Пермский филиал федерального государственного автономного образовательного учреждения высшего образования   
«Национальный исследовательский университет   
«Высшая школа экономики»

*Факультет экономики, менеджмента и бизнес-информатики*

Шешин Николай Андреевич

**Отчет по ЗАДАЧЕ №8**

студента образовательной программы «Программная инженерия»

по направлению подготовки *09.03.04 Программная инженерия*

Руководитель

доцент кафедры

информационных

технологий в бизнесе

к.т.н. О.Л. Викентьева

Пермь, 2020 год

1. **Постановка задачи**

Граф задан списком вершин и ребер. Найти в нем какой-либо пустой подграф из K вершин

1. **Анализ**

*Классы входных данных:*

vertexCount – целое положительное число, кол-во вершин

edgesCount – целое положительное число, кол-во ребер

vertexes – массив размером vertexCount, вершины

edges – массив размером edgesCount типа Edge, ребра

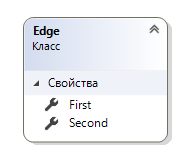
k – целое положительное число, кол-во пустых вершин, сколько надо вывести

*Классы выходных данных:*

vertexes – массив размером vertexCount, некоторые вершины в нем являются пустым графом, они и будут выводиться

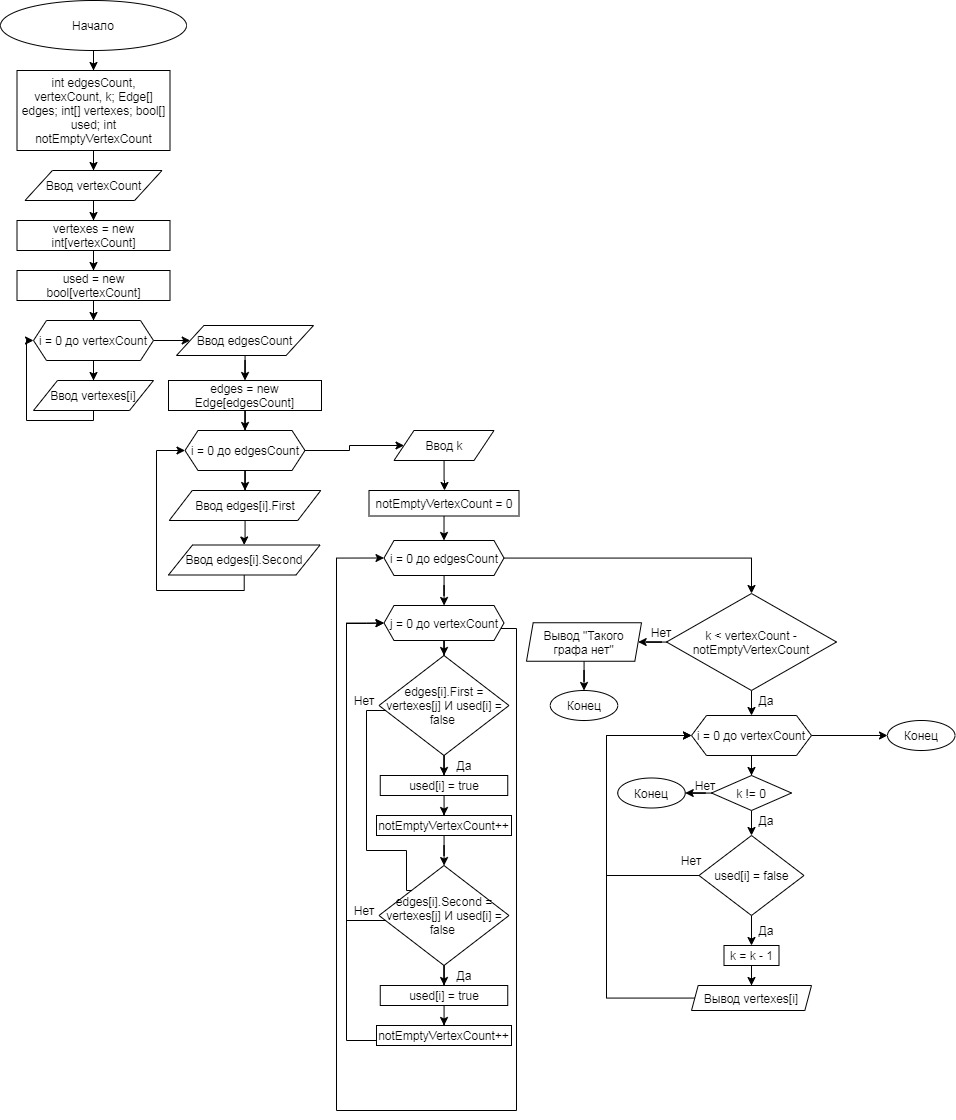
1. **Проектирование**

*Класс Edge:*



* First – целое положительное число, первая вершина ребра
* Second – целое положительное число, вторая вершина ребра

*Метод Main:*

**

1. **Программа**
2. **using** **System**;
3. **namespace** **Exersise8\_30**
4. {
5. **public** **class** **Edge**
6. {
7. **public** **int** First { **get**; **set**; }
8. **public** **int** Second { **get**; **set**; }
9. }
10. **class** **Program**
11. {
12. **static** **void** **Main**(**string**[] args)
13. {
14. Console.WriteLine("Введите кол-во вершин: ");
15. **int** vertexCount = **int**.Parse(Console.ReadLine());
16. **int**[] vertexes = **new** **int**[vertexCount];
17. **bool**[] used = **new** **bool**[vertexCount];
18. Console.WriteLine($"Введите {vertexCount} вершин: ");
19. **for** (**int** i = **0**; i < vertexCount; i++)
20. {
21. vertexes[i] = **int**.Parse(Console.ReadLine());
22. }
23. Console.WriteLine("Введите кол-во ребер: ");
24. **int** edgesCount = **int**.Parse(Console.ReadLine());
25. Edge[] edges = **new** Edge[edgesCount];
26. **for** (**int** i = **0**; i < edgesCount; i++)
27. {
28. edges[i] = **new** Edge();
29. Console.WriteLine($"Введите первую вершину {i + 1} ребра: ");
30. edges[i].First = **int**.Parse(Console.ReadLine());
31. Console.WriteLine($"Введите вторую вершину {i + 1} ребра: ");
32. edges[i].Second = **int**.Parse(Console.ReadLine());
33. }
34. Console.WriteLine("Введите K");
35. **int** k = **int**.Parse(Console.ReadLine());
36. **int** notEmptyVertexCount = **0**;
37. **for** (**int** i = **0**; i < edgesCount; i++)
38. {
39. **for** (**int** j = **0**; j < vertexCount; j++)
40. {
41. **if** (edges[i].First == vertexes[j] && !used[j])
42. {
43. used[j] = **true**;
44. notEmptyVertexCount++;
45. }
46. **if** (edges[i].Second == vertexes[j] && !used[j])
47. {
48. used[j] = **true**;
49. notEmptyVertexCount++;
50. }
51. }
52. }
53. **if** (k <= vertexCount - notEmptyVertexCount)
54. {
55. **for** (**int** i = **0**; i < vertexCount; i++)
56. {
57. **if** (k == **0**)
58. {
59. **break**;
60. }
61. **if** (!used[i])
62. {
63. Console.Write(vertexes[i] + " ");
64. k--;
65. }
66. }
67. }
68. **else**
69. {
70. Console.WriteLine("Такого графа не существует");
71. }
72. }
73. }
74. }