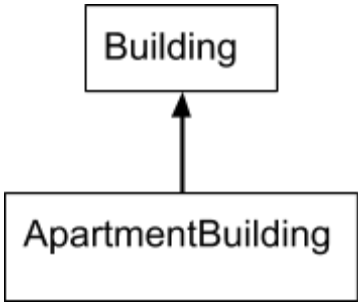
ApartmentBuilding FRQ

Consider a heirarchy of classes used by a power company to keep track of the buildings where they supply electricity. The hierarchy is represented by the following diagram:



Note that an `ApartmentBuilding` is a subclass of Building.
A building is represented by the class defined below.

```
public abstract class Building {
      private final double RATE = 3.25;
      private String address;
      private double wattHours; //units of electricity used in 1
month

      public Building ( String ad ){
            address = ad;
            wattHours = 0;
      }

      public double getRate()
      {
            return RATE;
      }

      /*returns the amount owed by this building*/
      double amtOwed( );

      //other methods not shown
}
```

An `ApartmentBuilding` is different from a regular building because instead of keeping track of the watt hours used for the whole building, it needs to keep track of the watt hours used by each of the individual apartments in the building.

Write a complete declaration of class `ApartmentBuilding` including the following:

- A private field `aptWattHours` (an array of doubles) to be used to store the apartments' watt hours.
- A constructor with two parameters: the address of the apartment building and the number of apartments. The constructor should initialize the building's address field as well as initializing the array to be big enough to store watt hours for each apartment in the building.
- An implementation of `amtOwed` that returns the amount of money owed by the entire building for the electricity used (calculated by multiplying the sum of the `wattHours` for the entire building by the RATE).

Finally, include the following method into your ApartmentBuilding class for testing purposes:

```
public void setApartmentHours( double[] hours )
{
      for(int i=0; i<aptWattHours.length; i++)
      {
            aptWattHours[i] = hours[i];
      }
}
```