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 №1**

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**

Агент Джеймс Бонд пошел на пенсию, но неугомонный характер требовал новых впечатлений. Поэтому Джеймс Бонд с удовольствием согласился провести мастер-класс в некоторых группах школы «Молодого агента». Тема одного из занятий – работа агента с напарником. В таком опасном деле, как разведка, важно иметь очень надёжного напарника, поэтому напарниками могут стать только агенты, которые максимально близки по возрасту (т.е. два агента не могут стать напарниками, если в группе существует третий агент, который старше одного и младше другого).

Задание Бонда состоит в том, чтобы агенты нашли друг другу напарников таким образом, чтобы у каждого агента был хотя бы один напарник (всего у агента может быть 2 напарника – один младше, и один старше него, но эти двое не считаются напарниками между собой). Очевидно, что группа из 4 и более агентов может поделиться несколькими способами.

После нескольких занятий Бонд узнал способности групп, обучающихся в школе «Молодого агента», и оценил риск раскрытия каждого агента в отдельности. Но специфика работы с напарником такова, что в паре риску подвергается только старший из двух агентов, поэтому группу надо распределить так, чтобы суммарный риск был минимален.

1. **Анализ**

*Классы входных данных:*

n – целое положительное число

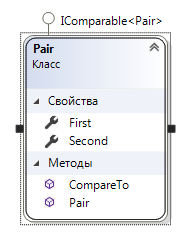
agentInfos – массив класса Pair, хранится информация о агентах

*Классы выходных данных:*

dp – массив типа int, в элементе с номером n – 1 будет находится ответ

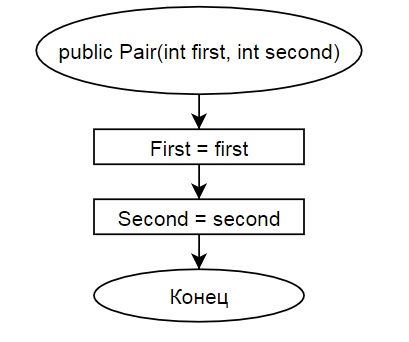
1. **Проектирование**

*Класс Pair:*

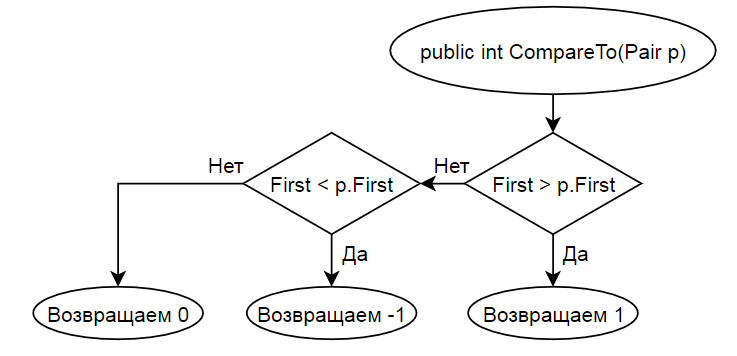


* First – переменная int, здесь хранится возраст для определенного агента
* Second – переменная int, здесь хранится риск для определенного агента

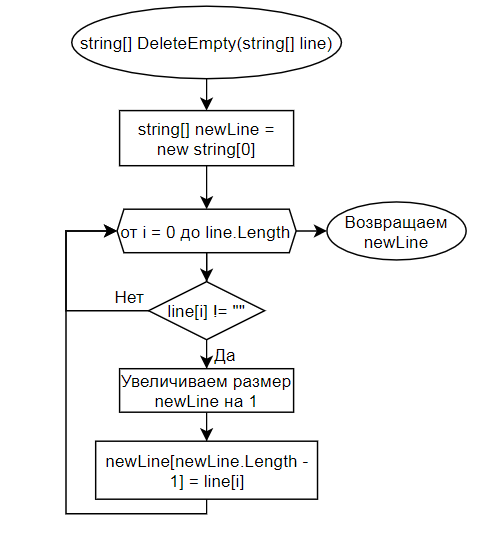
*Конструктор Pair:*



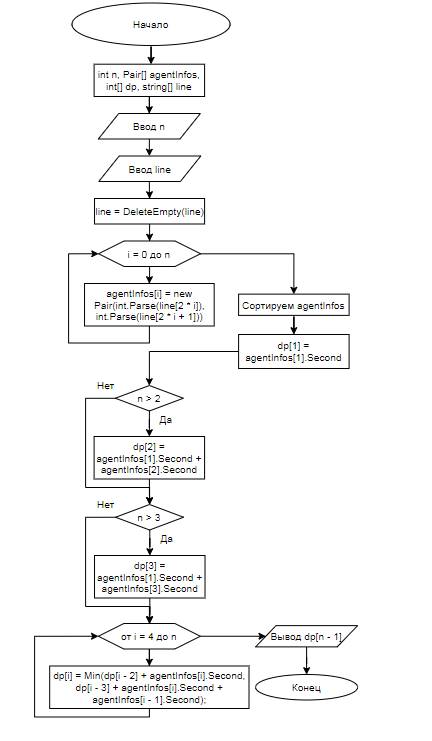
*Метод CompareTo от интерфейса IComparable<Pair>:*



*Метод DeleteEmpty (для удаления пустых строк из массива):*



*Метод Main:*



1. **Программа**
2. **using** **System**;
3. **using** **System.Collections.Generic**;
4. **using** **System.IO**;
5. **namespace** **Agent**
6. {
7. **public** **class** **Pair** : IComparable<Pair>
8. {
9. **public** **int** First { **get**; **set**; }
10. **public** **int** Second { **get**; **set**; }
11. **public** **Pair**(**int** first, **int** second)
12. {
13. First = first;
14. Second = second;
15. }
16. **public** **int** **CompareTo**(Pair pair)
17. {
18. **if** (First > pair.First)
19. **return** **1**;
20. **else** **if** (First < pair.First)
21. **return** -**1**;
22. **else**
23. **return** **0**;
24. }
25. }
26. **class** **Program**
27. {
28. **static** **string**[] **DeleteEmpty**(**string**[] line)
29. {
30. **string**[] newLine = **new** **string**[**0**];
31. **for** (**int** i = **0**; i < line.Length; i++)
32. {
33. **if** (line[i] != "")
34. {
35. Array.Resize(**ref** newLine, newLine.Length + **1**);
36. newLine[newLine.Length - **1**] = line[i];
37. }
38. }
39. **return** newLine;
40. }
41. **static** **void** **Main**(**string**[] args)
42. {
43. StreamReader streamReader = **new** StreamReader("INPUT.TXT");
44. StreamWriter streamWriter = **new** StreamWriter("OUTPUT.TXT");
45. **int** n = Convert.ToInt32(streamReader.ReadLine());
46. Pair[] agentInfos = **new** Pair[n];
47. **long**[] dp = **new** **long**[**10100**];
48. **string**[] line = streamReader.ReadLine().Split(**new** **char**[] { ' ' });
49. **if** (n != line.Length / **2**)
50. line = DeleteEmpty(line);
51. **for** (**int** i = **0**; i < n; i++)
52. {
53. agentInfos[i] = **new** Pair(Convert.ToInt32(line[**2** \* i]), Convert.ToInt32(line[**2** \* i + **1**]));
54. }
55. Array.Sort(agentInfos);
56. dp[**1**] = agentInfos[**1**].Second;
57. dp[**2**] = (n > **2**) ? agentInfos[**1**].Second + agentInfos[**2**].Second : **0**;
58. dp[**3**] = (n > **3**) ? agentInfos[**1**].Second + agentInfos[**3**].Second : **0**;
59. **for** (**int** i = **4**; i < n; i++)
60. {
61. dp[i] = Math.Min(dp[i - **2**] + agentInfos[i].Second, dp[i - **3**] + agentInfos[i].Second + agentInfos[i - **1**].Second);
62. }
63. streamWriter.WriteLine(dp[n - **1**]);
64. streamWriter.Close();
65. }
66. }
67. }
68. }
69. **Тестирование**

