**难度: 中等**

写一个函数 StrToInt，实现把字符串转换成整数这个功能。不能使用 atoi 或者其他类似的库函数。

首先，该函数会根据需要丢弃无用的开头空格字符，直到寻找到第一个非空格的字符为止。

当我们寻找到的第一个非空字符为正或者负号时，则将该符号与之后面尽可能多的连续数字组合起来，作为该整数的正负号；假如第一个非空字符是数字，则直接将其与之后连续的数字字符组合起来，形成整数。

该字符串除了有效的整数部分之后也可能会存在多余的字符，这些字符可以被忽略，它们对于函数不应该造成影响。

注意：假如该字符串中的第一个非空格字符不是一个有效整数字符、字符串为空或字符串仅包含空白字符时，则你的函数不需要进行转换。

在任何情况下，若函数不能进行有效的转换时，请返回 0。

说明：

假设我们的环境只能存储 32 位大小的有符号整数，那么其数值范围为 [−231,  231 − 1]。如果数值超过这个范围，请返回  INT\_MAX (231 − 1) 或 INT\_MIN (−231) 。

示例 1:

输入: "42"

输出: 42

示例 2:

输入: " -42"

输出: -42

解释: 第一个非空白字符为 '-', 它是一个负号。

  我们尽可能将负号与后面所有连续出现的数字组合起来，最后得到 -42 。

示例 3:

输入: "4193 with words"

输出: 4193

解释: 转换截止于数字 '3' ，因为它的下一个字符不为数字。

示例 4:

输入: "words and 987"

输出: 0

解释: 第一个非空字符是 'w', 但它不是数字或正、负号。

因此无法执行有效的转换。

示例 5:

输入: "-91283472332"

输出: -2147483648

解释: 数字 "-91283472332" 超过 32 位有符号整数范围。

  因此返回 INT\_MIN (−231) 。

来源：力扣（LeetCode）

链接：https://leetcode-cn.com/problems/ba-zi-fu-chuan-zhuan-huan-cheng-zheng-shu-lcof

著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。

**思路(注释在代码里面了):**

|  |
| --- |
| public int strToInt(String str) {  //去除前后空格  String trim = str.trim();  //没有任何字符  if (trim.equals("")) {  return 0;  }  byte sign = 1; //符号 默认加号  int i = 0; //下标  //判断第一个是不是符号  if (trim.charAt(0) == '-' || trim.charAt(0) == '+') {  sign = (byte) (trim.charAt(0) == '-' ? -1 : 1);  i = 1;  }  System.out.println((int)trim.charAt(i));  long res = 0;  for (; i < trim.length(); i++) {  //不是数字  //48对应的是0,47对应的是9  if (trim.charAt(i) < 48 || trim.charAt(i) > 57) {  break;  }  //叠加  res = (res \* 10) + (trim.charAt(i) - 48);  //超过最大值或者最小值  if (res < Integer.MIN\_VALUE || res > Integer.MAX\_VALUE) {  //判断是什么符号 输出int的最大值或者最小值  return sign == 1 ? Integer.MAX\_VALUE : Integer.MIN\_VALUE;  }  }  return (int) (sign \* res);  } |