Weight Converter Class Design (1.0)
Prepared by Dr. C. S. Tritt
Last revised 4/5/05

Description

This class permits conversion of weights on one planet to weights on a different planet. Weights can be in any units; output is in same units as input. It is an example of a class for abstract objects. “Weight converters” don’t really exist in the physical world.

The test program, in *Main.Java*, demonstrates how the *WeightConverter* class can be used. In particular it shows how responses to messages (method calls) can be “nested.”

Its only constructor is `WeightConverter(double ratio)` where *ratio* is the ratio of the gravitational acceleration on the planet of interest to that of Earth.

Its only data member is `double relativeG` in which the ratio described above is stored.

Its other methods (member functions) are `double convertTo(double weight)` that converts the specified weight on Earth to the weight on the planet of interest and `double convertFrom(double weight)` that converts the specified weight on the planet of interest to the weight on Earth.

Test “vector” (results of manual (hand) calculations…)

Weight = 20.0 →

On Mercury: 7.80
From Mercury: 51.28
On Jupiter: 51.60
On Mercury to on Jupiter: 132.31

Definition/Implementation

See attached source code files.

File: WeightConverter.Java

```java
/* File: WeightConverter.java
 * Created by C. S. Tritt, Ph.D.
 * Last revised 4/5/05 (Version 1.1)
 *
 * See WeightConvert.doc for design details.
 */

public class WeightConverter {

    private double relativeG; // Planet to Earth gravity ratio.

    public WeightConverter(double rG) {
        // Only constructor. Takes a ratio and stores it.
        relativeG = rG;
    }

    public double convertTo(double weight) {
        // Given weight on Earth, returns weight on planet.
        return weight*relativeG;
    }

    public double convertFrom(double weight) {
        // Given weight on planet, returns weight on Earth.
        return weight/relativeG;
    }
}
```

File: Main.Java

```
/* File: Main.java
 * Created by C. S. Tritt, Ph.D.
 * Last revised 4/5/05 (Version 1.0)
 *
 * See WeightConvert.doc for design details.
 */
import javax.swing.*; // For JOptionPane methods.
import java.text.*; // For DecimalFormat methods.

public class Main {

    public static void main(String[] args) {
        //Symbolic constants (from Fishbane, Gasiorowics & Thornton, 1993)
        final double MERCURY_RATIO = 0.39; // Weight ratio Mercury to Earth
        final double JUPITER_RATIO = 2.58; // Weight ratio Jupiter to Earth

        // Get weight from user.
        double weight;
        weight = Double.parseDouble(JOptionPane.showInputDialog(
                "Enter a weight (any units): "));

        // Convert and display weight on Mercury.
        WeightConverter mercuryWeight = new WeightConverter(MERCURY_RATIO);
        double onMercury = mercuryWeight.convertTo(weight);
        double onEarth = mercuryWeight.convertFrom(weight);
        DecimalFormat df = new DecimalFormat("0.00");
        JOptionPane.showMessageDialog(null, "An object weighing " +
               weight + " units on Earth, weighs " + df.format(onMercury) +
               " units on Mercury.");

        // Convert and display weight on Jupiter.
        WeightConverter jupiterWeight = new WeightConverter(JUPITER_RATIO);
        JOptionPane.showMessageDialog(null, "An object weighing " +
                weight + " units on Earth, weighs " + df.format(
                jupiterWeight.convertTo(weight)) + " units on Jupiter.");

        // Demonstrate how converters can be nested.
        JOptionPane.showMessageDialog(null, "An object weighing " +
                weight + " units on Mercury, weighs " + df.format(
                jupiterWeight.convertTo(mercuryWeight.convertFrom(
                weight))) +  " units on Jupiter.");
    }
}
```