|  |
| --- |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **"МИРЭА – Российский технологический университет"**  **РТУ МИРЭА** |

Институт комплексной безопасности и специального приборостроения

Кафедра КБ-3 «Управление и моделирование систем»

**ОТЧЕТ   
о выполнении лабораторной работы №2**

**«Реализация алгоритмов на графах»**

**по дисциплине   
«Программная реализация нелинейных структур»**

**Вариант № {{task\_num}}**

Выполнил: студент 2 курса

Группы **\_\_\_\_\_\_**

шифр \_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_  
*(фио студента)*

Проверил:

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

Москва 2022 г.

**Вариант № {{task\_num}}.**

**Алгоритм:** **{{task\_algo}}**

**Способ представления графа: {{graph\_definition\_type}}**

**Теория о Графах.**

Граф — это математический объект, который состоит из точек и линий, которые их соединяют. Точки называют вершинами графа, а линии — ребрами.

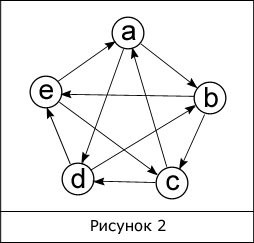
**Вот некоторые важные обозначения, используемые в теории графов:**

* G = (V, E), здесь G – граф, V – его вершины, а E – ребра;
* |V| – порядок (число вершин);
* |E| – размер графа (число рёбер).

В нашем случае (рис. 1) |V|=5, |E|=10;

Когда из любой вершины доступна любая другая вершина, то такой граф называется **неориентированным** связным графом (рис. 1). Если же граф связный, но это условие не выполняется, тогда такой граф называется **ориентированным** или орграфом (рис. 2).

В ориентированных и неориентированных графах имеется понятие степени вершины. **Степень вершины** – это количество ребер, соединяющих ее с другими вершинами. Сумма всех степеней графа равна удвоенному количеству всех его ребер. Для рисунка 2 сумма всех степеней равна 20.

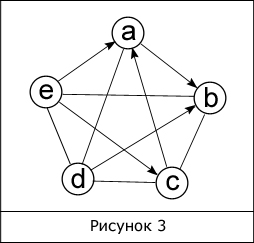
[](https://kvodo.ru/wp-content/uploads/orgraph.jpg)

В орграфе, в отличие от неориентированного графа, имеется возможность двигаться из вершины h в вершину s без промежуточных вершин, лишь тогда, когда ребро выходит из h и входит в s, но не наоборот.

**Ориентированные графы имеют следующую форму записи:**

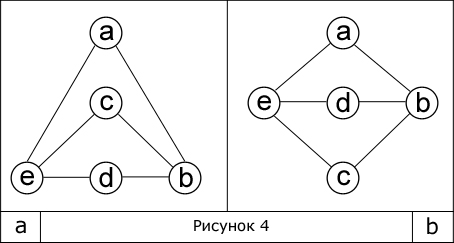
*G = (V, A), где V – вершины, A – направленные ребра.*

Третий тип графов – **смешанные** графы (рис. 3). Они имеют как направленные ребра, так и ненаправленные. Формально смешанный граф записывается так: G = (V, E, A), где каждая из букв в скобках обозначает тоже, что ей приписывалось ранее.

[](https://kvodo.ru/wp-content/uploads/mix_graph.jpg)

В графе на рисунке 3 одни дуги направленные [(e, a), (e, c), (a, b), (c, a), (d, b)], другие – ненаправленные [(e, d), (e, b), (d, c) …].

Два или более графов на первый взгляд могут показаться разными по своей структуре, что возникает вследствие различного их изображения. Но это не всегда так. Возьмем два графа (рис. 4).

[](https://kvodo.ru/wp-content/uploads/izomorph_graph.jpg)

Они эквивалентны друг другу, ведь не изменяя структуру одного графа можно построить другой. Такие графы называются **изоморфными**, т. е. обладающими тем свойством, что какая-либо вершина с определенным числом ребер в одном графе имеет тождественную вершину в другом. На рисунке 4 изображены два изоморфных графа.

Когда каждому ребру графа поставлено в соответствие некоторое значение, называемое весом ребра, тогда такой граф **взвешенный**. В разных задачах в качестве веса могут выступать различные виды измерений, например длины, цены маршруты и т. п. В графическом представлении графа весовые значения указываются, как правило, рядом с ребрами.

В любом из рассмотренных нами графов имеется возможность выделить путь и, причем не один. **Путь** – это последовательность вершин, каждая из которых соединена с последующей посредством ребра. Если первая и последняя вершины совпадают, то такой путь называется циклом. Длина пути определяется количеством составляющих его ребер. Например, на рисунке 4.а путем служит последовательность [(e), (a), (b), (c)]. Этот путь является подграфом, так как к нему применимо определение последнего, а именно: граф G’ = (V’, E’) является подграфом графа G = (V, E), только тогда, когда V’ и E’ принадлежат V, E.

{{p definition\_theory}}

**Листинг программы**

{{task\_code}}

**Скриншоты работы программы.**

{{graph\_png}}

Рисунок 3. - Пример графа

{{example\_out\_screenshot}}

Рисунок 4. - Пример работы программы в режиме example.

{{task\_out\_screenshot}}

Рисунок 5. - Пример работы программы в режиме task.

**Вывод:**

В результате выполнения данной работы была реализована структура Graph предоставляющая удобный интерфейс для работы с графами и их компонентами. Для этого класса был реализован весь необходимый функционал по заданию.

{{task\_outer}}

**Литература:**

1. Буркатовския Ю. Б. Теория графов. — Томск: Издательство Томского политехнического университета, 2014. — Т. 1. — 200 с

2. Дистель Р. Теория графов. — Новосибирск: Издательство Института математики им. С. Л. Соболева СО РАН, 2002. — 336 с. — ISBN 5-86134-101-Х.

3. Касьянов В. Н., Евстигнеев В.А. Графы в программировании: обработка, визуализация и применение. — СПб.: БХВ-Петербург, 2003. — С. 1104. — ISBN 5-94157-184-4.