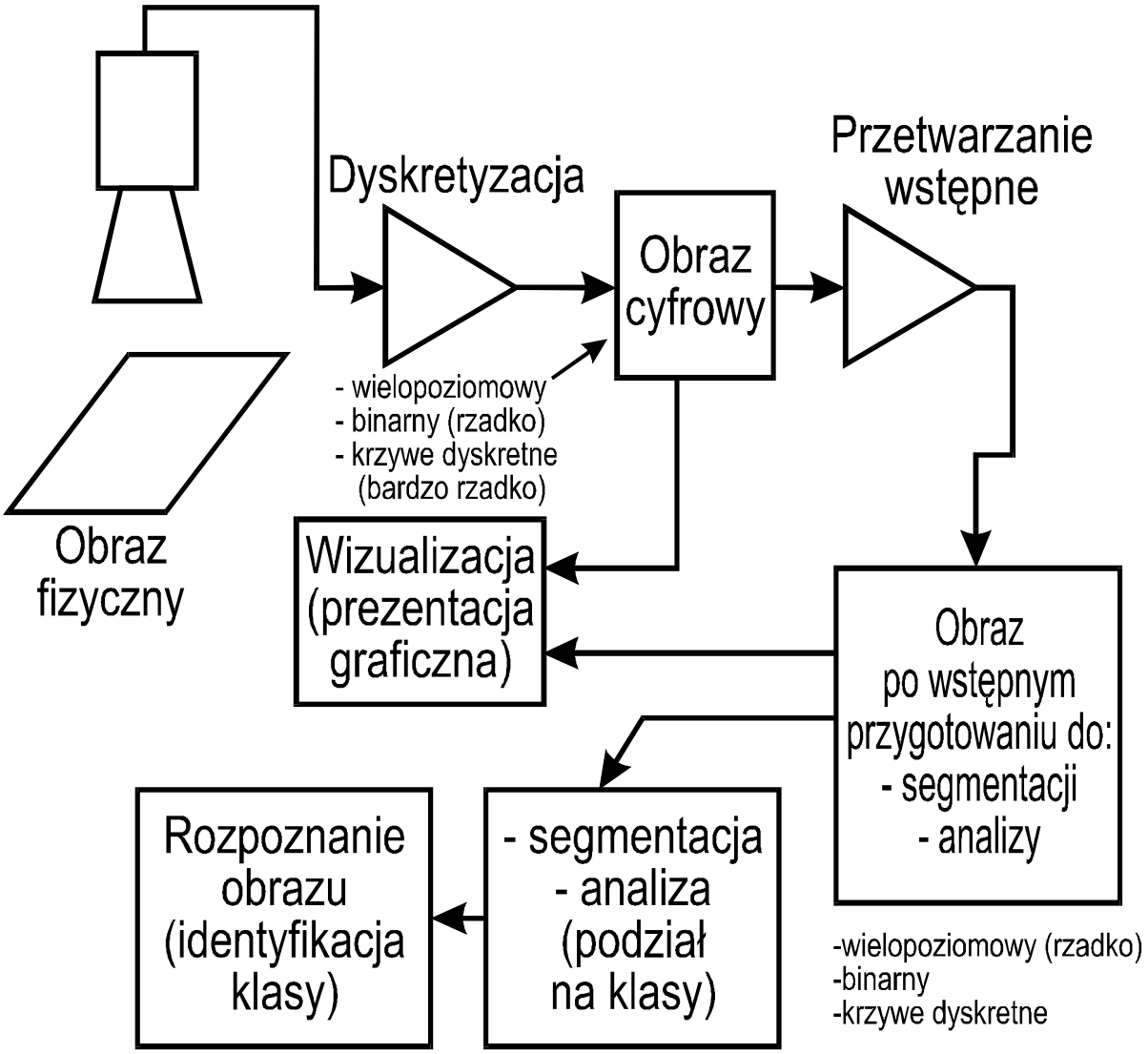
WYKŁAD 8

- **Reprezentacja obrazu**
- **Elementy edycji (tworzenia) obrazu**

Postacie obrazów na różnych etapach procesu przetwarzania



Klasy obrazów

- **Klasa 1:**

Obrazy o pełnej skali stopni jasności, typowe parametry: N=512, M=256
Reprezentacja rastrowa: np. tablica 512x512 jednobajtowych elementów (true color - 3 bajty NxN)

- **Klasa 2:**

Obrazy binarne: tablica NxN np. 512x512 elementów jednobitowych (również reprezentacja rastrowa).

- **Klasa 3:**

Krzywe dyskretne - zbiór punktów (pikseli) rastru prostokątnego z których każdy (oprócz punktów końcowych) posiada nie mniej niż 2 i nie więcej niż 3 sąsiadów odpowiednio skonfigurowanych. Punkty końcowe: 1-2 sąsiadów.
Krzywe otwarte, krzywe zamknięte.

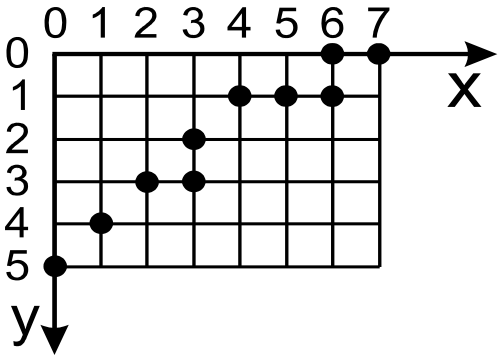
- **Klasa 4:**

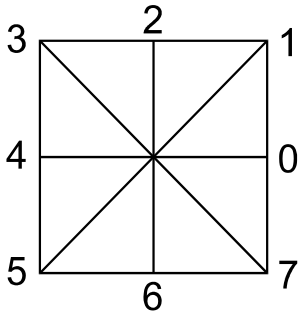
Punkty lub wieloboki. Punkty tak od siebie oddalone, że nie mogą być reprezentowane przez kod łańcuchowy.
Reprezentacja: tablica współrzędnych punktów. Łączenie prostymi lub krzywymi o zadanych parametrach.

Krzywa dyskretna - zbiór punktów (piksli) siatki prostokątnej (rastru prostokątnego) z których każdy (oprócz punktów końcowych) posiada **nie mniej** niż 2 i **nie więcej** niż 3 sąsiadów odpowiednio skonfigurowanych (w sensie sąsiedztwa 8-mio lub 4-spójnego). Punkty końcowe: 1-2 sąsiadów.

Krzywe dzielimy na: otwarte, zamknięte.

Kierunki:





Reprezentacja krzywych

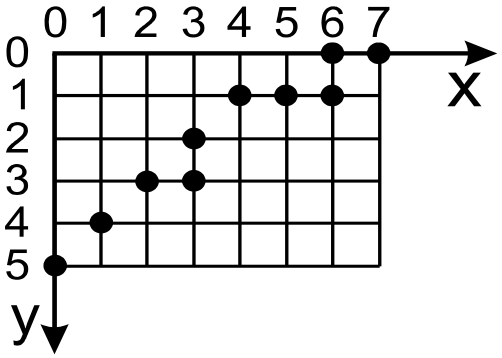
- Ciąg par współrzędnych x, y kolejnych punktów krzywej

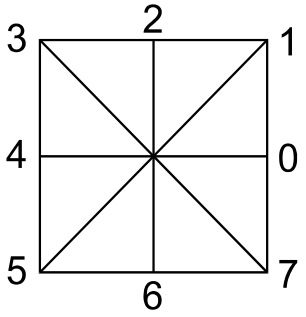
$(x_1, y_1), (x_2, y_2),....., (x_n, y_n),$
(0,5), (1,4),.....,(7,0) (krzywa z przykładu)

- Kod łańcuchowy (*chain code*) o stałej długości (3 bity/punkt)

(0,5) 0010010000010001000.....000
(0,5) - współrzędne punktu początkowego krzywej z przykładu
001 - kod kierunku „1”
Długość kodu **nie zależy** do kształtu krzywej (określonego zmianami kierunków pomiędzy kolejnymi punktami krzywej).

• Różnicowy kod łańcuchowy
(o zmiennej długości, średnio 2 bity / punkt, długość kodu zależy od kształtu krzywej).





Przypisania:

zmiana nachy-lenia	0	+1	-1	+2	-2	+3	-3	4
kod	0	01	011	0111	01111	011111	0111111	01111111

+1, +2, +3, 4 - **zmiana** nachylenia o 1,2,3,4 w kierunku dodatnim
- 1, -2, -3, 4 - **zmiana** nachylenia o 1,2,3,4 w kierunku ujemnym

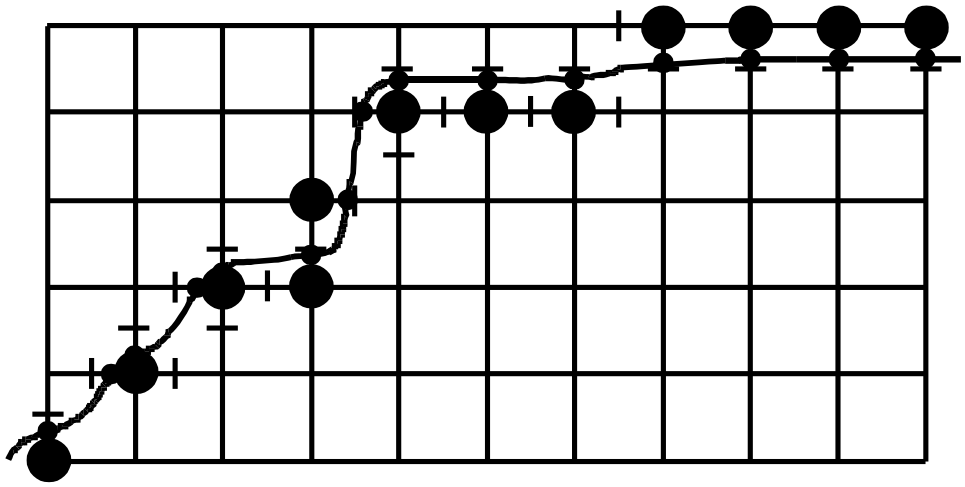
Krzywa z przykładu:

(0,5) 001 0 011011101101100111101111

Dla jednoznacznego opisu krzywej powyższy kod musi zawierać:
(0,5) - współrzędne punktu początkowego krzywej z przykładu
001 - kod łańcuchowy
0 - różnicowy kod łańcuchowy

Technika dyskretyzacji krzywych

Schemat ***dyskretyzacji Freemana*** (w procesie akwizycji obrazu)
Zasada: badanie każdego punktu przecięcia się krzywej z linią łączącą dwa kolejne węzły siatki (*rastru*). Wybór węzła rastru leżącego **bliżej** punktu przecięcia. Wybrany węzeł należy do piksli tworzących krzywą dyskretną.
Punkt niejednoznaczności (*ambiguity point*) - punkt przecięcia jednakowo odległy od obu rozważanych węzłów siatki (rastru). W tym przypadku wybór węzła do utworzenia krzywej dyskretnej następuje według dodatkowej reguły (np. prawy z dwóch węzłów tworzących odcinek poziomy lub górny z dwóch węzłów tworzących odcinek pionowy.



• Punkt przecięcia krzywej z linią siatki

● Punkt krzywej dyskretnej

Tworzenie obrazów na podstawie ich reprezentacji

- Grafika wektorowa (tworzenie obrazów **klas 3 i 4**)

Niech:
$S_1,S_2,....,S_n$ - sekwencje rozkazów odpowiadających poszczególnym ***n*** obiektom tworzącym obraz.
Wtedy:
Program sterujący wyświetlaniem obrazu jest postaci:

```
START:
        S1
        S2
        .....
        ......
        Sn
        if   brak przerwania then goto START
end.
```

Modyfikacja kształtu i - tego obiektu **bez ingerencji** w pozostałe sekwencje. Częstotliwość odtwarzania jest odwrotnie proporcjonalna do długości pętli.
Zajętość pamięci zależy od rodzaju obiektów na obrazie.

- Grafika rastrowa - tworzenie obrazów wielo- lub dwupoziomowych (binarnych), czyli **klasy 1 i 2**.

Rastrowe urządzenia obrazowe - brak możliwości wyświetlania wektorowego
W tym przypadku stosowana jest symulacja grafiki wektorowej
Cechy urządzeń rastrowych: duża pamięć, jedna komórka pamięci odpowiada jednemu pikselowi (pamięć obrazu).
Zajętość pamięci nie zależy od rodzaju obiektów na obrazie

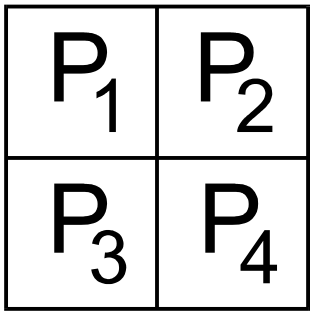
Instrukcje:
read (I,x,y,z) - czytaj komórkę **I** oraz określ **z** na podstawie zawartości **I** (x,y są określone adresem **I**)
Później wykonaj instrukcję zapisu **write** (x,y,z).
Główna pętla wyświetlania:

```
START:
        FOR wszystkich komórek I pamięci DO
                BEGIN
                READ (I,x,y,z)
                WRITE (x,y,z)
END.
```

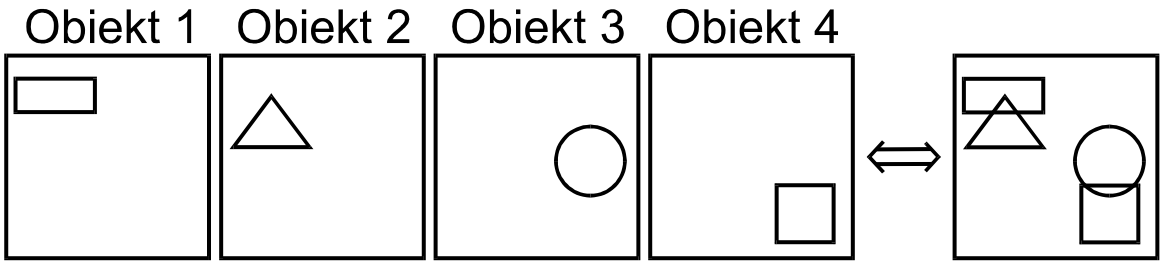
Jeśli przez N oznaczymy wielkość pamięci odnawiającej, to pętla zawiera N par instrukcji **read write**.
Modyfikacja kształtu obiektu **ingeruje** w postać innych obiektów.

Rozwiązanie problemu:

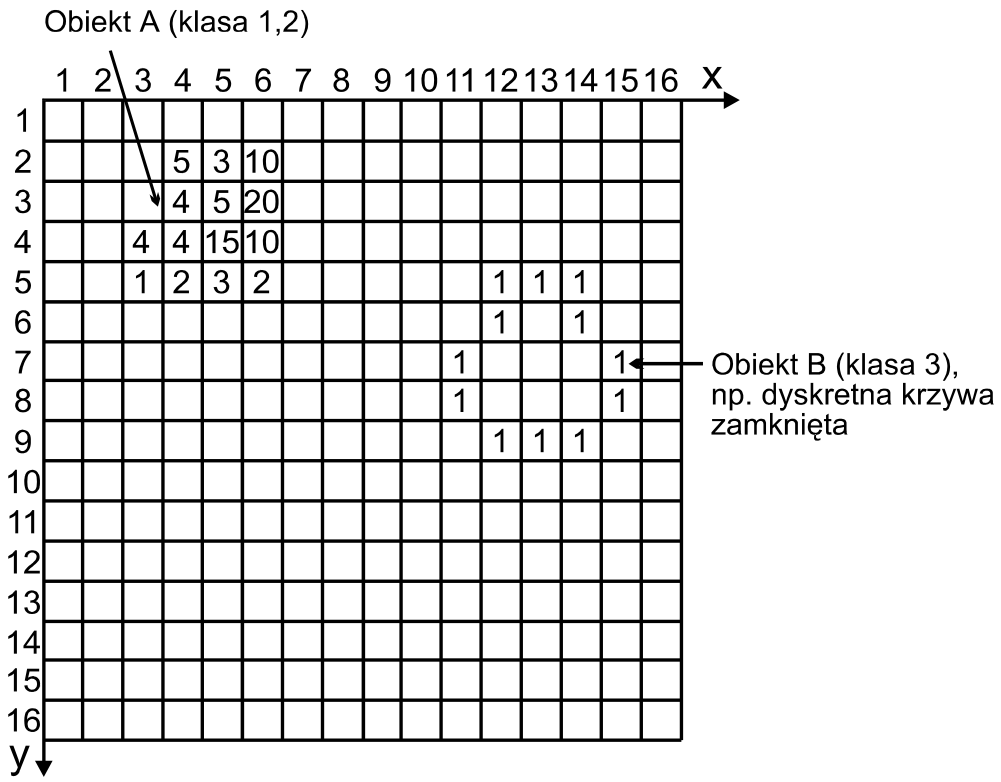
Podział pamięci na strefy i tworzenie obrazu każdego obiektu w oddzielnej strefie.



P_1 , P_2 , P_3 , P_4 - dotyczą tych samych współrzędnych ekranu



Przykład: Obraz 16x16 punktów (pikseli), 256 poziomów jasności.



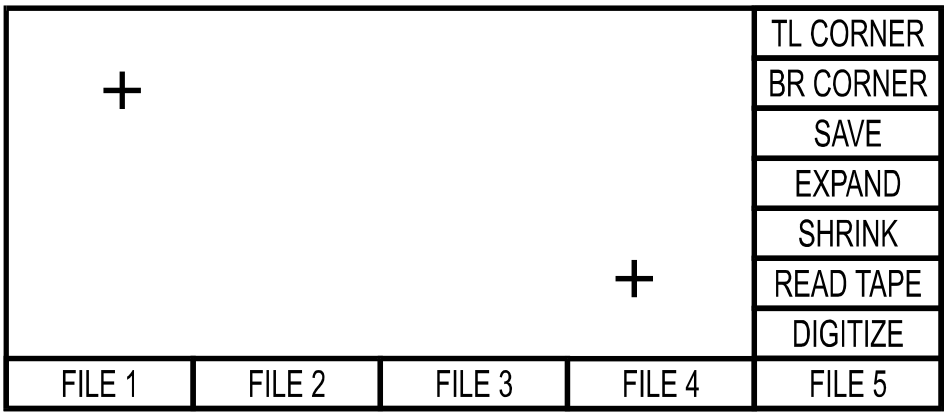
Reprezentacja rastrowa:
Jeden piksel obrazu zajmuje jedną komórkę (jednobajtową) pamięci.
Zawartość pamięci 16x16x1 bajt=256 bajtów.
Na oddzielne przechowywanie zarówno obiektu A jak i obiektu B potrzeba 256 bajtów.
Tablica jednowymiarowa: Obiekt A
nr elementu: 1 2 ..... 16 17 18 19 20 21 22 23 256
wartość elementu: 0 0 ..... 0 0 0 0 5 3 10 0 0
Reprezentacja wektorowa:
Jeden piksel obiektu - współrzędne x, y oraz poziom jasności z.
Obraz - tablica elementów trójskładnikowych:
Obiekt A
(4,2,5), (5,2,3), ..., (6,5,2)

Zadanie:
Obliczyć zajętość pamięci (w bajtach) przez obiekty A i B w obrazie 256x256 przy dwóch rodzajach reprezentacji.

Edytor obrazowy

Program do wprowadzania i modyfikacji danych obrazowych (dla porównania: edytor tekstowy służy do wprowadzania i modyfikacji tekstu)

Dyskretyzator, urządzenie wyświetlające, urządzenie do wczytywania elementów obrazu (urządzenie to realizuje rozkaz READP (x,y) umieszczenia w argumentach x, y współrzędnych punktu wskazanego przez użytkownika na ekranie)



Tryb działania:

READ TYPE DIGITIZE - program czyta zbiór z taśmy lub wprowadza dane z dyskretyzatora.

TL CORNER, BR CORNER - program wykonuje READP (x,y) dla drugiego znacznika aby uzyskać wartości współrzędnych lewego górnego lub prawego dolnego rogu obrazka.

SAVE - program czyta pozycję drugiego znacznika, w celu określenia nazwy zbioru, w którym będzie przechowywany obraz.

SHRINK - zmiana obrazu na jedną ćwiartkę.

EXPAND - powiększenie

Edytor punktowy

Program do wprowadzania, modyfikacji i adresowania (wskazywania) punktów.
Struktura danych dla edytora punktowego:
– Tablica dwuwymiarowa współrzędnych (x,y) punktów
– Połączona lista (łatwiejsza dla użytkownika).
Elementy listy są czteroskładnikowe:
– x - współrzędna x,
– y - współrzędna y,
– p - adres poprzedniego elementu na liście
– n - adres następnego elementu na liście.
Realizacja struktury - cztery tablice jednowymiarowe z indeksem tablicy służącym jako adres.

Przykład:

(5,8,0,2) - element poprzedni nie istnieje
(11,22,1,3)
(14,18,2,0) - element następny nie istnieje.

Wprowadzenie punktu (9,10) między pierwszym a drugim punktem daje następującą listę:

(5,8,0,4); (11,22,4,3); (14,18,2,0); (9,10,1,2)

Literatura podstawowa:

M. Doros, Przetwarzanie obrazów, Skrypt WSISIZ, Warszawa 2005.

(w szczególności rozwiązać Zadanie 10 na str.138)

Literatura uzupełniająca:

T.Pavlidis, Grafika i Przetwarzanie Obrazów, WNT Warszawa 1987.

Praca domowa

Zadanie

Na siatce o rozmiarach 8x8 węzłów naszkicować dwie sześciopikslowe krzywe i podać ich kody łańcuchowe o stałej długości (z uwzględnieniem lokalizacji piksla początkowego). Rozważyć dwa przypadki:

a) Możliwe jest wystąpienie każdego z 8 kierunków,
b) Możliwe jest wystąpienie jedynie 4 (wybranych z 8) kierunków.

Dla każdego z przypadków:

- zobrazować wykorzystywane kierunki oraz określić minimalną liczbę bitów wystarczającą do zakodowania każdego z tych kierunków,
- podać opis krzywej za pomocą kodu o minimalnej liczbie bitów oraz liczbę bitów w kodzie