This is a Java class named "B0076" that contains a single method, "minWindow", which takes in two strings as input: "s" and "t". The purpose of this method is to find the smallest substring in "s" that contains all the characters in "t".

The method begins by initializing an array, "tMap", with a length of 128. This array is used to keep track of the frequency of each character in the input string "t". The for loop iterates through each character in "t" and increments the corresponding index in "tMap" for each character.

The variable "required" is set to the length of the input string "t". This variable is used to keep track of the number of unique characters in "t".

The method then initializes two pointers, "left" and "right", which are used to create a sliding window over the input string "s". The variable "formed" is used to keep track of the number of unique characters in the current window that are also in "t".

Next, the method initializes an array, "windowMap", with a length of 128. This array is used to keep track of the frequency of each character in the current window.

The method also initializes an array, "ans", with the values [-1, 0, 0]. This array is used to store the information of the smallest window that contains all characters in "t". The array is in the format of {length, left, right}.

The method then enters a while loop that continues until the "right" pointer reaches the end of the input string "s". Within the loop, the character at the current "right" pointer index is stored in the variable "c". The "windowMap" array is then updated to increment the count of the character "c".

Then the method checks if the current character "c" is in "t" and the count of this character in the "windowMap" is less than or equal to the count of this character in "tMap". If this is true, the "formed" variable is incremented.

The method then enters another while loop that continues until the "left" pointer is less than or equal to the "right" pointer and the "formed" variable is equal to the "required" variable. Within this loop, the character at the current "left" pointer index is stored in the variable "c". The method then checks if the length of the current window is smaller than the previous smallest window and if so, updates the values in the "ans" array to the current length, left index, and right index.

The method then decrements the count of the character "c" in the "windowMap" array. If the current character "c" is in "t" and the count of this character in the "windowMap" is less than the count of this character in "tMap", then the "formed" variable is decremented. The "left" pointer is then incremented.

Finally, when the while loop is complete, the method returns the smallest substring in "s" that contains all the characters in "t", or an empty string if none is found.