

UML Diagram of `Measurer` Interface and Related Classes

Note that the `Rectangle` class is decoupled from the `Measurer` interface

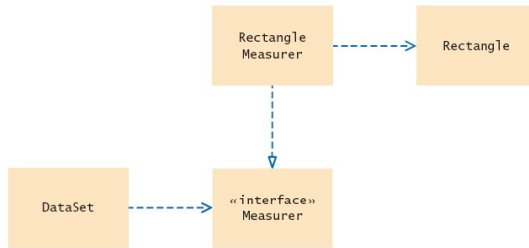


Figure 2 UML Diagram of the `DataSet` Class and the `Measurer` Interface

Big Java by Cay Horstmann
Copyright © 2008 by John Wiley & Sons. All rights reserved.

- You can convert from a class type to an interface type, provided the class implements the interface.
- Method calls on an interface reference are polymorphic. The appropriate method is determined at run time.
- You need a cast to convert from an interface type to a class type.
- Implement the `Comparable` interface so that objects of your class can be compared, for example, in a sort method.
- User-interface events include key

presses, mouse moves, button clicks, menu selections, and so on.

- An event listener belongs to a class that is provided by the application programmer. Its methods describe the actions to be taken when an event occurs.
- Event sources report on events. When an event occurs, the event source notifies all event listeners.
- Use `JButton` components for buttons. Attach an `ActionListener` to each button.
- Methods of an inner class can access local and instance variables from the surrounding scope.
- Local variables that are accessed by an inner class method must not change after they have been initialized.
- A mock object provides the same services as another object, but in a simplified manner.
- Both the mock class and the actual class implement the same interface

Implementing an Interface Type

- A variable of type `Measurable` holds a reference to an object of some class that implements the `Measurable` interface.

- `Country` class can also implement the `Measurable` interface:

```
public class Country implements Measurable {
    public double getMeasure() {
        return area;
    }
    ...
}
```

- Use interface types to make code more reusable.

EXAMPLE CODE :

```
public interface Named {

    String getName();

}
```

