

# Polymorphism

- What is polymorphism?
  - Examples
- public, private, and protected



Polymorphic - the quality or state of existing in or assuming different forms.

Polymorphism - In object-oriented programming, the term is used to describe a variable that may refer to objects whose class is not known at compile time and which respond at run time according to the actual class of the object to which they refer.



# How is it done in Java?

- Done automatically!!! COOL STUFF!!!
- Early Binding – the **compiler** knows what kind of object is used and calls the appropriate method.
- Late (or Dynamic) Binding – as the program is running, the **Java Virtual Machine** (JVM) selects the appropriate method.



# Example - GridWorld

```
public void step()
{
    < other code not shown >

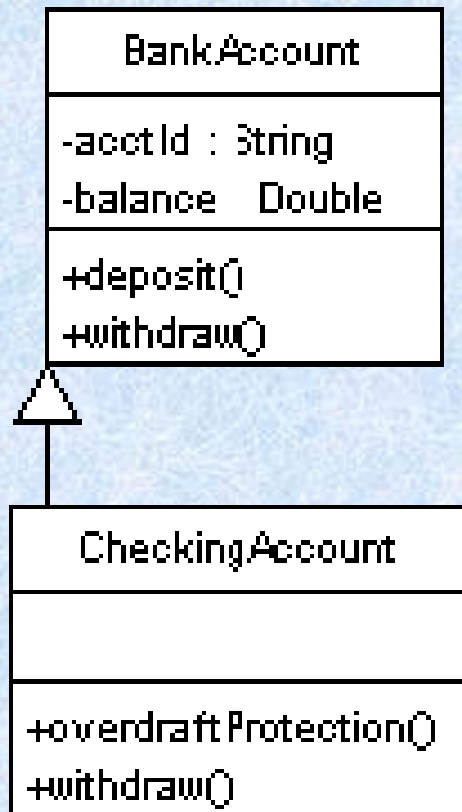
    // Get all the Locations where there are Actors in the
    // Grid and ask each one to perform the actions it does
    // in one timestep.

    ArrayList <Location> occupiedLocs =
        theGrid.getOccupiedLocations();

    for ( Location loc: occupiedLocs )
    {
        if( theGrid.get(loc) != null )
            theGrid.get(loc).act();
            // Note!! All kinds of objects here!!
    }
}
```



# Example - BankAccount



```
CheckingAccount cAcct =
new CheckingAccount ();
```

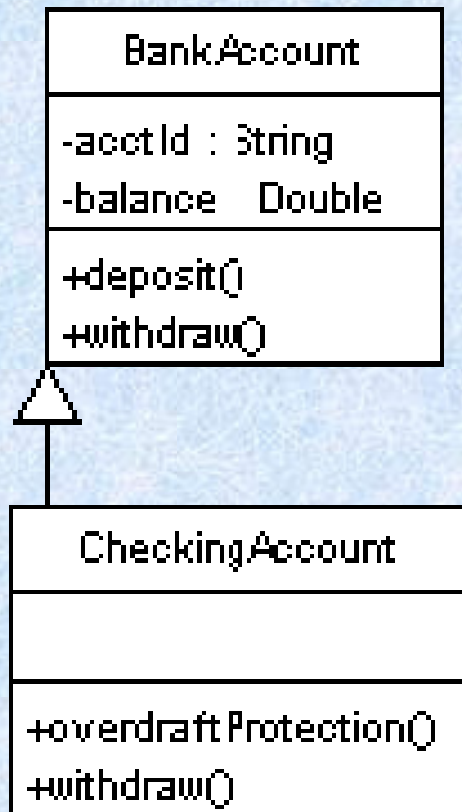
```
BankAccount myAcct;
```

```
myAcct = cAcct;
```

Is this ok???



# Example - BankAccount



```
CheckingAccount cAcct;  
  
cAcct = new BankAccount ();
```

How about this???



# A final word

- **private** members in the super class are not accessible in the subclass.
- **protected** members in the super class
  - Are accessible in the subclass.
  - Are not accessible outside the package.
- **public** members of a class are accessible from anywhere (that you are using an object)

