Match results

A simple check whether a string contains a substring matching our regular expression is not the only thing we can do with a Matcherobject.

Getting match results

As you know, the find method of Matcher can check whether a substring of a string matches the pattern. Here is an example.

```
String javaText = "Java supports regular expressions. LET'S USE JAVA!!!";

Pattern javaPattern = Pattern.compile("java", Pattern.CASE_INSENSITIVE);
Matcher matcher = javaPattern.matcher(javaText);

System.out.println(matcher.find()); // prints "true"
```

When find() method returns true it is possible to get some info about the substring matching the pattern. start() and end() return the starting and the last indices of the match respectively, while group() returns the matching substring itself.

```
System.out.println(matcher.start()); // 0, the starting index of match
System.out.println(matcher.end());   // 4, the index following the last index of match
System.out.println(matcher.group()); // "Java", a substring that matches the pattern
```

There is a special classMatchResult that comprises all this information about the match:

```
MatchResult result = matcher.toMatchResult(); // a special object containing match results

System.out.println(result.start()); // 0
System.out.println(result.end());   // 4
System.out.println(result.group()); // "Java"
```

Be careful, if you invoke the methods start, end, group before invokingfind() method or in case it was invoked and returned false, they will throw IllegalStateException.

```
if (matcher.find()) {
    System.out.println(matcher.start());
    System.out.println(matcher.end());
    System.out.println(matcher.group());
```

```
} else {
    System.out.println("No matches found");
}
```

Iterating over multiple matches

Sometimes more than one substring matches the same pattern. In the previous example, there are two suitable strings **"Java"** and **"JAVA"**, because the pattern is case insensitive.

```
String javaText = "Java supports regular expressions. LET'S USE JAVA!!!";

Pattern javaPattern = Pattern.compile("java", Pattern.CASE_INSENSITIVE);
Matcher matcher = javaPattern.matcher(javaText);

while (matcher.find()) {
    System.out.println("group: " + matcher.group() + ", start: " + matcher.start());
}
```

This code outputs the following lines:

```
group: Java, start: 0
group: JAVA, start: 45
```