# Inheritance

- What is inheritance?
- "Is-a" vs. "Has-a" relationship
- Programming example "CircleBug"

# Inheritance

A class (child class)

- **inherits** the functionality of another class (parent class), and then
- **adds** new functionality of its own.

- **Java supports single inheritance.**
- **Inheritance** is a requirement of all object-oriented systems.

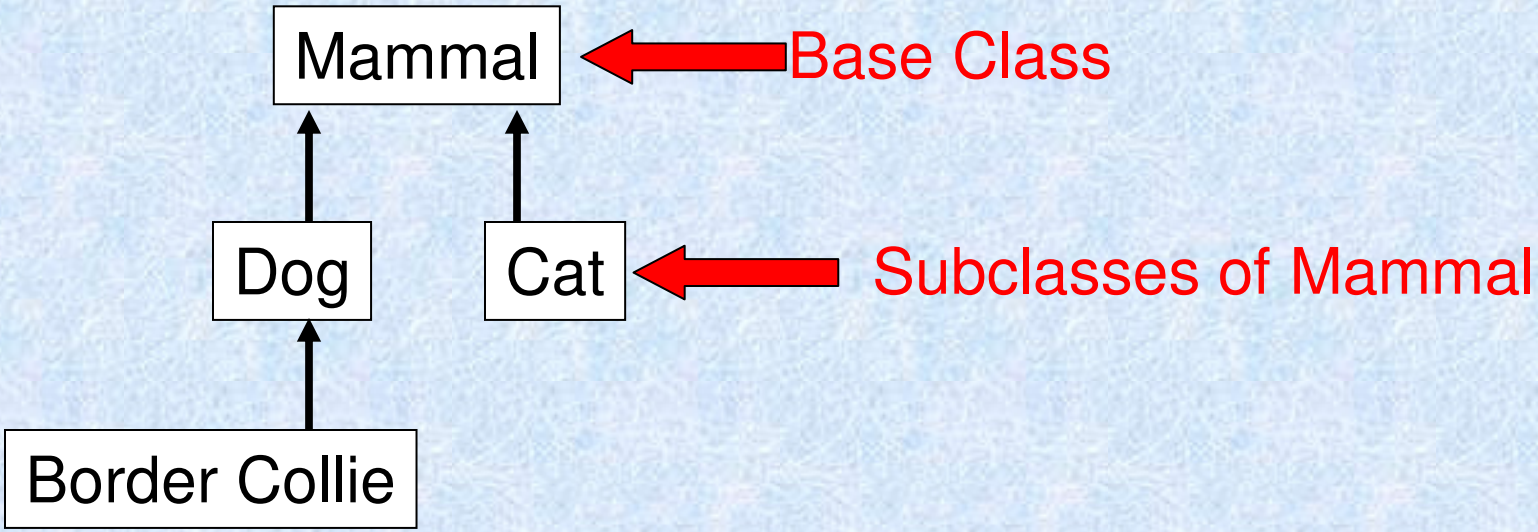
# Why inheritance?

- Inheritance is a technique for reuse.
- If a new class has a lot in common with a class that already exists, you can reuse parts of the existing class in the new class.
- The child class is defined by extending the parent class.
- The new "child" class has all characteristics of its "parent" plus any it adds itself.

# Inheritance (an is-a relationship)

- Inheritance is a parent-child relationship between classes.
- The parent class is also called a base class or a superclass.
- The child class is also called a derived class or subclass.

# Example: Inheritance Diagram



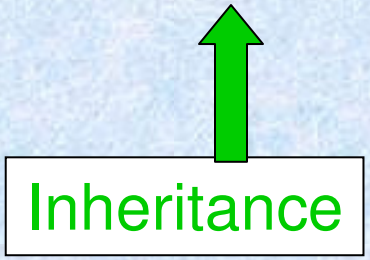
Example: In an OOP traffic simulation, you may have the following classes:

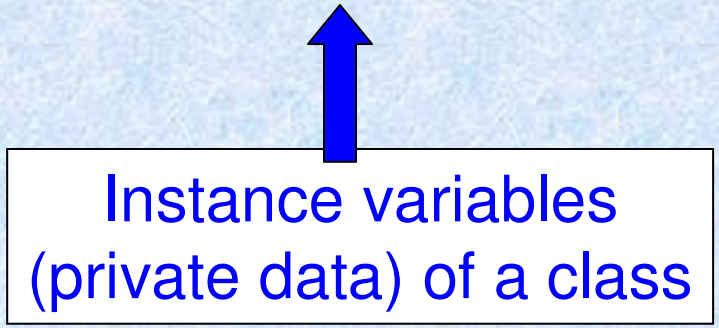
-Vehicle
-Car
-Truck
-Sedan
-Coupe
-PickupTruck
-SUV
-Minivan
-Motorcycle
-Bicycle

Make an inheritance diagram for the classes at the left.

# Important Concept in OOP!!!

"Is-a" vs. "Has-a"



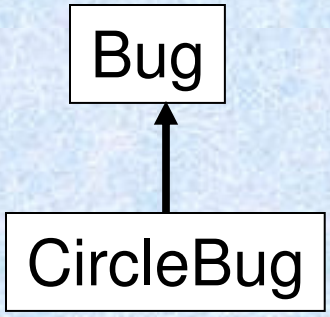



Example: **The CircleBug**

A **`CircleBug`** will inherit most of the qualities of a **`Bug`**. The only difference is how this bug moves.

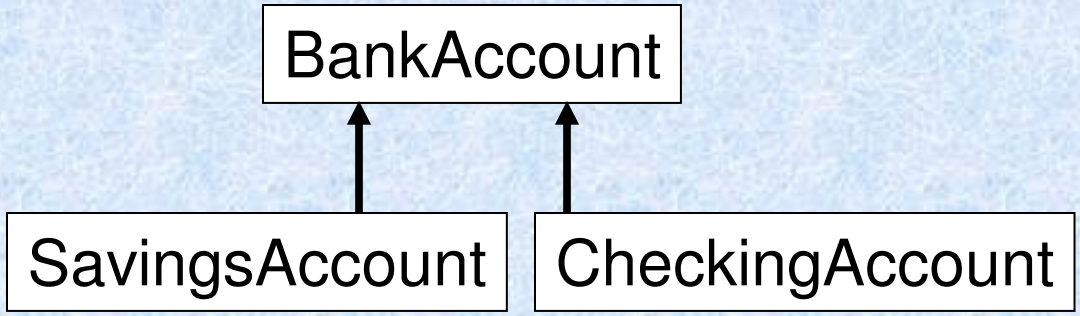
A **`CircleBug`** acts identical to the **`BoxBug`**, except that in the **`act`** method, the **`turn`** method is called once instead of twice. How is the behavior different than a **`BoxBug`**? Should it be called a **`CircleBug`**?

Inheritance Diagram

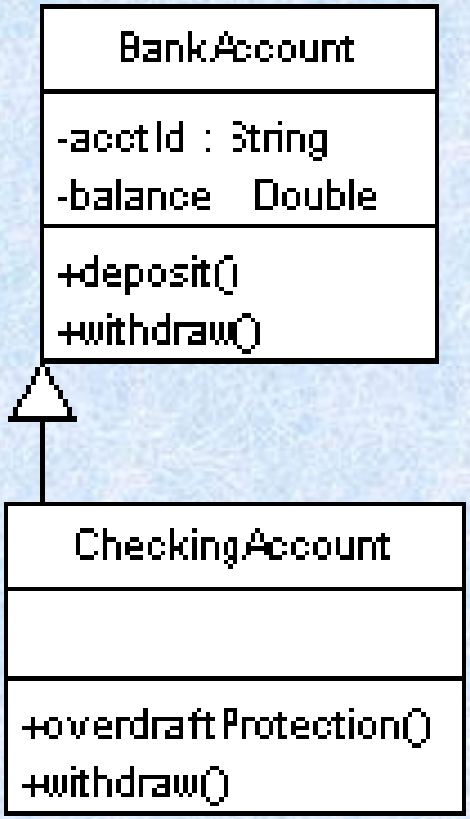



Example: Create a checking account class.
A checking account has all of the qualities of a **`BankAccount`**, but is assessed a fee of $2.00 for each transaction in excess of 3 transactions per month.

Inheritance Diagram




# Example - BankAccount



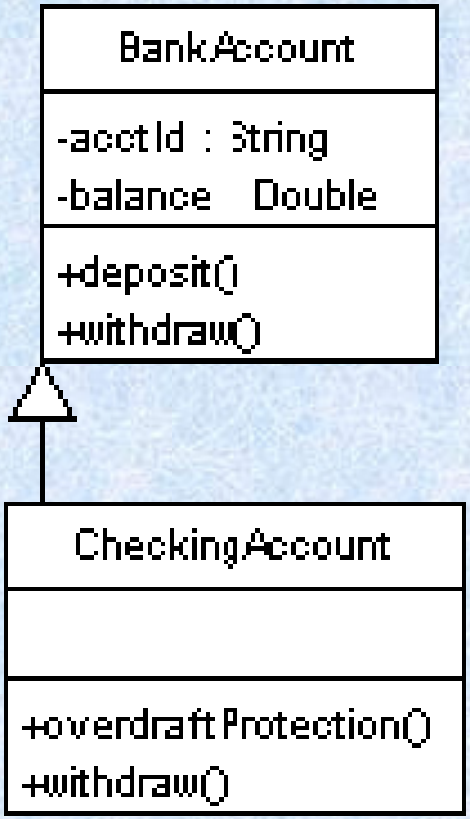

```
CheckingAccount cAcct =
new CheckingAccount ();

BankAccount myAcct;

myAcct = cAcct;
```

Is this ok???

# Example - BankAccount




```
CheckingAccount cAcct;

cAcct = new BankAccount();
```

How about this???