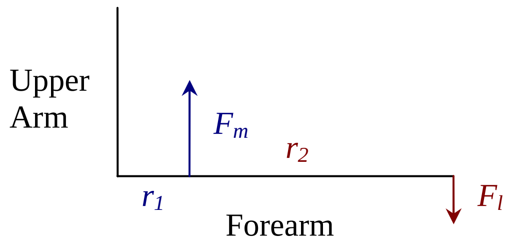
Arm Class Design (1.1)

Prepared by Dr. C. S. Tritt

Last revised 4/4/05

Description

This class represents an arm (particularly a human forearm). It uses a very (probably overly) simple model shown below:



where F_m is the muscle force, F_l is the load, r_1 is the distance from the joint to the insertion point of the muscle and r_2 is the distance from the joint to the point of action of the load and the forces are considered positive in the directions shown. The simplified static and kinematic equations for this situation are:

$$F_l = {}^{r_1}\!/_{r_2}\, F_m \qquad F_m = {}^{r_2}\!/_{r_1}\, F_l \qquad v_l = {}^{r_2}\!/_{r_1}\, v_m \qquad v_m = {}^{r_1}\!/_{r_2}\, v_l$$

Constructor is `Arm(double r1, double r2)`.

Data members are `double r_1 and double r_2`.

Other methods (member functions) are `double fLoad(double fMuscle)`, `double fMuscle(double fLoad)`, `double vLoad(double vMuscle)`, and `double vMuscle(double vLoad)`.

Test "vector"

r_1 = 5.00 cm, r_2 = 20.0 cm:

f_m = 10.0 → f_l = 2.5
f_l = 10.0 → f_m = 40.0
v_m = 1.0 → v_l = 4.00
f_m = 1.0 → f_l = 0.250

Definition/Implementation

See attached source code.

File: arm.java

```java
/* See my Arm Class documentation for details.
 * Prepared by C. S. Tritt, Ph.D.
 * Last revised 4/4/05
 */
public class Arm {

    // These are the data members (radii in this case).
    private double r_1; // Joint to insertion distance (cm).
    private double r_2; // Joint to load distance (cm).
    // Could just store ratio, but I might regret this later.

    public Arm(double r1, double r2) {
        // Only provided constuctor intializes r_1 and r_2
        r_1 = r1;
        r_2 = r2;
    }

    public double fLoad(double fMuscle) {
        // Returns load force given muscle force (both in N).
        return r_1*fMuscle/r_2;
    }

    public double fMuscle(double fLoad) {
        // Returns muscle force given load force (both in N).
        return r_2*fLoad/r_1;
    }

    public double vLoad(double vMuscle) {
        // Returns load velocity given muscle velocity (both in cm/s).
        return r_2*vMuscle/r_1;
    }

    public double vMuscle(double vLoad) {
        // Returns muscle velocity given load velocity (both in cm/s).
        return r_1*vLoad/r_2;
    }
}
```

File: main.java

```
/* Main.java
 * This class tests my Arm class.
 * Written by C. S. Tritt, Ph.D.
 * Last revised 4/4/05
 */
import java.util.*; // for Scanner class

public class Main {

    public static void main(String[] args) {

        // Console I/O to get test value...
        System.out.print("Enter \"x\" for Arm test: ");
        double x;
        Scanner sysin = new Scanner(System.in);
        x = sysin.nextDouble();
        System.out.println("Test value: " + x);

        // Continue to get r1 and r2...
        System.out.print("Enter \"r1\" and \"r2\"(cm): ");
        double r1, r2; // Distances (cm)
        r1 = sysin.nextDouble();
        r2 = sysin.nextDouble();
        System.out.println("Distances in cm: " + r1 + " and " + r2);

        // Construct an arm and call each method.
        Arm leftArm = new Arm(r1, r2);
        double fMuscle = leftArm.fMuscle(x);
        double fLoad = leftArm.fLoad(x);
        double vMuscle = leftArm.vMuscle(x);
        double vLoad = leftArm.vLoad(x);

        // Display results.
        System.out.println("fMuscle: " + fMuscle + " N");
        System.out.println("fLoad: " + fLoad + " N");
        System.out.println("vMuscle: " + vMuscle + " cm/s");
        System.out.println("vLoad: " + vLoad + " cm/s");
    }
}
```