



Розділ 4

кроків і повідомляє про це. Збережіть проект у вашій папці у файлі з іменем завдання 4.3.2.

3*. Намалюйте у вбудованому графічному редакторі три дерева в різні пори року. Створіть проект з використанням цих образів. Збережіть проект у вашій папці у файлі з іменем завдання 4.3.3.

4°. Складіть проект з двома виконавцями, які спочатку знаходяться в середині **Сцени**, а потім рухаються один від одного в напрямку країв **Сцени**. Збережіть проект у вашій папці у файлі з іменем завдання 4.3.4.



5°. Складіть проект із чотирма виконавцями, кожен з яких рухається вздовж відповідного краю **Сцени**. Збережіть проект у вашій папці у файлі з іменем завдання 4.3.5.



4.4. ПОВТОРЕННЯ (ЦИКЛИ). АЛГОРИТМИ З ПОВТОРЕННЯМ



1. Які явища природи, що неодноразово повторюються, ви спостерігали?
2. Які процеси у вашому житті неодноразово повторюються?
3. Які процеси називають циклічними? Наведіть приклади.

ПОВТОРЕННЯ (ЦИКЛИ)

Цикл (від грец. *kiklos* – коло) – сукупність взаємопов'язаних явищ, процесів, робіт, що створює закінчене коло дій.

Ви вже знаєте, що процеси, які повторюються, називають **циклічними**.

Циклічними, наприклад, є процеси щоденного сходу і заходу Сонця (мал. 4.32), змінення форми Місяця (мал. 4.33), уроки у школі щотижня протягом семестру та ін.

Кожний з вас бере участь у циклічних процесах. Кожного робочого дня у школі уроки й перерви тривають протягом одних і тих самих інтервалів часу. Кожного разу ви йдете або їдете з дому до спортивної секції або музичної школи одним і тим самим маршрутом. Кожного разу, коли вам потрібно закип'ятити воду в чайнику, ви виконуєте одну й ту саму послідовність дій.

На уроках української мови, розбираючи кілька речень, ви також кілька разів виконуєте одну й ту саму послідовність дій. Додаючи числа у стовпчик ви також виконуєте одну й ту саму послідовність дій.



І таких прикладів циклічних дій можна навести багато.



Мал. 4.32. Схід і захід Сонця



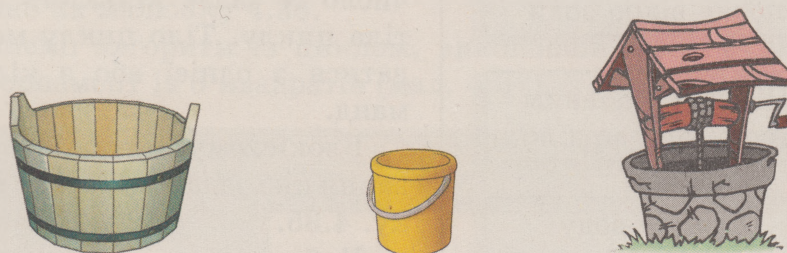
Мал. 4.33. Фази Місяця

ПОВТОРЕННЯ (ЦИКЛИ) В АЛГОРИТМАХ

В алгоритмах розв'язування багатьох задач потрібно виконати одну або кілька команд більше ніж один раз. Для цього такі алгоритми мають містити команди, які визначатимуть, які команди повинні виконатися неодноразово і скільки саме разів.

Розглянемо таку задачу.

Задача. У дворі є порожня діжка і відро місткістю 50 л і 10 л відповідно та колодезь (мал. 4.34). Потрібно наповнити діжку водою.



Мал. 4.34. Діжка, відро та колодезь

Очевидно, для розв'язування цієї задачі потрібно виконати такий алгоритм:

1. Узяти відро.
2. Повторити 5 разів:
 1. Підійти до колодезя.
 2. Набрати з колодезя повне відро води.
 3. Підійти з повним відром до діжки.
 4. Вилити воду з відра в діжку.
3. Поставити відро.



Розділ 4

Цей алгоритм містить команду 2:
Повторити 5 разів

1. Підійти до колодязя.
2. Набрати з колодязя повне відро води.
3. Підійти з повним відром води до діжки.
4. Вилити воду з відра в діжку.

Таку команду називають **командою циклу з лічильником**. Вона визначає, що під час виконання алгоритму команди:

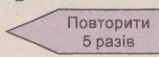
1. Підійти до колодязя.
2. Набрати з колодязя повне відро води.
3. Підійти з повним відром до діжки.
4. Вилити воду з відра в діжку повинні виконатися 5 разів поспіль. Вони утворюють тіло циклу.

Сама команда **Повторити 5 разів** задає кількість повторень тіла циклу. Її називають **заголовком циклу**.

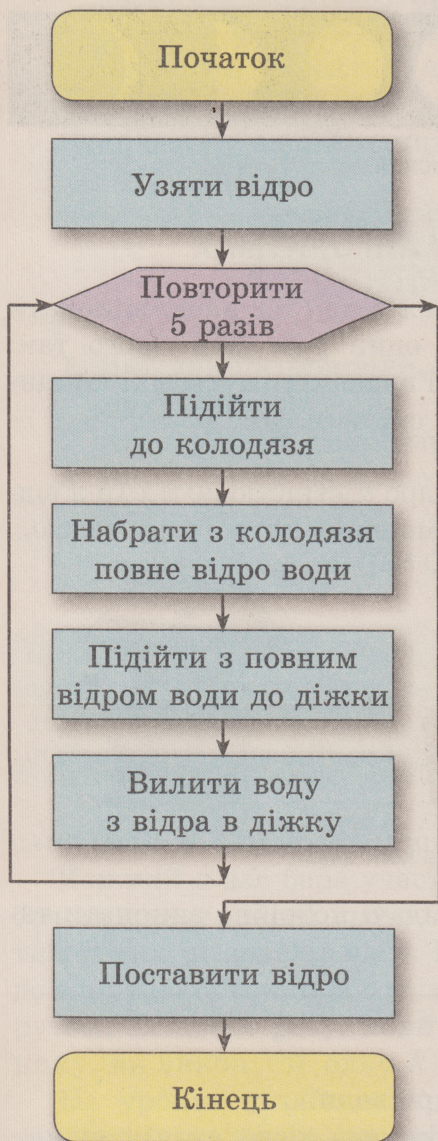
Загальний вигляд команди циклу з лічильником:

Повторити N разів Команди, де **Команди** утворюють тіло циклу, число **N** задає кількість повторень тіла циклу. Тіло циклу може складатися з однієї або з кількох команд.

Блок-схему алгоритму розв'язування цієї задачі наведено на малюнку. 4.35.

У блок-схемі використано блок **Повторити N разів** , який задає кількість повторень команд тіла циклу.

Фрагмент алгоритму, у якому одна або кілька команд можуть виконуватися більше ніж один раз, називають **циклом**. Алгоритм, який містить цикл, називають **алгоритмом із циклом**, або **алгоритмом з повторенням**.

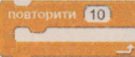


Мал. 4.35. Блок-схема алгоритму наповнення діжки водою

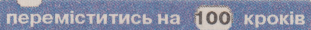



ЦИКЛИ З ЛІЧИЛЬНИКОМ У SCRATCH

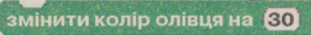
У середовищі **Scratch** також можна складати алгоритми із циклами. Для цього в системі команд виконавців є спеціальні команди. Зокрема, для організації в алгоритмі циклу з лічильником

можна використати команду , яка розміщена у групі **Керувати**. Її виконання приводить до виконання вказаної кількості разів команд, які містяться всередині цього блока (тіло циклу). Зрозуміло, що кількість повторень тіла циклу можна змінювати.

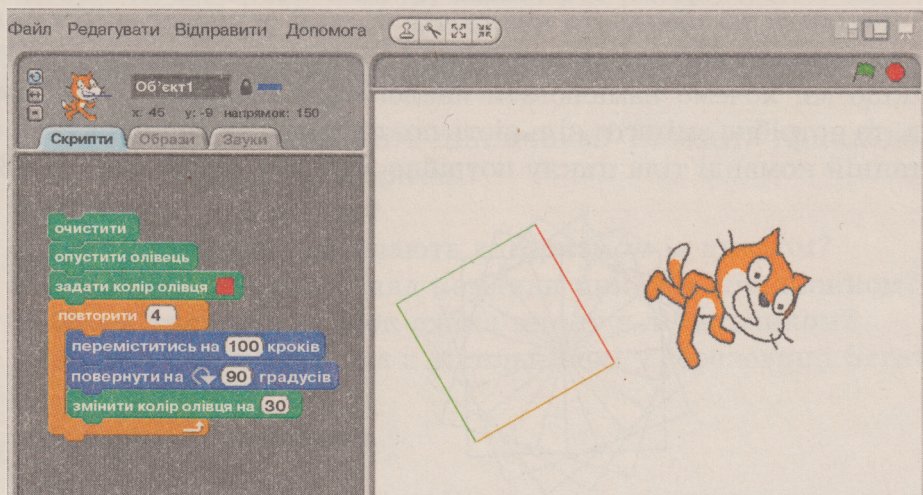
Ви вже знаєте, що для малювання квадрата зі стороною, наприклад, 100 кроків потрібно 4 рази поспіль виконати команди:

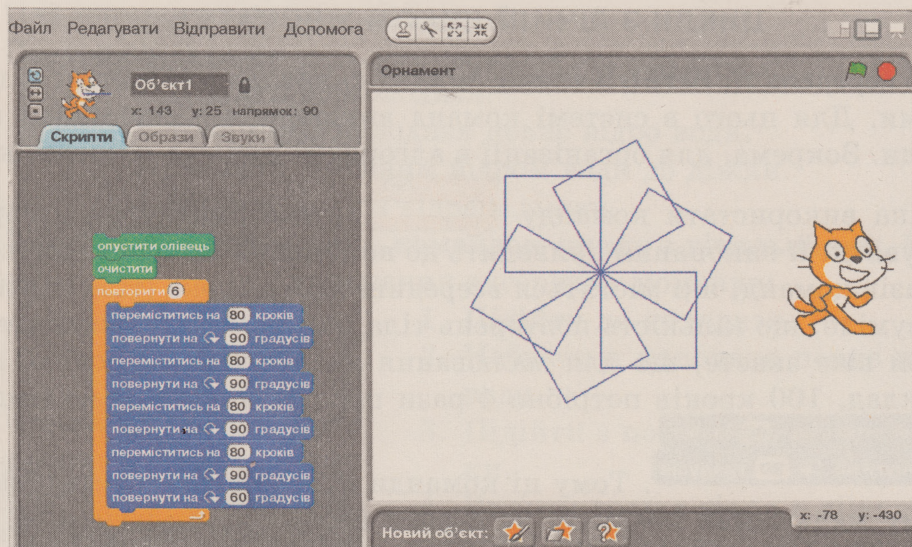
. Тому ці команди можна включити до алгоритму малювання квадрата не 4 рази поспіль, а включити їх до тіла циклу з лічильником, яке повинно виконатися 4 рази.

Крім того, до цього тіла циклу можна включити команду змінення кольору олівця для змінення кольору малювання. У **Scratch** кожному кольору олівця відповідає певне число, код цього кольору. В алгоритмі, наведеному на малюнку 4.36, перед командою циклу розміщено команду, яка задає початковий колір олівця – червоний. Під час виконання команди  у тілі циклу під час кожного виконання команди код кольору олівця збільшується на 30. Тим самим змінюється колір малювання, як це показано на малюнку 4.36.

Розглянемо алгоритм із циклом, виконавши який **Рудий кіт** намалює орнамент із 6 квадратів (мал. 4.37).



Мал. 4.36. Алгоритм малювання різнокольорового квадрата



Мал. 4.37. Алгоритм малювання орнаменту

Тіло циклу цього алгоритму містить команди малювання квадрата і поворота виконавця на кут 60° . Повторюватиметься виконання цього тіла циклу 6 разів.

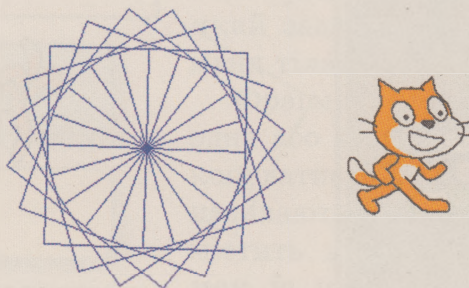
Орнамент (лат. *ornamentum* – прикраса) – прикраса у вигляді візерунка, що складається із симетрично розташованих художніх елементів.

Тому в отриманому орнаменті кожний наступний квадрат повернуто відносно попереднього на кут 60° .

В останній команді тіла циклу здійснюється поворот виконавця саме на 60° , тому що орнамент

складається із 6 квадратів, і якщо кут повного повороту по колу 360° поділити на кількість квадратів 6, то й отримаємо кут повороту виконавця 60° для малювання наступного квадрата.

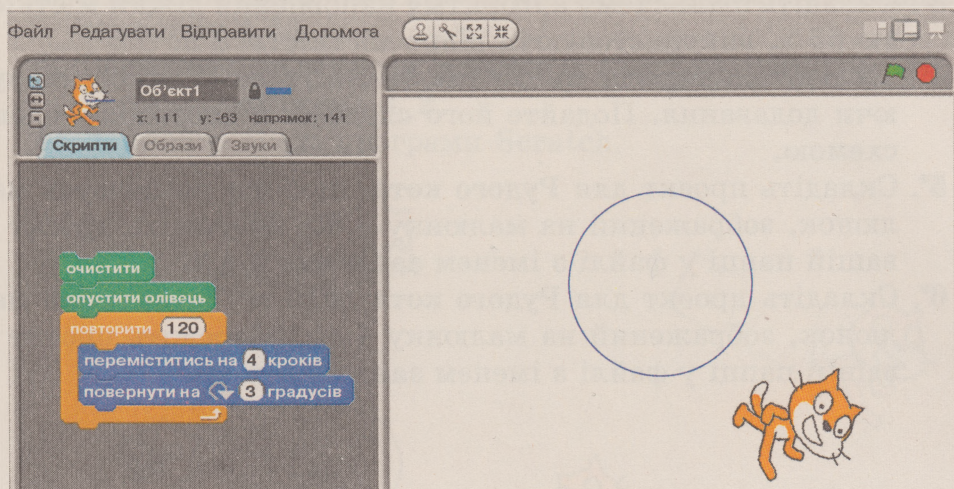
Якщо ми хочемо намалювати аналогічний орнамент із 20 квадратів, то потрібно змінити кількість повторень тіла циклу на 20 і кут в останній команді тіла циклу потрібно змінити на 18° (мал. 4.38).



Мал. 4.38. Орнамент, утворений двадцятьма квадратами



Наведемо ще приклад алгоритму, виконавши який Рудий кіт намалює коло (мал. 4.39).



Мал. 4.39. Алгоритм малювання кола

! Найважливіше в цьому пункті

Процеси, які повторюються, називають **циклічними**.

Команду виду **Повторити N разів Команди** називають **командою циклу з лічильником**. **Команди** утворюють **тіло циклу**, а число **N** задає **кількість виконань команд тіла циклу**.

Фрагмент алгоритму, у якому одна або кілька команд можуть виконуватися більше ніж один раз, називають **циклом**.

Алгоритм, що містить цикл, називають **алгоритмом із циклом**.

? Дайте відповіді на запитання


- 1°. Який процес називають циклічним? Наведіть приклади.
- 2°. Що таке цикл в алгоритмі?
- 3°. Що таке тіло циклу?
- 4°. Який алгоритм називають алгоритмом із циклом?
- 5°. Який загальний вигляд команди циклу з лічильником?
- 6°. Який вигляд має блок-схема циклу з лічильником?
- 7°. Як організувати цикл з лічильником у середовищі Scratch?

✍ Виконайте завдання


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

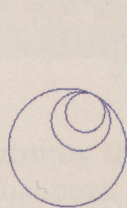
2*. Наведіть приклади циклів під час застосування правил математики, української мови.

3°. Складіть блок-схему алгоритму наповнення діжки місткістю 56 л, використовуючи відро місткістю 8 л.

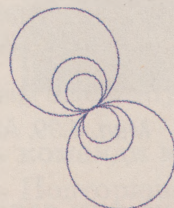
 4*. Складіть алгоритм множення числа 12 на 4, використовуючи додавання. Подайте його словесним способом і блок-схемою.

5°. Складіть проект для Рудого кота, щоб він намалював малюнок, зображений на малюнку 4.40. Збережіть проект у вашій папці у файлі з іменем завдання 4.4.5.

 6°. Складіть проект для Рудого кота, щоб він намалював малюнок, зображений на малюнку 4.41. Збережіть проект у вашій папці у файлі з іменем завдання 4.4.6.



Мал. 4.40




Мал. 4.41



7°. Складіть проект, у якому виконавець малюватиме прямокутник, змінюючи при цьому колір олівця і товщину лінії. Збережіть проект у вашій папці у файлі з іменем завдання 4.4.7.

8°. Складіть проект для двох виконавців, які рухаються назустріч один одному від країв Сцени, у якому буде використано команду циклу з лічильником. Збережіть проект у вашій папці у файлі з іменем завдання 4.4.8.

 9°. Складіть проект, у якому виконавець малюватиме прямокутник, змінюючи при цьому свої образи. Збережіть проект у вашій папці у файлі з іменем завдання 4.4.9.

ПРАКТИЧНА РОБОТА № 5

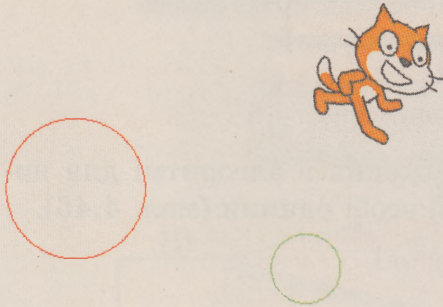
«Складання та виконання алгоритмів з використанням циклів з лічильником»

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

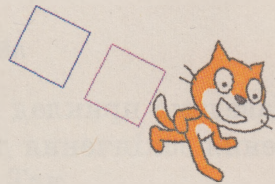
1. Відкрийте середовище Scratch.
2. Складіть алгоритм, щоб отримати зображення, подане на малюнку 4.42.



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.