

Java SE 8 関数インターフェース

[Java SE 8]

- <http://docs.oracle.com/javase/specs/jls/se8/html/jls-9.html#jls-9.8>
- 関数インターフェースは、ただ一つの抽象メソッド (オブジェクトから継承するメソッドはさておいて) を持ったインターフェース
- 一つの関数契約を表現する
- この「一つ」のメソッドは、複数の抽象メソッドを、スーパーインターフェースから継承されたオーバーライド同等のシグネチャをとまって形成する。
- この例では、継承されたメソッドは論理的に一つのメソッドを表現する。

例

■ 簡単な関数インターフェースの例

```
interface Runnable {  
    void run();  
}
```

以下は、関数インターフェースではない。Object のメンバーであってははいけない。

```
interface NonFunc {  
    boolean equals(Object obj);  
}
```

しかしながら、上記を継承したサブインターフェースは関数インターフェースとなる。宣言された抽象メソッドは、Object のメンバーではない。

```
interface Func extends NonFunc {  
    int compare(String o1, String o2);  
}
```

同様に、java.util.Comparator<T> は一つの抽象メソッドをもち、かつ Object のメソッドではないため、関数インターフェースとなる。

```
interface Comparator<T> {  
    boolean equals(Object obj);  
    int compare(T o1, T o2);  
}
```

次のインターフェースは関数インターフェースではない。一つの抽象メソッドをもち、それは Object のメンバーではないが、Object のメンバーではない、2 つの public な抽象メソッドをもつ。

```
interface Foo {  
    int m();  
    Object clone();  
}
```