

## *Практична робота*

**Тема:** складання алгоритмів опрацювання подій з використання структури слідування та виконання їх у визначеному навчальному середовищі виконання алгоритмів Scratch.

**Мета:** навчитися складати алгоритми з використання структури слідування. Навчитися виконувати створені алгоритми у навчальному середовищі програмування Scratch.

Після вивчення теми *учень*

*знає:*

виконавців алгоритму та прості команди виконавців у визначеному навчальному середовищі виконання алгоритмів;

*уміє:*

- складати алгоритм у словесній формі;
- складати алгоритм у графічному вигляді;
- формально виконувати алгоритми з навчальної діяльності та побуту;
- записувати алгоритм у вигляді послідовності команд виконавця;
- складати і виконувати алгоритми у визначеному навчальному середовищі виконання алгоритму;
- складати план дій з повсякденного життя та з використанням матеріалу навчальних предметів (математики, української мови тощо).

**Обладнання:** ПК із встановленими ОС і середовищем Scratch, інтерактивна дошка та дошка для писання маркером.

### **Структура уроку**

1. Організаційний момент.
2. Актуалізація опорних знань.
3. Вивчення нового матеріалу.
4. Інструктаж з ТБ.
5. Закріплення вивченого матеріалу.
6. Підбиття підсумків уроку.
7. Домашнє завдання.

### **Хід уроку**

#### **1. Організаційний момент.**

Вітання з класом. Перевірка присутності і готовності учнів до уроку.  
Перевірка виконання домашнього завдання.

#### **2. Актуалізація опорних знань.**

*Дати відповіді на запитання.*

1. Що таке алгоритм?
2. Що таке виконавець?

3. Назвіть і поясніть властивості алгоритму.
4. Що називається системою вказівок виконавця?
5. Що таке середовище виконання алгоритму?
6. Поясніть, що називається словесною формою подання алгоритму.
7. Поясніть, що називається графічною формою подання алгоритму.

### 3. Вивчення нового матеріалу.

Спостерігаючи за поведінкою різних об'єктів, люди виявили такі основні типи алгоритмів:

- лінійний алгоритм;
- алгоритм з розгалуженням;
- циклічний алгоритм.

**Лінійний алгоритм (структура слідування)** — *алгоритм, в якому дії виконуються один раз послідовно одна за одною від початку до кінця.* Прикладами таких алгоритмів є перехід пішоходом вулиці, обчислення суми чисел, кип'ятіння води в чайнику тощо.

Виконавцем алгоритму може бути *будь-який об'єкт* — жива істота чи технічний пристрій. В інформатиці універсальним виконавцем є *комп'ютер*.

Як ви гадаєте, чому пральна машина не випікає тістечок, а холодильник не пере білизни?

А тому, що кожний об'єкт має обмежену систему команд виконавця. Пральна машина виконує команди прання, натомість команди випікання є для неї нездійсненними.

У навчальному середовищі виконання алгоритмів Scratch є лише два види виконавців алгоритмів, а саме *спрайти* та *сцени*. Це дає їм змогу виконувати численні команди різноманітних алгоритмів.

Основа Scratch — *графічна мова програмування*, що дозволяє керувати діями (операціями) та взаємодією між різними медіа об'єктами.

Ви можете самостійно розробляти і створювати різноманітні проекти, комбінуючи графіку, фотографії, музику і звук в інтерактивній розробці.

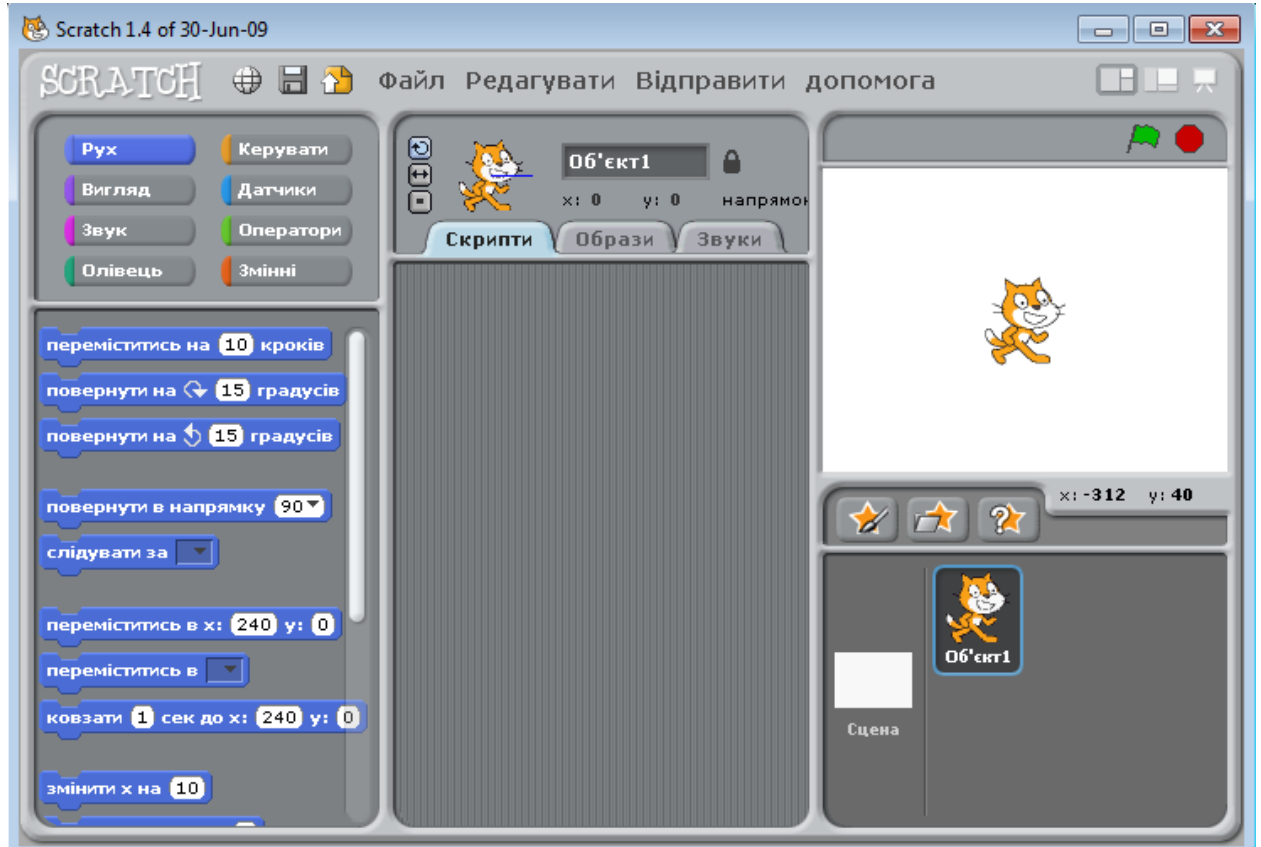
Програмувати в Scratch набагато простіше, ніж у традиційних мовах програмування. Так, щоб створити скрипт, потрібно лише об'єднати різнокольорові графічні блоки, як цеглинки в LEGO або пазли.

*Основне завдання Scratch* — зробити математику, основи алгоритмізації та програмування більш привабливим і доступним для дітей, підлітків, а також тих, хто хоче навчитися програмувати. *Scratch* використовує кращі зразки архітектурних підходів та дизайну інтерфейсів.

## Запустити Scratch можна різними способами.

1. Знайдіть на робочому столі піктограму із зображенням *Рудого Кота* та двічі клацніть на ньому покажчиком миші.

Якщо ви все зробили правильно, то через деякий час завантажиться обране середовище.



2. Якщо на робочому столі ви не знайшли Scratch, то пошукайте його в головному меню системи, яке ховається за кнопкою «Пуск».

## 4. Інструктаж з ТБ.

## 5. Закріплення вивченого матеріалу.

**Примітка.** Повідомляти вчителя про виконання кожного завдання і продемонструвати результати своєї роботи.

**Завдання 1 («Спрайт-художник»).** Використовуючи спрайта Рудого Кота створіть такі геометричні фігури: квадрат, трикутник, шестикутник.

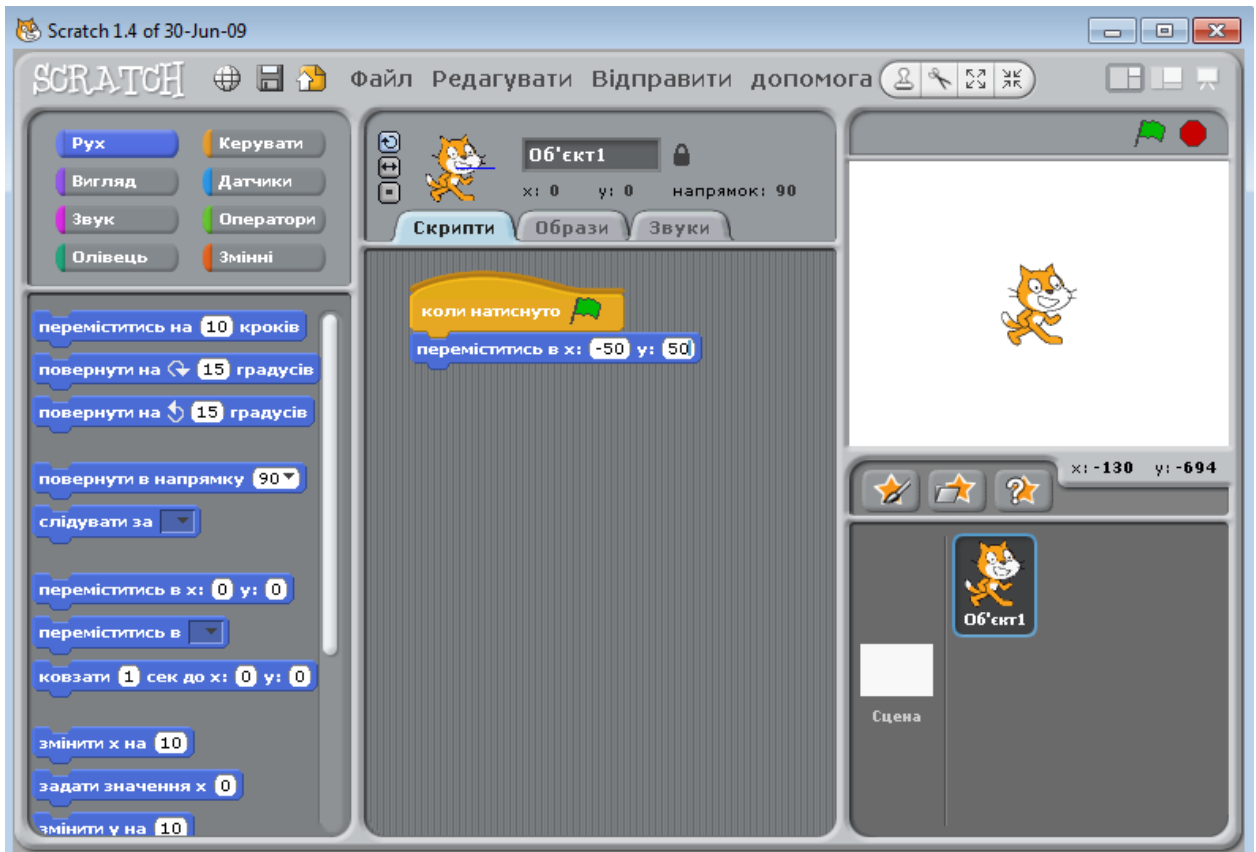
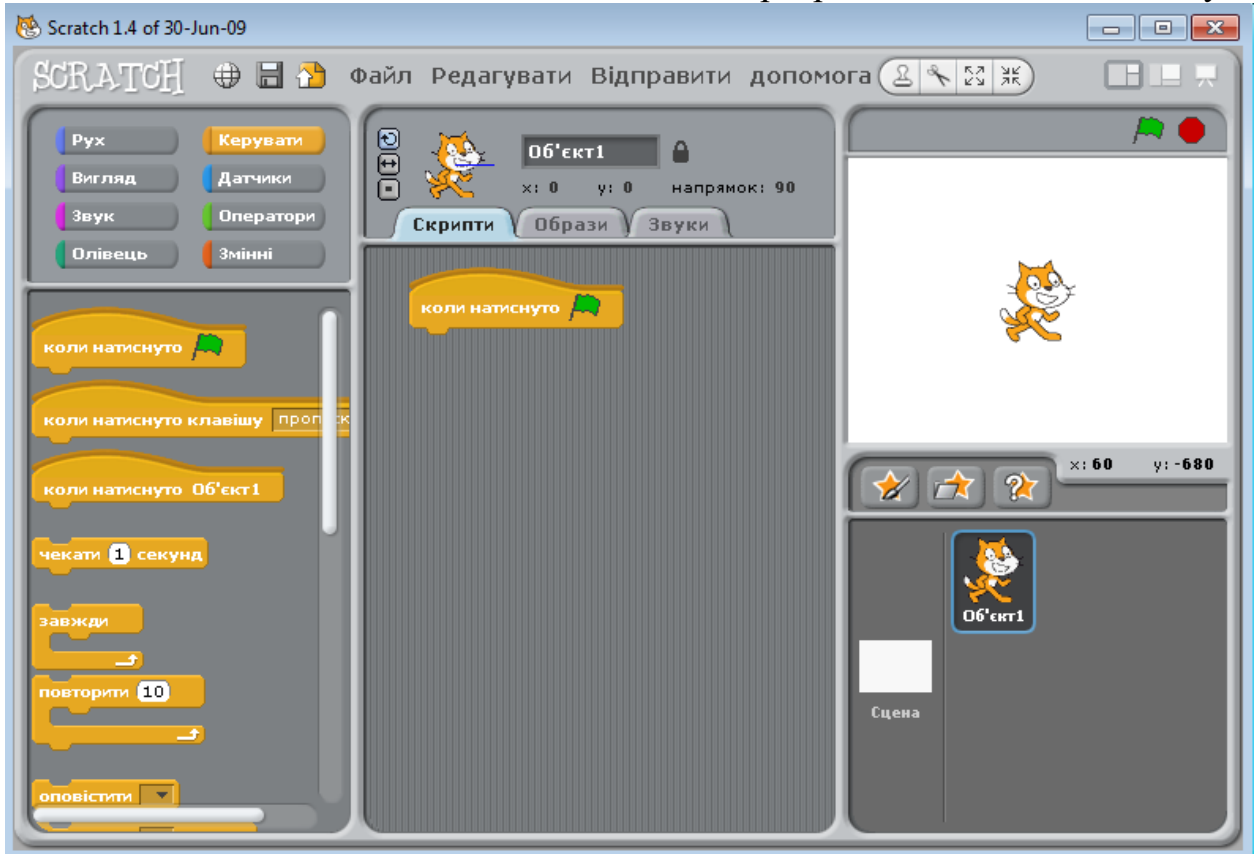
1. Створюємо геометричну фігуру "квадрат", складаючи сценарій поведінки об'єкта з блоків-команд руху, малювання та керування.

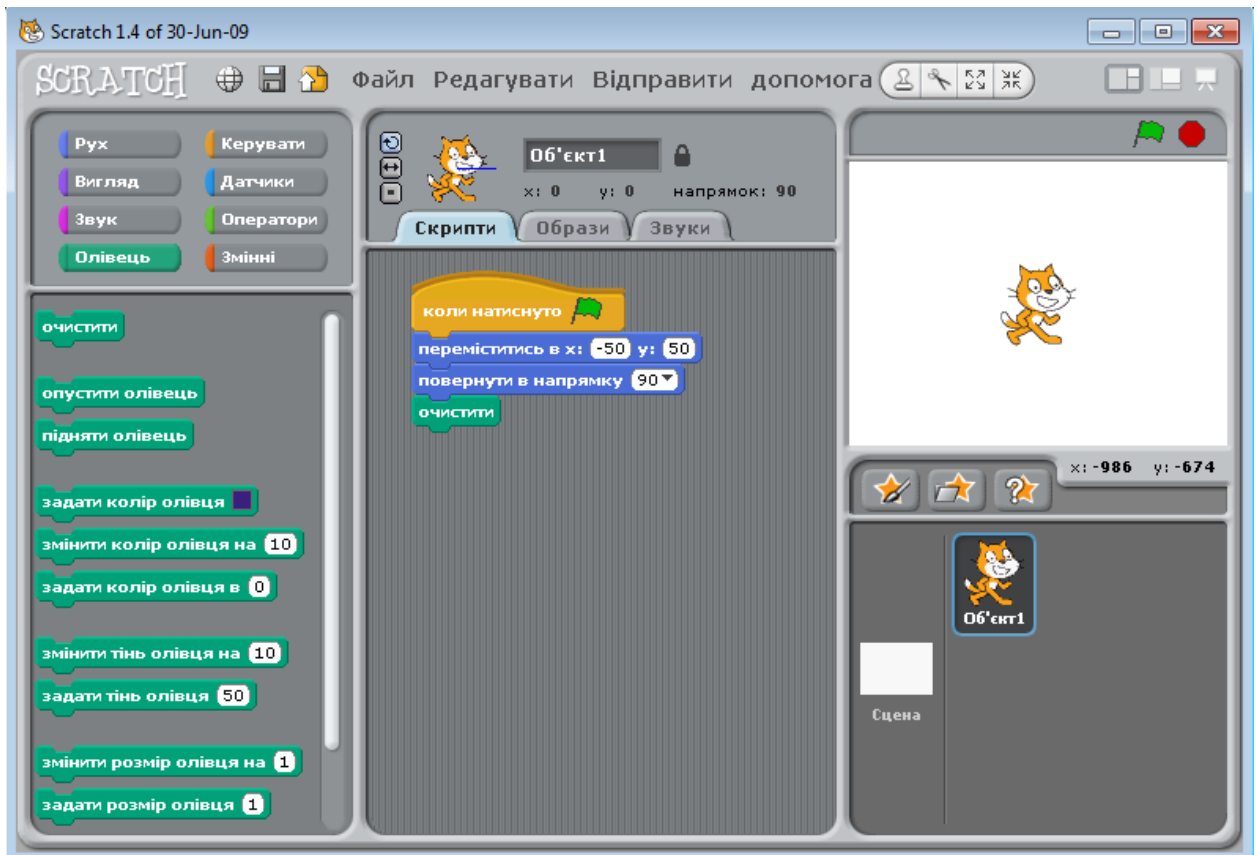
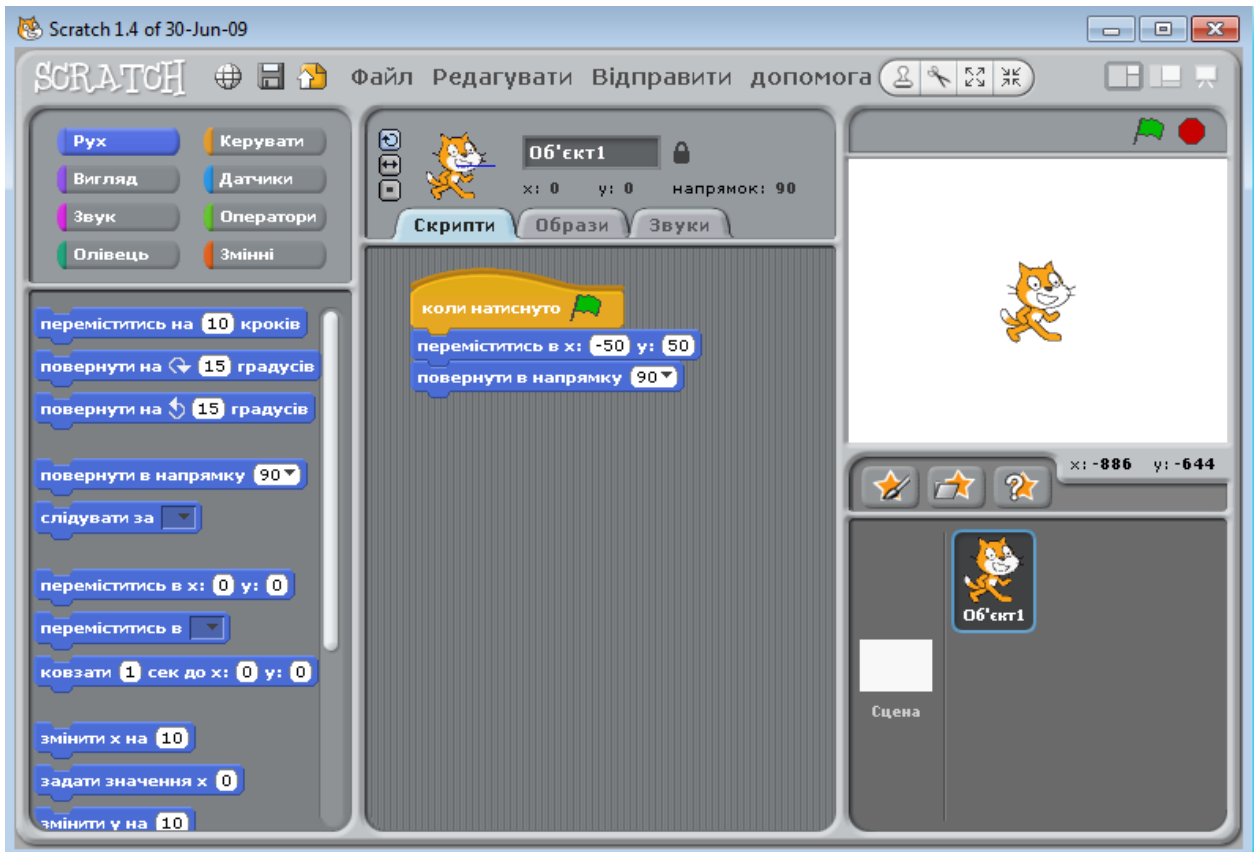
Алгоритм:

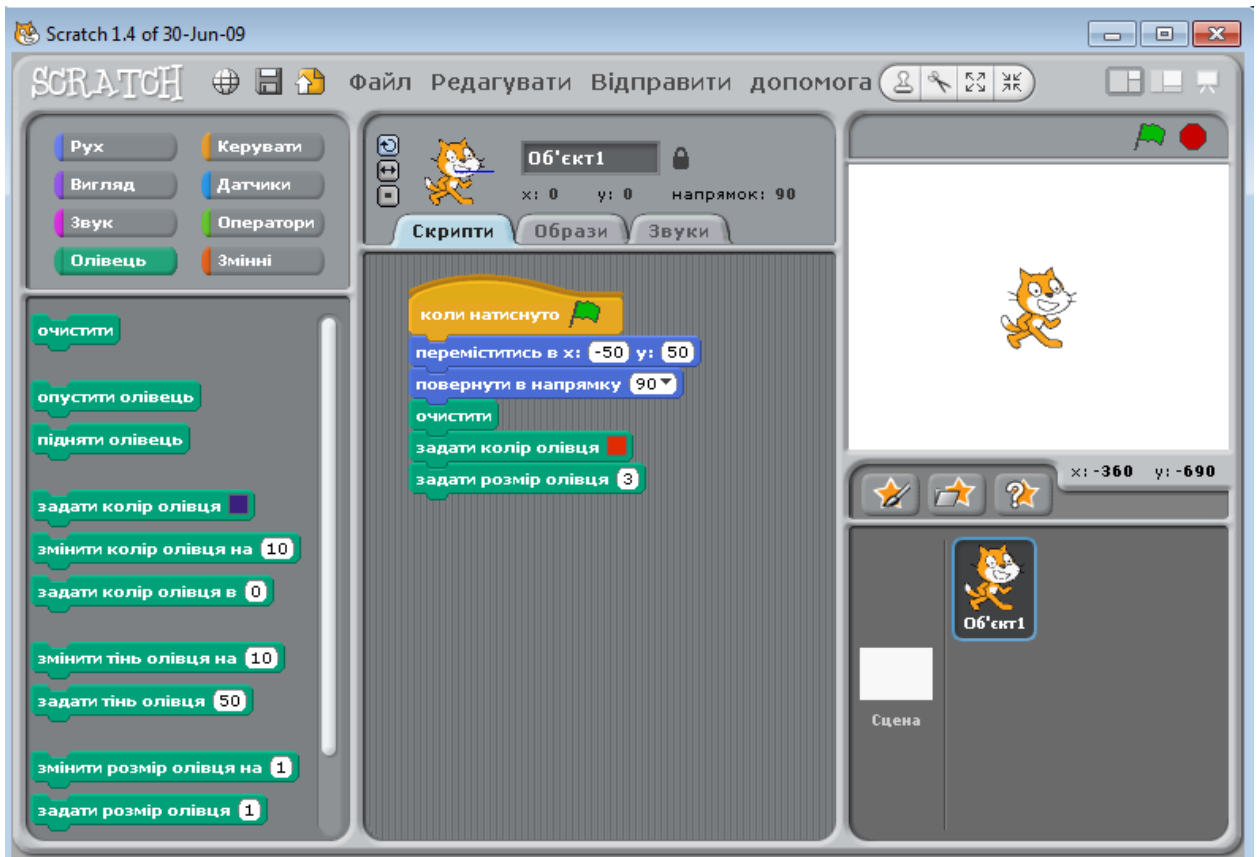
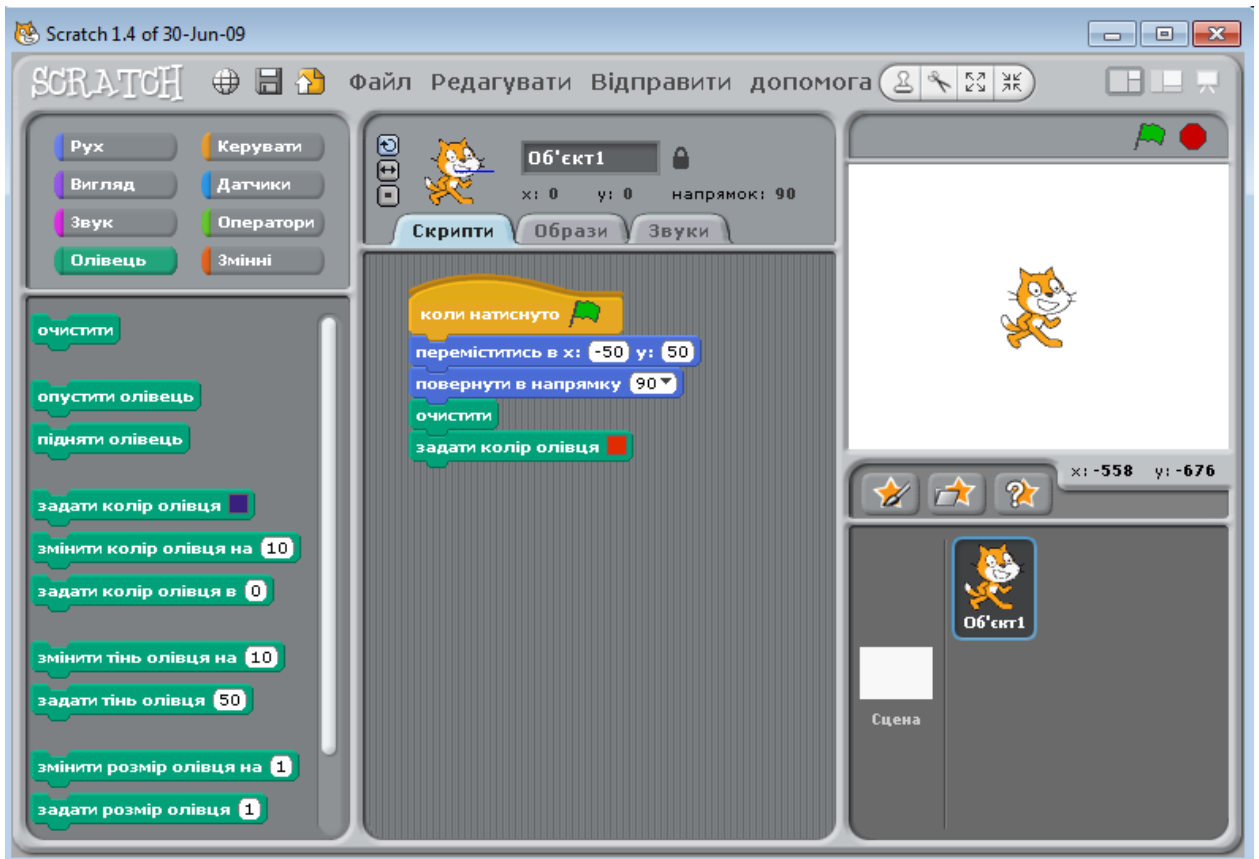
- запустити скрипт;
- перемістити спрайт на вказану позицію  $x: -50$  і  $y: 50$  на сцені;
- повернути спрайт вправо;
- очистити всі сліди олівця та штампи зі сцени;
- встановити червоний колір олівця з кольорової палітри;
- встановити товщину олівця;

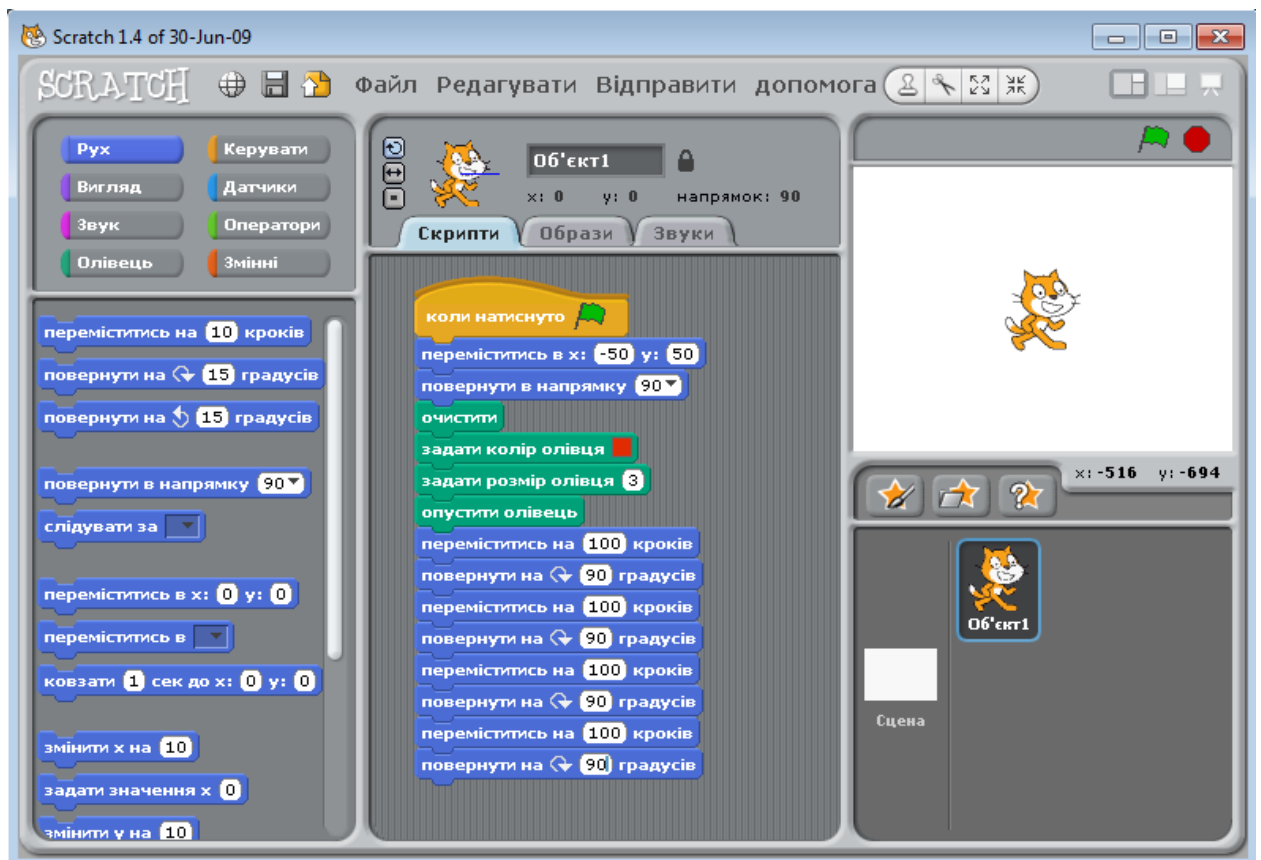
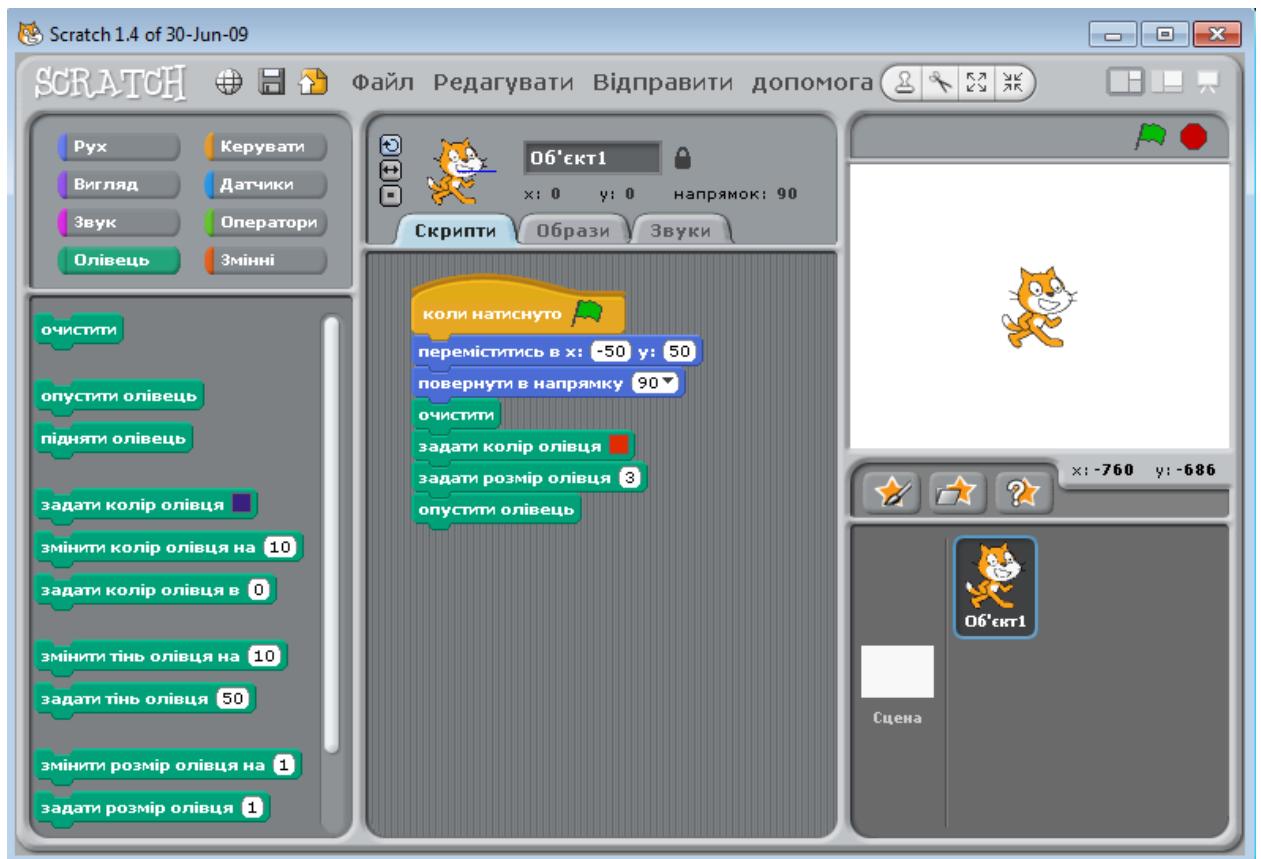
- опустити олівець, щоб спрайт малював під час переміщення;
- перемістити спрайт вперед на 100 кроків;
- обертати спрайт за годинниковою стрілкою на 90 градусів;
- повторити останні дві команди 4 рази.

На малюнках нижче подано вигляд вікон програми після кожного етапу.

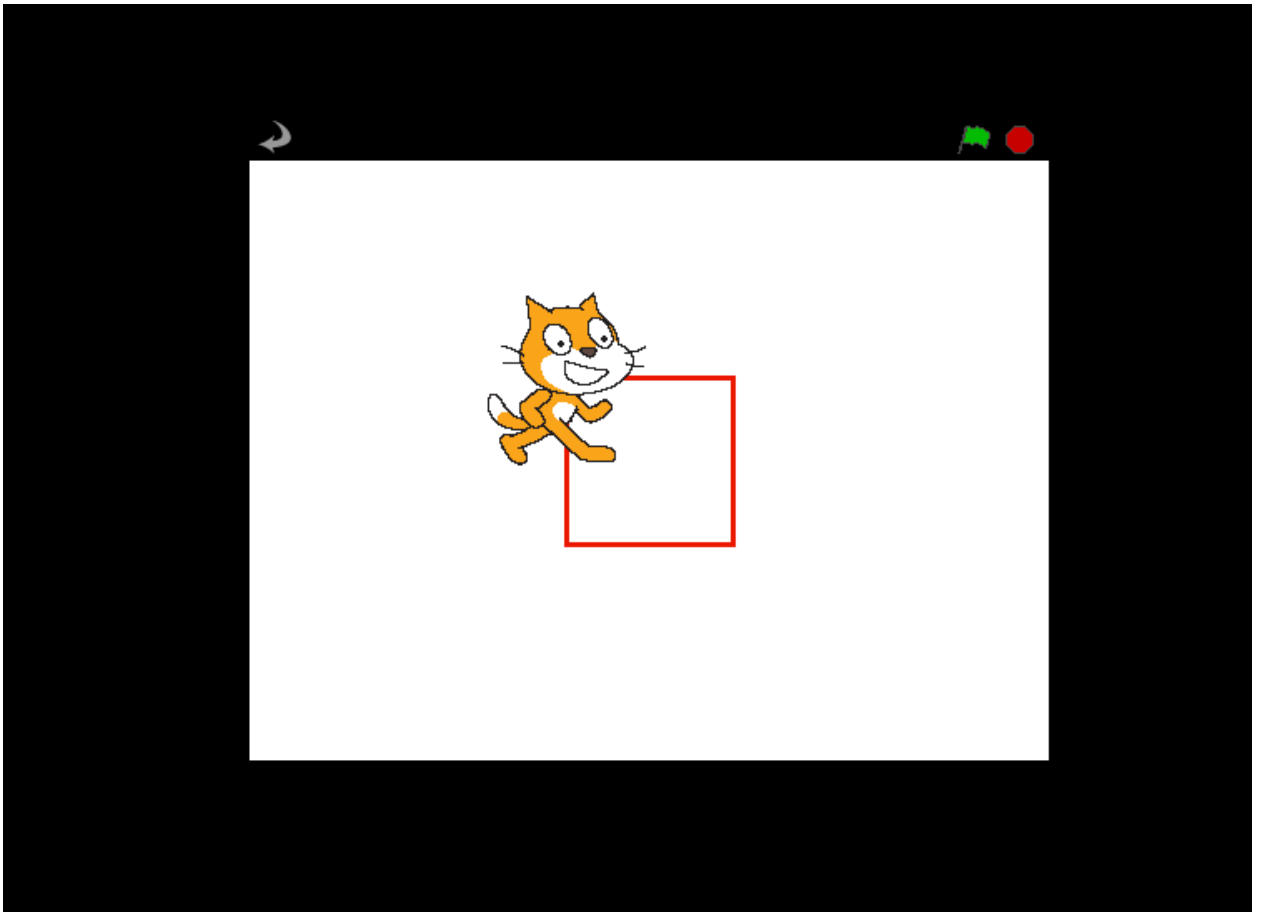






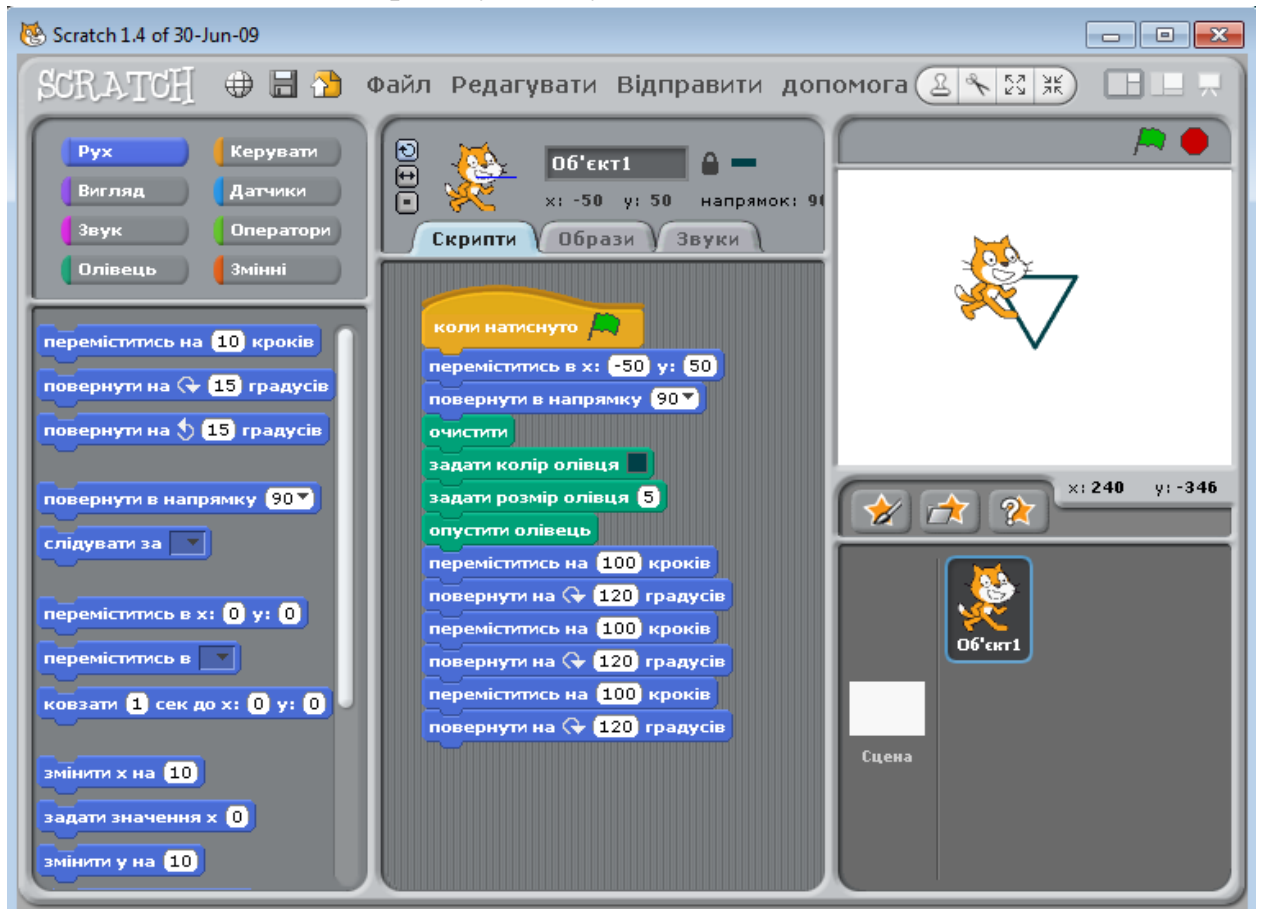


Запускаємо проект, натиснувши в правому верхньому кутку *режим перегляду*. Результатом виконання лінійного алгоритму буде:



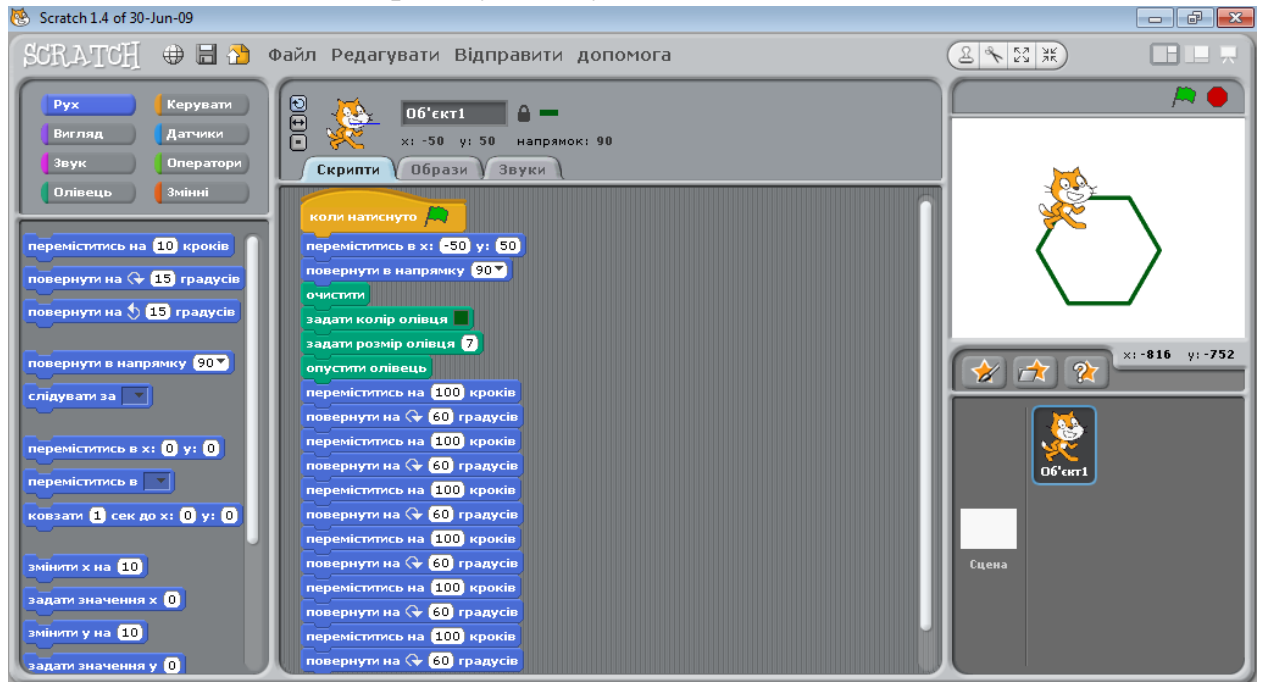
2. Створюємо геометричну фігуру "трикутник", використовуючи блоки команди руху, малювання та керування.

Загальний вигляд проекту має бути таким.



3. Створюємо геометричну фігуру "шестикутник", використовуючи блоки-команди руху, малювання та керування.

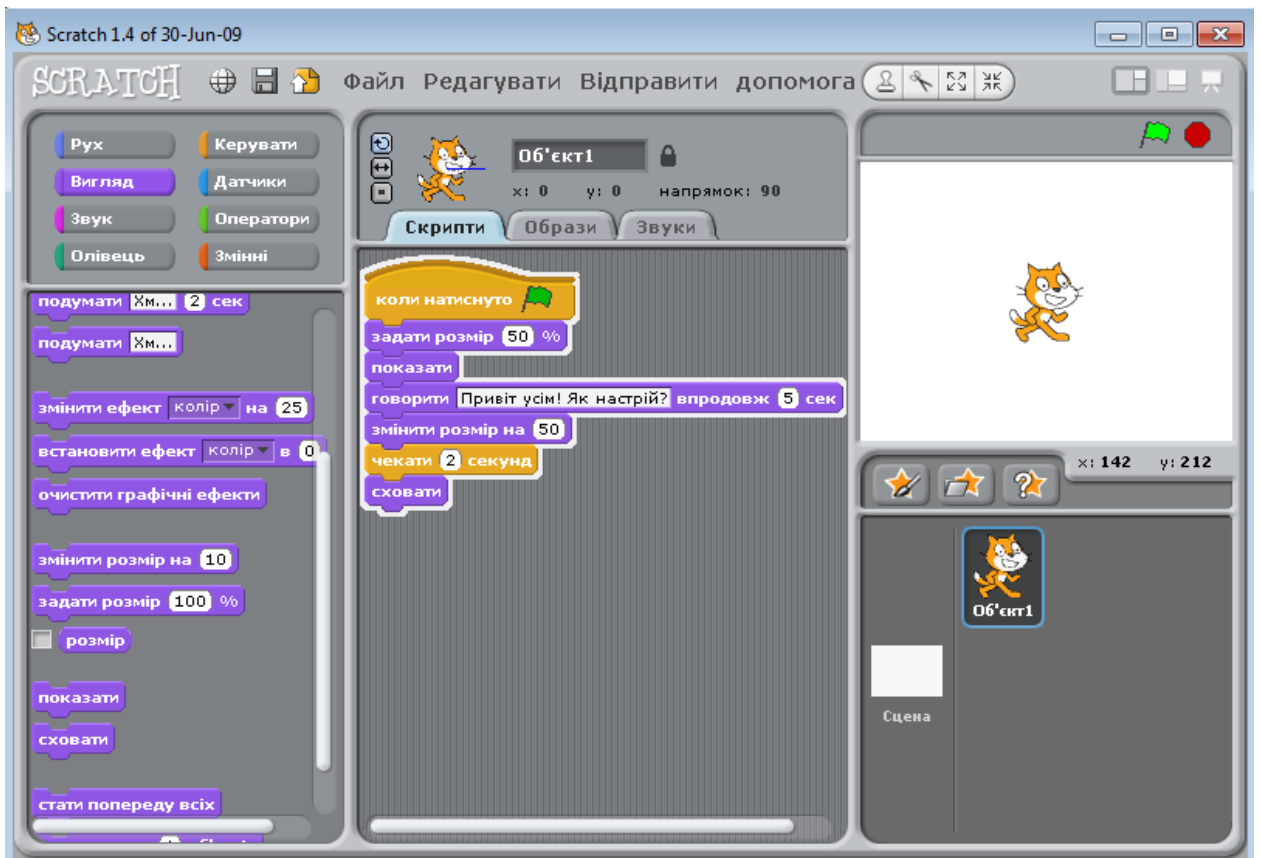
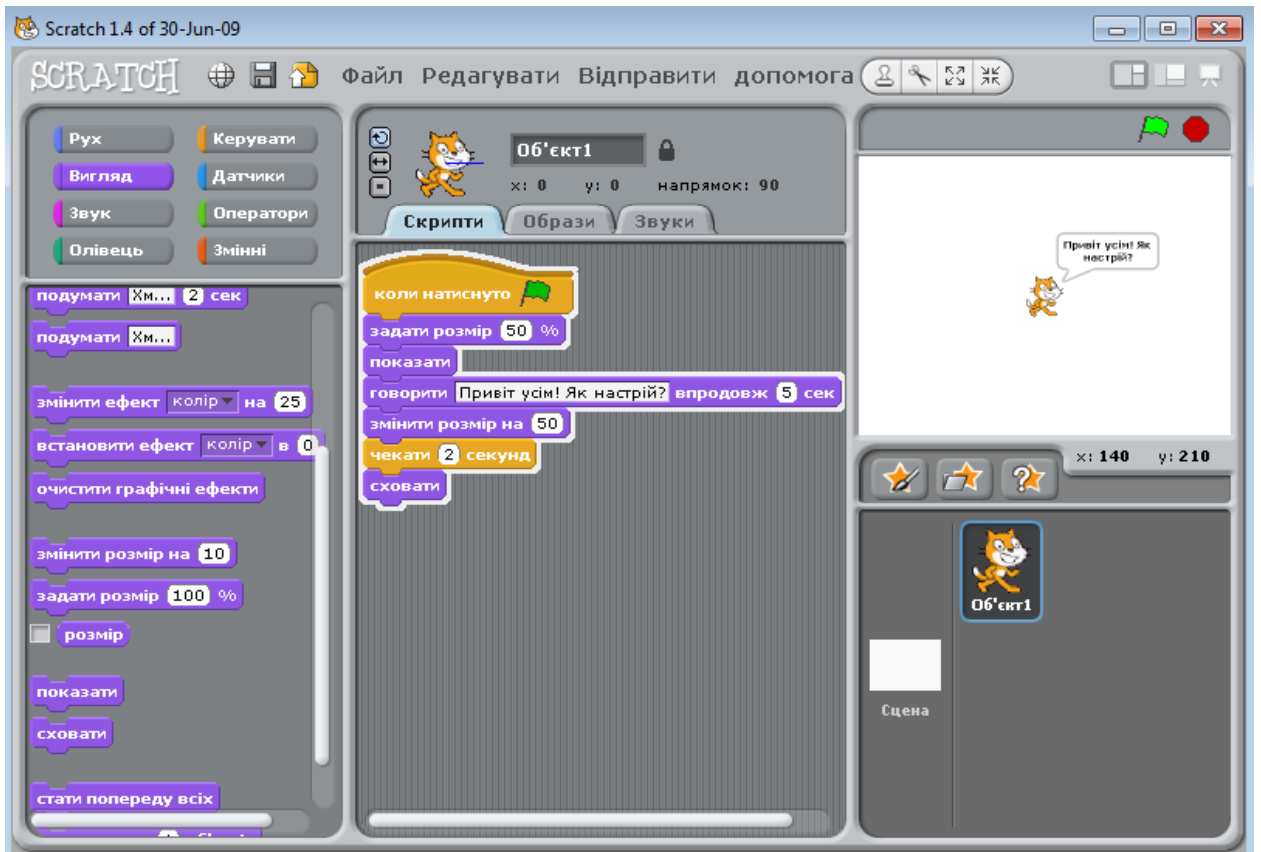
Загальний вигляд проєкту має бути таким.

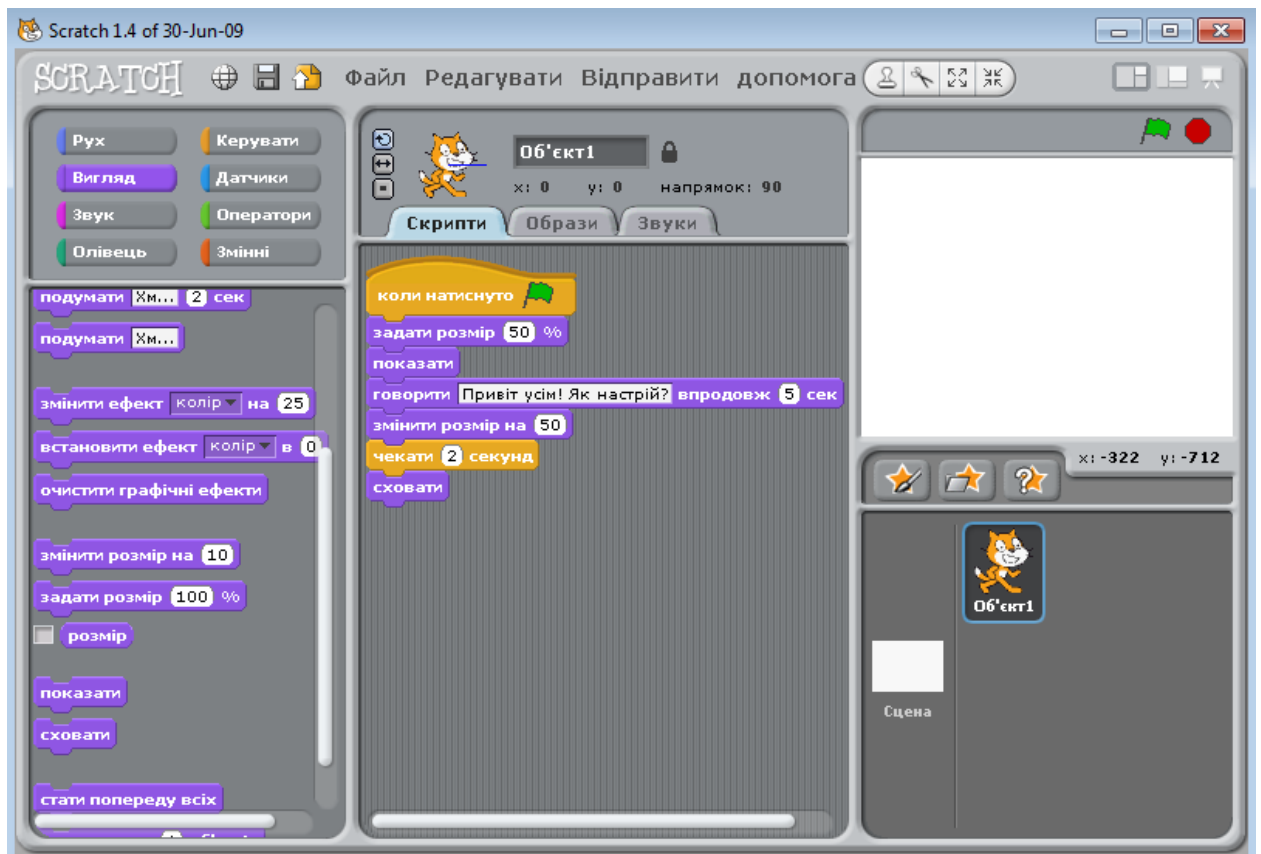


**Завдання 2 («Спрайт в шапці-невидимці»).** Щоб перевірити, як працюють блоки-команди *Показати* та *Сховати*, складіть у вікні нового проєкту скрипт за таким алгоритмом:

- запустити скрипт;
- встановити розмір спрайта у 50% скриптного розміру;
- показати спрайта на сцені;
- дати спрайту можливість сказати «Привіт усім! Як настрій?» впродовж 5 секунд;
- змінити розмір спрайта на 50;
- зробити затримку на 2 секунди;
- сховати спрайта.

Після запуску проєкту на виконання отримаєте такий результат:



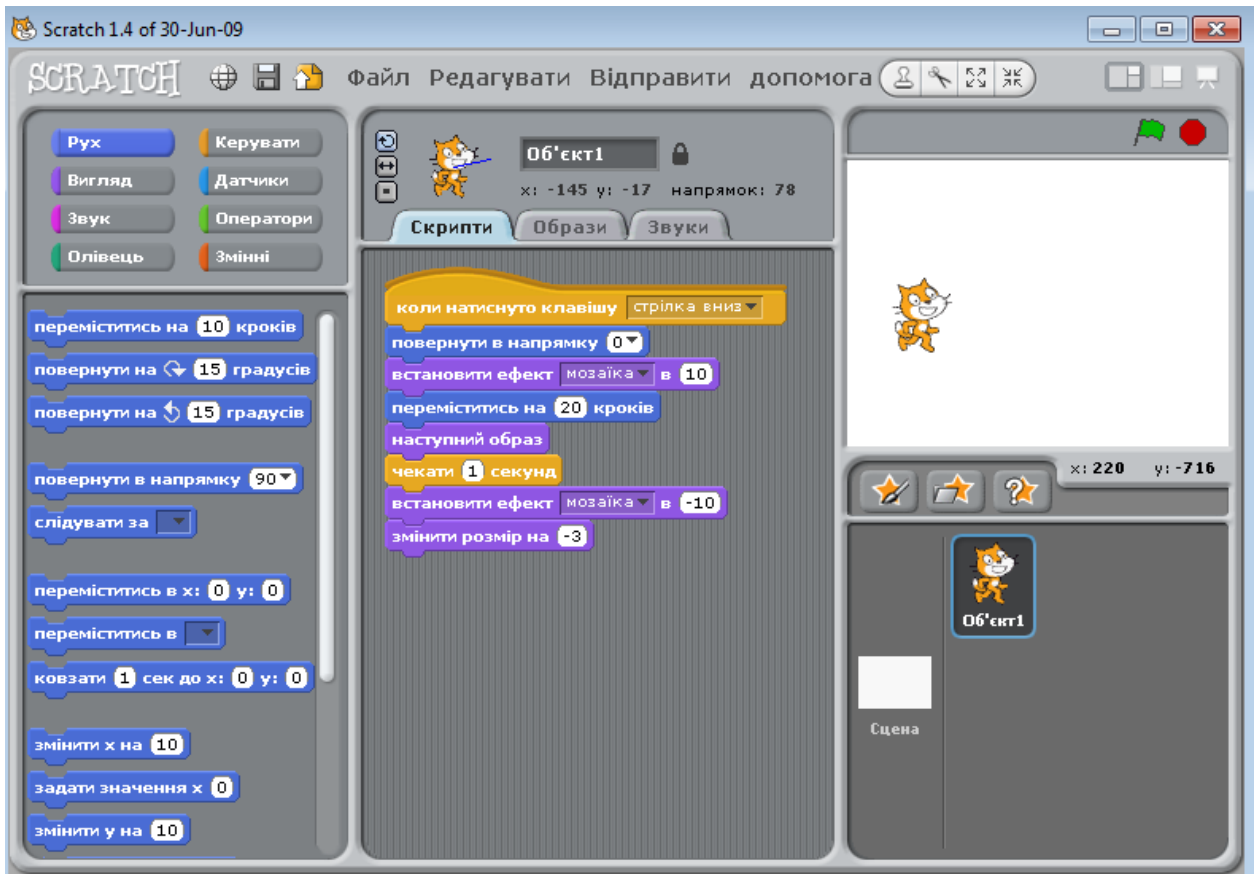


**Завдання 3 («Мультфільм»).** Створити анімацію спрайта, додавши графічний ефект *мозаїка*.

*Алгоритм сценарію:*

- користувач запускає скрипт, натиснувши на клавішу "стрілка вниз";
- повернути спрайт вгору;
- встановити візуальний ефект «мозаїка»;
- перемістити спрайт вперед на 10 кроків;
- спрайт на мить завмирає;
- зробити затримку на 1 секунду;
- встановити візуальний ефект «мозаїка»;
- зменшити розмір спрайта.

Результат виконання даного алгоритму:



## **6. Підбиття підсумків уроку.**

Аналіз результатів роботи учнів на уроці та виставлення оцінок.

## **7. Домашнє завдання.**

Повторити матеріал, вивчений на уроці. Створити проєкт, у якому б розміри одних спрайтів збільшувались, а інших, навпаки, зменшувались.