

Для реалізації даного проекту буде введено два абстрактних типи даних: Point та Graph. Для початку розглянемо тип Point.

## Point

АТД Point використовує модуль math, з якого дістаються функції, методи та атрибути: pi, sin, cos, radians, для повороту точки навколо іншої точки. Кожна точка має координати, якими відрізняється від інших екземплярів класу.

Ініціалізація типу через перенос точки на задані координати. Точка має атрибути x та y для визначення координат.

Метод **move** просто переносить точку на нові координати.

Метод **listing** повертає список x координат

Метод **rotate** повертає одну точку навколо іншої. Цей метод використовує імпортований модуль.

Також цей АТД має методи для створення стрічки з координат та метод для виводу екземплярів.

Цей тип даних є дуже важливим для даного проекту, бо він використовується для утворення малюнка, який шифрує текст. Так, як малюнок будує модуль matplotlib, який використовує список точок на вході, цей тип даних дозволяє доцільно замінити кортеж з координатами, адже він має зручні методи.

Тип Кратка реалізований на основі операцій зі змінними і їх окремому збереженні в пам'яті. Обчислювальна здатність наступна:

Переміщення точки -  $O(1)$

Поворот однієї точки навколо іншої -  $O(1)$

## Graph

АТД Graph уособлює собою пайтонівський список, який складається з екземплярів класу Tuple. Завдяки цьому можна проводити обчислення над точками чи зображувати цей граф через бібліотеку matplotlib. Як було зазначено вище, функція plot, що будує графічне зображення, отримує як аргументи два списки відповідних координат усіх точок графа.

Граф ініціалізується, як список списків відповідних екземплярів, де внутрішній список представляє одне слово. Кожне слово складається з точок, які представляють літеру, тому структура має трохи громіздкий вигляд.

Абстрактний тип граф має лише два методи. Перший з них це **draw**. Він зображує граф на екрані, як малюнок. Цей метод не приймає додаткових аргументів окрім self.

Другий метод **save** зберігає зображення графа у файл. Цей метод приймає як аргумент назву файлу, в який збережеться граф.

Також у модулі graph.py є допоміжні функції, які шукають найбільше і найменше значення серед списку списків, вважаючи, що останні складаються з кортежів. Ці функції отримують першим аргументом номер комірки, яку нам треба оглянути, а другим - квадратний список.

Тип Граф створений на основі двовимірного масиву C. Завдяки цьому ми отримуємо наступну складність:

Вставка/видалення нового слова -  $O(n)$

Вставка/видалення нової літери -  $O(n^2)$

Доступ до слова -  $O(1)$

Доступ до літери -  $O(n)$

Заміна слова -  $O(1)$

Заміна літери -  $O(n)$

Обхід по словам -  $O(n)$

Обхід по літерам -  $O(n^2)$

Пошук літери -  $O(n^2)$