

Групове завдання 23

Завдання продовжує завдання 22

Бінарне зображення на прямокутнику $[(0,0), (m, n)]$ складається з точок. Кожна точка може мати значення 1 (зафарбовано) та 0 (не зафарбовано).

Для стиснення бінарного зображення використовують ланцюговий код, який визначається наступним чином: для послідовності зафарбованих точок у рядку i , починаючи з позиції j , довжиною k зберігають кортеж (i, j, k) .

Фігура у ланцюгових кодах визначається наступним чином: два сусідніх рядки фігури мають хоча б один перетин (тобто, мають ланцюгові коди, які перетинаються). Приклад перетину – на рисунку нижче

```
.....*****.....  
.....***.....
```

Описати клас ChainCodeFigure, який зберігає одну фігуру у ланцюгових кодах. Цей клас повинен мати методи:

Конструктор – створює порожню фігуру

add_code – додати ланцюговий код code до фігури

weight – обчислити та повернути «вагу» фігури, кількість точок у фігурі

mass_center – обчислити та повернути центр мас фігури (суми значень по відповідних координатах розділити на вагу)

codes – властивість (property), повертає усі коди фігури

intersects – перевіряє, чи перетинається фігура з заданим ланцюговим кодом code

merge – злити фігуру з іншою фігурою figure

__lt__ - чи менше поточна Фігура1, ніж Фігура2, other. Фігура1 < Фігура2, якщо вона знаходиться вище (спочатку) та лівіше (потім), ніж Фігура2

show – показати фігуру у масштабі scale кольором color

З використанням класів ChainCodeReader, ChainCodePicture, ChainCodeFigure розв'язати задачу. У текстовому файлі в архіві ht23.zip записано бінарне зображення. Порахувати та показати (print) кількість фігур, координати центру мас та вагу кожної фігури. Показати зображення кожної окремої фігури у turtle у заданому масштабі чорним кольором.

Нижче наведено завдання 22

Бінарне зображення на прямокутнику $[(0,0), (m-1, n-1)]$ складається з точок. Кожна точка може мати значення 1 (зафарбовано) або 0 (не зафарбовано).

Для стиснення бінарного зображення використовують ланцюговий код, який визначається наступним чином: для послідовності зафарбованих точок у рядку i , починаючи з позиції j , довжиною k зберігають кортеж (i, j, k) .

Нехай у текстовому файлі один рядок відповідає одному рядку зображення. Зафарбовані точки позначені зірочками ('*'), не зафарбовані - крапками ('.'). Наприклад, початковий рядок файлу може мати вигляд:

```
....***...*****.....****.....
```

Для рядка файлу, зображеного вище, маємо таку послідовність ланцюгових кодів: (0, 4, 3), (0, 10, 8), (0, 23, 4)

Описати клас ChainCodeReader, який читає бінарне зображення з текстового файлу (ім'я файлу передається у конструкторі) та формує послідовність ланцюгових кодів. Цей клас також має бути ітератором то повертати усі прочитані коди у порядку слідування. Окрім цього клас має містити властивість (property) codes, що повертає список прочитаних кодів

Для показу за допомогою turtle одну зафарбовану точку показувати зафарбованим квадратом розміром a (для незафарбованої точки відповідно пропустити квадрат розміром a).

Описати клас ChainCodePicture, який має метод show для показу бінарного зображення, представленого у вигляді послідовності ланцюгових кодів, за допомогою turtle. У конструктор класу передаються параметри:

- codes - послідовність кодів
- scale – масштаб відображення точки (значення a)
- color – колір відображення

Врахувати, що у turtle вісь OY спрямована угору.

Для зображення одного зафарбованого квадрату довжиною a чорним кольором, починаючи з поточної позиції, можна використати такі команди turtle:

```
turtle.color('black', 'black')
```

```
turtle.begin_fill()
```

```
for i in range(4):
```

```
    turtle.fd(a)
```

```
    turtle.right(90)
```

```
turtle.end_fill()
```

З використанням класів ChainCodeReader та ChainCodePicture розв'язати задачу. Нехай у текстовому файлі записано бінарне зображення для сьогоднішньої дати (у форматі “dd mm yyyy”, dd - день, mm – місяць, yyyy - рік). Перетворити це зображення у ланцюговий код та показати.