

Вариант 4.

Написать программу, которая получает на вход набор идентификаторов, организует таблицу по заданному методу и позволяет осуществлять многократный поиск идентификаторов в этой таблице. Список идентификаторов задан в виде текстового файла.

Простое рехэширование

Бинарное дерево

The screenshot shows a Visual Studio Code editor with a C# project named "Programcs - IdentifierTableProgram". The main file is "IdentifierTableProgram.cs", which contains a class `Program` with a static `Main` method. The `Main` method reads a list of identifiers from a file named "identifiers.txt" and stores them in a table. The table is a 2D array of strings, where each row represents a bucket in a hash table. The identifiers are stored in the buckets based on their first character. The `Main` method also prints the contents of the table to the console.

```
9 {
10     0 references
11     class Program
12     {
13         0 references
14         static void Main(string[] args)
15         {
16             // Read identifiers from file
17             string[] identifiers = File.ReadAllLines("identifiers.txt");
18             // Create a table to store identifiers
19             string[,] table = new string[26, 100];
20             // Store identifiers in the table
21             for (int i = 0; i < identifiers.Length; i++)
22             {
23                 char firstChar = identifiers[i][0];
24                 int index = firstChar - 'a';
25                 if (index < 26)
26                 {
27                     table[index, i % 100] = identifiers[i];
28                 }
29             }
30             // Print the table
31             for (int i = 0; i < 26; i++)
32             {
33                 for (int j = 0; j < 100; j++)
34                 {
35                     Console.Write(table[i, j] + " ");
36                     if (j % 10 == 9)
37                     {
38                         Console.WriteLine();
39                     }
40                 }
41             }
42         }
43     }
44 }
```

The console output shows the contents of the table, with identifiers grouped by their first character. For example, the 'a' bucket contains identifiers like "123asd", "123asd", "123asd", etc.