**Вступ**

Формування змісту української освіти ХХІ ст. спрямовано на розвиток в учнів цілісної картини світу. Реалізація цієї задачі тісно зв'язана із інтеграцією різних предметів в освітній процес. Це процес характеризується широким впровадженням соціально-комунікативних та інформаційних технологій. ІТ-технології знаходять своє застосування в різних предметних областях, на всіх вікових рівнях, допомагаючи кращому засвоєнню як окремих тем.

Вивчення основ програмування у шкільному курсі інформатики має велике методологічне значення. У зв’язку із цим знайомство із засадами алгоритмізації і програмування у молодшій школі заслуговує на окрему увагу. У педагогічній літературі відмічається, що курс інформатики у молодшій школі є пропедевтичним і має забезпечувати розвиток загально навчальних умінь та навичок, а нові поняття даного курсу повинні вводиться з опорою на знання, одержані при вивченні інших предметів. Цей курс повинен базуватися на науково обґрунтованій концепції вивчення інформатики в початковій школі, на основі якої визначається зміст навчання, розробляється навчальна програма, яка враховує вимоги дитячої психології та лікарів-гігієністів. Програма з інформатики побудована лінійно-концентрично, тож змістовна лінія «Алгоритми» розкривається поступово, у декілька етапів. Аналіз підручників інформатики для 2-4 класу, засвідчує, що навчальний матеріал, який розкриває цю змістову лінію, добре структуровано і належним чином ілюстровано, що полегшує учням зрозуміння і засвоєння. Наявні також запитання і завдання логічного характеру. Проте підручники містять недостатню кількість завдань практичного характеру, що передбачають розв’язання із використанням комп’ютера. Наявні в підручниках практичні завдання не дозволяють ефективно відпрацювати необхідні навички програмування. Відчувається нестача різнорівневих завдань та покрокових рекомендацій до їх виконання, які б підвищували мотивацію учнів до роботи з побудови алгоритмів та програм, спонукали б їх до активної пізнавальної діяльності. Велика частина знань, умінь і навичок, здобутих дітьми на традиційних уроках, не використовується ними в практичній діяльності. Тому їх практична цінність втрачається, а міцність – суттєво знижується. Використання знань, умінь і навичок в ігровому, комп’ютерному середовищі призводить до їх актуалізації, а бажання гратися – до мотивації засвоєння знань.

Педагогічні спостереження засвідчують також, що опанування основ програмування викликають в учнів молодшої школи значні утруднення. На їх вивчення відводиться за навчальною програмою 2-4 класу обмежений академічний час, що недостатньо для одержання уяви про особливості різних алгоритмів та формування стійких навичок їх програмної реалізації. Актуальним є поглиблення вивчення цього матеріалу та проведення пропедевтичної роботи щодо практичного застосування різноманітних алгоритмів при розв’язанні практично-значущих задач. У сучасній методиці навчання інформатики виділяють труднощі, пов’язані з формуванням чіткого уявлення про алгоритм: – в уяві дитини є цілісний образ розв’язання завдання, а вимагається дати чіткий опис цього процесу у формі послідовності спонукальних речень. – у задачах такого типу важко виділити "елементарні" дії, з яких складатиметься алгоритм. – при виконанні алгоритмів, виконавець не повинен демонструвати елементи творчості – лише чітко виконувати команди у вказаному порядку. Для їх подолання методисти пропонують використання рольового методу навчання. Для цього виділяються ролі пояснюючого та виконавця алгоритму. Пояснюючий складає алгоритм, зачитує команди, виконавець програє накази. При неможливості виконання якоїсь команди, вона уточнюється чи деталізується до тих пір, поки не стане зрозумілою і виконуваною – "елементарною". Щоб виробити навички формального виконання алгоритму (виконавець не повинен розуміти суть проблеми чи завдання, намагатися додати власні дії, яких не має у записі алгоритму), вчителі-практики використовують прийом складання алгоритмів для виконавця-робота (програвання ролі робота, згодом перенесення його основних властивостей і на саме поняття виконавця).

Напередодні вивчення алгоритмів із розгалуженням школярів знайомлять з елементами математичної логіки: в учнів формують уявлення про висловлювання, з’ясовують, що висловлювання можуть бути істинними або хибними, навчають аналізувати висловлювання та розв’язувати найпростіші логічні задачі. При формуванні поняття алгоритму з розгалуженням, учитель повинен звертатися до життєвого досвіду дітей, досить вдало ілюструють алгоритми з розгалуженням казки. Спочатку вводяться елементи блок-схем у вигляді прямокутників, всередині яких записують команди. Далі пропонуються елементи у вигляді овалів для позначення початку та кінця алгоритмів.

Пізніше при знайомстві з циклічними алгоритмами вводиться елемент ромб для позначення команди перевірки умови. Завдання із складання алгоритму реалізується через побудову ланцюжка моделей задачі: текст задачі – стислий словесний опис алгоритму – блок-схема – програма для конкретного виконавця. Для здійснення цих модельних переходів школярам необхідно навчитися виділяти основні змістові частини задачі; виявляти зв'язки між ними; складати модель задачі, що має розв'язок.

Після опрацювання лінійних алгоритмів вводиться поняття циклу – частини алгоритму, яка повторюється кілька разів. Учні розглядають циклічні алгоритми, складають для них блок-схеми. Команда перевірки умови в циклах має форму запитання, відповідь на яке може бути "так" чи "ні", наприклад: "Малюнки залишилися?", "Кошик повний?" тощо. Учням можна запропонувати розробити алгоритми для виконання різних повторюваних побутових дій з використання циклічних структур: миття посуду, виготовлення паперових гірлянд для новорічної ялинки та ін.

Ознайомлення з кожною алгоритмічною структурою здійснюється за схожою схемою: наведення прикладів алгоритмів відповідної структури, з’ясування ключових слів для запису структури та їх позначень на блок-схемах, реалізація системи вправ на виконання, відшукування та виправлення помилок, конструювання та видозміни алгоритмів. У ході виконання вправ учні вчаться аналізувати ситуації, синтезувати, порівнювати та оцінювати результати. Формується алгоритмічне та структурно-логічне мислення школярів.

Викладене вище висуває необхідність розробки комплекту практичних завдань, що дозволять подолати окреслені проблеми.

Мета роботи – представити комплект практичних завдань для навчання молодших школярів алгоритмізації у середовищі Scratch та інтеграції між предметних зв’язків. Розроблений комплект охоплює низку різнорівневих завдань, які дозволяють учням усвідомити сутність побудови різновидів алгоритму, розвине використання добутих знань з математики та творчої уяви.

Відповідно до мети, сформулюємо такі завдання роботи:

1. Створити комплект практичних завдань для навчання учнів 2-4 класів основ програмування у середовищі Scratch.

2. Надати методичні рекомендації щодо використання розробленого комплекту на уроках інформатики у 2-4 класах.

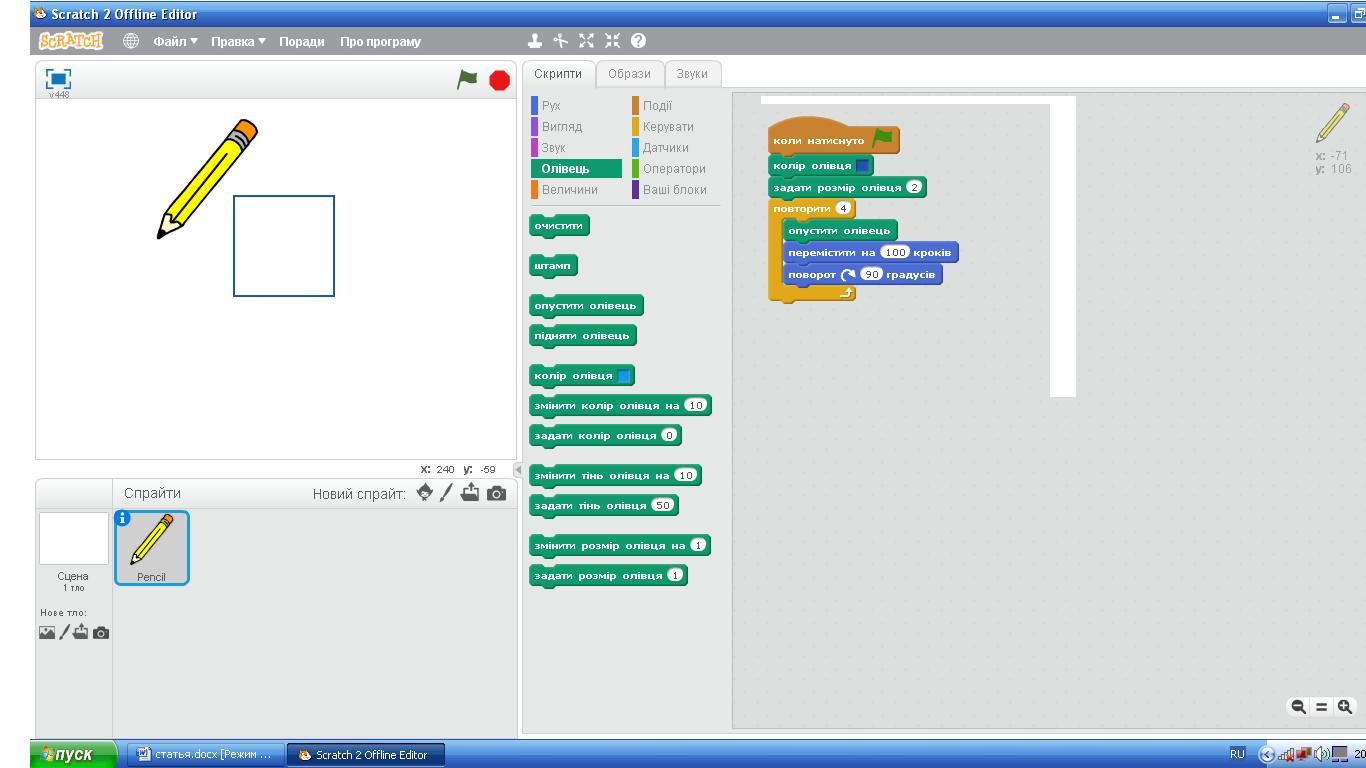
### При аналізі умови кожного із завдань комплекту вчитель може поставити учням запитання:

1. Які дії будуть повторюватись при виконанні завдання?
2. Скільки разів їх повторити, щоб одержати певну фігуру або малюнок?
3. За допомогою якої команди з блоку «Керування» можна реалізувати малювання пропонованої геометричної фігури (квадрата, кола, п’ятикутника)? Який параметр треба замінити і як?

Далі учні мають сформувати скрипт, реалізувати його у середовищі Scratch, проаналізувати результат, виправити помилки, якщо потрібно.

**комплект практичних завдань по створенню алгоритмів у середовищі Scratch з інтегративним піходом**

**Завдання 1. «Квадрат».** Намалювати квадрат із стороною 100 кроків (Рис. 1.1-1.2).



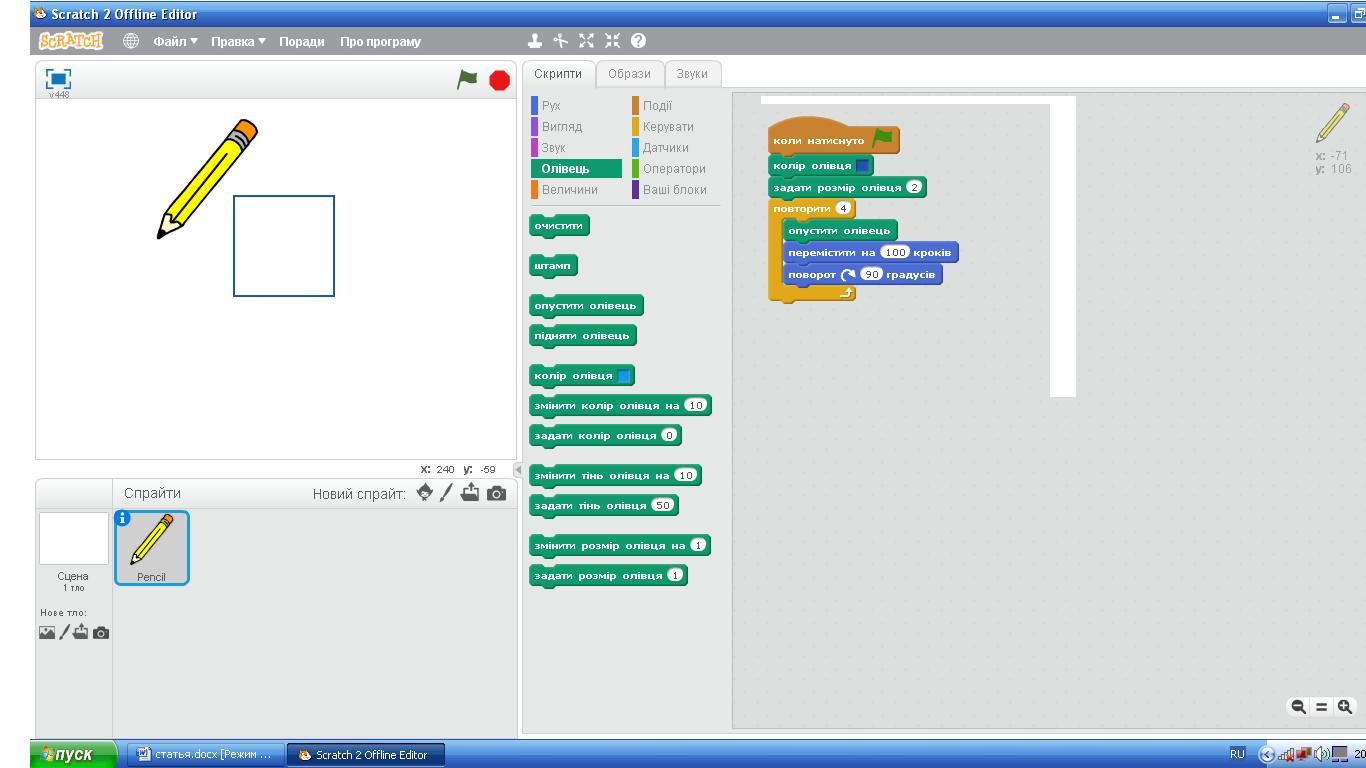
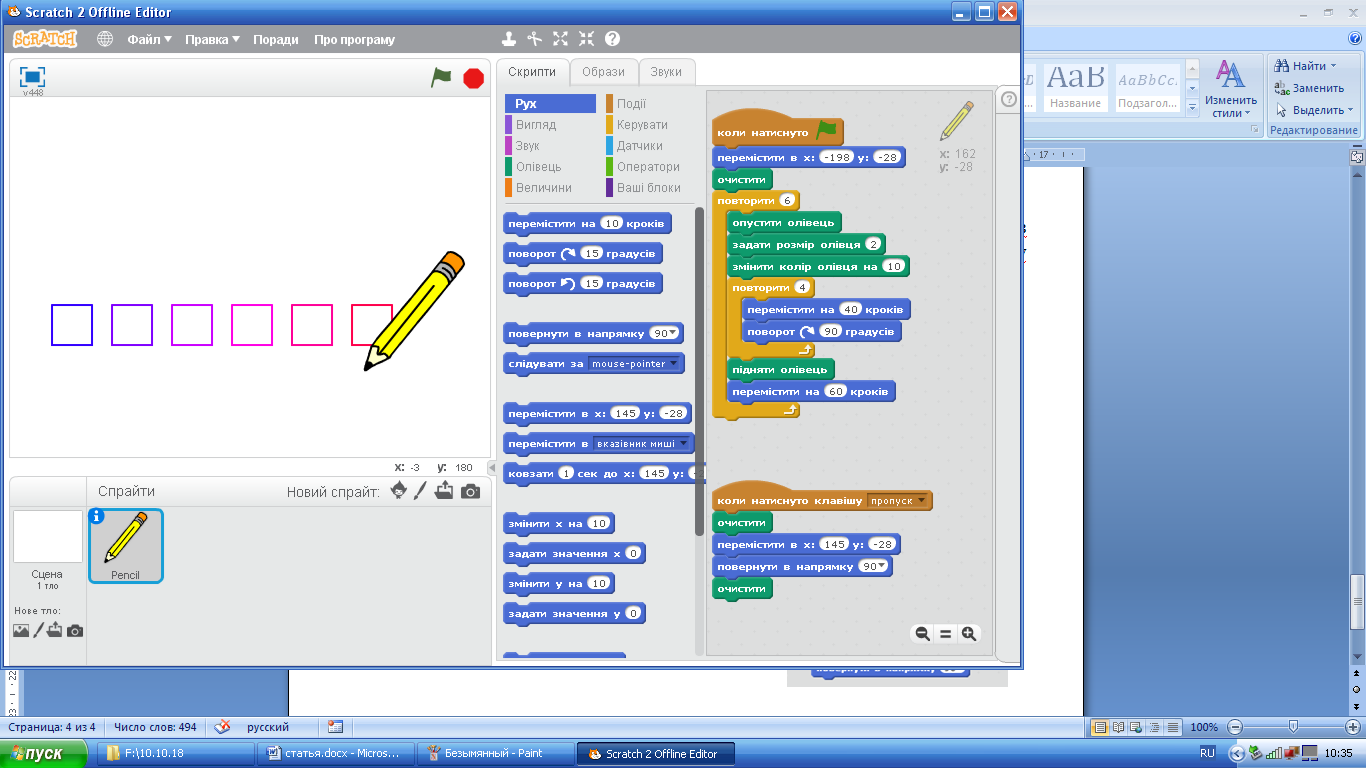


Рис.1.1. Малюнок до завдання 1 Рис.1.2. Розв’язок завдання 1 у Scratch

### **Завдання 2. «Квадрати»**

В цьому завданні потрібно намалювати шість однакових квадратів з довжиною сторони 40 кроків, відстань між квадратами 20 кроків (Рис. 2.1-2.2). При цьому олівець малює кожен квадрат різного кольору. При розв’язанні завдання слід сфокусувати учнів на зв'язок цієї задачі із завданням 1 та підвести їх до розуміння вкладених циклів. Доцільно обговорити такі питання.

1. Яку фігуру слід намалювати першою?
2. За допомогою якої команди з блоку «Керування» можна реалізувати малювання квадрату?
3. Скільки разів малювання квадрату повториться (див. умову завдання)?
4. Скільки вкладених циклів ми отримали?
5. Складіть скрипт та реалізуйте проект.

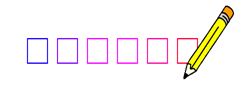


Рис. 2.1. Малюнок до завдання 2 Рис.2.2. Розв’язок завдання №2 у Scratch

**Завдання 3. «П’ятикутник»**

Намалюйте геометричну фігуру з п’ятьма кутами. Кожна сторона має дорівнювати 50 кроків, сума усіх кутів дорівнює 360 градусів. (Рис. 3.1-3.2)

Потрібно націлити школярів на зв'язок цієї задачі із завданням 1. Доречними будуть такі запитання:

1. За допомогою якої команди з блоку «Керування» можна реалізувати малювання п’ятикутнику?
2. Скільки дорівнює один із п’яти кутів цієї фігури (див. умову завдання)?
3. Скільки повторень необхідно зробити?

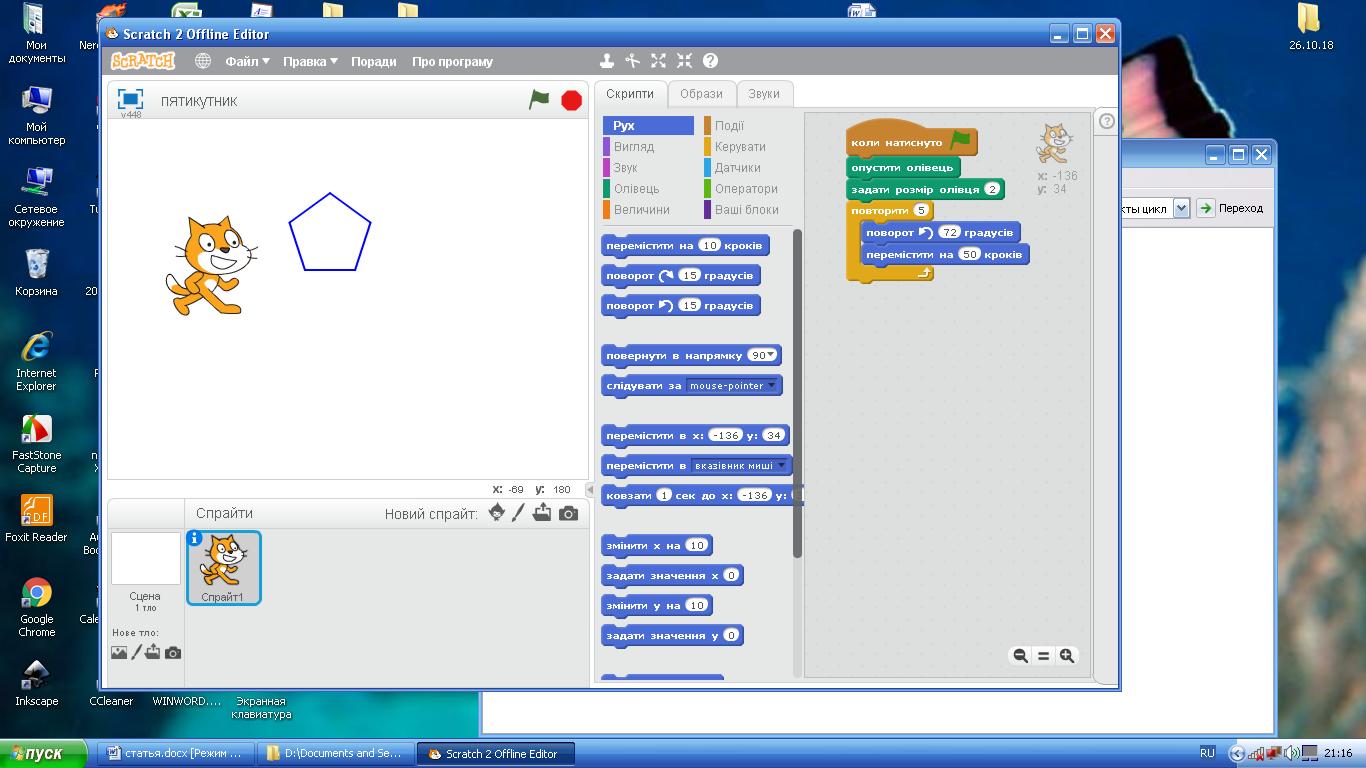
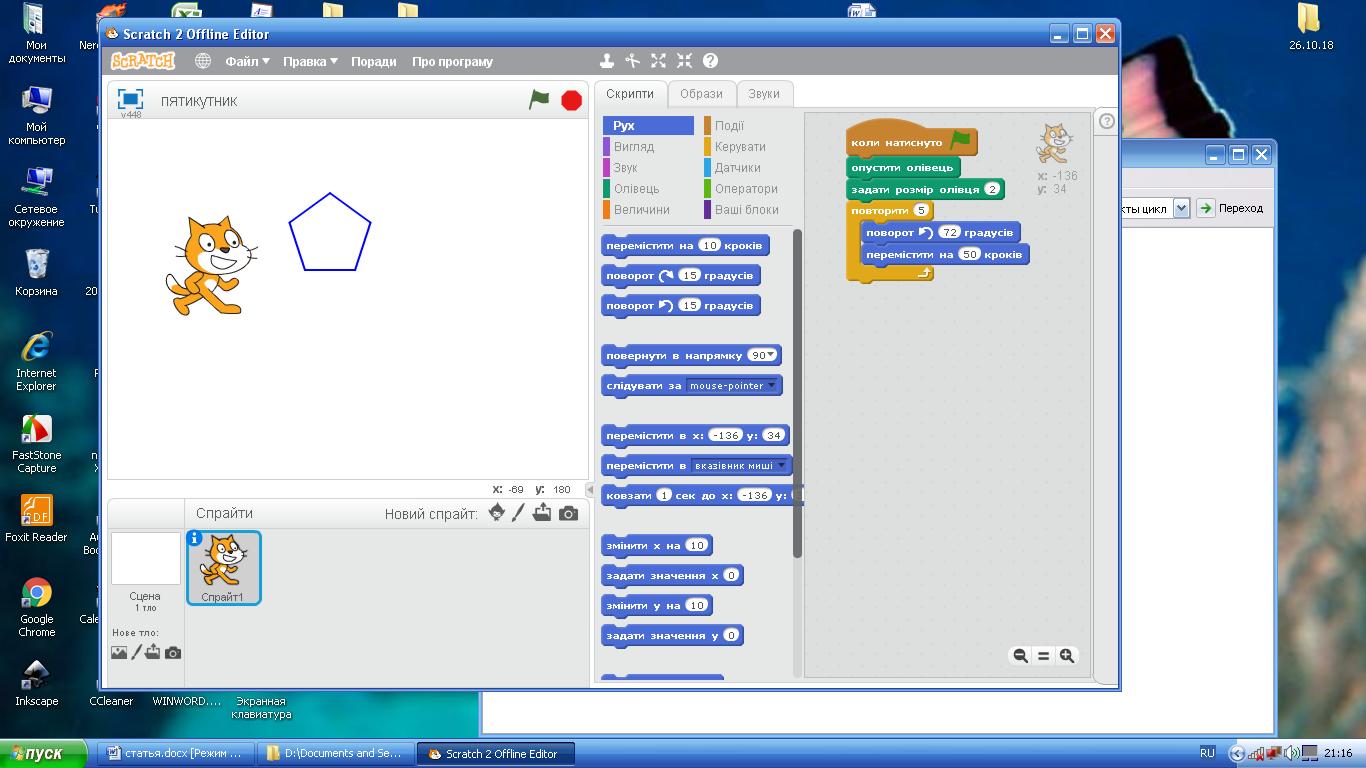
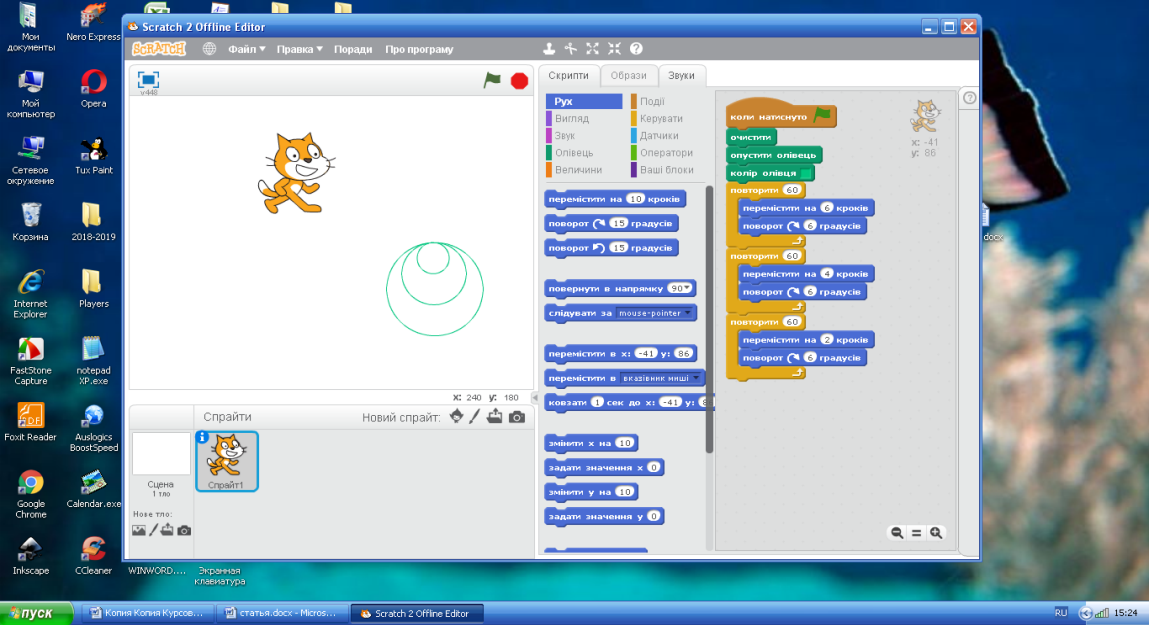


Рис. 3.1 Малюнок до завдання 3 Рис. 3.2. Розв’язок завдання 3 у Scratch

### **Завдання 4. «Вкладені кола»**

Намалюйте три вкладених кола різних діаметрів зеленого кольору (Рис. 4.1-4.2).

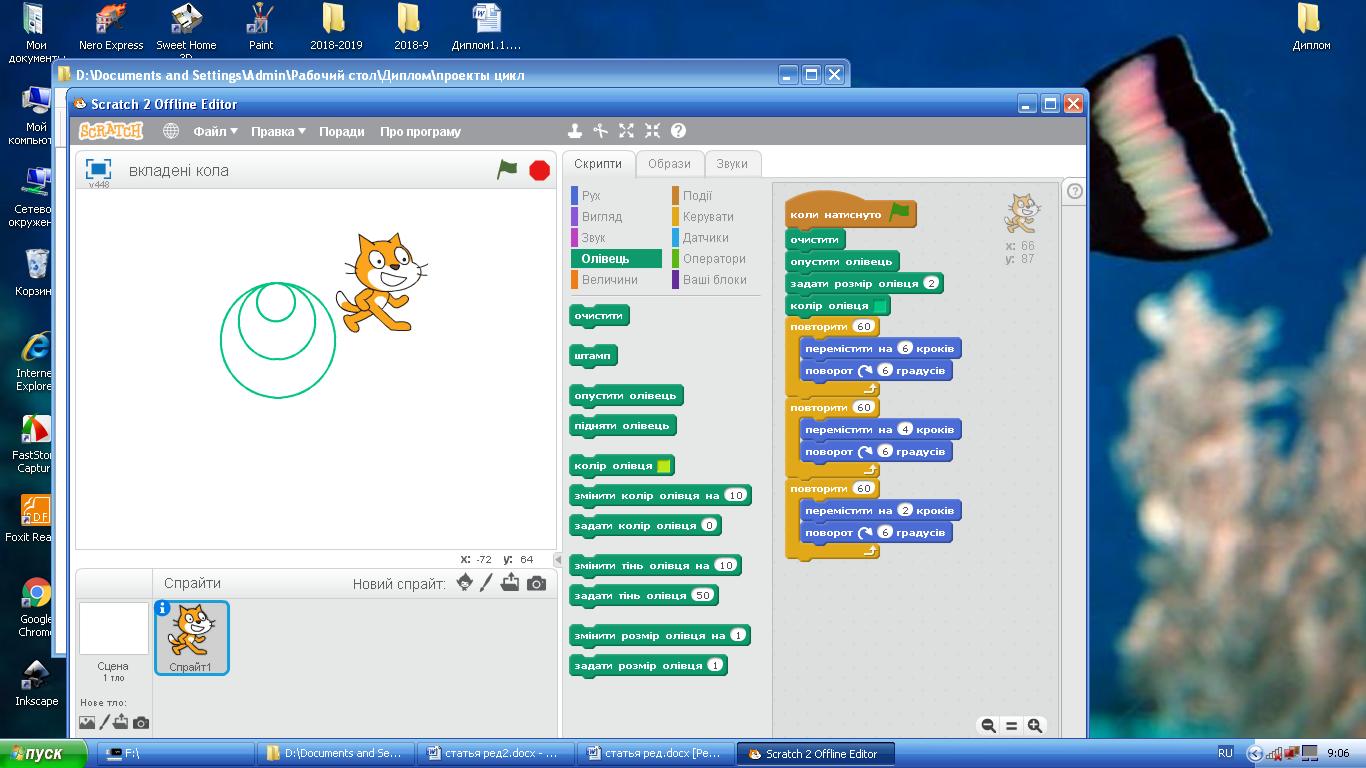
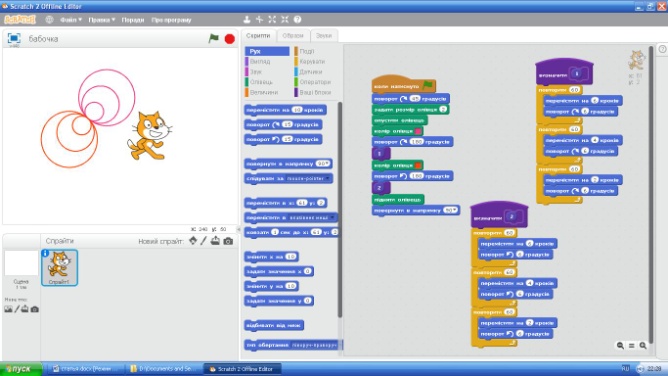
****

Рис. 4.1 Малюнок до завдання 4 Рис. 4.2. Розв’язок завдання 4 у Scratch

**Завдання 5. «Метелик з вкладених кіл»**

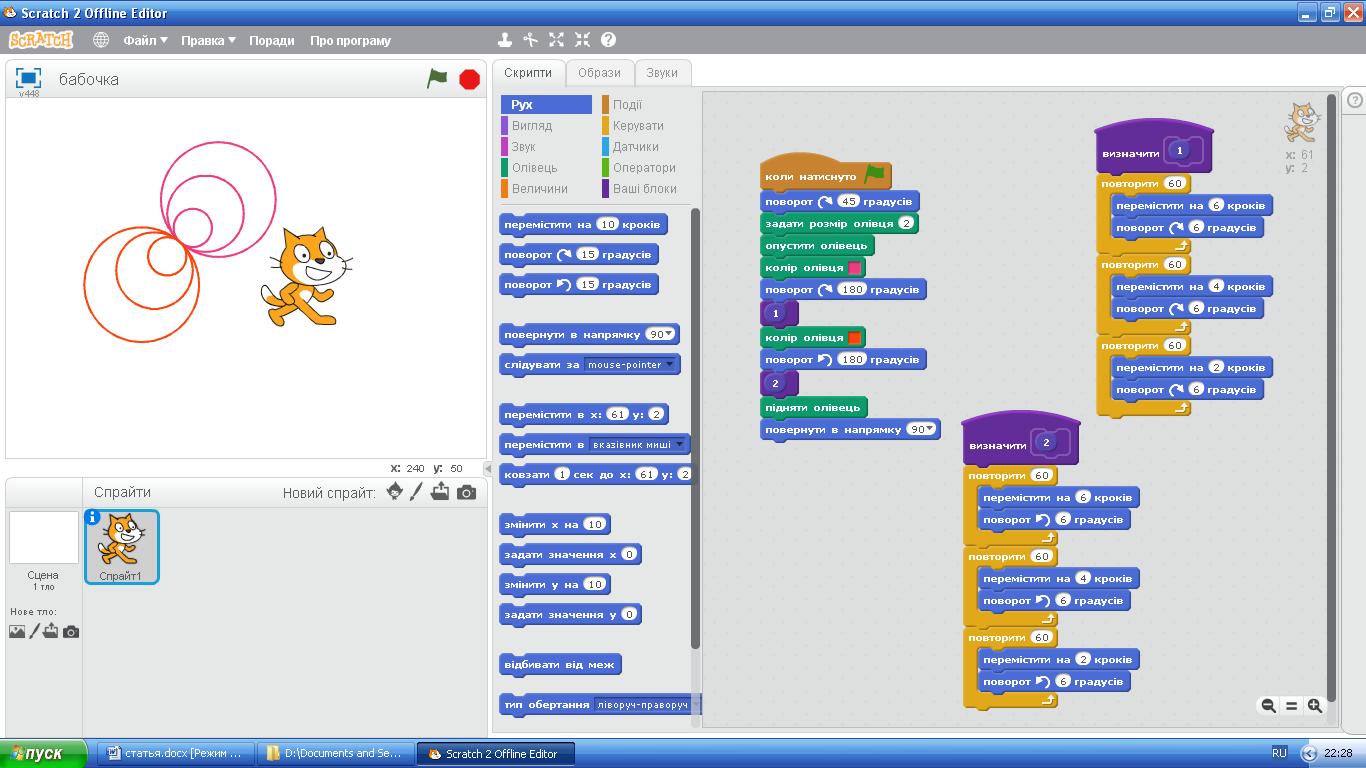


Рис. 5.1.Малюнок до завдання 5 Рис. 5.2**.** Розв’язок завдання 5 у Scratch

**Завдання 6. «24 квадрати»**

Намалюйте 24 квадрати із стороною у 40 кроків. При малюванні кожного наступного квадрата потрібно повернути вправо на 180 градусів (Рис.6.1-6.2).

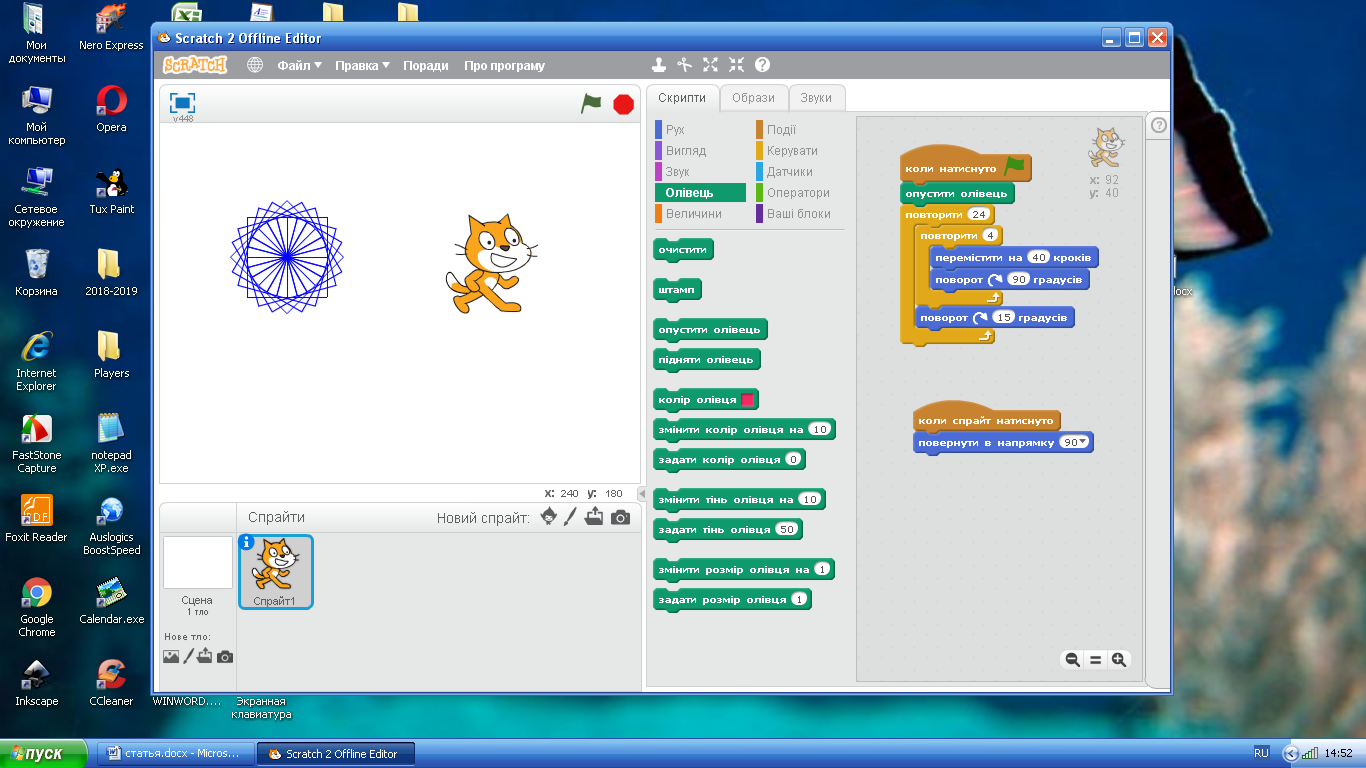
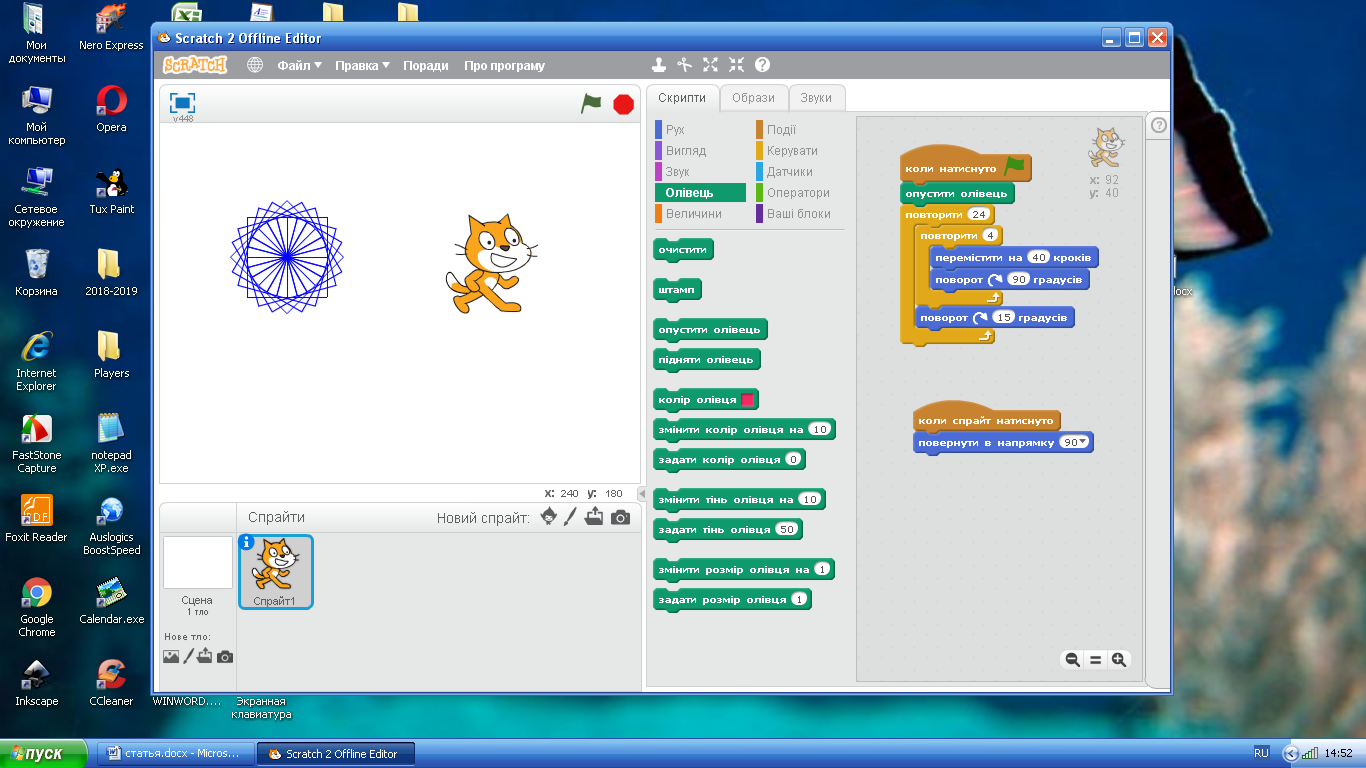
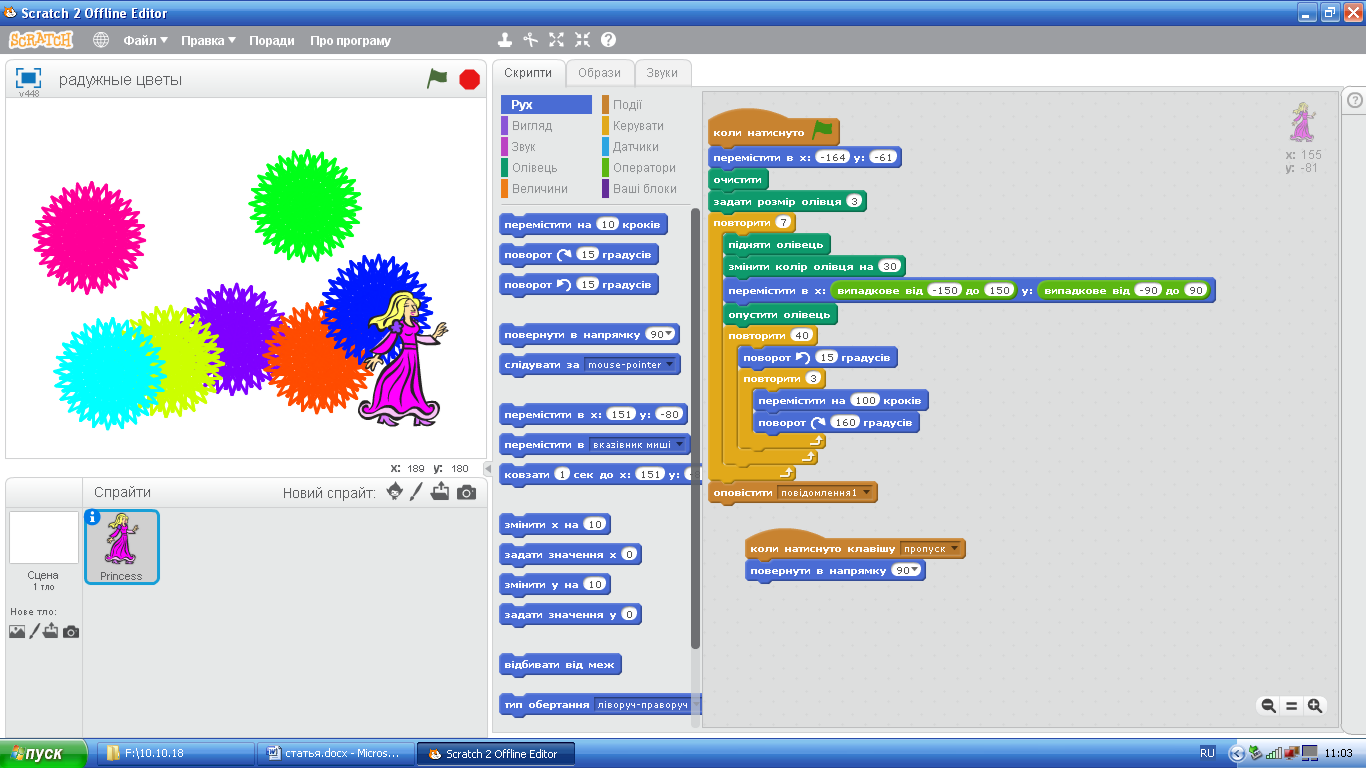


Рис.6.1. Малюнок до завдання 6 Рис.6.2. Розв’язок завдання 6 у Scratch

### **Завдання 7. «Сім квіток»**

Намалюйте квіти семи кольорів райдуги, використовуючи необхідну кількість циклів з повторенням.

При розв’язанні задачі слід наголосити учням, що необхідно спочатку скласти скрипт для побудови одної квітки, а потім застосувати вкладений цикл, щоб намалювати потрібну кількість квіток (Рис.7.1-7.3).



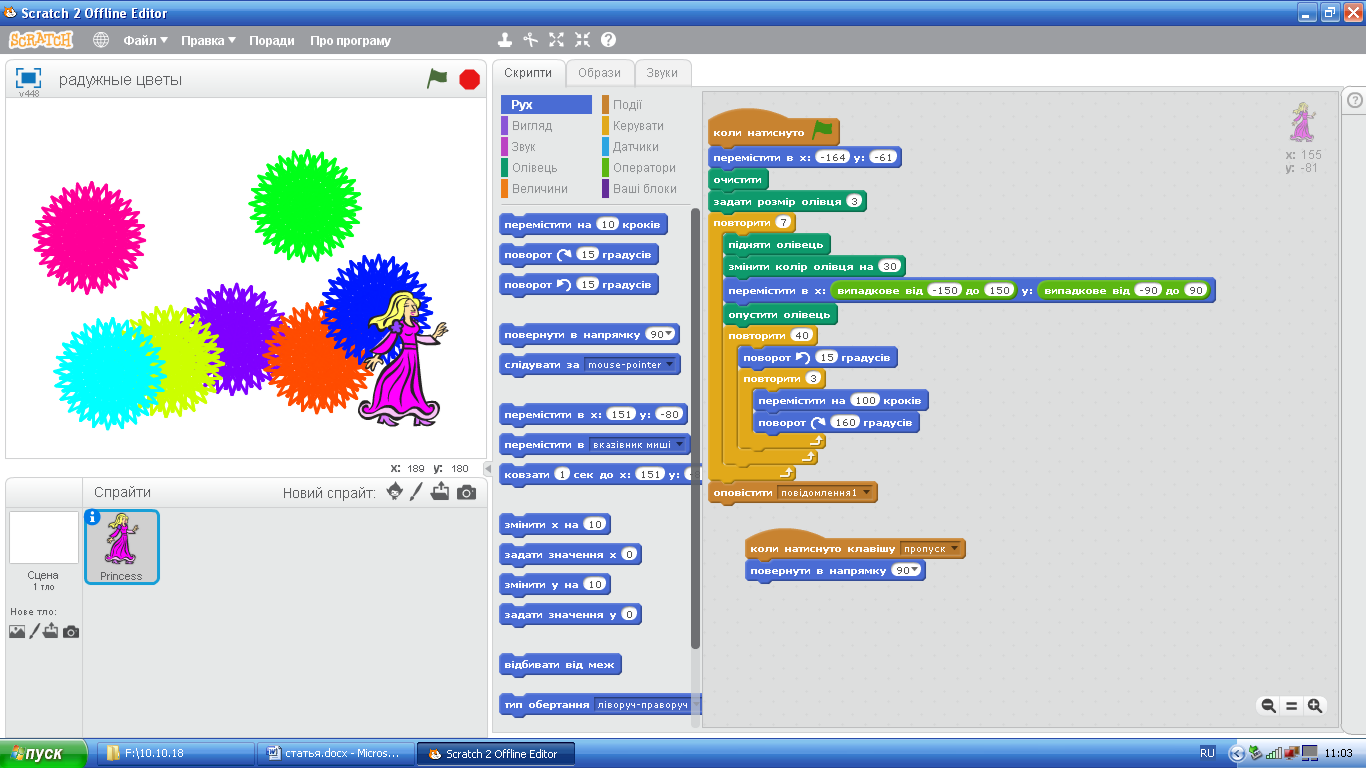
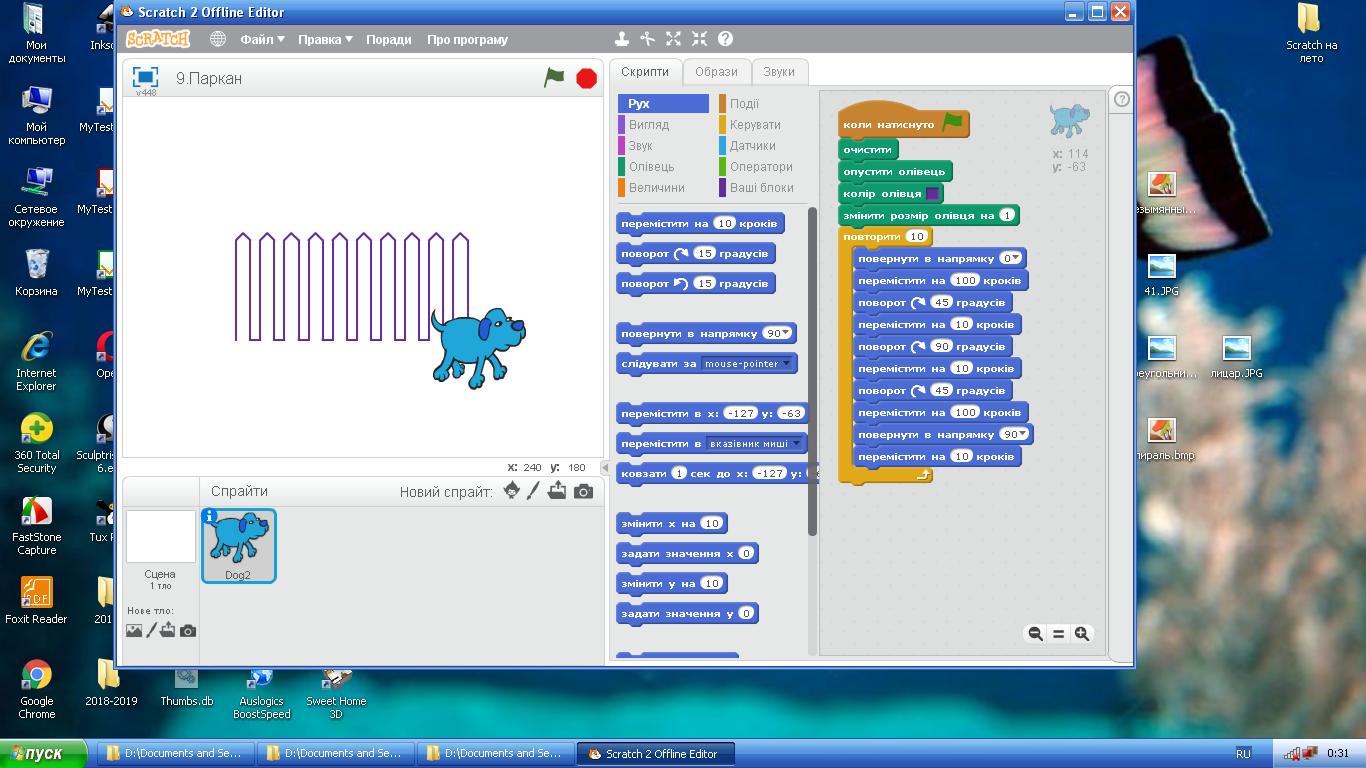
****

Рис. 7.1. Малюнок до завдання 7 Рис. 7.2 Скрипт для створення однієї квітки

Зауважимо, що виконання цих та подібних завдань не лише вчить застосовувати циклічні алгоритми, а й закріплює знання з геометрії.

**Завдання 8. «Паркан»**

З 10 узьких довгих дощечок (100 кроків) намалювати паркан.



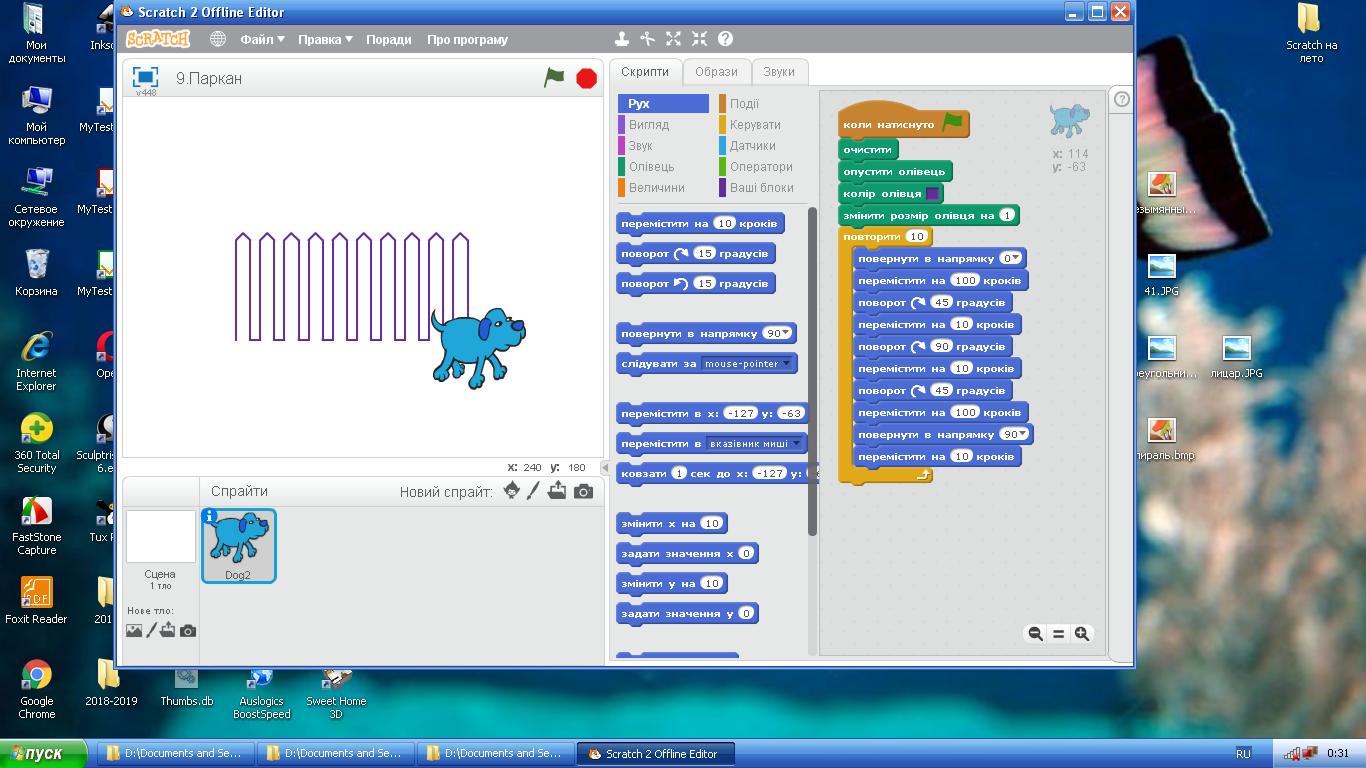
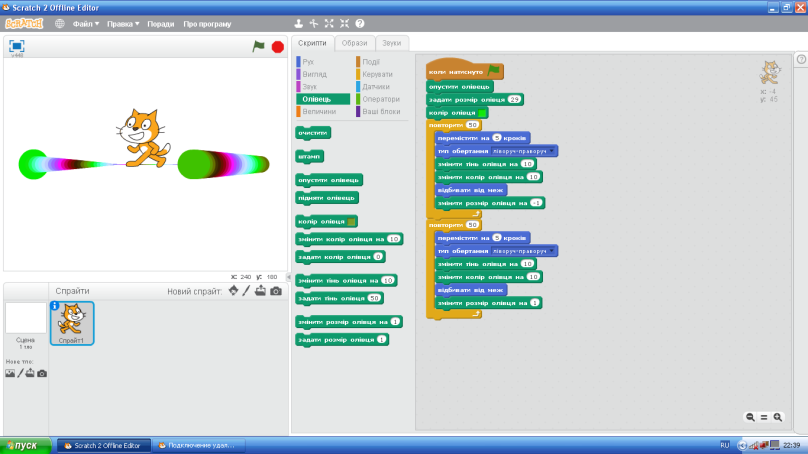


Рис.8.1. Малюнок до завдання 8 Рис.8.2. Розв’язок завдання 8 у Scratch

Виконуйте таку послідовність роботи:

1. використайте команди «коли натиснуто зелений прапорець»;
2. оберіть спрайт Пес;
3. використовуйте блоки команд «Керування» та «Олівець»;
4. введіть команду «повторити», використайте команду «поворот на ліво» та «поворот на право»;
5. оберіть потрібний колір паркана;
6. сформуйте такий скрипт та встановіть на виконання.

**Завдання 9. «Орнамент»** (Рис.9.1-9.2)

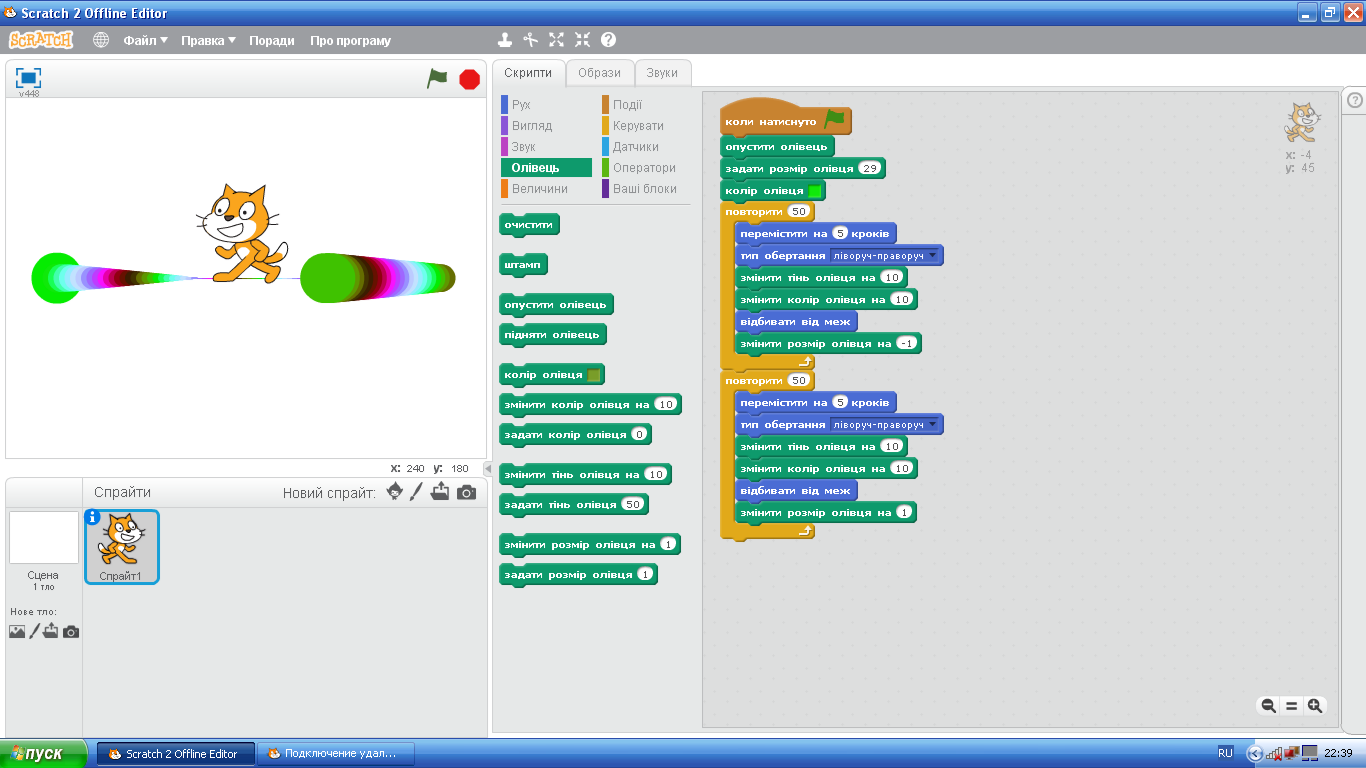
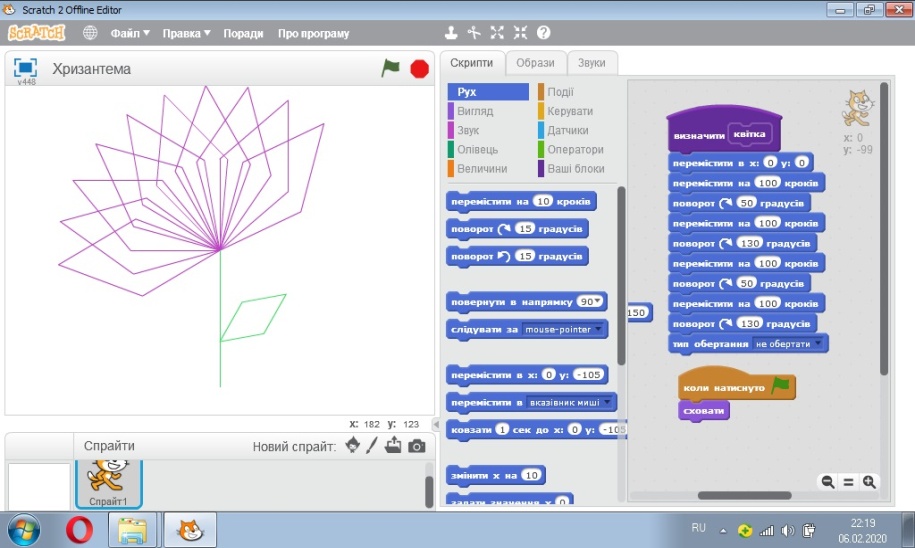


Рис.9.1. Малюнок до завдання 9 Рис.9.2. Розв’язок завдання 9 у Scratch

Виконуйте таку послідовність роботи:

1. використайте команди «підняти олівець» та «опустити олівець»;
2. оберіть колір та відтінок для малювання;
3. введіть команду «повторити». використайте команду «очистити»;
4. використайте такі команди «змінити тінь олівця на 10», «тип обертання ліворуч-праворуч»;
5. сформуйте такий скрипт та встанови на виконання;
6. змініть кількість повторень та зберіжіть новий проект.

**Завдання 10. «Хрізантема»**

Намалювати хрізантему з ромбовидними пелюстками та з зеленим листочком.

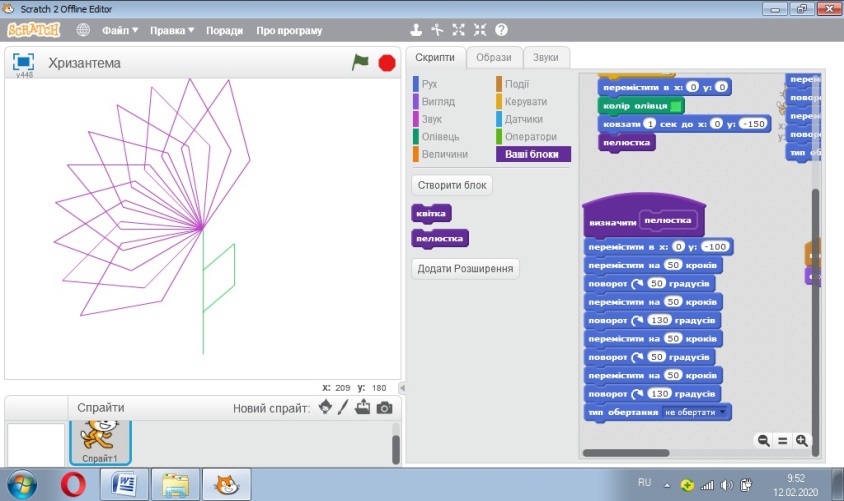


Рис.10.1. Малюнок до завдання 10 Рис.10.2. Розв’язок завдання 10 у Scratch

Виконуйте таку послідовність роботи:

1. використайте команди «коли натиснуто зелений прапорець»;
2. оберіть спрайт «Олівець» та введіть команду заховати;
3. використовуйте блоки команд «Керування» та «Олівець»;
4. введіть команду «повторити», використайте команду «поворот на право»;
5. оберіть потрібні кольори пелюстки та для листочка;
6. сформуйте такий скрипт та встановіть на виконання.

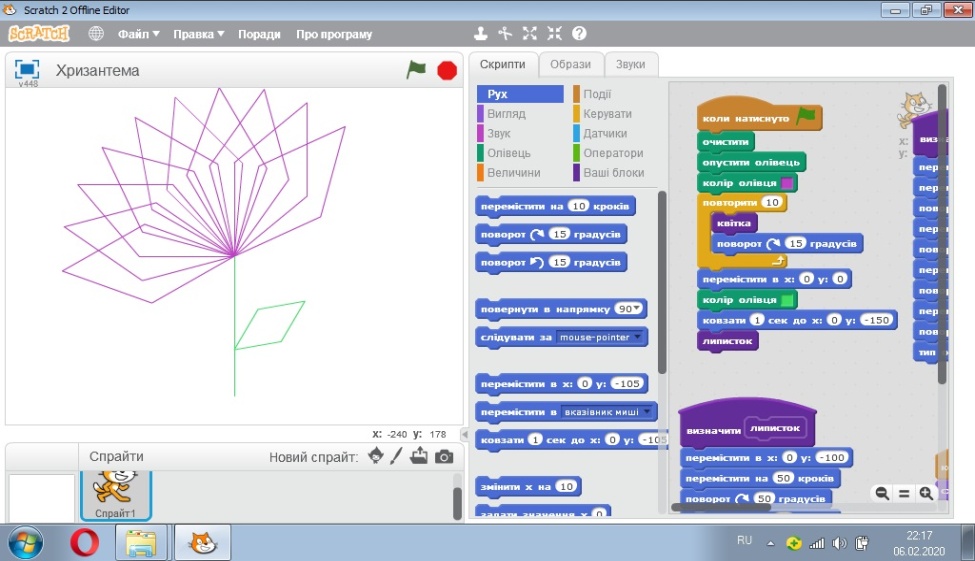


Рис.10.3. Розв’язок завдання 10 у Scratch (друга частина)

**Завдання 11. «Незвичайне малювання»**

Намалюйте візерунок, яка складається з певної кількості відрізків, розташованих під кутом 850 один від одного. Перед виконанням обговоріть наступні питання:

1. Як змінити кількість повторень циклу?
2. Як створити змінну?
3. За допомогою якої команди можна змінювати відтінок малювання?

Виконуйте таку послідовність роботи:

1. створіть будь-який спрайт;
2. зібріть наведений скрипт (Рис.11.2);
3. запустити проект.

Внесіть у проект такі зміни:

1. змініть назву змінної;
2. змініть кількість заданих повторень;
3. додайте блок зміни відтінку олівця;
4. додайте команду «штамп»;
5. проаналізуйте отримані зміни, зробіть висновки та збережіть новий проект.

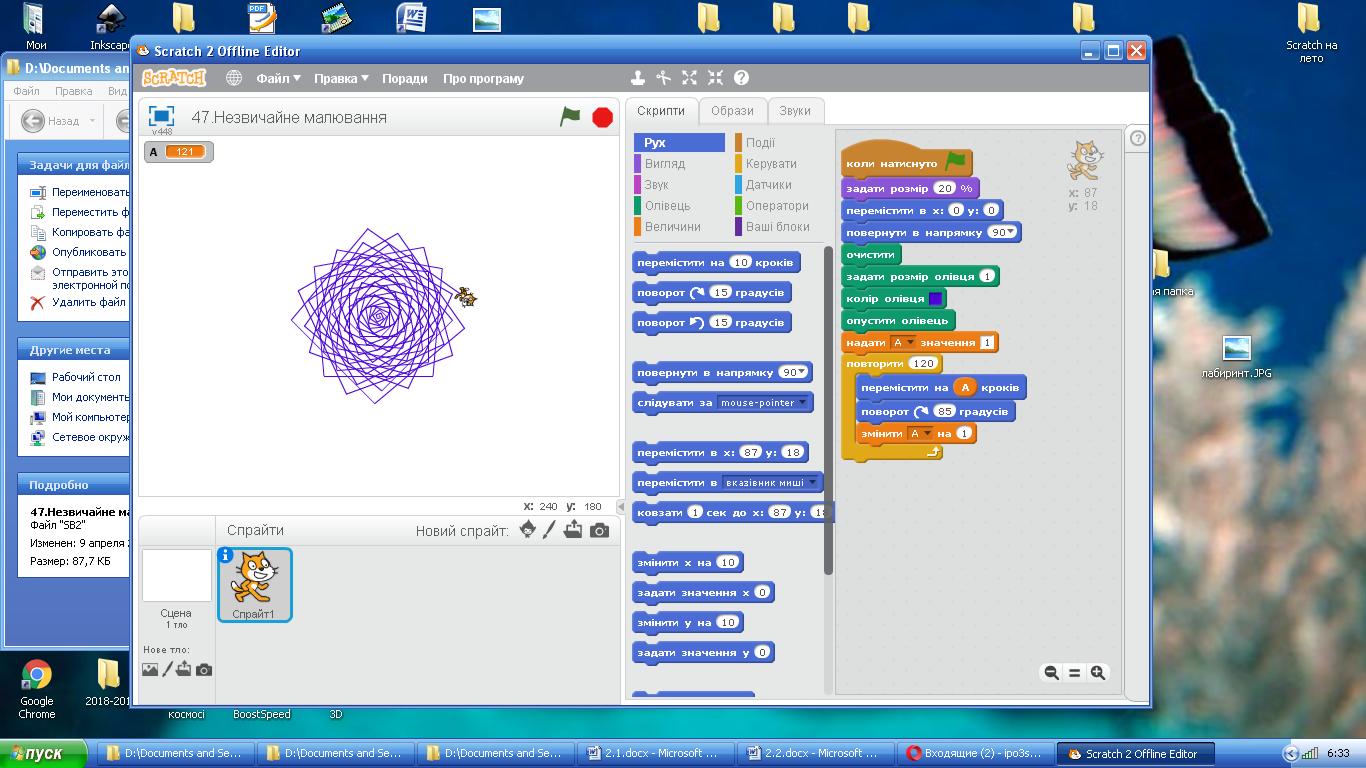
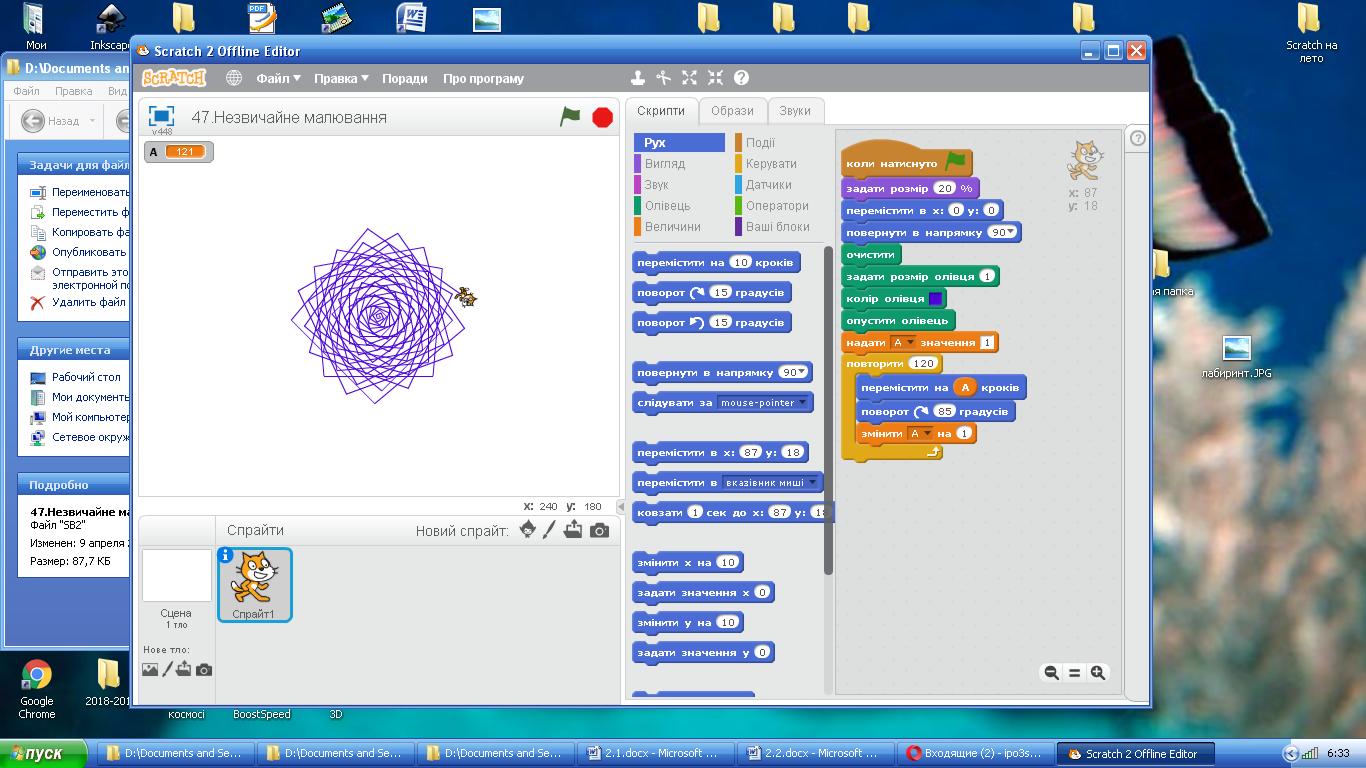


Рис.11.1 Малюнок до завдання 11 Рис.11.2 Скрипт до завдання 11

**Завдання 12. «Райдужна зірка»**

Намалюйте візерунок, яка складається з певної кількості відрізків, розташованих під кутом 142380. Перед виконанням обговоріть наступні питання:

1. Як змінити кількість повторень циклу?
2. Як створити змінну?
3. За допомогою якої команди можна змінювати відтінок малювання?

Виконуйте таку послідовність роботи:

1. додайте спрайт «Ball»;
2. зберіть наведений скрипт (Рис.12.2);
3. запустити проект.

Внесіть у проект такі зміни:

1. змініть назву змінної;
2. змініть кількість заданих повторень;
3. змініть кут відрізу на інші значень, розглянь отриманий результат (Рис.12.3);
4. додайте блок зміни відтінку олівця;
5. додайте команду «штамп»;
6. проаналізуйте отримані зміни, зробіть висновки та збережіть новий проект.

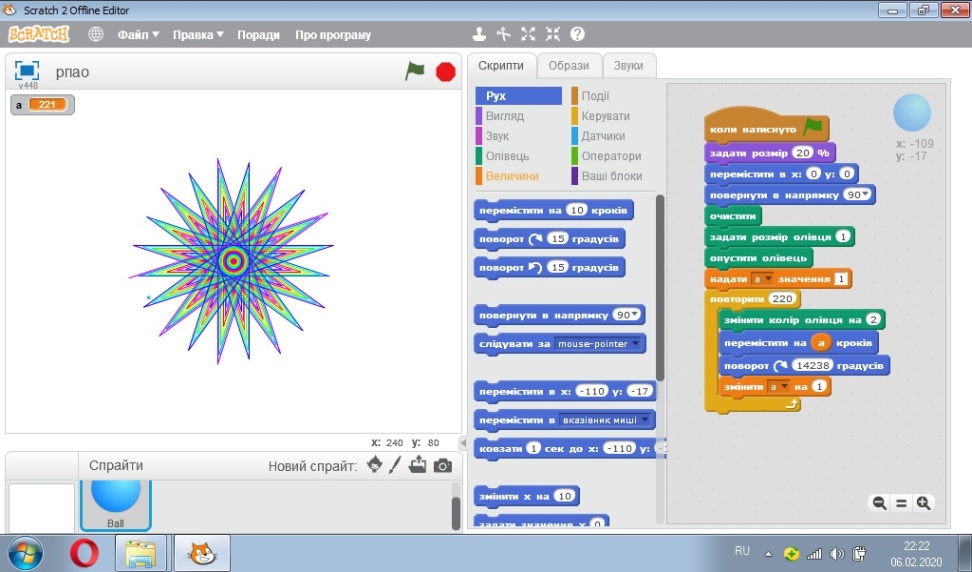
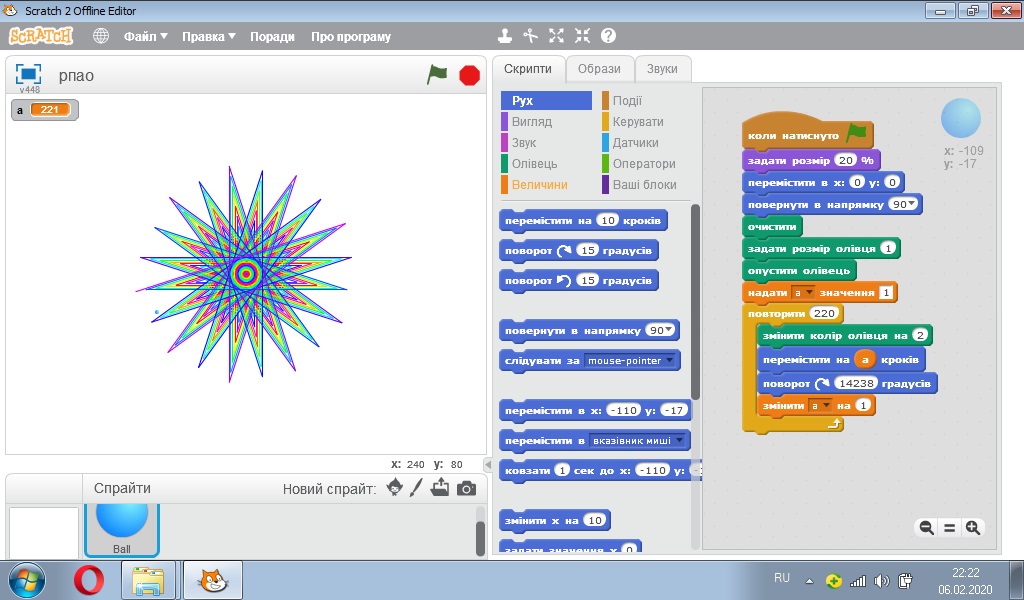


Рис.12.1 Малюнок до завдання 12 Рис.12.2 Скрипт до завдання 12

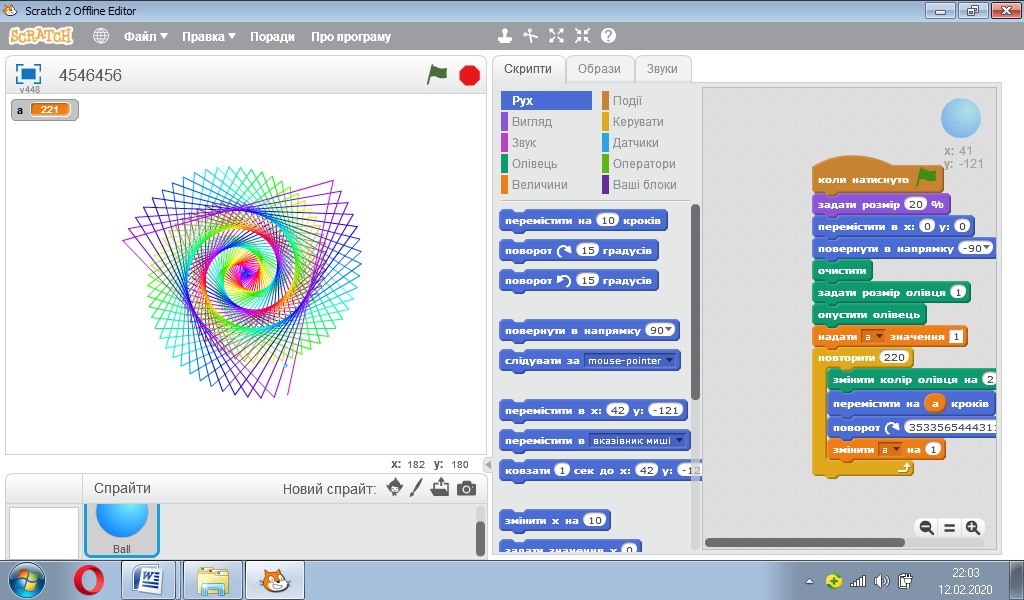
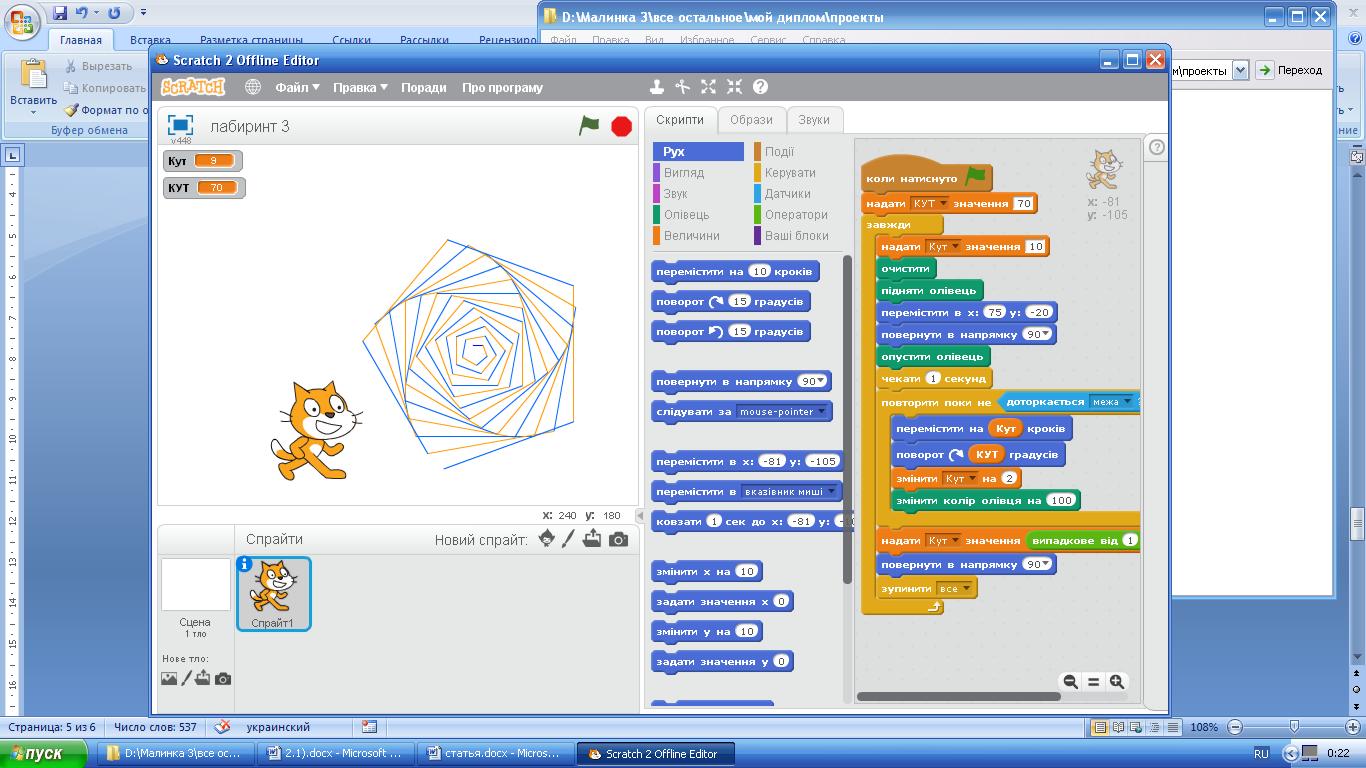


Рис.12.3 Малюнок до завдання 12 (малюнок, який з’явився в результаті внесених змін у кількість повторень та зміни градусу кута повороту)

**Завдання 13. «Лабіринт».** Це завдання носить тренувальний характер, під час виконання якого учні застосовують на практиці вкладені цикли (Рис.13.1-13.2).

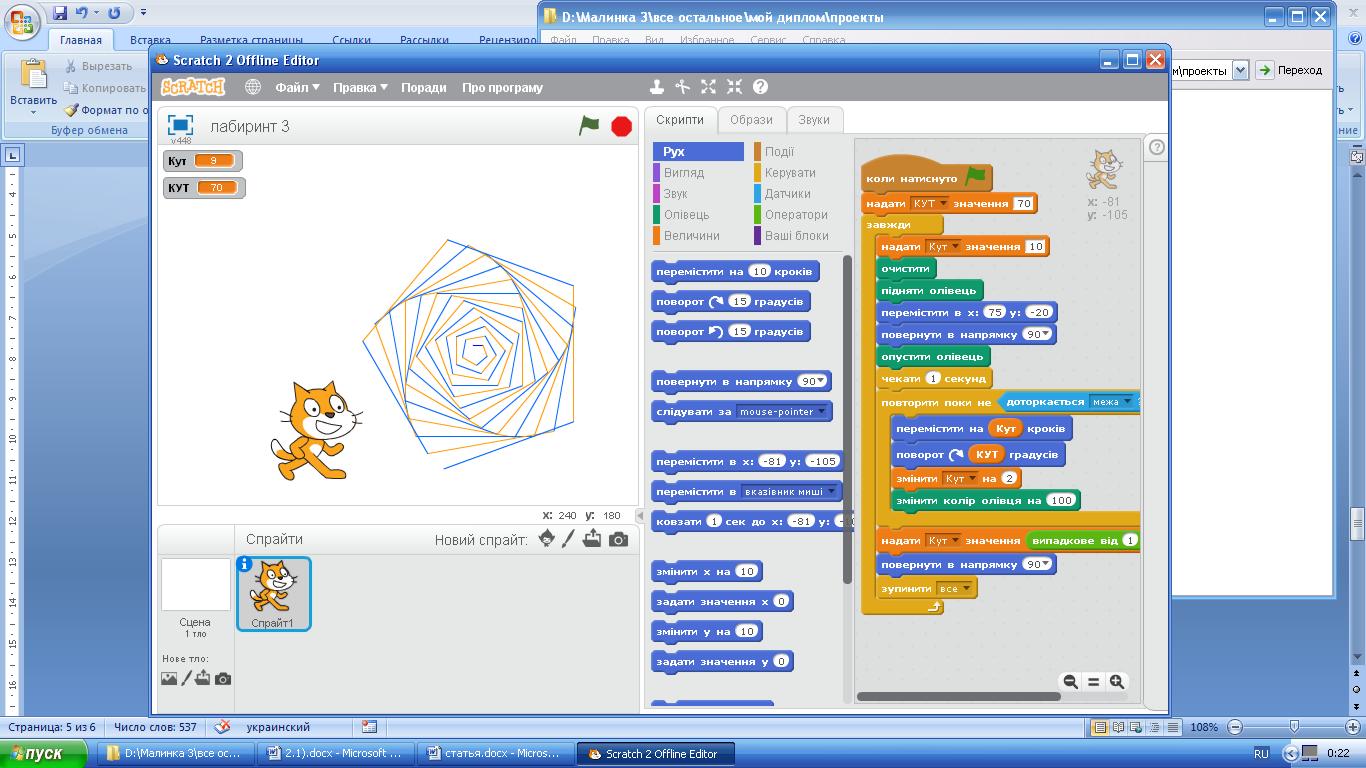


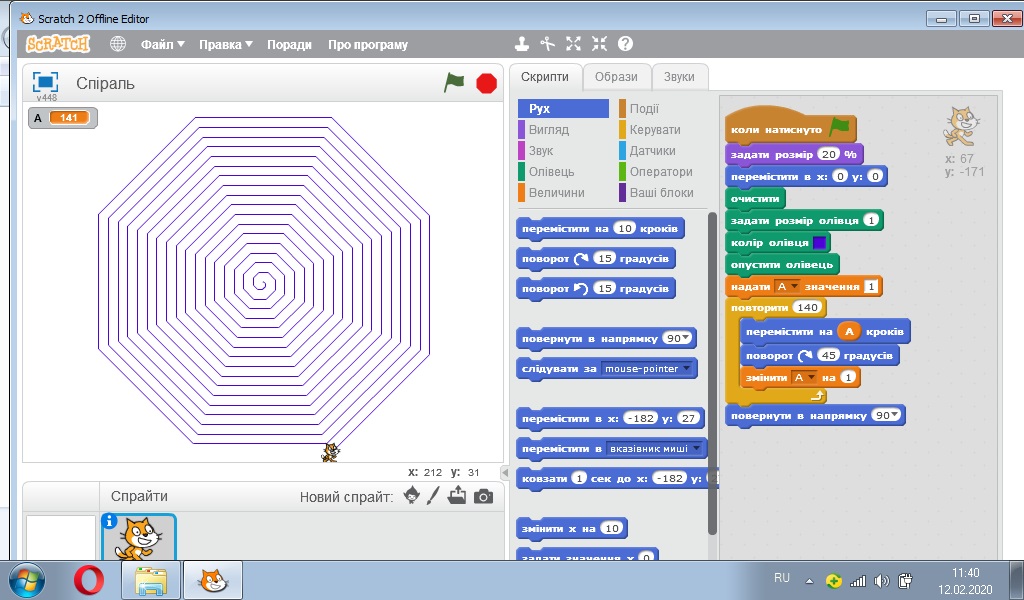
Рис.13.1. Малюнок до завдання 13 Рис.13.2. Розв’язок завдання 13 у Scratch

Для закріплення при роботі із завданням доцільно задати учням запитання.

1. Які команди середовища Scratch застосують в алгоритмах із фіксованою кількістю циклів?
2. Що вважається тілом циклу?
3. Наведіть приклад використання команди «повторити поки не».
4. Опишіть сутність команди «доторкається» з блоку «Датчики».

**Завдання 14. «Спіраль»**

Намалювати спіралеподібний малюнок. Це завдання також, як і завдання 13, носить тренувальний характер, під час виконання якого учні застосовують на практиці цикли та додають змінну «А». (Рис.14.1-14.2).



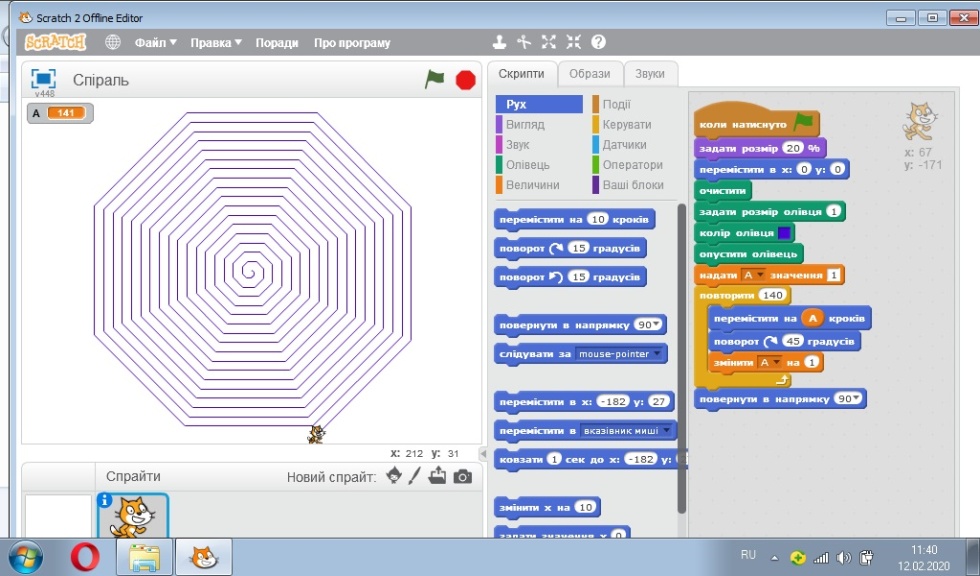
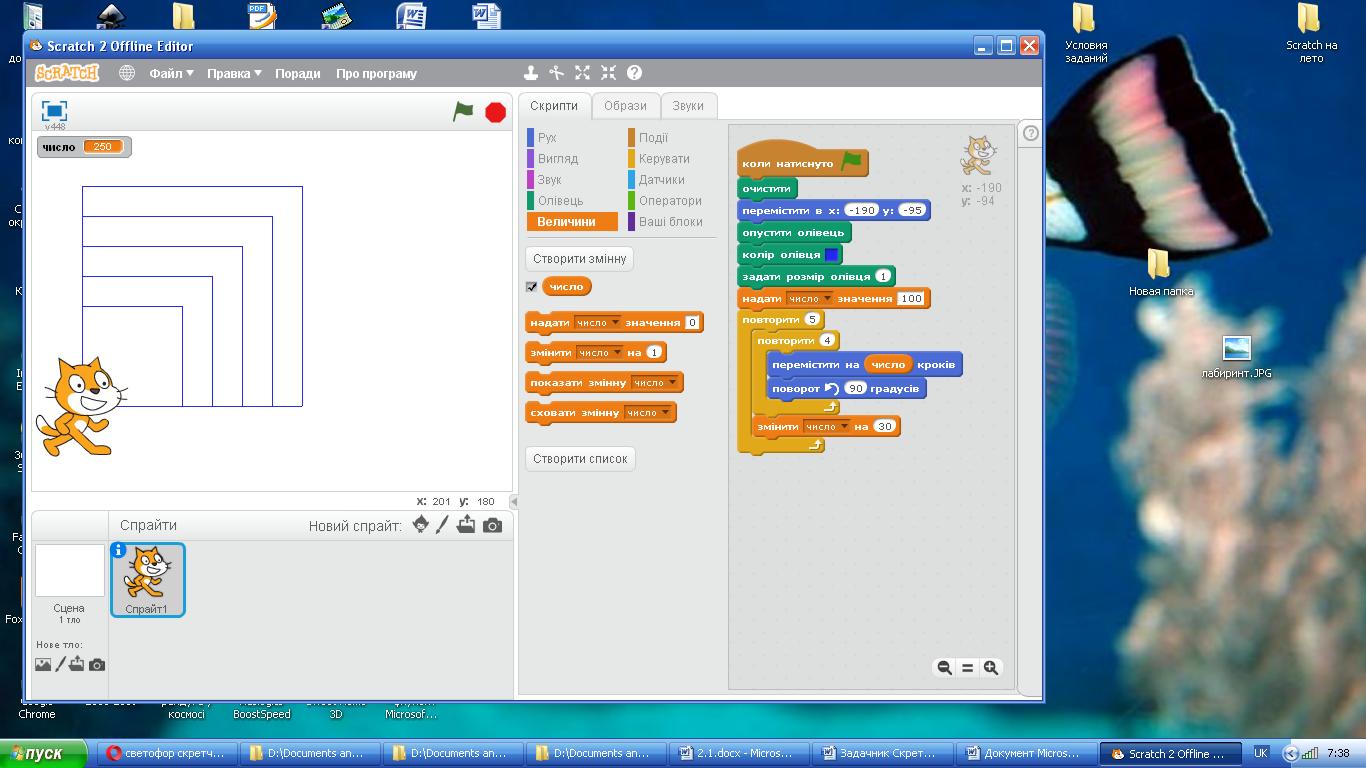


Рис.14.1. Малюнок до завдання 14 Рис.14.2. Розв’язок завдання 14 у Scratch

**Завдання 15. «Паркет»** (Рис. 15.1-15.2).

Намалюйте п’ять вкладених квадратів різної довжини будь-якого однакового кольору. Це завдання також, як і завдання 13 та 14, носить тренувальний характер, під час виконання якого учні застосовують на практиці вкладені цикли та додають змінну «А». (Рис.15.1-15.2).

Виконуйте таку послідовність роботи:

1. оберіть спрайт з бібліотеки спрайтів;
2. зберіть наведений скрипт (Рис.15.2);
3. запустіть проект.

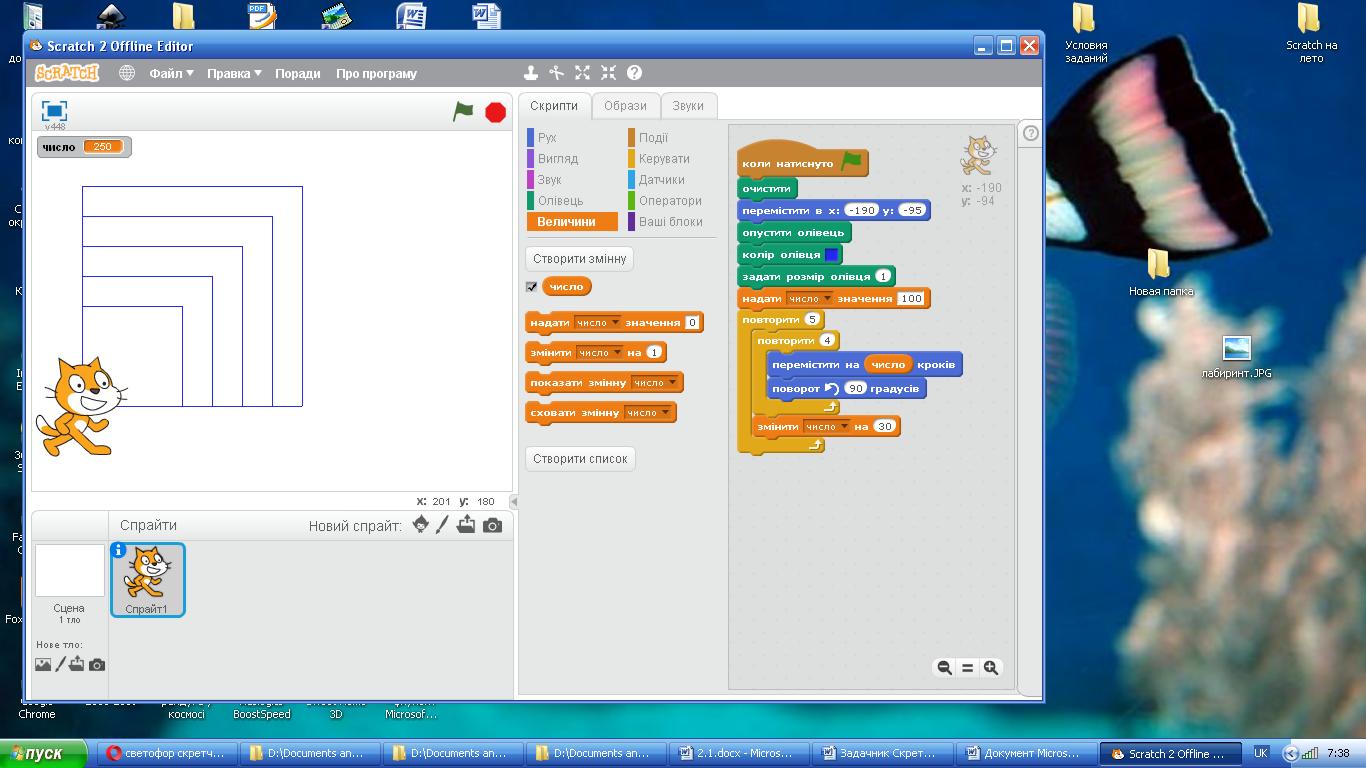
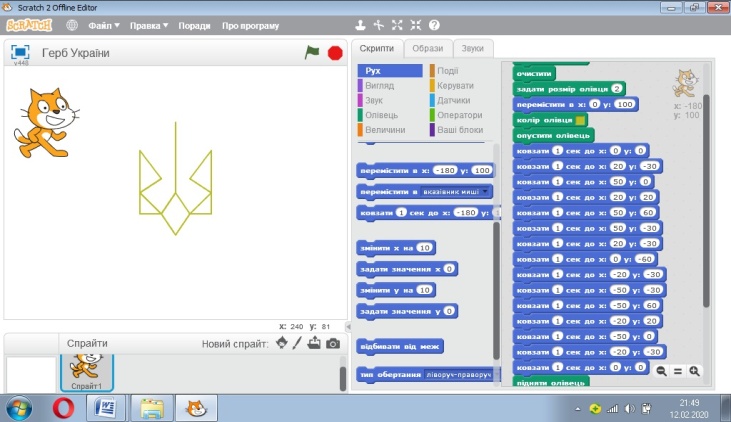


Рис. 15.1 Малюнок до завдання 15 Рис.15.2. Розв’язок завдання 15 у Scratch

Дайте відповіді на запитання.

1. Скільки градусів дорівнює кут квадрату ?
2. Скільки сторін має квадрат? Та чи рівні вони між собою?
3. За допомогою якої команди з блоку «Керування» можна реалізувати малювання квадрату?
4. Скільки квадратів потрібно намалювати по умові завдання?
5. Скільки вкладених циклів ми отримали?
6. Складіть скрипт, реалізуйте проект та збережіть його.

**Завдання 16. «Герб України»** (Рис. 16.1-16.2).

Намалювати Тризуб, який символізує Герб нашої Батьківщини – України. Це творче завдання, носить тренувальний характер, тому що учні застосовують лінійній алгоритм та працюють з блоком команд «Олівець».

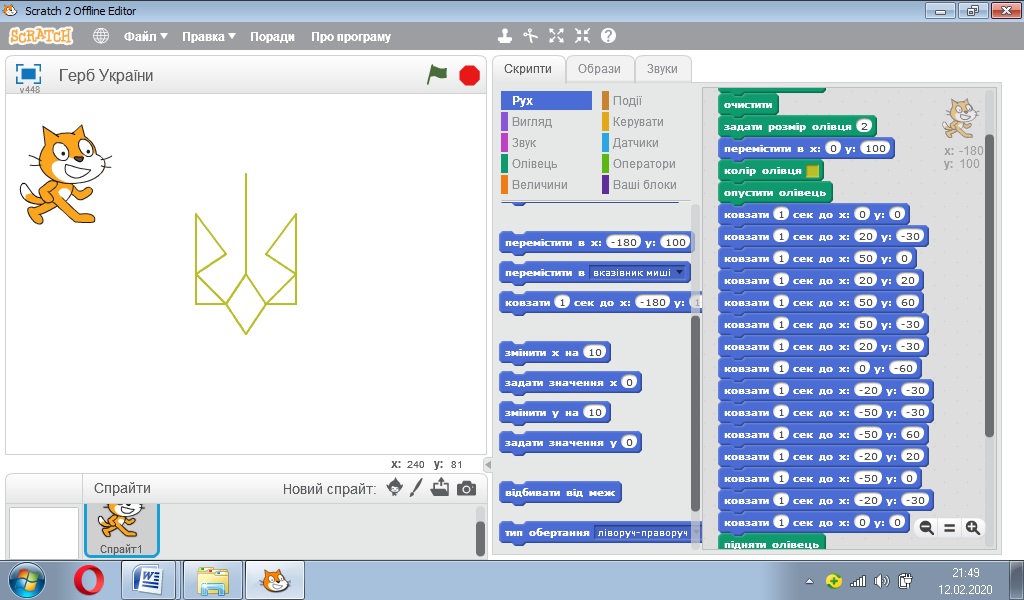


Рис. 16.1 Малюнок до завдання 16 Рис.16.2. Розв’язок завдання 16 у Scratch

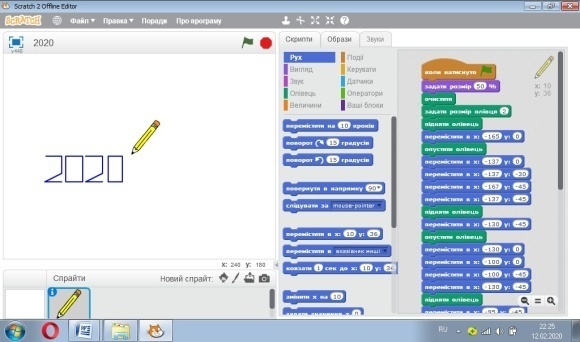
**Завдання 17. «2020».**

Намалювати на сцені число 2020 (Рис. 17.1). Використовуйте наданий скриптРис. (17.2).

Дайте відповіді на запитання.

1. Як система координат працює в Scratch?
2. В якій частині сцени знаходиться початок координат?
3. Які команди блоку «Олівець» знадобляться для виконання завдання?

Виконуйте таку послідовність роботи:

1. обрати тло з бібліотеки xy-grid з категорії «Інші»;
2. зібрати скрипт;
3. запустити проект.

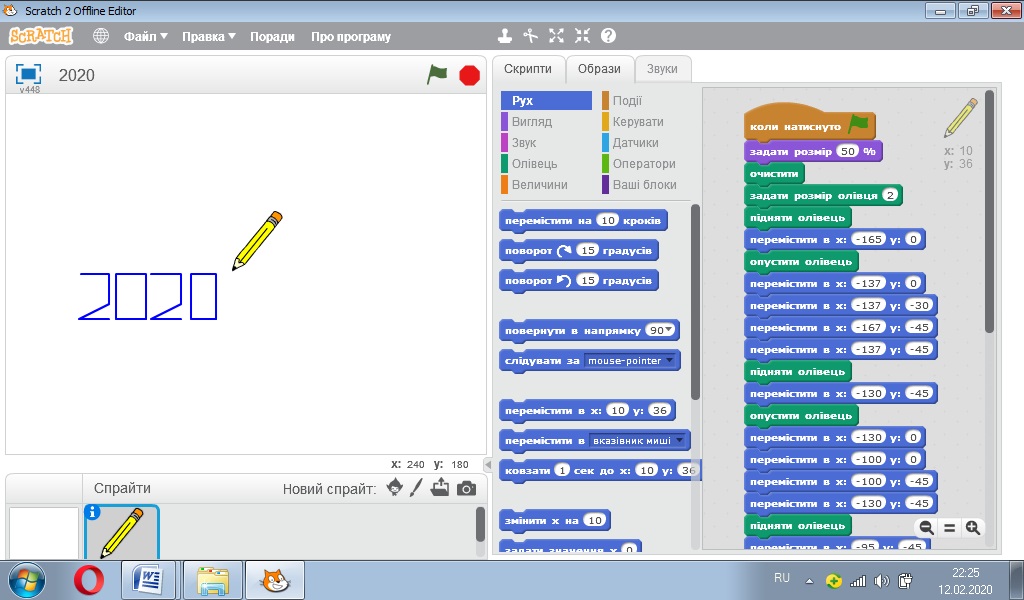


Рис.17.1. Малюнок до завдання 17 Рис. 17.2. Розв’язок завдання 17 у Scratch

Дайте відповіді на запитання.

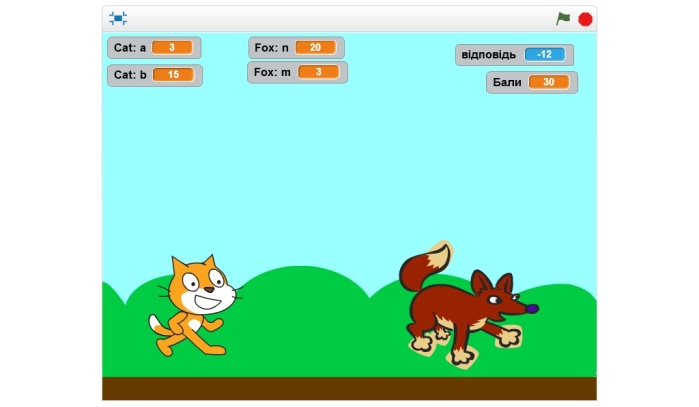
1. Яка частина скрипту малює числа 2, 0?
2. Як намальоване число 9 можна замінити на число 8?
3. Як можна використати блок «Повторення», якщо потрібно намалювати число 2010?
4. Яка команда з блоку «Олівець» може очистити тло?
5. Як можна намалювати число 2020 різними кольорами?

**Завдання 18. «Віднімання та додавання»** (Рис. 18.1-18.3)

Створіть проект на приклади з віднімання та додавання випадкових чисел. Додайте двох різних спрайтів, обов’язково введіть перевірку відповіді.

Дайте відповіді на запитання.

1. Як зробити, щоб спрайт вибирав правильну відповідь?
2. Як зробити підрахунок правильних відповідей та ввести змінну «Бали»?
3. Які команди блоку «Оператори» знадобляться для виконання завдання?



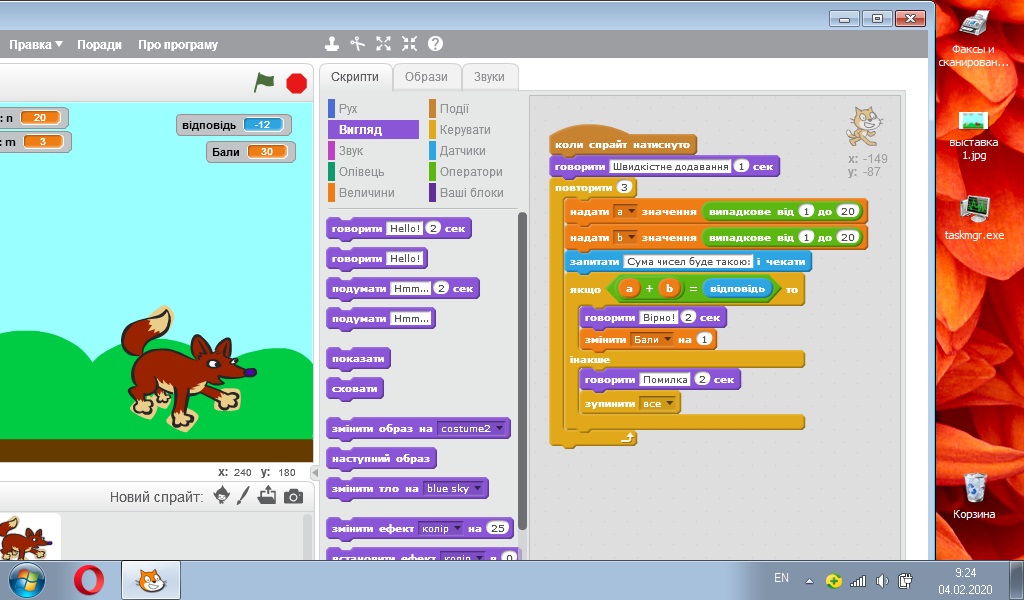
****

Рис.18.1. Малюнок до завдання 18

Рис. 18.2. Розв’язок завдання

18 у Scratch (для спрайта для додавання)

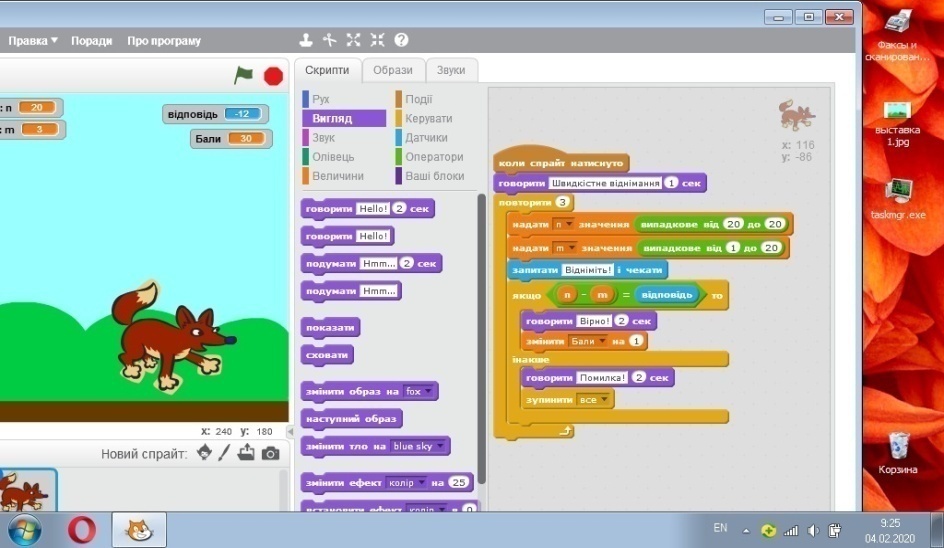


Рис. 18.2. Розв’язок завдання 18

у Scratch (для спрайта для віднімання)

Після виконання завдання потрібно дати відповіді на запитання:

1. Чи працює програма?
2. Як змінити значення для змінних?
3. Як в програмі використовується команда «Якщо, то, інакше»?
4. Додайте третього та четвертого спрайтів для інших математичних операцій, таких як множення та ділення?
5. Як обрати інше тло для програми?

У ході заключного уроку в 4 класі на тему «Розв’язок задач у середовищі програмування для дітей» учні демонструють свої набути знання і навички з складання скриптів різного типу складності. Тому на етапі закріплення знань можна запропонувати такі завдання пізнавального характеру, як «Цвітіння сакури» та розробка гри «Змійка».

**Завдання 19. «Цвітіння сакури»**

Потрібно намалювати гілку сакури. Квітка сакури повинна мати 5 пелюстків, при цьому кожна пелюстка має форму кола. Колір квітки обирається рожевий. Квітки розміщуються на гільці. Гілка має таку структуру: основну частину – півколо, елемент розгалуження – кілька четверті кола. Колір гілки обрати коричневим. (Рис.19.1-19.1)

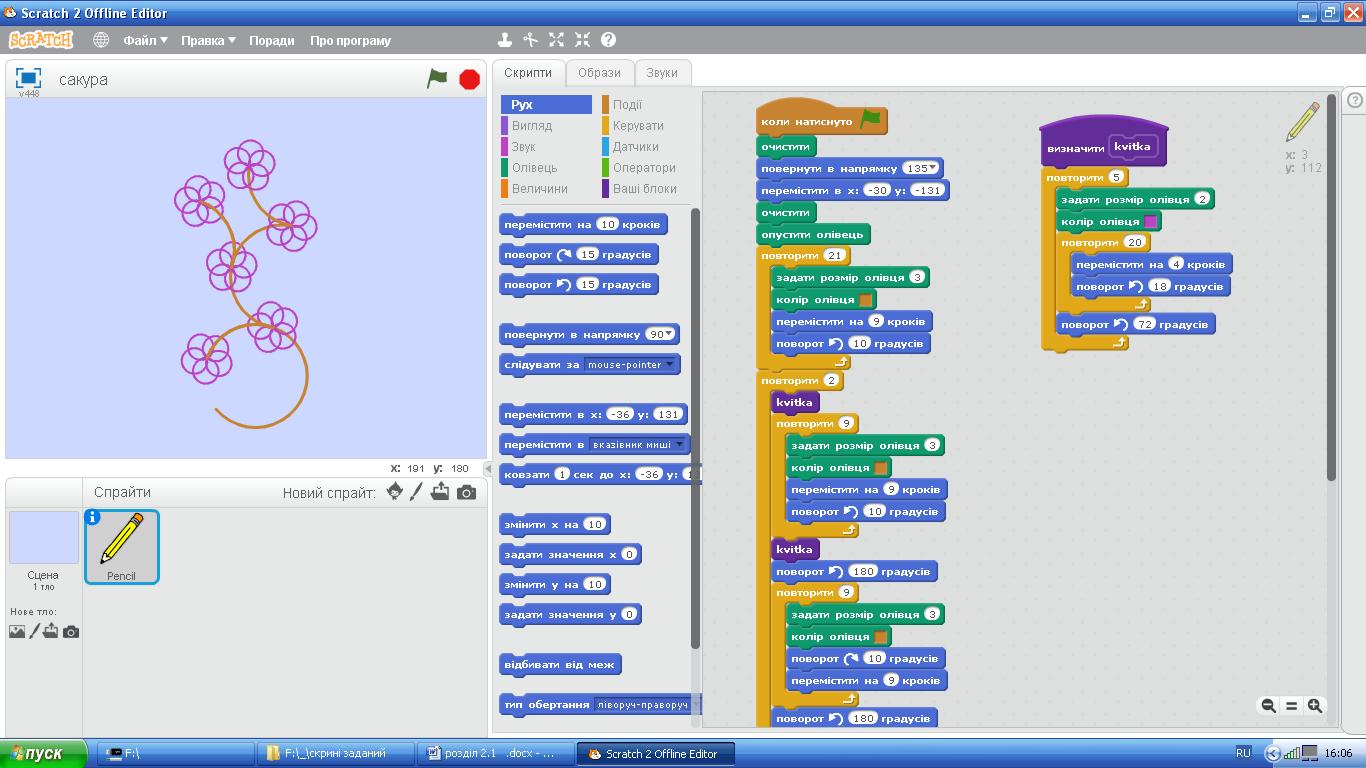
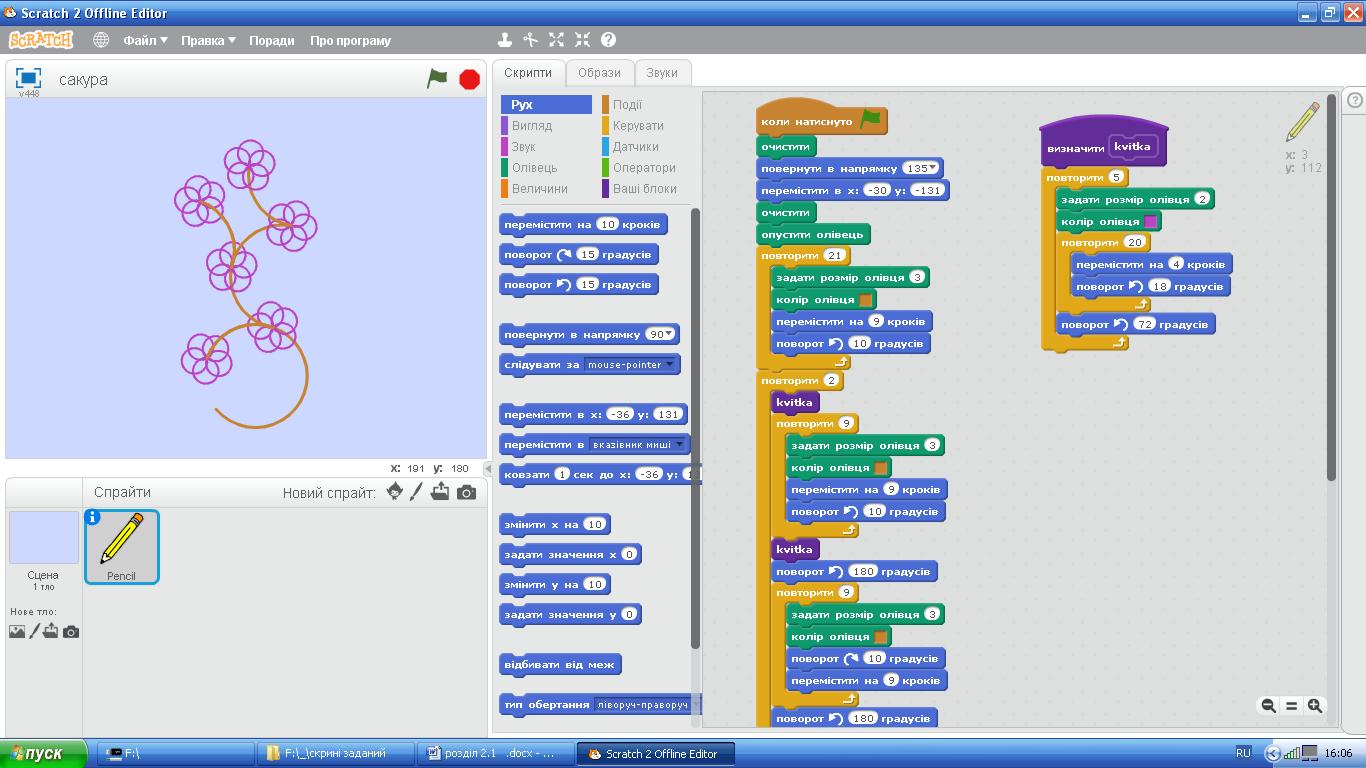


Рис.19.1. Малюнок сакури Рис.19.2. Скрипт для створення однієї квітки

Це завдання носить пізнавальний характер, закріплює такі види алгоритмів, як розгалуження та циклічність, навчає застосувати блок «Ваші блоки». Розвиває творчу уяву учнів та математичні навики.

**Завдання 20. «Гра «Змійка»**

Гра «Змійка» - це давно відома багатьом гра, де змія збільшується, ковтаючи яблука. Гра скінчується, коли змія стає занадто великою і не поміщається на сцені. Створення гри поетапне. Спочатку створюється новий спрайт «Голова» за допомогою графічного редактора. Потім частина тіла, яка буде розмножуватись шляхом клонування. Також додається третій готовий спрайт з бібліотеки – Яблуко. Змія рухається за допомогою клавіш зі стрілками, направляючись до яблука, яке з’являється на сцені у різному місці. Як і кожна з багатьох ігор, гра «Змійка» має рахунок. (Рис.20.1-20.3)

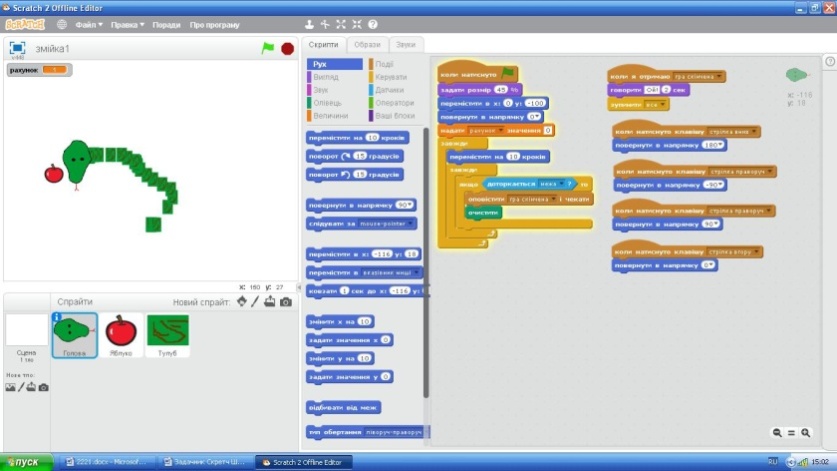
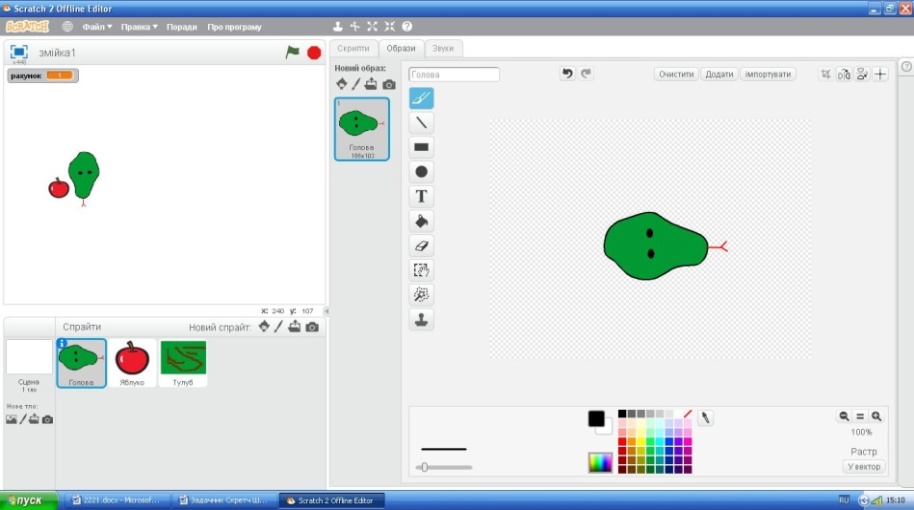


Рис.20.1. Малюнок до завдання 20

Рис.20.2. Малюнок спрайта «Голова»

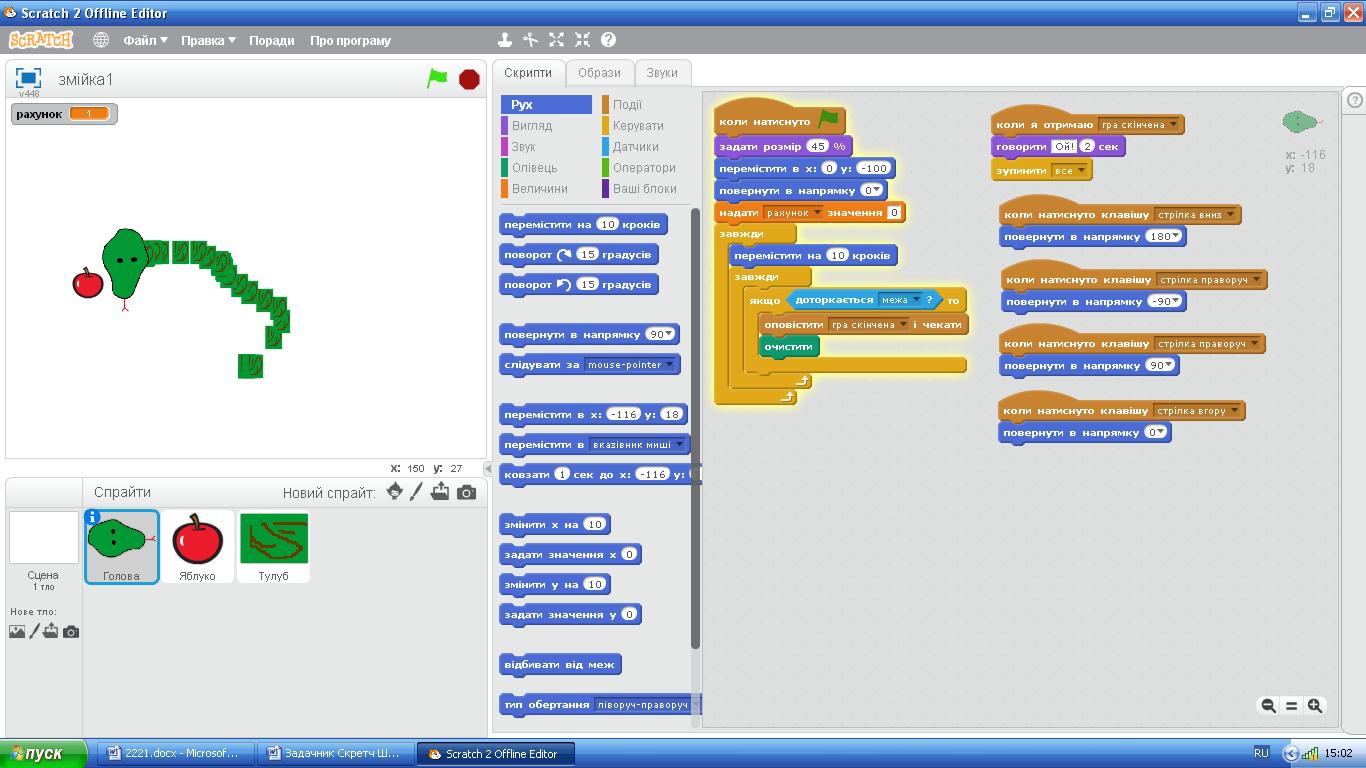


Рис.20.3. Розв’язок завдання 20 у Scratch

При аналізі умови цього завдань доцільно поставити учням такі запитання:

1. У чому різниця між блоками «Коли натиснуто клавішу» та «якщо клавішу натиснуто»?

2. Що робить блок «Переміститьсь в х: видати випадкове від -220 до 220 у: видати випадкове від -160 до 160»?

3. В чому різниця між блоками «Ковзати» та «Переміститись в»?

4. Чому намальована голова змії має напрямок на право та центральне положення на тлі гри?

5. Як зробити клонування та рух частин тіла змії?

6. Яким чином відбувається рахунок з’їдених яблук?

Виконання цього пізнавального завдання вимагає від учнів залучення низки навичок та умінь роботи у середовищі програмування Scratch. Зокрема, учням необхідно вміти працювати з вбудованим графічним редактором в режимах векторної та растрової графіки, застосовувати команду «створи клон з самого себе», змінювати дані лічильника тощо. Розробка скрипту, що реалізує гру, розширює набуті знання та навички програмування та алгоритмізації, розвиває творчу уяву в учнів, роблячи програмування захопливим та корисним.

ВИСНОВКИ

У ході роботи виконано усі поставлені завдання, що дозволяє зробити такі висновки.

1. Виявлено значущість інтеграційного підходу до навчання в концепції базової української школи.

2. Зроблено частковий аналіз особливостей навчання основ програмування у початковій школі. Встановлено основні методичні проблеми, що виникають при викладанні базових понять алгоритмізації і програмування. Проведено аналіз чинної програми і підручників з інформатики для 2-4 класів, який засвідчує нестачу різнотипових практичних завдань та покрокових рекомендацій до їх виконання і висуває необхідність розробки комплекту таких завдань.

2. Створено комплект різнотипових практичних завдань для навчання учнів 2-4 класів основ програмування у середовищі Scratch з інтеграційним підходом.

Розроблений комплект має структуру, яка відповідає календарному плануванню уроків з інформатики у 2-4 класах. Він охоплює в цілому 20 завдання, які можна застосовувати при вивченні різних розділів курсу. Також можна використовувати під час гурткових занять.

3. Надано методичні рекомендації щодо використання розробленого комплекту на уроках інформатики у 2-4 класах. Наведено приклади застосування завдань різних типів на відповідних етапах уроків інформатики.

Матеріали та результати дослідження було апробовано шляхом публікації статті в збірнику наукових праці Харківського державного педагогічного університету ім. Г.С. Сковороди та у навчально-виховному процесі загальноосвітньої школи № 8 І-ІІІ ступенів міста Павлограда Дніпропетровської області.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Барболіна Т.М. Шкільний курс інформатики та методика його викладання: навч. посіб. Полтавський держ. пед. університет ім. В.Г. Короленка. Полтава. 2007. Ч. 1. Загальна методика. 124 с.
2. *Вікіпідручник по Scratch*. URL: <http://uk.wikibooks.org>/ (дата звернення 29.10.2018)
3. Вдовенко В.В. Формування алгоритмічного мислення молодших школярів на уроках інформатики. Наукові записки Кіровоградський держ. пед.. університет імені В. Винниченка. 2016. 11(IV). С. 23-26.
4. Голиков Д.В. 40 проектов на Scratch для юных программистов. СПб.: БХВ-Петербург. 2018. – 192 с.: ил.
5. Голиков Д.В. 42 проекта на Scratch для юных программистов. СПб.: БХВ-Петербург. 2019. – 184 с.: ил.
6. Журибеда О. Scratch очима вчителя. *Інформатика. Шкільний світ.* 2018. № 6. С. 6-19.
7. *Інформатика. Навчальна програма для загальноосвітніх навчальних закладів в 2-4 класах.* URL: <https://mon.gov.ua/ua/osvita/zagalna-serednya-osvita/navchalni-programi/navchalni-programi-dlya-pochatkovoyi-shkoli/> (дата звернення 31.08.2019)
8. Кузнецова І.В. Розвиток пізнавального інтересу і творчої активності учнів. *Компютер у школі та сімї.* 2011. №1. С. 6-15.
9. Лернер И.Я., Шахмаев Н.М. Теоретические проблемы современного школьного учебника. *Сб. науч. тр.* *НИИОП*. Москва: НИИОП, 1989. С. 63-67.
10. *Офіційний сайт проекту Scratch*. URL: <http://scratch.mit.edu/> (дата звернення 04.02.2020)
11. Писарчук О.Т. Активізація навчально-пізнавальної діяльності молодших школярів в умовах освітньо-розвивального середовища: проблеми і шляхи розв’язання. *Science and Education a New Dimension. Pedagogy and Psychology.* 2014. Vol. 20. P. 90-92.
12. Пономарьова Н.О. Педагогічні умови використання пізнавальних задач у навчанні інформатиці 13.00.01-теорія та історія педагогіки. *Дис. на здобуття наукового ступеня канд.пед.наук*. Харків: ХНПУ імені Сковороди, 1998. С. 38
13. Руденко В.Д. Алгоритмізація і програмування: навч. посіб. / В.Д. Руденко; за заг.ред. акад. НАПН України, докт. тех. наук, проф. В.Ю. Бикова.- Х.: Ранок. 2017. 128с.
14. *Середовище програмування Скретч для створення власних проектів.* URL: <http://informatic.sumy.ua/scratch.php/> (дата звернення 15.04.2019)
15. Створюй анімації за допомогою Скретч; пер. з англ. О.Ю. Григорович. - Х: Ранок, 2019.- 128с.: іл.
16. Торгашева Ю.В. Первая книга юного программиста. Учимся писать программы на Scratch. СПб.: Питер. 2016. 128 с.