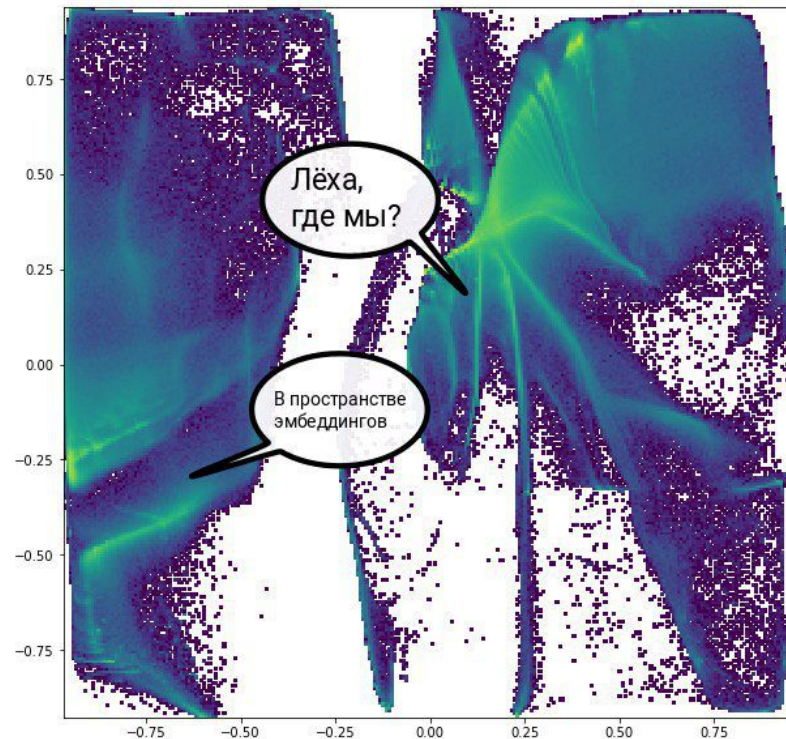


Визуализация графа пакетом Graphviz

Кучин Александр

Важность визуализации графов

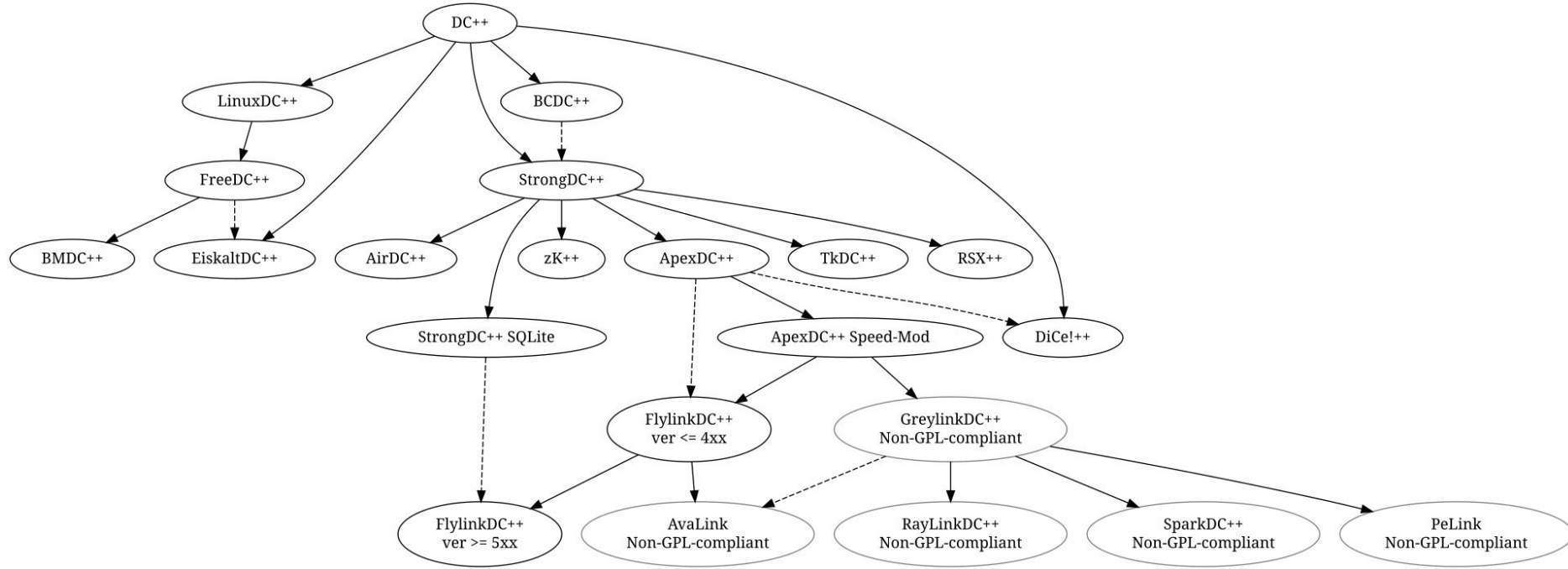
- Наглядность
- Выявление закономерностей



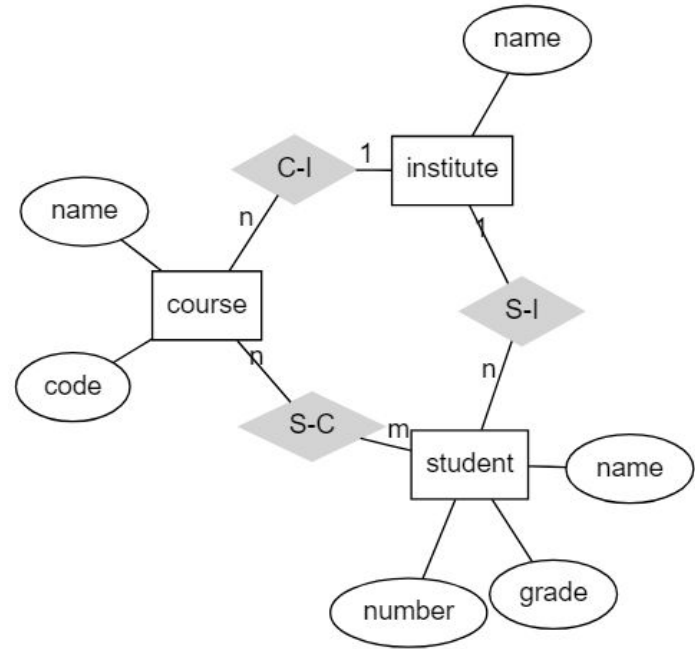
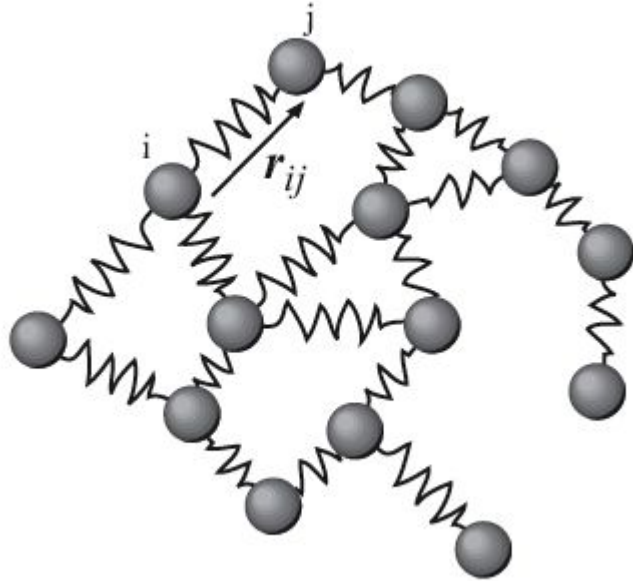
Graphviz

- Пакет утилит для автоматической визуализации графов
- Основные компоненты:
 - dot – для направленных графов
 - neato – для ненаправленных графов небольших размеров
 - twopi – для радиальных графов
 - circo – круговые диаграммы
 - fdp – для ненаправленных графов в общем случае

dot



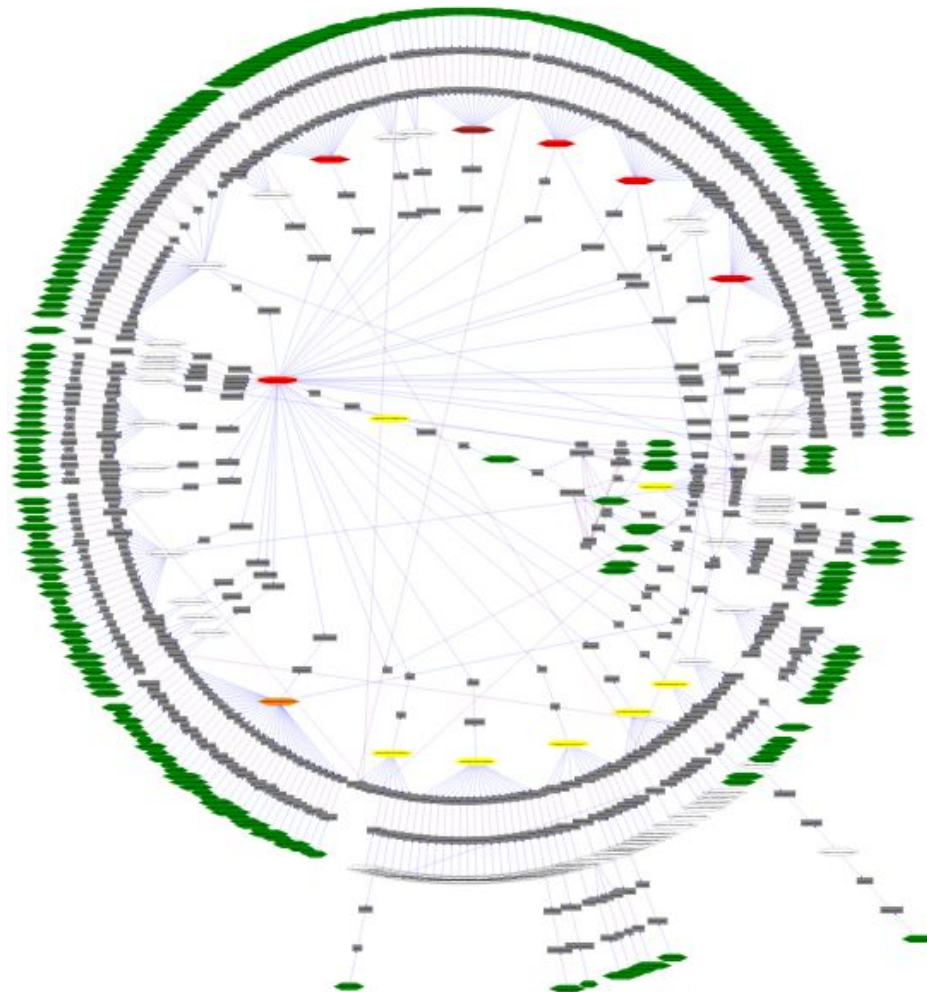
neato



Entity Relation Diagram
drawn by NEATO

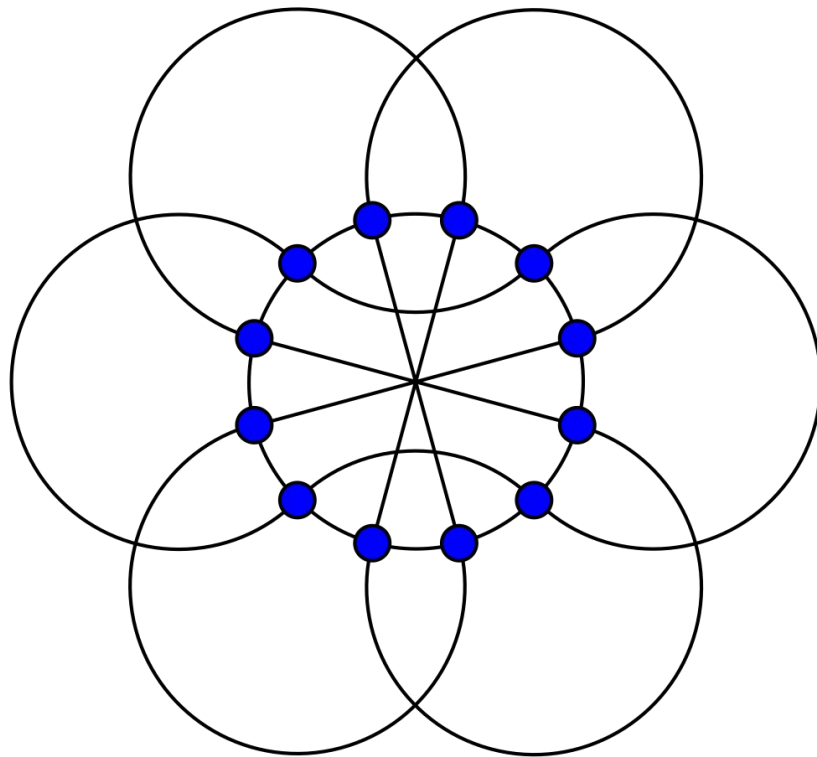
twopi

- узлы по
концентрическим
окружностям в
зависимости от
расстояния от ядра



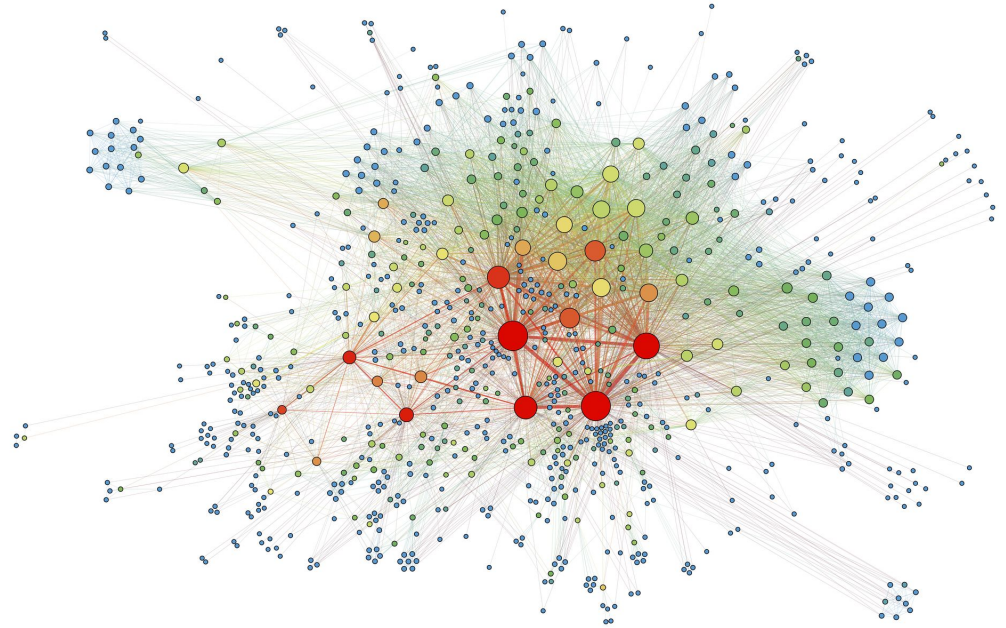
circo

- циклические структуры



fdp

- алгоритмы force directed
- используются физические законы



Визуализация графа с помощью dot

- Основные термины:
 - digraph – ориентированный граф
 - graph – неориентированный граф
 - subgraph - самостоятельный подграф
 - node - узел
 - edge - ребро

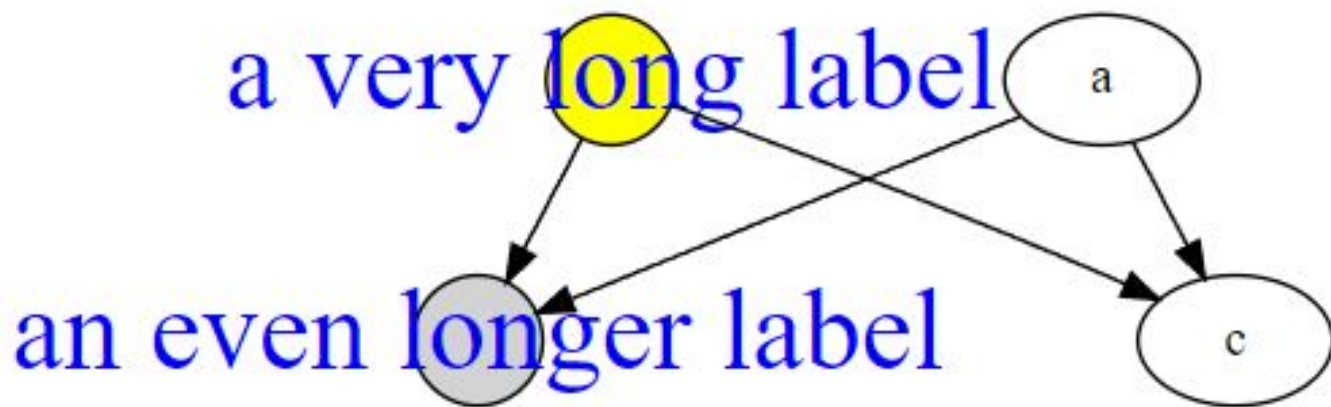
Описание графа на dot

```
digraph G {  
  A -> B;  
  B -> C;  
  A -> C;  
}
```

Атрибуты

- формат атрибута – subject[name = value]

```
digraph G {  
  {  
    node [margin=0 fontcolor=blue fontsize=32 width=0.5 shape=circle style=filled]  
    b [fillcolor=yellow fixedsize=true label="a very long label"]  
    d [fixedsize=shape label="an even longer label"]  
  }  
  a -> {c d}  
  b -> {c d}  
}
```





box



polygon



ellipse



oval



circle



point



egg



triangle

plaintext

plaintext

plain

plain



diamond



trapezium



parallelogram



house



pentagon



hexagon



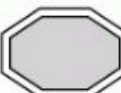
septagon



octagon



doublecircle



doubleoctagon



tripleoctagon



invtriangle

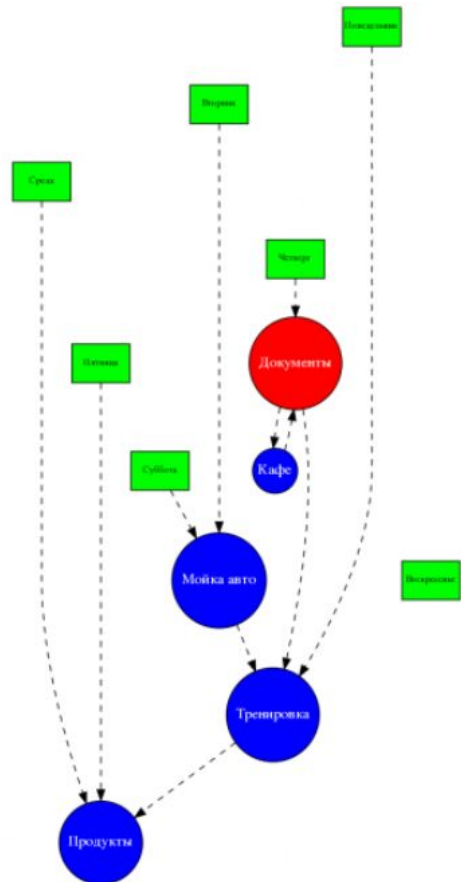


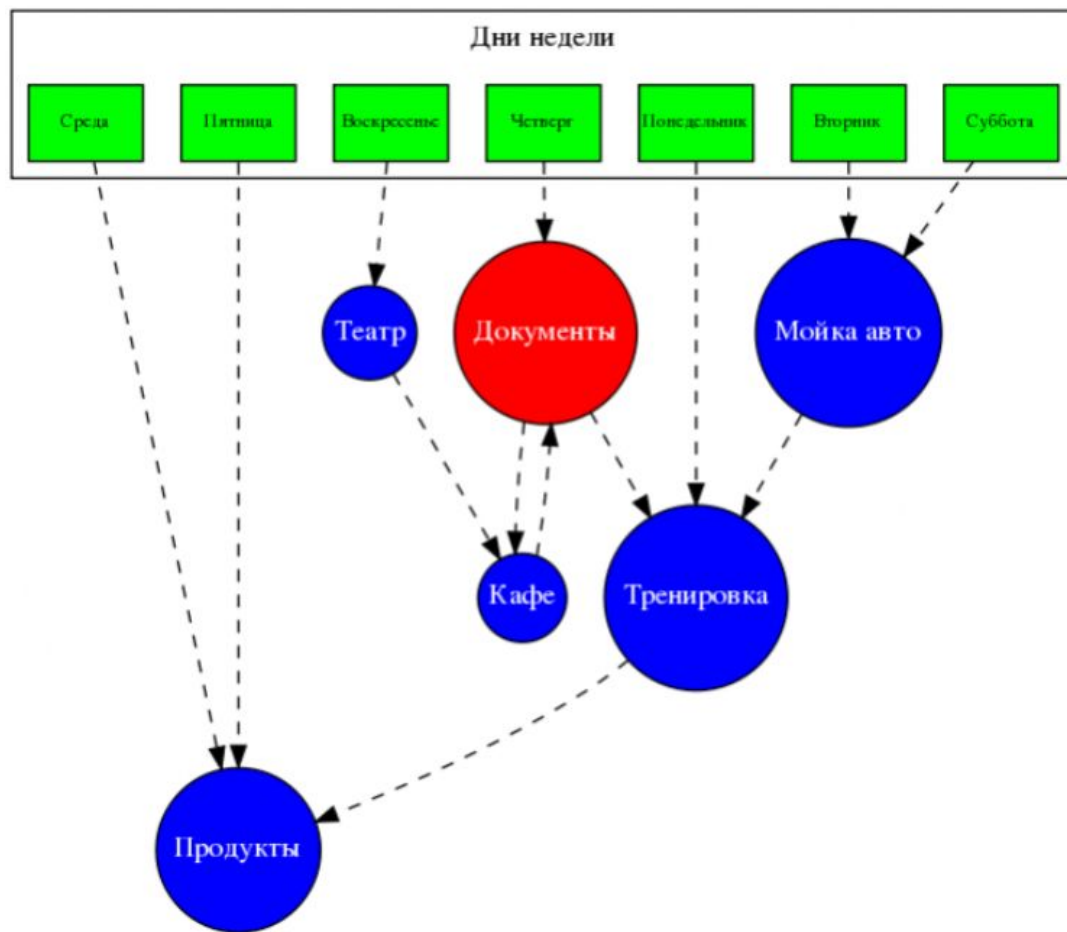
invtrapezium



invhouse

```
{ rank="same"; "1"; "Понедельник"; }  
{ rank="same"; "2"; "Вторник"; }  
{ rank="same"; "3"; "Среда"; }  
{ rank="same"; "4"; "Четверг"; }  
{ rank="same"; "5"; "Пятница"; }  
{ rank="same"; "6"; "Суббота"; }  
{ rank="same"; "7"; "Воскресенье"; }
```





```

digraph MyPlan {
  node [shape="circle", style="filled", fillcolor="blue", fontcolor="#FFFFFF", margin="0.01"];
  edge [style="dashed"];

  "Документы" [fillcolor="red"];

  subgraph cluster_week {
    node [shape="box", style="filled", fillcolor="green", fontcolor="black", fontsize="9"];
    label = "Дни недели";
    "Понедельник";
    "Вторник";
    "Среда";
    "Четверг";
    "Пятница";
    "Суббота";
    "Воскресенье";
  }

  "Понедельник"->"Тренировка"->"Продукты";
  "Вторник"->"Мойка авто";
  "Среда"->"Продукты";
  "Четверг"->"Документы"->"Тренировка";
  "Документы"->"Кафе"->"Документы";
  "Пятница"->"Продукты";
  "Суббота"->"Мойка авто"->"Тренировка";
  "Воскресенье"->"Театр"->"Кафе";
}

```


Вывод графа

- `cmd [flags] input files`
- Пример: `circo -Tpng cycle_graph.dot -O`
- Пример: `$ dot -Tsvg input.dot`

Автоматизация создания dot файла

```
void printDotFile(int graph[MAX_CITIES][MAX_CITIES], int numCities, const char *filename) {  
    FILE *file = fopen(filename, "w");  
    if (!file) {  
        exit(EXIT_FAILURE);  
    }  
  
    fprintf(file, "graph G {\n");  
  
    for (int i = 0; i < numCities; i++) {  
        for (int j = 0; j < numCities; j++) {  
            if (graph[i][j] == 1) {  
                fprintf(file, "    City%d -> City%d;\n", i, j);  
            }  
        }  
    }  
  
    fprintf(file, "}\n");  
    fclose(file);  
}
```

Интеграция и библиотека

```
#include <stdio.h>
#include <graphviz/cgraph.h>

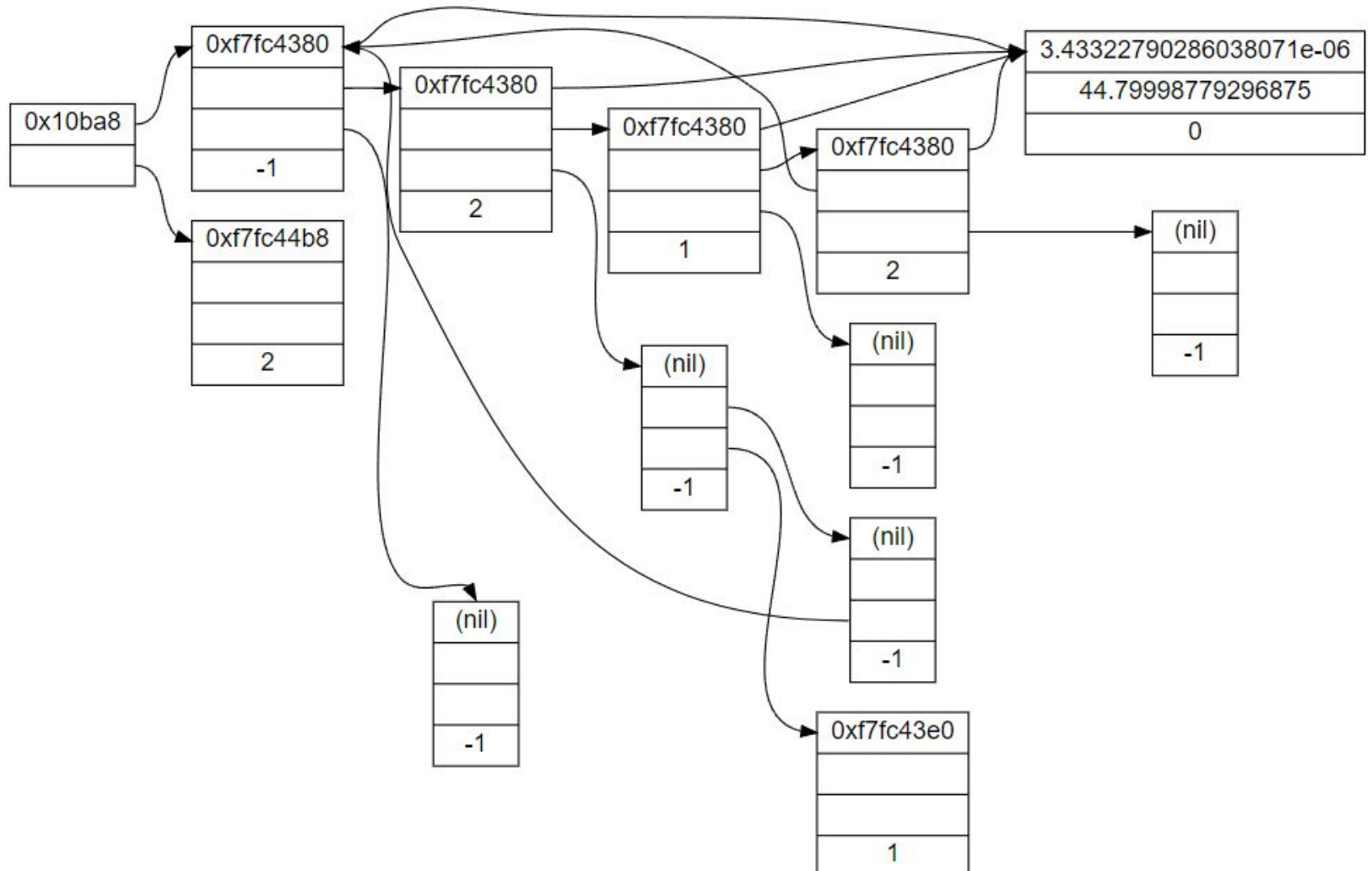
int main() {
    Agraph_t *g = agopen("G", Agdirected, NULL);
    Agnode_t *n1 = agnode(g, "Node1", TRUE);
    Agnode_t *n2 = agnode(g, "Node2", TRUE);

    aedge(g, n1, n2, NULL, TRUE);

    FILE *f = fopen("graph.dot", "w");
    agwrite(g, f);

    fclose(f);
    agclose(g);

    return 0;
}
```



```

"node9" [
  label = "<f0> (nil) | | -1"
  shape = "record"
];
"node10" [
  label = "<f0> (nil) | <f1> | <f2> | -1"
  shape = "record"
];
"node11" [
  label = "<f0> (nil) | <f1> | <f2> | -1"
  shape = "record"
];
"node12" [
  label = "<f0> 0xf7fc43e0 | | 1"
  shape = "record"
];
"node0":f0 -> "node1":f0 [
  id = 0
];
"node0":f1 -> "node2":f0 [
  id = 1
];

```