//fix this /

2014年5月/6月号では、オラクルのJavaエバンジェリストであるSimon Ritterから、Java SE 8のStream APIに関する課題が出されました。Ritterは、ストリームを使用してテキスト・ファイル内でもっとも長い行の長さを返すコードを提示し、長さではなく行そのものを返すように変更する方法を問いました。正解は2番です。単一の結果（このケースでは、ファイル内でもっとも長い文字列）を生成するには、ストリームに対してリデュース処理を実行する必要があります。_ reduce() メソッドは折りたたみ関数の形式を取っており、1つの関数をストリーム内の各要素に対して適用します。2番の回答では、2つの文字列の長さを比較して長い方を返す関数が適用されます。

Stringにはメソッド参照として使用できるlongest() メソッドがないため、1番は動作しません。3番はコンパイルに失敗します。reduce() メソッドの形式は2つのパラメータを受け取りますが、パラメータとラムダ式の戻り型がすべてString型である必要があります。4番と5番は両方とも動作しますが、長さの引き算が逆になっているため、ファイル内でもっとも短い行を返します。

,今回は、シニア・アイデンティティ / アクセス・エンジニアであるAbhishek Gupta氏から、ジェネリクスに関する出題です。

1 問題

Java SE 5でジェネリクスが導入されましたが、JDK 1.4以前を使用して作成されたコードに対する下位互換性を提供するために、引き続きraw型もサポートされています。では、ジェネリクスとraw型を混在させるとどうなるでしょうか。

2 コード

次のコードは、JDK 1.5以降でコンパイルされて実行されるものとします。JDK 1.5以降ではジェネリクスがサポートされています。

```
import java.util.ArrayList;
import java.util.List;
class Test{
private static void add(Object obj, List target){
target.add(obj);
}
public static void main(String[] args) throws Exception {
List<String> strings = new ArrayList<String>();
add(1, strings);
System.out.println("First element in the list:: "+ strings
  .get(0));
}
}
```

3 正しい修正方法はどれですか

1）プリミティブのint型をListに追加しようとするため、クラスのコンパイルに失敗する。

2）コンパイルに失敗するが、raw型のListの代わりにジェネリクス型（List<Object>）を使用するようにaddメソッドのシグネチャを変更することで解決する。

3）コンパイルは成功するが、実行時にjava.lang.ClassCastExceptionが発生する。ジェネリクスを使用するようにaddメソッドのシグネチャを変更し、適切な型の引数を使用してaddメソッドを呼び出すことで解決する。

4）コンパイルは成功するが、実行時にjava.lang.ClassCastExceptionが発生する。addメソッドのシグネチャを変更して、raw型のListの代わりにジェネリクス型（List<Object>）を使用すると解決する。

わかりましたか？

正解は次号に掲載されます。もしくはチャレンジした内容を電子メールでお送りください。

画像：I-HUA CHEN