**Тема уроку :**  **Об'єкти, властивості, конструктори, методи.**

**Мета:**

* ***формування предметних компетентностей***: сформувати уявлення про основні властивості та події, уміння задавати значення властивостей класу використовувати об’єкти для виведення інформації про хід виконання програми; виховувати інформаційну культуру.
* ***формування ключових компетентностей:***
  + *спілкування рідною мовою*: уміння формулювати запитання та дискутувати; уміння оперувати термінами і поняттями з основ програмування;
  + *спілкування іноземними мовами*: уміння використовувати іноземні джерела інформації;
  + *уміння вчитися:* навички конструювати власну освітню траєкторію;
  + *ініціативність і підприємливість*: уміння брати відповідальність за результат роботи на уроці.

**Тип уроку:** Комбінований.

**Обладнання та наочність:** дошка, комп’ютери з підключенням до мережі Інтернет.

**Програмне забезпечення:** Середовище програмування Processing ,браузер.

**Хід уроку**

**І. Організаційний етап**

* привітання
* перевірка присутніх
* перевірка готовності учнів до уроку

**ІІ. Актуалізація опорних знань**

Опитуання.

* Як пишеться виклик методу об'єкта?
* Як створити декілька об'єктів одного класу?
* Чим зручна робота з об'єктами?
* Яку можливість ми отримуємо оголошуючи свої власні класи?

**ІІІ. Мотивацій навчальної діяльності**

Оголошуючи свої власні класи, ми отримуємо можливість створювати свої власні унікальні геометричні об'єкти.

**IV. Вивчення нового матеріалу**

**Пояснення вчителя з елементами демонстрування презентації**

*(використовуються можливості локальної мережі кабінету або проектор)*

Якщо раніше ми були обмежені геометричними примітивами Processing, то тепер можемо описати клас своєї власної лінії. Розглянемо код:

class MyLine {

float centerX , centerY , length , angle , weight;

public void render () {

float x1 = centerX - cos(angle)\*length /2;

float y1 = centerY + sin(angle)\*length /2;

float x2 = centerX + cos(angle)\*length /2;

float y2 = centerY - sin(angle)\*length /2;

stroke (50, 100);

strokeWeight(weight);

line(x1 , y1 , x2 , y2);

strokeWeight (5);

stroke (50);

line(x2 , y2 , x2 , y2);

line(x1 , y1 , x1 , y1);

}

}

MyLine lineObj;

float counter;

void setup () {

size (500, 500);

smooth ();

background (255);

lineObj = new MyLine ();

lineObj.centerX = width /2;

lineObj.centerY = height /2;

lineObj.length = 350;

lineObj.angle = PI/4;

lineObj.weight = 1;

}

void draw () {

counter += 0.05;

if (counter > TWO\_PI) {

counter = 0;

}

lineObj.centerX = width /2 + sin(counter)\*50;

lineObj.centerY = width /2 + cos(counter)\*50;

lineObj.angle = counter \*2;

lineObj.render ();

}

При запуску програми ми бачимо, що лінія рухається по певній траєкторії. Причому лінія має унікальну форму з круглим закінченням. Лінія - це об'єкт класу MyLine, який ми оголошуємо з 1-го по 16-й рядок.

Клас MyLine містить кілька властивостей і один метод render ().

Цей метод служить для відтворення ліній. Об'єкт класу MyLine створюється в 24-му рядку (праворуч від знака рівності) і потім за допомогою знака рівності пов'язується з посиланням lineObj, яку ми оголосили в 18-му рядку. Після цього з 25-ї по 29-ю рядок ми встановлюємо значення властивостей нашого об'єкта lineObj.

У методі draw () відбувається оновлення лічильника counter для того, щоб сформувати координати для центру лінії. Центр лінії встановлюється в рядках 36 і 37. У 38-му рядку ми встановлюємо нове значення властивості angle і промальовується лінію в 39-му рядку. При наступному виклику методу draw () ми оновимо лічильник і у тій же самій лінії поміняємо її центр і кут. Таким чином, ми працюємо з одним об'єктом, тобто з однієї і тієї ж лінією: ми як би зберігаємо її між викликами методу draw ().

Змінимо код, додавши рух від центру полотна до його краям:

17 MyLine lineObj;

18 float counter , radius;

28 lineObj.weight = 1;

29 radius = 50;

30 }

33 counter = 0;

34 radius = radius + 50;

35 }

36 lineObj.centerX = width /2 + sin(counter)\*radius;

37 lineObj.centerY = height /2 + cos(counter)\*radius;

Ми внесли незначні зміни: додали властивість radius (рядок 29) і використовували його при формуванні координат центру нашої лінії (рядки 36 і 37). В іншому код залишився колишнім.

Видозмінюючи параметри відтворення в коді, можна домогтися цікавих результатів. Для того щоб вийшов результат необхідно включити логіку роботи з кольором в клас MyLine. Ви можете змінювати значення кольорів застосовуючи принципи тригонометрії.

У цьому параграфі ми познайомилися з можливостями опису своїх власних класів об'єктів, створення самих об'єктів і маніпулювання ними. Клас - це сутність (фрагмент коду), який визначає структуру (набір властивостей) всіх об'єктів цього класу. Всі об'єкти одного класу повинні мати однаковий набір методів, однаковий набір властивостей, але значення властивостей можуть бути різними. об'єкт можна уявити як єдиний пакет даних і функціональності. Як правило, єдиний шлях дістатися до даних (властивостей) об'єкта - викликати одну з наданих їм функцій. Ці функції називаються методами.

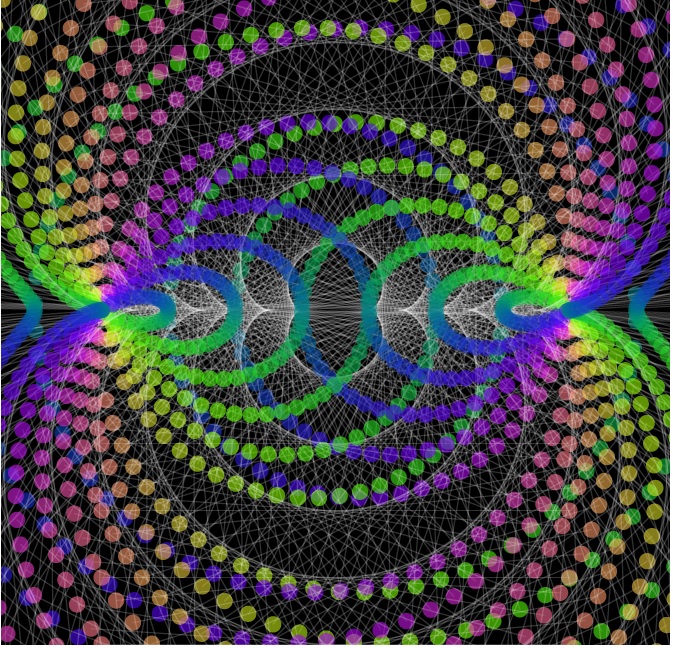
Рядків програмного коду виходить більше, але його ясність стає вище. Взаємодія з об'єктами відбувається на інтуїтивному рівні за рахунок методів. При роботі з Processing найчастіше доведеться використовувати вже розроблені класи, а не створювати власні. Однак все залежить від ваших амбіцій і завдань.

**V. Фізкультхвилинка**

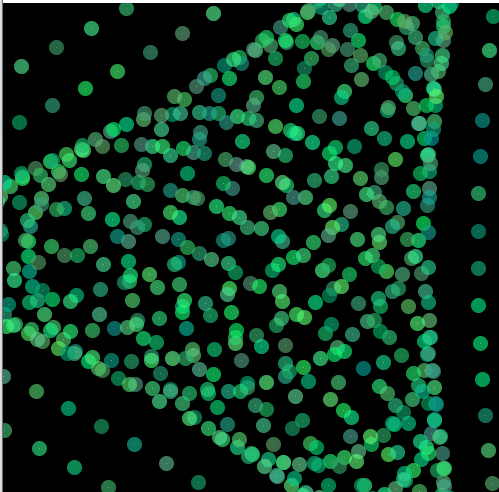
**VI. Засвоєння нових знань, формування вмінь**

Практичне завдання. Робота за комп’ютером

1. Повторення правил безпечної поведінки за комп’ютером.
2. Інструктаж учителя.
3. Практична робота за комп’ютерами.

**Завдання 1.** Змініть код, реалізувавши в ньому роботу з кольором, таким чином, щоб в результаті у вас вийшло зображення схоже з малюнком нижче.

**Завдання 2.** Змініть код таким чином, щоб в результаті у вас вийшло зображення схоже з малюнком нижче.



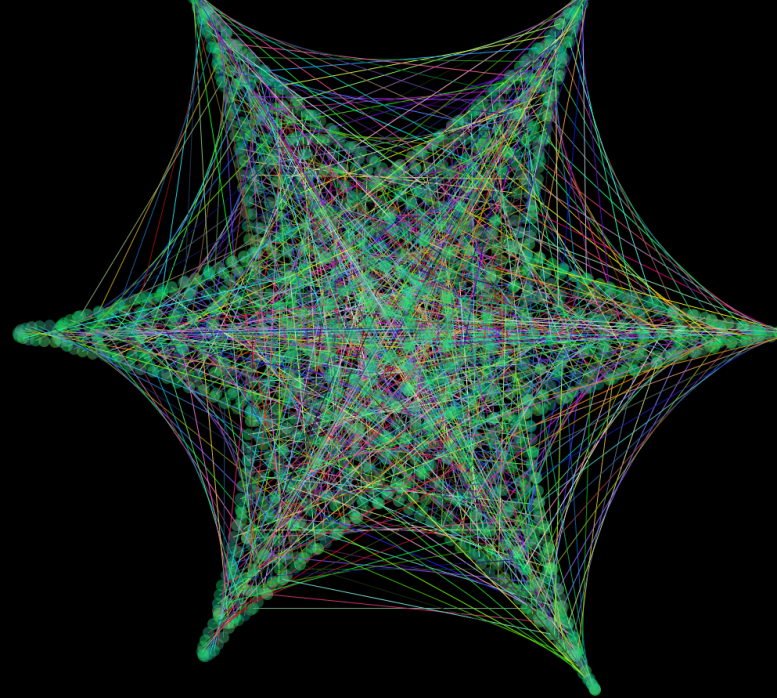
float x1 = centerX + cos(angle)\*length /2;

float y1 = centerY + sin(angle)\*length /2;

float x2 = centerX + cos(angle)\*length /2;

float y2 = centerY + sin(angle)\*length /2;

**Завдання 3.** Змініть код таким чином, щоб в результаті у вас вийшло зображення схоже з малюнком нижче.



float x1 = centerX - cos(angle)\*length /2;

float y1 = centerY - sin(angle)\*length /2;

float x2 = centerX + cos(angle)\*length /2;

float y2 = centerY + sin(angle)\*length /2;

1. Вправи для очей.

**VIІ. Підсумки уроку**

Рефлексія

1. Що нового ви сьогодні дізналися?

2. Чого навчилися?

3. Чи виникали труднощі?

**VІІI. Домашнє завдання**

Опрацювати теоретичний матеріал з теми уроку

**IX. Оцінювання роботи учнів**