

一、 比较匹配方法

Indexof.java:

```
package javaHomework3_1;
```

```
public class Indexof {
    private String mainstr; // 主串字段

    public Indexof(String mainString) {
        this.mainstr = mainString;
    }

    public int indexOf(String str) {
        int substrLength = str.length(); // 获取子串的长度
        int mainstrLength = this.mainstr.length(); // 获取主串的长度
        if(substrLength <= 0 || substrLength > mainstrLength) {
            return -1;
        } // 若子串长度大于等于主串则直接返回 -1
        else {
            int i=0;
            for(int h=0;h<mainstrLength;h++) { // 遍历主串查找
                if(str.charAt(i) == mainstr.charAt(h)) {
                    i++;
                    if(i == substrLength) {
                        return h-substrLength+1; // 若匹配成功则返回首位置
                    }
                }
            }
            return -1; // 全部匹配失败，返回 -1
        }
    }

    public int indexOf(String str,int fromIndex) {
        int substrLength = str.length(); // 获取子串的长度
        int mainstrLength = this.mainstr.length(); // 获取主串的长度
        if(substrLength <= 0 || substrLength > mainstrLength) {
            return -1;
        } // 若子串长度大于等于主串则直接返回 -1
        else {
            int i=0;
```

```

        for(int h=fromIndex;h<mainstrLength;h++) { //遍历主串查找
            if(str.charAt(i) == mainstr.charAt(h)) {
                i++;
                if(i == substrLength) {
                    return h-substrLength+1; //若匹配成功则返回首位置
                }
            }
        }
        else {
            i=0;
            continue; //匹配失败则将主串指针向后移一位，子串指针重新指向头部
        }
    }
}
return -1; //全部匹配失败，返回-1
}

public int indexOf(int ch) {
    int mainstrLength = this.mainstr.length(); // 获取主串的长度
    for(int h = 0; h < mainstrLength; h++) {
        if(ch == mainstr.charAt(h)) {
            return h;
        }
    }
    return -1;
}

public int indexOf(int ch, int fromIndex) {
    int mainstrLength = this.mainstr.length(); // 获取主串的长度
    for(int h = fromIndex; h < mainstrLength; h++) {
        if(ch == mainstr.charAt(h)) {
            return h;
        }
    }
    return -1;
}

public static void main(String args[]) {
    Indexof str = new Indexof("hello world");
    System.out.println(str.indexOf("world"));
    System.out.println(str.indexOf("world", 7));
    System.out.println(str.indexOf(111)); //查找'o'所在位置
    System.out.println(str.indexOf(111, 6));
}
}

```

运行结果：

```
75 public static void main(String args[]) {
76     Indexof str = new Indexof("hello world");
77     System.out.println(str.indexOf("world"));
78     System.out.println(str.indexOf("world",7));
79     System.out.println(str.indexOf(111)); //查找"o"所在位置
80     System.out.println(str.indexOf(111,6));
81 }
82 }
83 }
84 }
85 }
```

控制台 ×

```
<已终止> Indexof [Java 应用程序] E:\Java\Eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full\
6
-1
4
7
```

二、 KMP 算法（区别于原方法之处已标红处理）

KMPIndexof.java:

```
package javaHomework3_2;
```

```
public class KMPIndexof {
    private String mainstr; // 主串字段
    public KMPIndexof(String mainString) {
        this.mainstr = mainString;
    }

    void get_Next(String s, int next[]) //这个函数对字符串 s 进行预处理得到 next 数组
    {
        int j = 0;
        next[0] = 0; //初始化
        for(int i = 1; i < s.length(); i++) { //i 指针指向的是后缀末尾, j 指针指向的是前缀末尾
            while(j > 0 && s.charAt(i) != s.charAt(j)) j = next[j-1]; //前后缀不相同, 去找 j 前一位
            if(s.charAt(i) == s.charAt(j)) j++; //前后缀相同, j 指针后移
            next[i] = j; //更新 next 数组
        }
    }

    public int indexOf(String str) {
        int substrLength = str.length(); // 获取子串的长度
        int mainstrLength = this.mainstr.length(); // 获取主串的长度
        int next[] = new int[substrLength];
        get_Next(str, next);
        if(substrLength <= 0 || substrLength > mainstrLength) {
            return -1;
        } //若子串长度大于等于主串则直接返回-1
        else {
```

```

int i=0;
for(int h=0;h<mainstrLength;h++){//遍历主串查找
    while(i > 0 && str.charAt(i) != mainstr.charAt(h)) {
        i=next[i-1];//前后缀不相同，去找 j 前一位的最长相等前后缀
    }
    if(str.charAt(i) == mainstr.charAt(h)) {
        i++;
        if(i == substrLength) {
            return h-substrLength+1;
        }
    }
}
}
return -1;//全部匹配失败，返回-1
}

```

```

public int indexOf(String str,int fromIndex) {
    int substrLength = str.length(); // 获取子串的长度
    int mainstrLength = this.mainstr.length(); // 获取主串的长度
    int next[]=new int[substrLength];
    get_Next(str,next);
    if(substrLength <= 0 || substrLength > mainstrLength) {
        return -1;
    }//若子串长度大于等于主串则直接返回-1
    else {
        int i=0;
        for(int h=0;h<mainstrLength-fromIndex;h++){//将 fromIndex 位视作第 0 位

```

进行判断

```

        while(i > 0 && str.charAt(i) != mainstr.charAt(h+fromIndex)) {
            i=next[i-1];//前后缀不相同，去找 j 前一位的最长相等前后缀
        }
        if(str.charAt(i) == mainstr.charAt(h+fromIndex)) {
            i++;
            if(i == substrLength) {
                return h+fromIndex-substrLength+1;
            }
        }
    }
}
return -1;//全部匹配失败，返回-1
}

```

```

public int indexOf(int ch) {
    int mainstrLength = this.mainstr.length(); // 获取主串的长度

```

```

        for(int h = 0;h<mainstrLength;h++) {
            if(ch == mainstr.charAt(h)) {
                return h;
            }
        }
        return -1;
    }

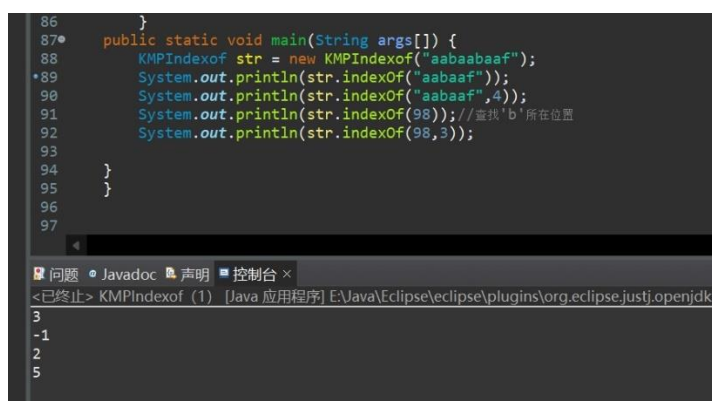
    public int indexOf(int ch,int fromIndex) {
        int mainstrLength = this.mainstr.length(); // 获取主串的长度
        for(int h = fromIndex;h<mainstrLength;h++) {
            if(ch == mainstr.charAt(h)) {
                return h;
            }
        }
        return -1;
    }

    public static void main(String args[]) {
        KMPIndexof str = new KMPIndexof("aabaabaaf");
        System.out.println(str.indexOf("aabaaf"));
        System.out.println(str.indexOf("aabaaf",4));
        System.out.println(str.indexOf(98)); // 查找'b'所在位置
        System.out.println(str.indexOf(98,3));

    }
}

```

运行结果：



The screenshot shows an IDE with a Java file named `KMPIndexof.java`. The code is as follows:

```

86     }
87     public static void main(String args[]) {
88         KMPIndexof str = new KMPIndexof("aabaabaaf");
89         System.out.println(str.indexOf("aabaaf"));
90         System.out.println(str.indexOf("aabaaf",4));
91         System.out.println(str.indexOf(98)); // 查找'b'所在位置
92         System.out.println(str.indexOf(98,3));
93     }
94 }
95
96
97

```

The IDE's console window shows the output of the program:

```

<已终止> KMPIndexof (1) [Java 应用程序] E:\Java\Eclipse\plugins\org.eclipse.justi.openjdk...
3
-1
2
5

```