



```
public class Data
{
    /**
     * Computes the average of the measures of the given objects.
     * @param objects an array of Measurable objects
     * @return the average of the measures
     */
    public static double average(Measurable[] objects)
    {
        double sum = 0;
        int count = 0;
        for (Measurable obj : objects)
        {
            if (obj instanceof Filter){
                Filter filterObj = (Filter) obj;
                if (filterObj.Accept(1000.0)){
                    sum += obj.getMeasure();
                    count++;
                }
            }
            else {
                sum += obj.getMeasure();
                count++;
            }
        }
        if (count > 0) { return sum / count; }
        else { return 0; }
    }
}
```

Class compiled - no syntax errors

```

    @param amount the amount to deposit
    */
    public void deposit(double amount)
    {
        balance = balance + amount;
    }

    /**
     Withdraws money from the bank account.
     @param amount the amount to withdraw
    */
    public void withdraw(double amount)
    {
        balance = balance - amount;
    }

    /**
     Gets the current balance of the bank account.
     @return the current balance
    */
    public double getBalance()
    {
        return balance;
    }

    public double getMeasure()
    {
        return balance;
    }

    public boolean Accept(Object Y)
    {
        if (Y instanceof Double){
            double threshold = (Double) Y;
            return balance > threshold;
        }
        return false;
    }
}

```

ass compiled - no syntax errors

saved