

```
str -> str.length()           // int apply(String str);
book -> book.title()          // String apply(Book book);
```

str は String 型のインスタンスで、str.length() は文字列の長さを返すインスタンスメソッドです。また、book は Book 型のレコードで、book.title() は本のタイトル名を返すインスタンスメソッドです。

1つのメソッドだけ呼び出すので、この場合もメソッド参照に直せますが、引数を省略するのがメソッド参照ですから、str::length とか book::title とは書けません。

そこで、やはり、**クラス名::メソッド名** の形式で書くことになっています。

```
str -> str.length()   ───────────▶ String::length
book -> book.title()  ───────────▶ Book::title
```

## 外部で定義された変数のインスタンスメソッド

次は、doit() メソッドの中にラムダ式を書いています。

このラムダ式も、1つのメソッドを呼び出しているだけですから、やはりメソッド参照に直せます。

```
String words = "apple banana cherry peach";
doit(str -> words.contains(str));          // boolean test (String str);
```

**words を str に訂正します**

ただ、これまでと違うのは、ローカル変数 words のインスタンスメソッドを使っている所です。メソッド参照は、ラムダ式の引数を省略するだけですから、ローカル変数はそのまま使えます。そこで、**外部変数::メソッド名** の形でメソッド参照にします。

```
String words = "apple banana cherry peach";
doit(words::contains);
```

この場合、words は実質的に final な変数でなければいけません。

なお、words.contains(str) は、words の中に str が含まれているかどうか調べるメソッドです。2章で解説した文字列のインスタンスメソッドを参照してください。