

リスト 10-2-1： インタフェース RateIntf

```
package chapter10_2;
public interface RateIntf {
    double taxRate(double gaku); // 所得額(gaku)により、税率を計算して返す ❶
}
```

int を double に訂正します

❶のように taxRate() メソッドを定義しています。
次は、税率を計算する Rate1 クラスです。

リスト 10-2-2： 税率計算メソッドを持つ Rate1 クラス

```
package chapter10_2;
public class Rate1 implements RateIntf { ❶

    @Override
    public double taxRate(double gaku) { ❷
        return gaku > 100 ? 0.1 : 0.05; // 所得額が 100 より大なら 0.1、それ以外は 0.05
    }
}
```

Rate1 クラスは、❶のように RateIntf インタフェースを実装し、❷で taxRate() メソッドを実装しています。内容は、税額(gaku)が 100 より大きければ 0.1、そうでなければ 0.05 を税率として返します。簡単にするために条件演算子を使っています。

Rate1 クラスの定義はこれだけです。taxRate() メソッドを引き渡すためのクラスですから、他には何も定義しません。

次は、zeigaku() メソッドを持つ Tax クラスです。税額計算を扱うこのクラスは、6 章でも使いましたが、ここでは少し簡単にしています。

リスト 10-2-3： Tax クラス

```
package chapter10_2;
public class Tax {
    private long number;
    private String name;
    private int shotoku;

    public Tax(long number, String name, int shotoku) {
        this.number = number; // 納税者番号
        this.name = name; // 氏名
        this.shotoku = shotoku; // 所得額
    }
}
```