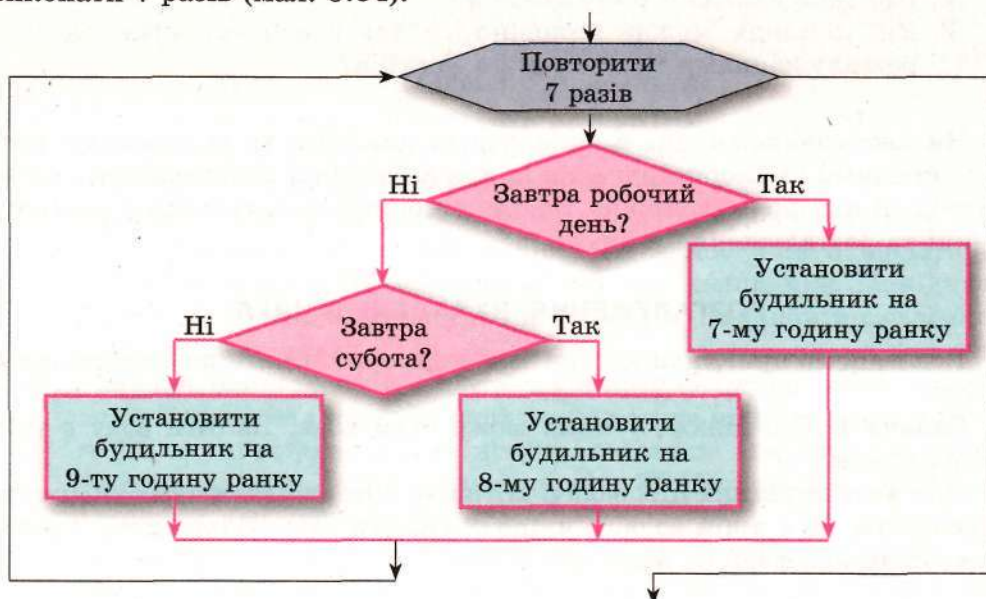
Для розв'язування цієї задачі потрібно 5 разів узяти чергову діжку, перевірити, чи є в ній вода, і, якщо є, вилити воду. Блок-схему такого алгоритму наведено на малюнку 3.33.



Мал. 3.33. Блок-схема алгоритму з розгалуженням у циклі

У задачі 1 у цикл з лічильником вкладено неповне розгалуження. У циклі можуть міститися також вкладені повні розгалуження.

У попередньому пункті ви розглядали алгоритм встановлення будильника. Але встановлювати будильник потрібно не один раз, а кожного дня тижня. Тому вищевказаний фрагмент алгоритму потрібно виконати 7 разів (мал. 3.34):

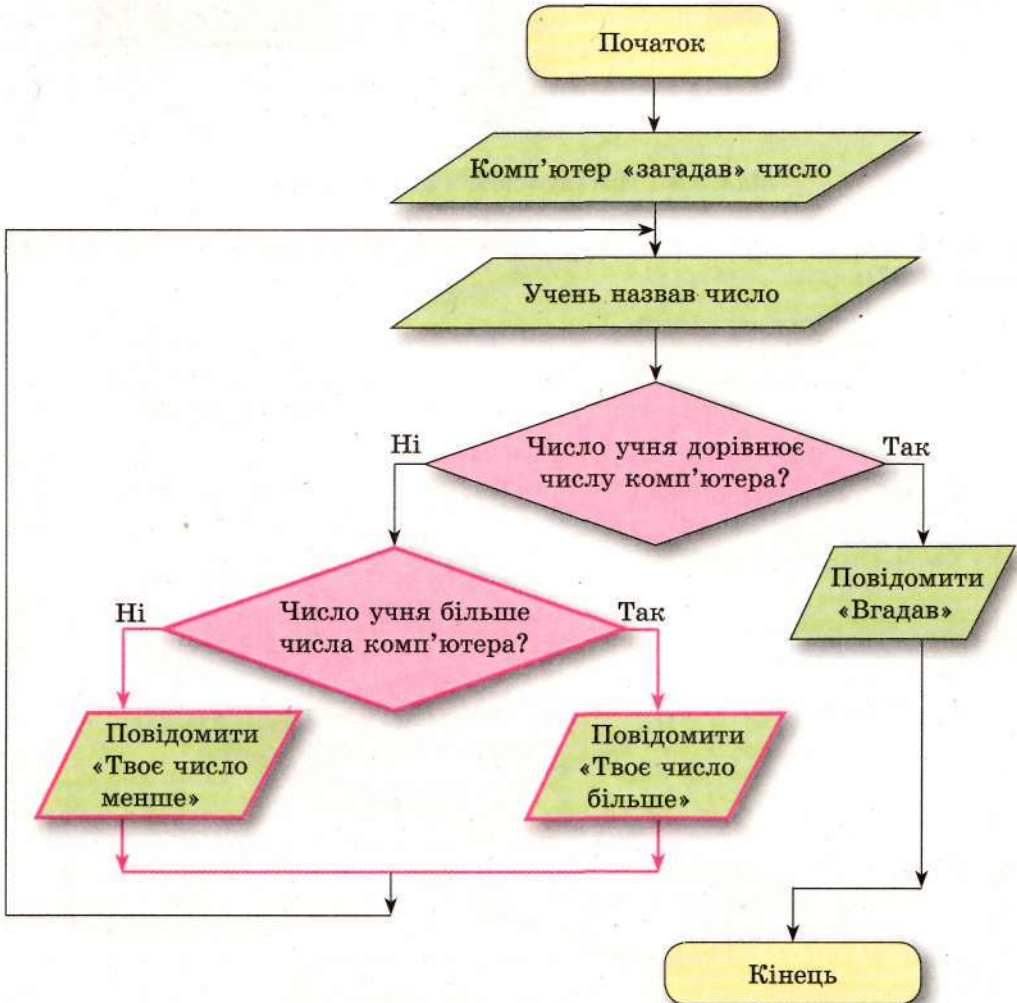


Мал. 3.34. Вкладені розгалуження в циклі



Розгалуження, у тому числі вкладені, можуть входити не тільки до циклу з лічильником, а й до циклу з передумовою.

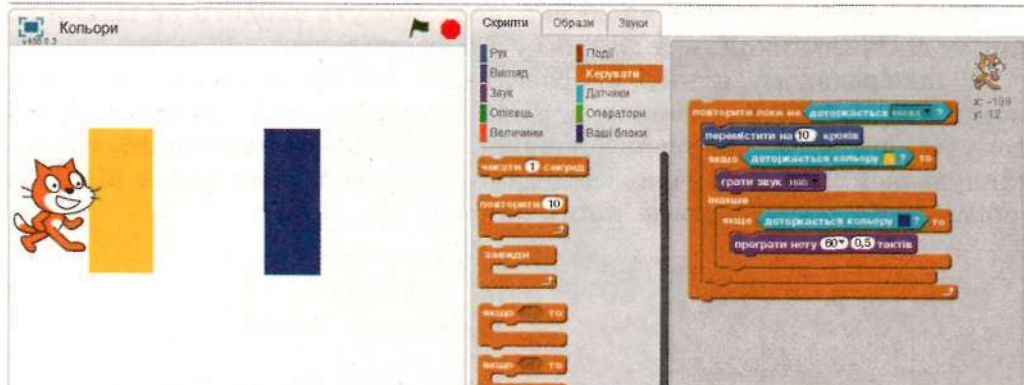
У попередньому пункті ви розглядали алгоритм задачі-гри, коли комп'ютер «загадав» число, а учень намагався його відгадати. Але в тому алгоритмі, якщо учень з першої спроби не відгадав число, гра закінчувалася. Якщо ж учень матиме багато спроб, поки він не відгадає число, то алгоритм матиме такий вигляд (мал. 3.35):



Мал. 3.35. Розгалуження в циклі з передумовою

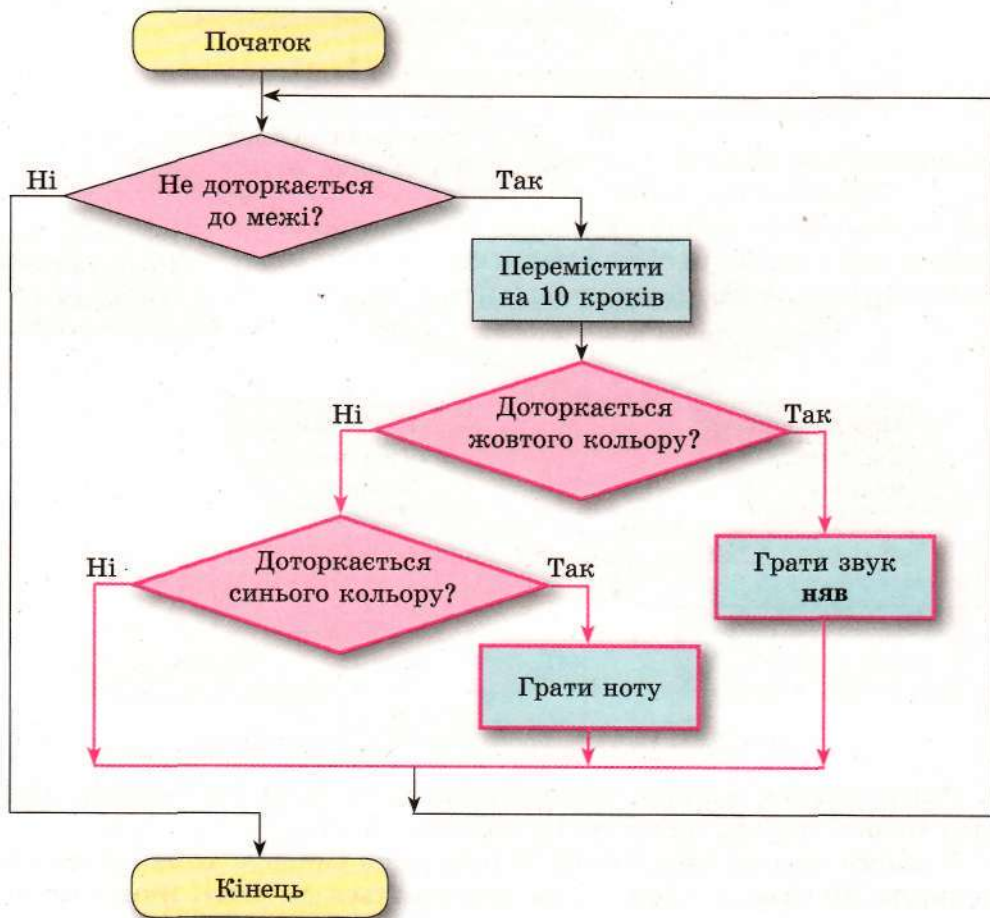
Розгалуження в циклі можна використовувати і в **Scratch**. Приклад такого проекту наведено на малюнку 3.36.

У цьому проекті виконавець кілька разів виконує команду переміщення на 10 кроків, доки він не доторкається до межі. Якщо під час руху він доторкається жовтого кольору, він грає звук няв. А якщо він під час руху доторкається синього кольору, то грає певну ноту.



Мал. 3.36. Розгалуження в циклі в Scratch

Блок-схема алгоритму наведеного проекту має такий вигляд (мал. 3.37):



Мал. 3.37. Блок-схема алгоритму проекту



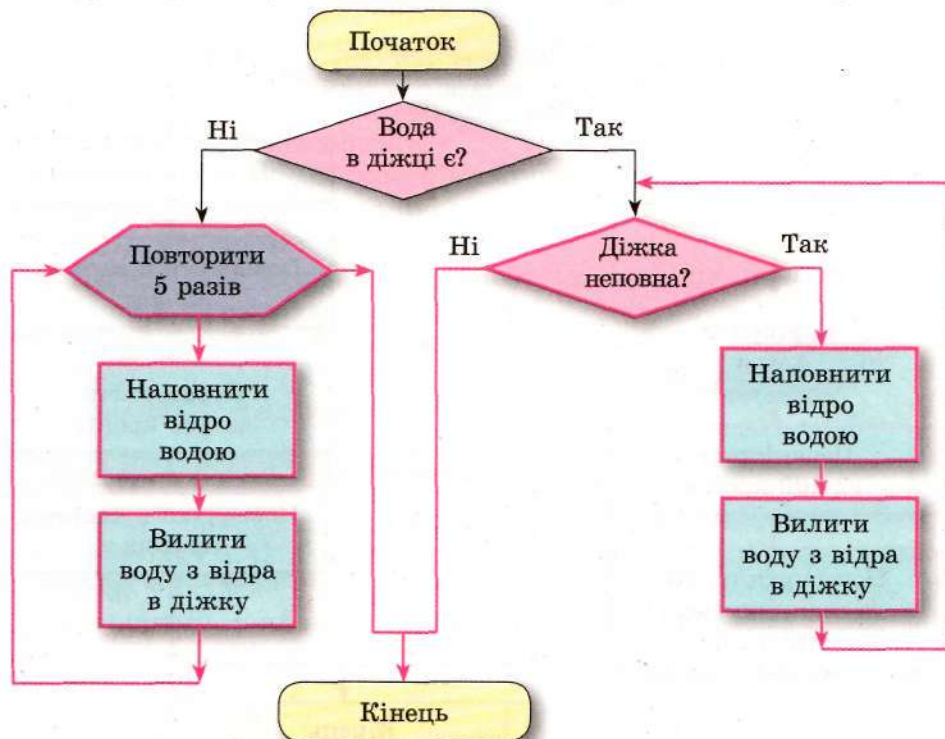
ЦИКЛ, ВКЛАДЕНИЙ У РОЗГАЛУЖЕННЯ

Тепер розглянемо приклади алгоритмів з використанням циклів у розгалуженні.

Задача 2. Є діжка місткістю 50 літрів. Невідомо, чи є в ній вода. Є відро місткістю 10 л. Наповнити діжку водою.

Оскільки діжка може бути не порожньою (а можливо, навіть і повною), то спочатку перевіримо, чи є в ній вода. Якщо є, то наповнюватимемо її водою, доки вона буде неповною. Якщо спочатку води в діжці немає, то 5 разів наллємо в неї відро води.

Алгоритм розв'язування цієї задачі наведено на малюнку 3.38.



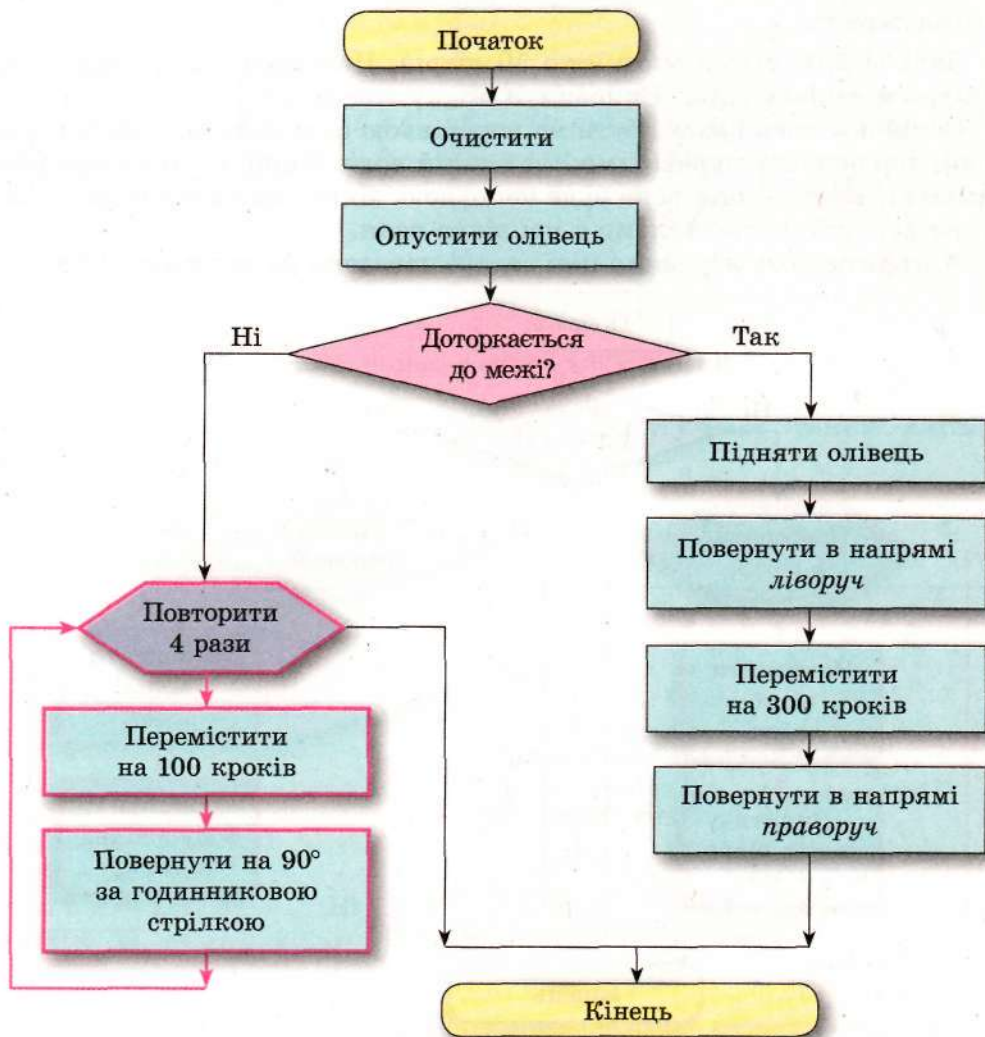
Мал. 3.38. Блок-схема алгоритму розв'язування задачі 2

Цикли в розгалуженні можна використовувати і в Scratch. Наведемо приклад проєкту, у якому виконавець відходить від правої межі Сцени, якщо він до неї доторкається, або малює квадрат, якщо він до межі не доторкається (мал. 3.39).

Мал. 3.39. Цикл у розгалуженні в Scratch



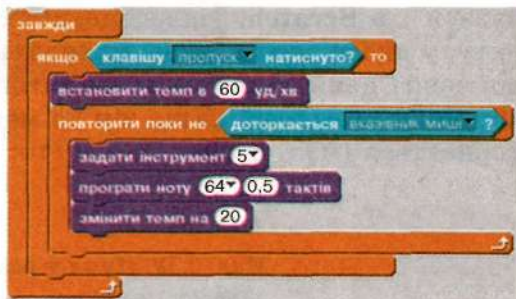
Блок-схема алгоритму наведеного проекту матиме такий вигляд (мал. 3.40):



Мал. 3.40. Блок-схема алгоритму проекту

Розглянемо ще один проект у **Scratch** з використанням циклів і розгалужень. У ньому, якщо натиснуто клавішу **пропуск**, виконавець у циклі, поки його не доторкається вказівник миші, грає певну ноту зі збільшенням темпу на 20 під час кожного виконання тіла циклу (мал. 3.41).

Мал. 3.41. Цикли та розгалуження в Scratch





Працюємо з комп'ютером

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

1. Відкрийте середовище **Scratch**.
2. Розмістіть **Рудого kota** посередині **Сцени**.
3. Розмістіть в області скриптів алгоритм, наведений на малюнку 3.39.
4. Збережіть проект у вашій папці у файлі з іменем **вправа 3.4_1**.
5. Виконайте проект.
6. Поясніть результат виконання проекту.
7. Розмістіть **Рудого kota** біля правої межі **Сцени**.
8. Запустіть проект на виконання.
9. Поясніть результат виконання проекту.
10. Відкрийте новий проект.
11. Розмістіть в області скриптів алгоритм, наведений на малюнку 3.41.
12. Збережіть проект у вашій папці у файлі з іменем **вправа 3.4_2**.
13. Запустіть проект на виконання.
14. Натисніть та утримуйте натиснутою клавішу **пропуск**.
15. Стежте за виконанням проекту.
16. Підведіть вказівник до виконавця.
17. Поясніть, чому виконавець припинив грати ноти.
18. Закрийте середовище **Scratch**.

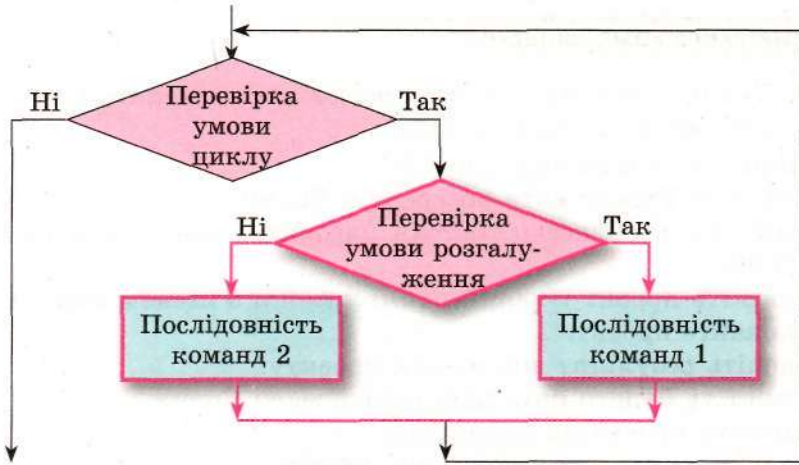
Найважливіше в цьому пункті

Розгалуження може бути вкладеним у цикл.

На малюнку 3.42 наведено фрагмент алгоритму, у якому повне розгалуження вкладено в цикл з лічильником, а на малюнку 3.43 – фрагмент алгоритму, у якому повне розгалуження вкладено в цикл з передумовою. Аналогічно у цикли може бути вкладено й неповні розгалуження.



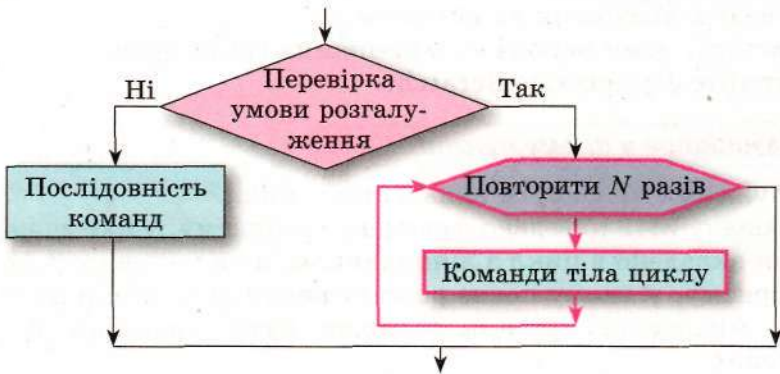
Мал. 3.42. Повне розгалуження, вкладене в цикл з лічильником



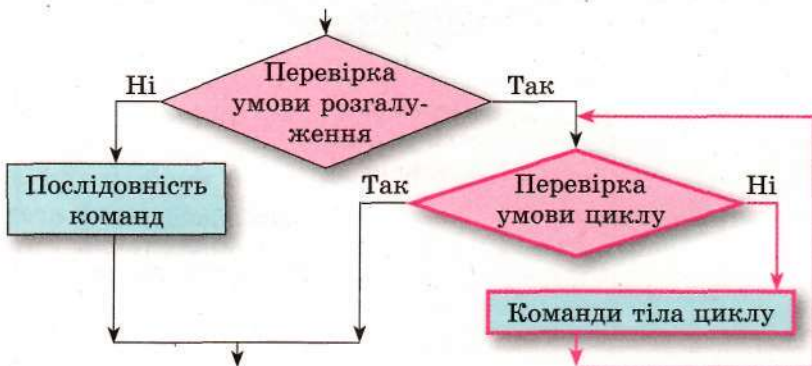
Мал. 3.43. Повне розгалуження, вкладене в цикл з передумовою

Також у розгалуження може бути вкладено цикл.

На малюнку 3.44 наведено фрагмент алгоритму, у якому цикл з лічильником вкладено в повне розгалуження, а на малюнку 3.45 – фрагмент алгоритму, у якому в повне розгалуження вкладено цикл з передумовою.



Мал. 3.44. Цикл з лічильником, вкладений у повне розгалуження



Мал. 3.45. Цикл з передумовою, вкладений у повне розгалуження



У наведених на малюнку 3.44 і 3.45 фрагментах алгоритмів цикл виконується, якщо результат перевірки умови розгалуження **Так**. Аналогічно цикл може входити в розгалуження так, щоб він виконувався, якщо результат перевірки умови розгалуження **Ні**. Також у повне розгалуження цикли можуть входити так, щоб один з них виконувався, якщо результат перевірки умови розгалуження **Так**, а другий – якщо результат перевірки умови розгалуження **Ні**.

Зазначимо, що цикли можуть бути вкладені не тільки в повне розгалуження, а й у неповне розгалуження.



Дайте відповіді на запитання

- 1°. Який цикл називають вкладеним у розгалуження?
- 2°. Яке розгалуження називають вкладеним у цикл?
- 3°. Як виконується фрагмент алгоритму, наведений на малюнку 3.42?
- 4°. Як виконується фрагмент алгоритму, наведений на малюнку 3.43?
- 5°. Як виконується фрагмент алгоритму, наведений на малюнку 3.44?
- 6°. Як виконується фрагмент алгоритму, наведений на малюнку 3.45?
- 7°. Як виконується наведений фрагмент алгоритму (мал. 3.46, а) в Scratch?
- 8°. Як виконується наведений фрагмент алгоритму (мал. 3.46, б) в Scratch?
- 9°. Як виконується наведений фрагмент алгоритму (мал. 3.46, в) в Scratch?
- 10°. Як виконується наведений фрагмент алгоритму (мал. 3.46, г) в Scratch?



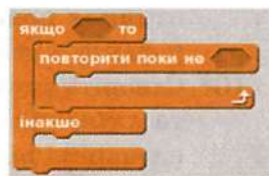
а)



б)



в)



г)

Мал. 3.46. Вкладені цикли і розгалуження в Scratch



Виконайте завдання

- 1°. Складіть блок-схему фрагмента алгоритму, у якому в цикл з лічильником вкладено неповне розгалуження. Поясніть виконання складеного фрагмента.
- 2°. Складіть блок-схему фрагмента алгоритму, у якому в цикл з передумовою вкладено неповне розгалуження. Поясніть виконання складеного фрагмента.

- 3°. Складіть блок-схему фрагмента алгоритму, у якому цикл з лічильником вкладено в неповне розгалуження. Поясніть виконання складеного фрагмента.
- 4°. Складіть блок-схему фрагмента алгоритму, у якому цикл з лічильником вкладено в повне розгалуження і який виконується, якщо результат перевірки умови розгалуження **Ні**. Поясніть виконання складеного фрагмента.
- 5°. Складіть блок-схему фрагмента алгоритму, у якому цикл з передумовою вкладено в неповне розгалуження. Поясніть виконання складеного фрагмента.
- 6°. Складіть блок-схему фрагмента алгоритму, у якому цикл з передумовою вкладено в повне розгалуження і який виконується, якщо результат перевірки умови розгалуження **Ні**. Поясніть виконання складеного фрагмента.
- 7°. Складіть проект у середовищі **Scratch**, у якому виконавець намалює три квадрати різних кольорів зі сторонами 100 кроків, якщо до нього доторкається вказівник миші. Збережіть проект у вашій папці у файлі з іменем **завдання 3.4.7**.
- 8°. Складіть проект у середовищі **Scratch**, у якому виконавець відійде від верхньої межі **Сцени** на 200 кроків і намалює прямокутник зі сторонами 100 і 80 кроків, якщо він стоїть біля цієї межі, і підійде до верхньої межі **Сцени**, якщо він біля неї не стоїть. Збережіть проект у вашій папці у файлі з іменем **завдання 3.4.8**.
- 9°. Складіть проект у середовищі **Scratch**, у якому, поки виконавець не доторкнеться до межі, грає мелодія, якщо він отримає повідомлення **Зіграй**, або рухається на 150 кроків уперед, якщо отримає повідомлення **Уперед**. Збережіть проект у вашій папці у файлі з іменем **завдання 3.4.9**.
- 10*. Створіть сценарій, у якому використовуються події *змінення тла* та *змінення образу*, і складіть проект, у якому реалізується цей сценарій і використовуються вкладені цикли і розгалуження. Збережіть проект у вашій папці у файлі з іменем **завдання 3.4.10**.
- 11*. Створіть сценарій з двома виконавцями та складіть проект, у якому реалізується цей сценарій і використовуються вкладені цикли та розгалуження. Збережіть проект у вашій папці у файлі з іменем **завдання 3.4.11**.
- 12*. Створіть сценарій і складіть проект, у якому реалізується цей сценарій і використовуються вкладені цикли та розгалуження. Збережіть проект у вашій папці у файлі з іменем **завдання 3.4.12**.



ПРАКТИЧНА РОБОТА № 6

«Проекти з вкладеними циклами та розгалуженнями»

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

1. Відкрийте середовище **Scratch**.
2. Складіть проект, у якому виконавець 4 рази зіграє деяку мелодію, якщо буде натиснуто клавішу **пропуск**, або переміститься на 100 кроків вниз, якщо буде натиснуто клавішу **стрілка вниз**.
3. Збережіть проект у вашій папці у файлі з іменем **практична 6_1**.
4. Складіть проект, у якому виконавець, поки до нього не **доторкнеться миша**, буде повторювати такі дії: рухатися вправо, якщо буде натиснуто клавішу **стрілка вправо**, або рухатися вліво, якщо буде натиснуто клавішу **стрілка вліво**.
5. Збережіть проект у вашій папці у файлі з іменем **практична 6_2**.
6. Закрийте середовище **Scratch**.

ПРАКТИЧНА РОБОТА № 1

«Створення простих векторних зображень»

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

Варіант 2

1. Запустіть на виконання графічний редактор **Draw**.
2. Створіть векторний малюнок відповідно до зразка.
3. Збережіть створений малюнок у вашій папці у файлі з іменем **практична 1.odg**.



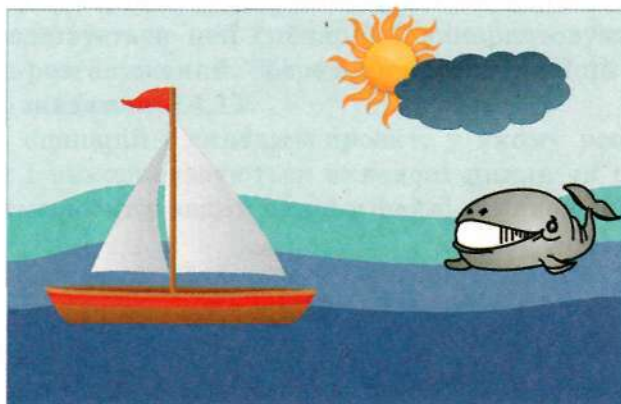
ПРАКТИЧНА РОБОТА № 2

«Створення векторних зображень»

Варіант 2

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

1. Запустіть на виконання програму графічного редактора **Draw**.
2. Створіть векторний малюнок відповідно до зразка, використовуючи інструменти малювання графічних примітивів з груп **Основні фігури**, **Фігури-символи**, а також вставлення об'єктів з групи **Транспорт** (**Бічна панель** ⇒ **Галерея**).
3. Під час малювання кита використайте редагування кривих Без'є.
4. Відформатуйте об'єкти та встановіть значення властивостей максимально наближено до зразка.
5. Згрупуйте всі об'єкти.
6. Збережіть створений малюнок у вашій папці у файлі з іменем **практична 2.odg**.





ПРАКТИЧНА РОБОТА № 3

«Створення комп'ютерної презентації»

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

Варіант 2

1. Створіть презентацію *Художні промисли України* на основі шаблону оформлення *Вишукана* з групи *Інсталювані теми* для виступу з повідомленням на класній годині. Потрібні зображення містяться у файлах папки **Розділ 2\Практична робота 3\Художні промисли**.

Сценарій презентації:

1. Титульний слайд. Заголовок слайда – *Художні промисли України*, підзаголовок – *ваше прізвище та ім'я, клас*.
 2. Слайд 2. Макет – *Порівняння*. Заголовок слайда – *Музеї художніх промислів*. Текст підписів – *Музей гончарства в Опішні, Музей писанкарства в Коломиї*, вирівнювання – *по центру*. Вставте на слайд фотографії відповідних музеїв.
 3. Слайд 3. Макет – *Два об'єкти*. Заголовок слайда – *Ткацтво*. Вставте на слайд зображення ткацького верстата та два абзаци тексту: *Ткацтво – виготовлення тканин на ткацькому верстаті. Одне з ремесел Київської Русі*. Установіть розмір символів – *24*. Вирівнювання – *по лівому краю*.
 4. Слайди 4, 5. Макет – *Зображення з підписом*. Рисунки – фотографії художніх виробів, у заголовках слайдів – *назви промислів Вишивка, Різьблення*. Текст слайда: перший абзац – *Вироби з тканини* або *Вироби з дерева*, другий абзац – *Рушники* або *Посуд*. Установіть розмір символів у заголовках – *40*, у підписах – *28*. Вирівнювання – *по ширині*.
 5. Слайд 6. Макет – *Заголовок розділу*. Заголовок слайда – *Ціную українське!*, текст слайда – *Художні промисли – скарб України!* Установіть розмір символів у тексті слайда – *28*. Вирівнювання тексту та заголовка – *по лівому краю*.
2. Збережіть комп'ютерну презентацію у вашій папці у файлі з іменем **практична робота 3**.



Зразок виконання



ПРАКТИЧНА РОБОТА № 4

«Налаштування показу комп'ютерної презентації»

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

Варіант 2

1. Відкрийте комп'ютерну презентацію з файла **Розділ 2\Практична робота 4\практична 4_2.pptx**.
2. Уведіть у підзаголовок титульного слайда ваше прізвище та ім'я, клас.
3. Додайте до кожного із зображень на слайді 2 ефект анімації *Фігура*, напрям ефекту – *Назовні*, початок – *Після клацання*.
4. Додайте на слайді 3 до написів *Музей гончарства* та *Музей писанкарства* гіперпосилання для переходу відповідно на слайди 4 та 5.
5. Додайте до тексту слайда 3 ефект анімації *Панорама*, початок відтворення ефекту – *З попереднім*.
6. Установіть для всіх слайдів ефект змінення слайдів *Розділення*.
7. Установіть тривалість демонстрації кожного слайда *7 секунд*.
8. Перегляньте комп'ютерну презентацію у режимі демонстрації.
9. Збережіть комп'ютерну презентацію у вашій папці у файлі з іменем **практична 4_2**.

ПРАКТИЧНА РОБОТА № 5

«Проекти з вкладеними циклами»

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

Варіант 2

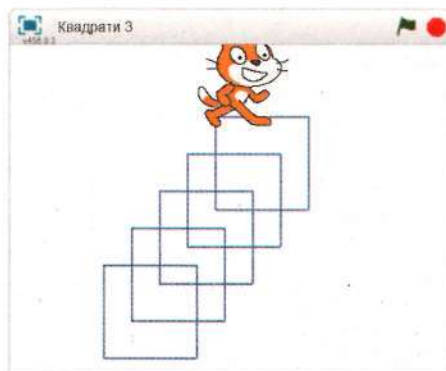
1. Відкрийте середовище **Scratch**.
2. Складіть проект для малювання чотирьох різнокольорових кіл, зображених на зразку 1.
3. Збережіть проект у вашій папці у файлі з іменем **практична 5_1**.



4. Складіть проект для малювання квадратів, поки виконавець не доторкнеться до межі Сцени (Зразок 2).
5. Збережіть проект у вашій папці у файлі з іменем **практична 5_2**.
6. Закрийте середовище **Scratch**.



Зразок 1



Зразок 2

ПРАКТИЧНА РОБОТА № 6

«Проекти з вкладеними циклами і розгалуженнями»

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

Варіант 2

1. Відкрийте середовище **Scratch**.
2. Складіть проект, у якому виконавець 4 рази змінить образ, якщо буде натиснуто клавішу **пропуск**, або переміститься на 20 кроків управо, якщо буде натиснуто клавішу **стрілка вправо**.
3. Збережіть проект у своїй папці у файлі з іменем **практична 6_1**.
4. Складіть проект, у якому виконавець, поки він не доторкнеться до **межі**, буде повторювати такі дії: рухатися вгору та повідомляти деяке повідомлення, якщо буде натиснуто клавішу **стрілка вгору**, або рухатися вниз і повідомляти інше повідомлення, якщо буде натиснуто клавішу **стрілка вниз**.
5. Збережіть проект у вашій папці у файлі з іменем **практична 6_2**.
6. Закрийте середовище **Scratch**.

СЛОВНИЧОК

А

Анімація – вид кіномистецтва, твори якого створюються шляхом зйомки послідовних етапів руху об'єктів, с. 66.

В

Векторне зображення будується з окремих геометричних фігур – **графічних примітивів**: відрізків, багатокутників, кривих, овалів тощо. Основними властивостями векторних зображень є види та кількість графічних примітивів, з яких будується зображення, та кількість кольорів, с. 16.

Вкладені розгалуження – фрагмент алгоритму, у якому розгалуження містяться всередині іншого розгалуження, с. 110.

Вкладені цикли – фрагмент алгоритму, у якому цикли містяться в тілі іншого циклу, с. 99.

Властивості (*параметри, атрибути, характеристики*) – відомості про об'єкт, за якими його можна описати та відрізнити від інших об'єктів, с. 7.

Е

Ефекти змінення слайдів – візуальні ефекти, які відтворюються під час появи слайдів комп'ютерної презентації, с. 80.

К

Кнопка дій – графічний об'єкт на слайді комп'ютерної презентації, призначений для переходу на деякий слайд або запуску певного файлу після вибору цієї кнопки або наведення на неї вказівника, с. 78.

Комп'ютерна графіка – розділ інформатики, який вивчає способи та засоби створення та опрацювання графічних зображень з використанням комп'ютерної техніки, с. 15.

Комп'ютерна презентація – набір слайдів, підготовлених з використанням інформаційних технологій і призначених для демонстрації текстових, графічних, відео, звукових даних з певної теми, найчастіше під час усного виступу або з іншою метою, с. 48.

Криві Безьє – криві, що використовуються у векторній графіці; редагування форми кривої відбувається шляхом зміни напрямку та довжини відрізків дотичних, с. 38.

М

Макет – схема розміщення текстових, графічних та інших об'єктів на слайді комп'ютерної презентації, с. 50.



Н

Незалежні події – дії, які жодним чином не впливають на настання одна одної, с. 10.

О

Об'єкти – тіла й явища оточуючого світу, с. 6.

Образ (костюм) – зовнішній вигляд об'єкта, с. 89.

П

Подія – дія або явище, що приводить до змінення значень властивостей об'єктів, с. 9.

Показчик місця заповнення – об'єкт на слайді комп'ютерної презентації, призначений для введення тексту або вставлення графічних об'єктів, відео, таблиць або діаграм, с. 59.

Презентація – представлення чогось нового, важливого групі зацікавлених осіб, с. 48.

Причина – подія, після настання якої обов'язково повинна наступити інша подія з якимось об'єктом, с. 9.

Програмна подія – дія, яка розпізнається програмним забезпеченням, обробляється за допомогою певних інструкцій і може стати причиною для виконання наступних дій, с. 11.

Програмні об'єкти – об'єкти, які використовуються в комп'ютерних програмах, с. 10.

Р

Растрове графічне зображення складається з окремих маленьких прямокутників – **пікселів**. Растрове графічне зображення має такі властивості: **розміри, роздільність, глибина кольору** та інші, с. 16.

Редактор презентацій – програма, призначена для створення комп'ютерних презентацій, їх редагування та форматування, збереження, перегляду тощо, с. 48.

Розгалуження, вкладені в цикл – фрагмент алгоритму, у якому розгалуження містяться в тілі циклу, с. 118.

С

Скрипт – алгоритм проекту в середовищі **Scratch**, с. 89.

Слайд – основний об'єкт комп'ютерної презентації. На слайдах комп'ютерної презентації можуть розміщуватися текстові, графічні, звукові, відео та інші об'єкти, с. 49.

Спрайт – виконавець у середовищі **Scratch**, с. 89.



Ц

Цикл, вкладений у розгалуження – фрагмент алгоритму, у якому цикл міститься в розгалуженні, с. 122.

L

LibreOffice Draw – векторний графічний редактор, що входить до складу офісних програм **LibreOffice**. **Draw** використовує багатопланову структуру малюнків, має засоби для створення креслень і діаграм, простих об'ємних фігур, для редагування складних кривих тощо, с. 25.

S

Scratch – програмне середовище для складання алгоритмів для різних виконавців, с. 11.



ЗМІСТ

Дорогі друзі!	3
---------------------	---

РОЗДІЛ 1

КОМП'ЮТЕРНА ГРАФІКА

1.1. Об'єкти та події	6
1.2. Основні поняття комп'ютерної графіки	15
1.3. Векторний графічний редактор	25
<i>Практична робота № 1.</i> Створення простих векторних зображень	35
1.4. Опрацювання векторних графічних зображень	36
<i>Практична робота № 2.</i> Створення векторних зображень	46

РОЗДІЛ 2

КОМП'ЮТЕРНІ ПРЕЗЕНТАЦІЇ

2.1. Комп'ютерні презентації. Об'єкти комп'ютерної презентації. Етапи створення комп'ютерної презентації	48
2.2. Макети слайдів. Додавання об'єктів до слайдів	56
<i>Практична робота № 3.</i> Створення комп'ютерної презентації	65
2.3. Анімація об'єктів на слайді	66
2.4. Налаштування показу комп'ютерної презентації	76
<i>Практична робота № 4.</i> Налаштування показу комп'ютерної презентації	87

РОЗДІЛ 3

АЛГОРИТМИ ТА ПРОГРАМИ

3.1. Програмні об'єкти та програмне опрацювання події в Scratch	89
3.2. Вкладені цикли	98
<i>Практична робота № 5.</i> Проекти з вкладеними циклами	107
3.3. Вкладені розгалуження	108
3.4. Вкладені цикли та розгалуження	117
<i>Практична робота № 6.</i> Проекти з вкладеними циклами та розгалуженнями	127
ДОДАТОК	128
СЛОВНИЧОК	132

Навчальне видання

РИВКІНД Йосиф Якович
ЛИСЕНКО Тетяна Іванівна
ЧЕРНІКОВА Людмила Антонівна
ШАКОТЬКО Віктор Васильович

ІНФОРМАТИКА

**Підручник для 6 класу
загальноосвітніх навчальних закладів**

Рекомендовано Міністерством освіти і науки України

Головний редактор *Наталія Заблоцька*

Редактор *Наталія Дашко*

Обкладинка *Тетяни Куц*

Макет, художнє оформлення,

комп'ютерна обробка ілюстрацій *Василя Марущинця*

Технічний редактор *Цезарина Федосіхіна*

Комп'ютерна верстка *Лариси Кулагіної, Людмили Ємець*

Коректори *Лариса Леуська, Любов Федоренко*

Формат 70×100/16.

Ум. друк. арк. 8,5. Обл.-вид. арк. 7,56.

Тираж 3023 пр. Вид. № 1903.

Зам. № 7-09-0521.

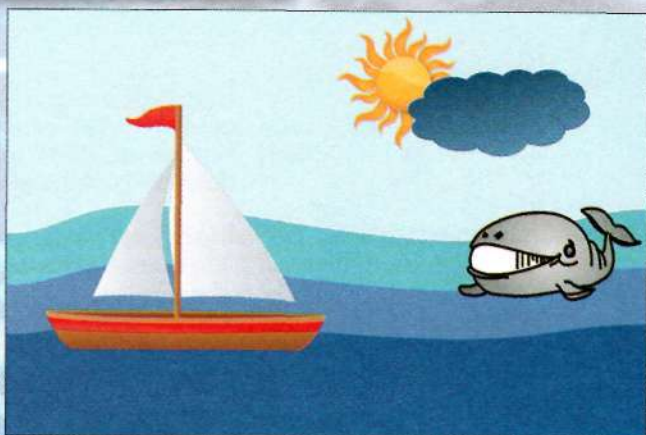
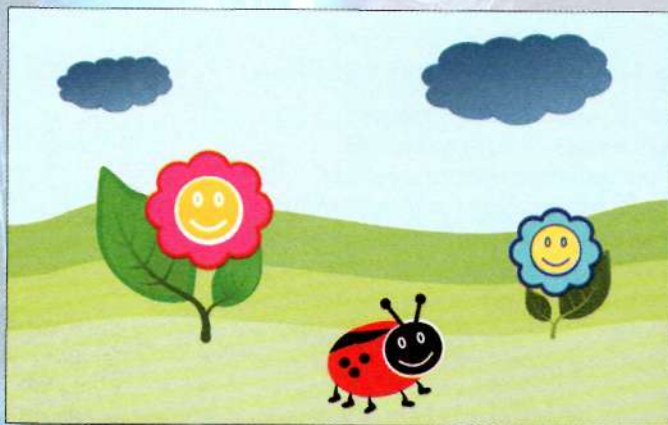
Видавництво «Генеза», вул. Тимошенка, 2-л, м. Київ, 04212.
Свідоцтво суб'єкта видавничої справи серія ДК № 5088 від 27.04.2016.

Віддруковано на ПРАТ «Харківська книжкова фабрика «Глобус»»,
вул. Різдвяна, 11, м. Харків, 61052.
Свідоцтво суб'єкта видавничої справи серія ДК № 3985 від 22.02.2011.
www.globus-book.com

ЗРАЗКИ ДО ВИКОНАННЯ ПРАКТИЧНОЇ РОБОТИ № 1



ЗРАЗКИ ДО ВИКОНАННЯ ПРАКТИЧНОЇ РОБОТИ № 2





ЗБЕРЕЖЕМО ПРИРОДУ УКРАЇНИ

Людмила Олександрівна

Червона книга України



- Заснована у 1976 році.
- Вперше опублікована у 1980 році.
- Містить відомості про зникаючі види тварин та рослин.

Вони потребують захисту

Тваринний світ



Рослинний світ



Чорний журавель

Тваринний світ.
Птах



Сон-трава

Рослинний світ.
Квітка

Пам'ятай!

Вони потребують нашого захисту!

ХУДОЖНІ ПРОМИСЛИ УКРАЇНИ

Валентина Марічка

МУЗЕЙ ХУДОЖНІХ ПРОМИСЛІВ



Музей гончарства
в Одесі



Музей писарництва
в Калуші

ТКАЦТВО

- Ткацтво – виготовлення тканин на ткацькому верстаті.
- Одне з ремесел Київської Русі.



ВИШИВКА
Вироби з тканини.
Рушники

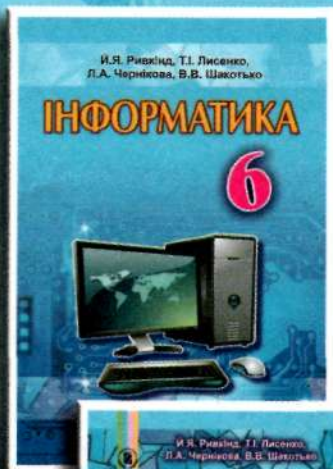


РІЗЬБЛЕННЯ
Вироби з дерева.
Посуд

Художні промисли – скарб України!
ЦІНІТЬ УКРАЇНСЬКЕ!

ЗА НОВОЮ ПРОГРАМОЮ ДЛЯ УЧНІВ, ЯКІ ПОЧАЛИ ВИВЧАТИ ІНФОРМАТИКУ З 2-ГО КЛАСУ

Видавництво «ГЕНЕЗА» пропонує
навчальний комплект
«ІНФОРМАТИКА.
6 клас»:



- ПІДРУЧНИК
автори
Й.Я. Ривкінд,
Т.І. Лисенко,
Л.А. Чернікова,
В.В. Шакоцько

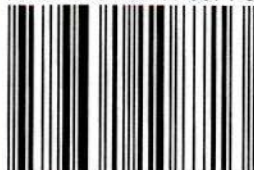


- РОБОЧИЙ ЗОШИТ
автори
Й.Я. Ривкінд,
Т.І. Лисенко,
Л.А. Чернікова,
В.В. Шакоцько

*Учням — тільки якісні
знання!*

ТОВ «Видавництво «Гене́за»
тел.: (044) 379-14-07
(044) 379-10-54
e-mail: sales@geneza.ua
Сайт: www.geneza.ua
www.shop.geneza.ua

ISBN 978-966-11-0871-3



9 789661 108713 >