

# kurs języka Java

## edycja grafu

Instytut Informatyki  
Uniwersytetu Wrocławskiego

Paweł Rzechonek

---

Co to jest graf? Graf to kropki i kreski, które łączą kropki. Trochę mniej infantylne wyjaśnienie jest takie, że graf to obiekty (kropki) i relacje między tymi obiektami (kreski). W teorii grafów obiekty są nazywane wierzchołkami a relacje krawędziami – sam graf  $G = (V, E)$  jest zatem parą dwóch zbiorów: zbioru wierzchołków  $V$  i zbioru krawędzi  $E$ . Liczbę wierzchołków w grafie oznaczamy przez  $n = |V|$  a same wierzchołki najczęściej numerujemy kolejnymi liczbami naturalnymi  $V = \{0, 1, \dots, n-1\}$ . Pojedyncza krawędź to para wierzchołków  $e = \{v_i, v_j\}$ , gdzie  $v_i, v_j \in V$  oraz  $i \neq j$ ; liczba krawędzi w grafie prostym (bez pętli i krawędzi wielokrotnych) jest oznaczana przez  $m = |E|$ , przy czym  $0 \leq m \leq n \cdot (n-1)/2$ ; zbiór krawędzi jest podzbiorem wszystkich możliwych krawędzi  $E \subseteq \{\{v_i, v_j\} : v_i, v_j \in V, i \neq j\}$ ; jeśli  $m = 0$ , to graf jest pusty (nie zawiera żadnej krawędzi) a jeśli  $m = n \cdot (n-1)/2$  to graf jest pełny (każda para wierzchołków jest połączona krawędzią).

Napisz aplikację okienkową w technologii Swing, która pozwoli graficznie prezentować i edytować grafy. Graf ma być przedstawiony w postaci kółek (wierzchołki) połączonych odcinkami (krawędzie) na powierzchni kreślarskiej. Jako obiektu, na którym będzie ilustrowany graf użyj obiektu `JLabel` osadzonego w panelu przewijalnym `JScrollPane`, a sam rysunek wykonaj na obiekcie `BufferedImage` o rozmiarach 1024x1024.

```
class Rysunek extends JLabel {  
    private BufferedImage obraz = new BufferedImage(1024, 1024, ...);  
    private Graf graf;  
    public Rysunek(Graf g) {  
        setIcon(new ImageIcon(obraz));  
        // ...  
    }  
}
```

Graf konstruujemy za pomocą myszy, co wiąże się z funkcjonalnością graficzną aplikacji:

- podwójne kliknięcie (double click) w pusty obszar powoduje wstawienie nowego wierzchołka do grafu (numer nowego wierzchołka to kolejna liczba naturalna) i umiejscowienie go w miejscu kliknięcia;

- podwójne kliknięcie (double click) w obszar wierzchołka powoduje usunięcie tego wierzchołka z grafu (co może się wiązać z przenumerowaniem części wierzchołków);
- przeciągnięcie myszy (drag and drop) z obszaru jednego wierzchołka na obszar innego powoduje dodanie (jeśli nie istnieje) albo usunięcie (jeśli istnieje) krawędzi;
- przeciągnięcie myszy (drag and drop) z obszaru jednego wierzchołka na obszar pusty powoduje zmianę lokalizacji wierzchołka na rysunku (nowa pozycja ustalona w miejscu puszczenia przycisku myszy).

Problem trafiania myszą w wierzchołek można rozwiązać w taki sposób, żeby współrzędne wierzchołka na rysunku zawsze były podzielne przez 64; ze względu na czytelność rysunku grafu wierzchołki nie powinny być umieszczane na skraju powierzchni kreślarskiej; sam okrąg reprezentujący wierzchołek powinien gwarantować nienakładanie się na siebie sąsiadujących bezpośrednio na rysunku wierzchołków, więc średnica takiego okręgu może wynosić 32 (wewnątrz będzie więc wystarczająco miejsca na wpisanie numeru wierzchołka). Można też ograniczyć liczbę wszystkich wierzchołków w grafie do 100.

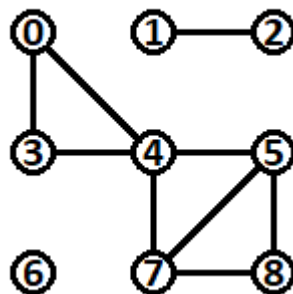
Najważniejszą funkcjonalnością aplikacji ma być utrwalanie grafu w pliku na dysku. Edytowany graf powinno dać się w dowolnym momencie działania aplikacji zapisać w pliku tekstowym oraz dodatkowo wykorzystać serializację do chwilowego przzerwania edycji (zamknięcie programu a potem jego uruchomienie z ostatnio edytowanym grafem):

- przed zamknięciem aplikacji program może zapisać na dysku (decyzję podejmuje użytkownik) w pliku `graf.ser` stan rysowanego grafu (wierzchołki wraz z ich lokalizacją na rysunku i krawędzie) za pomocą mechanizmu serializacji;
- po uruchomieniu aplikacji program może odczytać z dysku z pliku `graf.ser` stan rysowanego grafu (wierzchołki wraz z ich lokalizacją na rysunku i krawędzie) za pomocą mechanizmu deserializacji (o ile użytkownik zapisał ten stan kończąc pracę z aplikacją ostatnim razem);
- w trakcie działania aplikacja powinna mieć możliwość zapisania grafu do określonego pliku tekstowego w następującym formacie:

- w pierwszej linijce zapisz liczbę wierzchołków  $n$  (numerowanych od 0 do  $n-1$ ),
  - w następnych  $n$  liniijkach zapisz sąsiadów kolejnych wierzchołków;
- na przykład dla poniższego grafu w pliku tekstowym powinien znaleźć się taki zapis:

```
9
3 4
2
1
0 4
0 3 5 7
4 7 8

4 5 8
5 7
```



Graf w aplikacji pamiętaj w postaci zbioru krawędzi i zbioru wierzchołków używając kolekcji standardowych. Wierzchołek przechowuje swój identyfikator i współrzędne na rysunku:

```

class Wierzchołek implements Serializable, Comparable<Wierzchołek> {
    int nr, x, y;
    // ...
    public int compareTo(Wierzchołek w) {
        return nr < w.nr ? -1 : nr > w.nr ? 1 : 0;
    }
}

```

Krawędź przechowuje referencje do wierzchołków, które ze sobą łączy (zachowując niezmiennik, że numer wierzchołka początkowego jest mniejszy od numeru wierzchołka końcowego, łatwiej będzie zdefiniować metodę porównującą):

```

class Krawędź implements Serializable, Comparable<Krawędź> {
    Wierzchołek pocz, kon;
    // ...
    public int compareTo(Krawędź k) {
        switch (pocz.compareTo(k.pocz)) {
            case -1: return -1;
            case 1: return 1;
            default: return kon.compareTo(k.kon);
        }
    }
}

```

Funkcjonalność grafu powinna pozwalać na dynamiczne wprowadzanie zmian:

```

class Graf implements Serializable {
    private List<Wierzchołek> wierzch = new ArrayList<>();
    private Set<Krawędź> kraw = new TreeSet<>();
    public void wstawWierzchołek(int x, int y) { ... }
    public void usuńWierzchołek(int n) { ... }
    public void wstawKrawędź(int u, int v) { ... }
    public void usuńKrawędź(int u, int v) { ... }
    // ...
}

```

Wstawiając do grafu nowy wierzchołek należy nadać mu numer największy i dopisać na koniec listy wierzchołków (numer wierzchołka powinien taki sam jak jego pozycja na liście). Natomiast usuwając wierzchołek należy przenieść numery wierzchołki, które miały większe numery (zmniejszenie każdego numeru o 1); pamiętaj by przed usunięciem wskazanego wierzchołka usunąć wszystkie krawędzie połączone z tym wierzchołkiem.