

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана»
(МГТУ им. Н.Э. Баумана)

Факультет: Информатика и системы управления

Кафедра: Теоретическая информатика и компьютерные технологии

Лабораторная работа №7
по курсу: «Языки и методы программирования»
«Разработка простейшего класса на C++»

Выполнил:
Студент группы ИУ9-21Б

Проверил:
Посевин Д. П.

Москва, 2024

1. Цель

Целью данной работы является изучение объектно-ориентированных возможностей языка C++.

2. Персональный вариант

Матрица смежности для простого орграфа с операциями:

1. определение, ведёт ли дуга из i -той вершины в j -тую (операция индексации)
2. добавление вершины;
3. удаление вершины;
4. определение, содержит ли граф циклы.

3. Решение

3.1. Код

```
#include <iostream>
#include <vector>

using namespace std;

typedef vector<int> vi;

class Graph {
private:
    vector<vi> mat_;
    int connections_ = 0;

    bool DFS(int v, vector<bool> &used) {
        if (used[v])
            return true;

        bool cycle = false;

        used[v] = true;
        for (int u = 0; u < mat_.size(); ++u) {
            if (mat_[v][u]) {
                cycle = cycle || DFS(u, used);
            }
        }

        return cycle;
    }
}
```

```

void setSize(int n) {
    mat_.resize(n);
    for (auto &l : mat_)
        l.resize(n);
}

public:
Graph() {}

void addEdge(int u, int v) {
    if (!(u < mat_.size() && v < mat_.size()))
        setSize(max(u + 1, v + 1));
    connections_ += mat_[u][v] == 0;
    mat_[u][v] = 1;
}

void removeEdge(int u, int v) {
    if (!(u < mat_.size() && v < mat_.size()))
        return;
    connections_ -= mat_[u][v] == 1;
    mat_[u][v] = 0;
}

bool hasConnections(int u, int v) {
    if (!(u < mat_.size() && v < mat_.size()))
        return false;
    return mat_[u][v];
}

void print() {
    cout << "Current matrix:\n";
    for (auto &l : mat_) {
        for (auto &v : l) {
            cout << v << " ";
        }
        cout << "\n";
    }

    cout << "Has cycle? " << (hasCycle() ? "yes" : "no") <<
        "\n";
    cout << "Current count of connections: " << connections_
        << "\n";
}

void printDOT() {

```

```

cout << "digraph {\n";

for (int v = 0; v < mat_.size(); ++v)
    cout << "\t" << v << "\n";

for (int v = 0; v < mat_.size(); ++v)
    for (int u = 0; u < mat_.size(); ++u)
        if (mat_[v][u])
            cout << "\t" << v << " -> " << u << "\n";

cout << "}";
}

bool hasCycle() {
    vector<bool> used(mat_.size(), false);
    bool cycle = false;

    for (int v = 0; v < mat_.size(); ++v) {
        if (!used[v]) {
            cycle = cycle || DFS(v, used);
        }
    }
}

return cycle;
}
};

int main() {
    Graph g;

    g.addEdge(0, 1);
    g.addEdge(1, 2);
    g.addEdge(0, 3);
    g.addEdge(0, 4);
    g.addEdge(4, 5);
    g.addEdge(5, 4);
    g.addEdge(4, 6);
    g.print();

    g.removeEdge(5, 4);
    g.print();
}

```

Код 1: main.cc

3.2. Скриншоты

```
Current matrix:  
0 1 0 1 1 0 0  
0 0 1 0 0 0 0  
0 0 0 0 0 0 0  
0 0 0 0 0 0 0  
0 0 0 0 0 1 1  
0 0 0 0 1 0 0  
0 0 0 0 0 0 0  
Has cycle? yes  
Current count of connections: 7  
Current matrix:  
0 1 0 1 1 0 0  
0 0 1 0 0 0 0  
0 0 0 0 0 0 0  
0 0 0 0 0 0 0  
0 0 0 0 0 1 1  
0 0 0 0 0 0 0  
0 0 0 0 0 0 0  
Has cycle? no  
Current count of connections: 6
```

Рис. 1: Пример работы программы

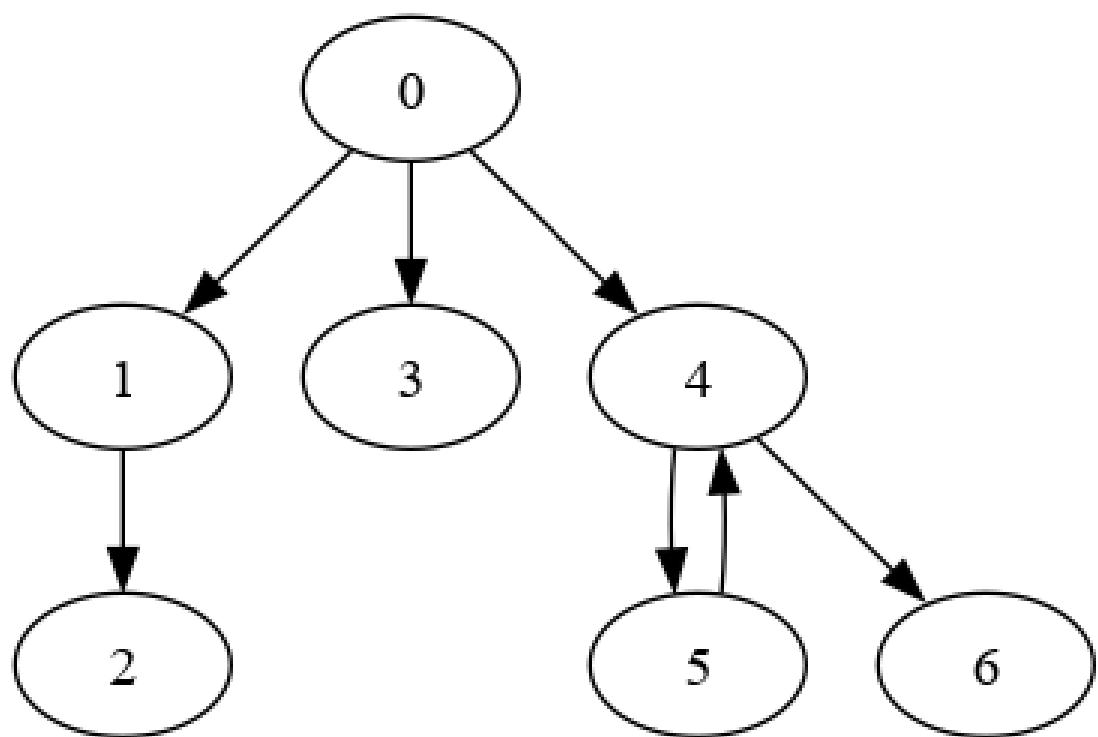


Рис. 2: Граф до удаления ребра

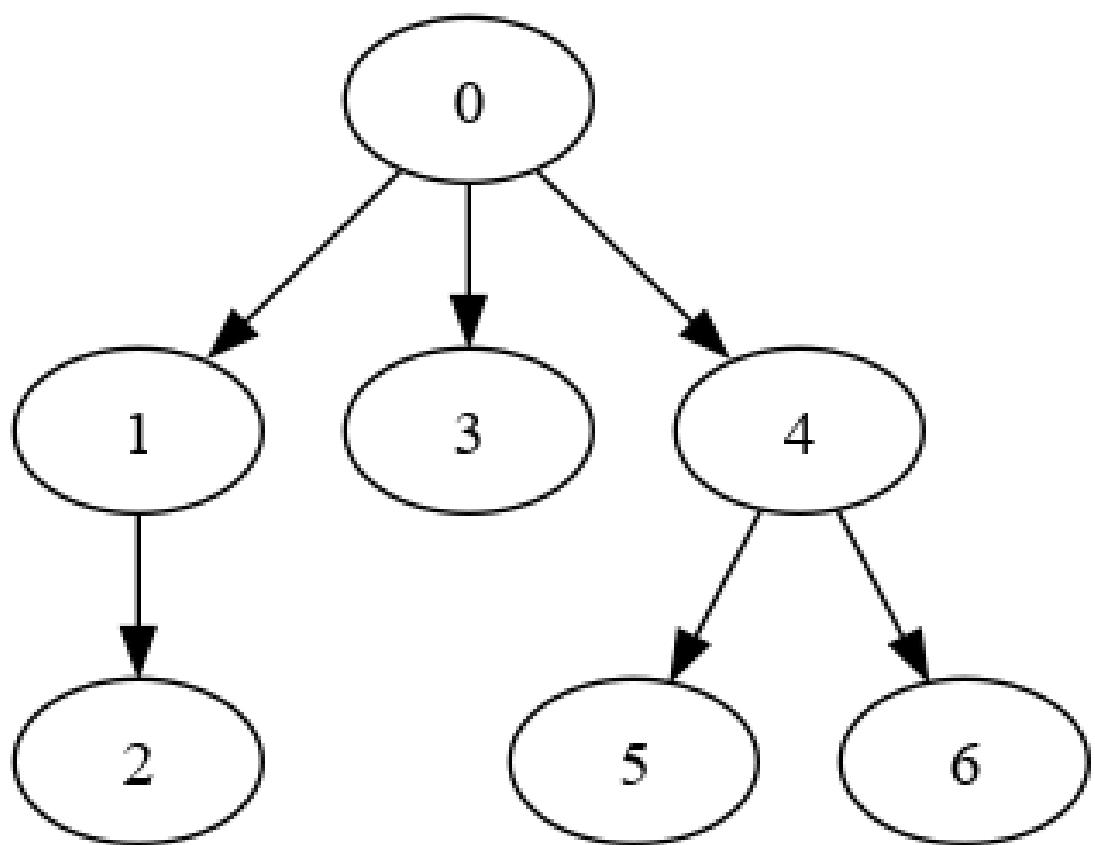


Рис. 3: Граф после удаления ребра