Inheritance

Apr. 4. 2017

Very *Very* Basic Inheritance

- Making a Game

```
public class Dude {
    public String name;
    public  int hp = 100;
    public  int mp = 0;

    public void sayName() {
          System.out.println(name);
    }
    public void punchFace(Dude target) {
          target.hp -= 10;
    }
}
```

Inheritance..

- Now create a Wizard…

```
public class Wizard {
    // ugh, gotta copy and paste
    // Dude's stuff
}
```

Inheritance?

- Now create a Wizard…

But Wait!

A Wizard does and has everything a Dude does and has!

Inheritance?

- Now create a Wizard…

Don’t Act Now!

You don’t have to Copy & Paste!

Buy Inheritance!

- Wizard is a **subclass** of Dude

```
public class Wizard extends Dude {
}
```

Buy Inheritance!

- Wizard can use everything* the Dude has!

```
wizard1.hp += 1;
```

*except for private fields and methods

Buy Inheritance!

- Wizard can do everything* Dude can do!

```
wizard1.punchFace(dude1);
```

*except for private fields and methods

Buy Inheritance!

- You can use a Wizard like a Dude too!

```
dude1.punchface(wizard1);
```

*except for private fields and methods

Buy Inheritance!

- Now augment a Wizard

```
public class Wizard extends Dude {
    ArrayList<Spell> spells;
    public class cast(String spell) {
        // cool stuff here
        ...
        mp -= 10;
    }
}
```

Inheriting from inherited classes

- What about a Grand Wizard?

```
public class GrandWizard extends Wizard {
    public void sayName() {
      System.out.println("Grand wizard" + name)
    }
}

grandWizard1.name = "Flash"
grandWizard1.sayName();
((Dude)grandWizard1).sayName();
```

How does Java do that?

- What Java does when it sees

```
grandWizard1.punchFace(dude1)
```

1. Look for `punchFace()` in the GrandWizard class
2. It's not there! Does GrandWizard have a parent?
3. Look for `punchFace()` in Wizard class
4. It's not there! Does Wizard have a parent?
5. Look for `punchFace()` in Dude class
6. Found it! Call `punchFace()`
7. Deduct hp from dude1

How does Java do that? pt2

- What Java does when it sees

```
((Dude)grandWizard1).sayName()
```

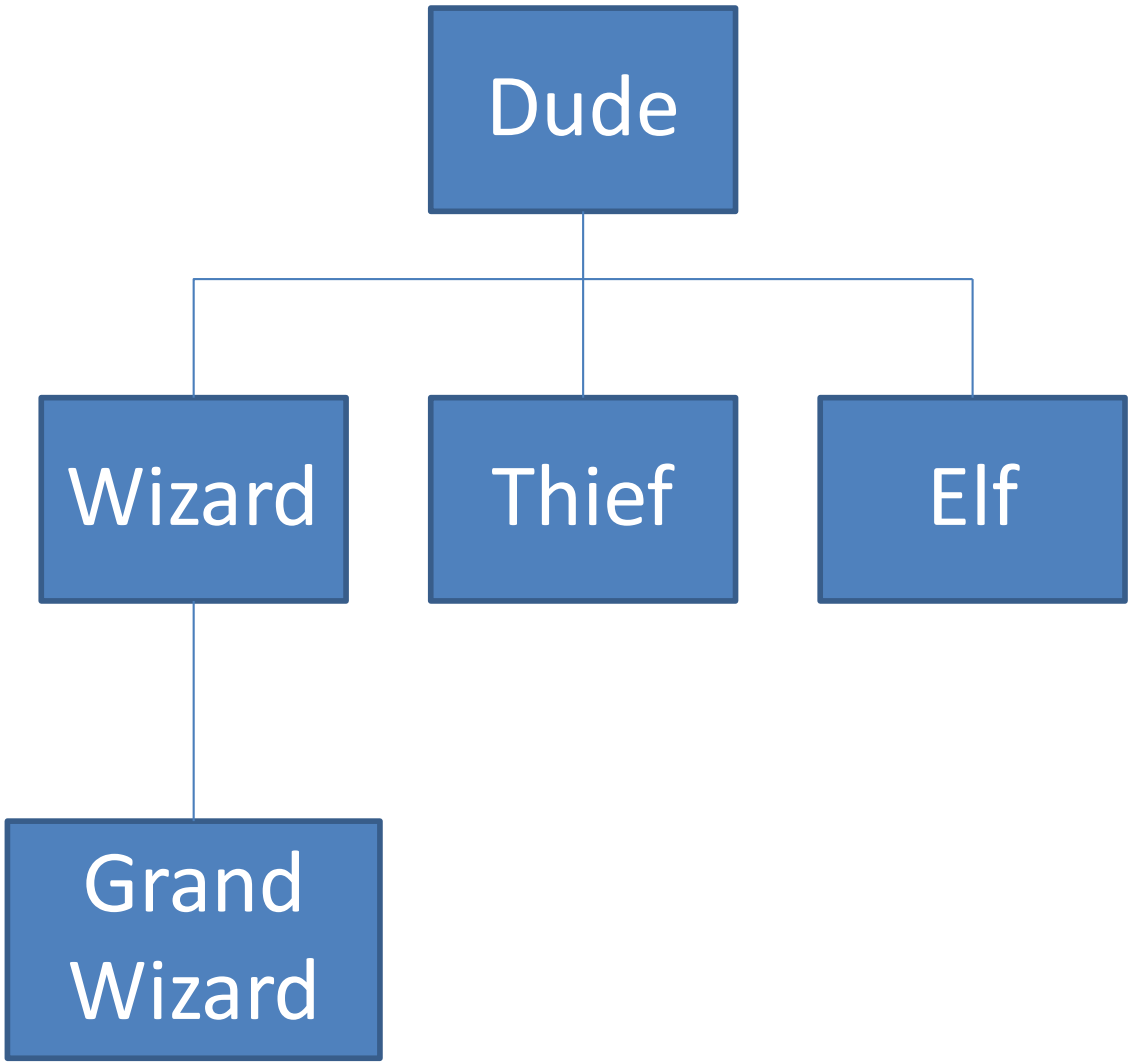
1. Cast to `Dude` tells Java to start looking in `Dude`
2. Look for `sayName()` in Dude class
3. Found it! Call `sayName()`

What's going on?

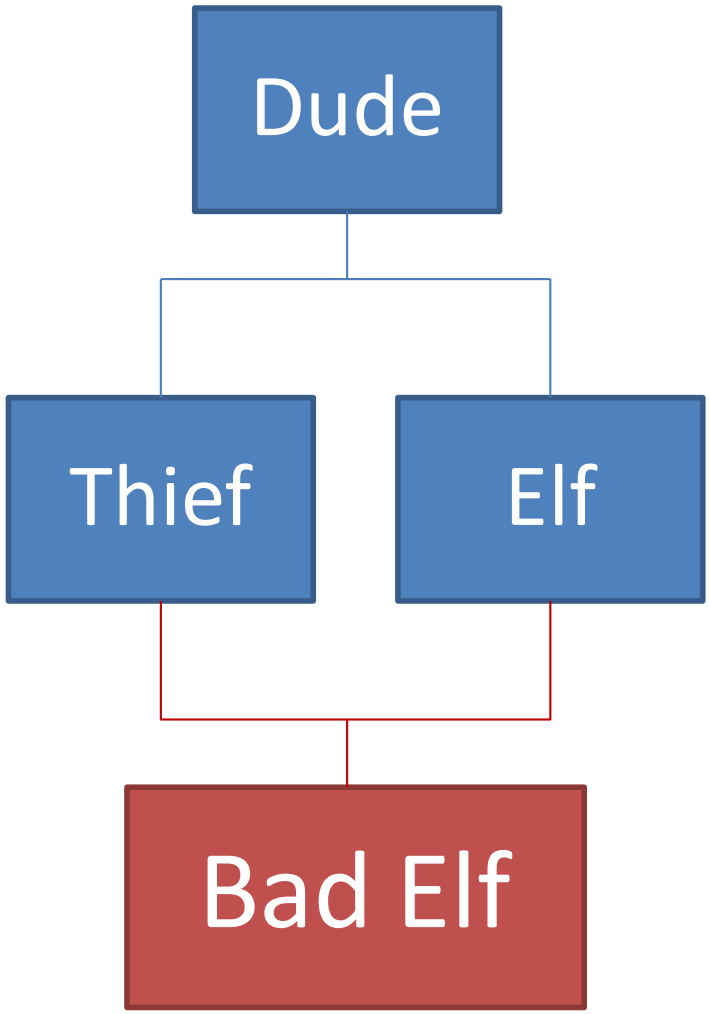
Parent of Wizard, Elf..

Subclass of Dude

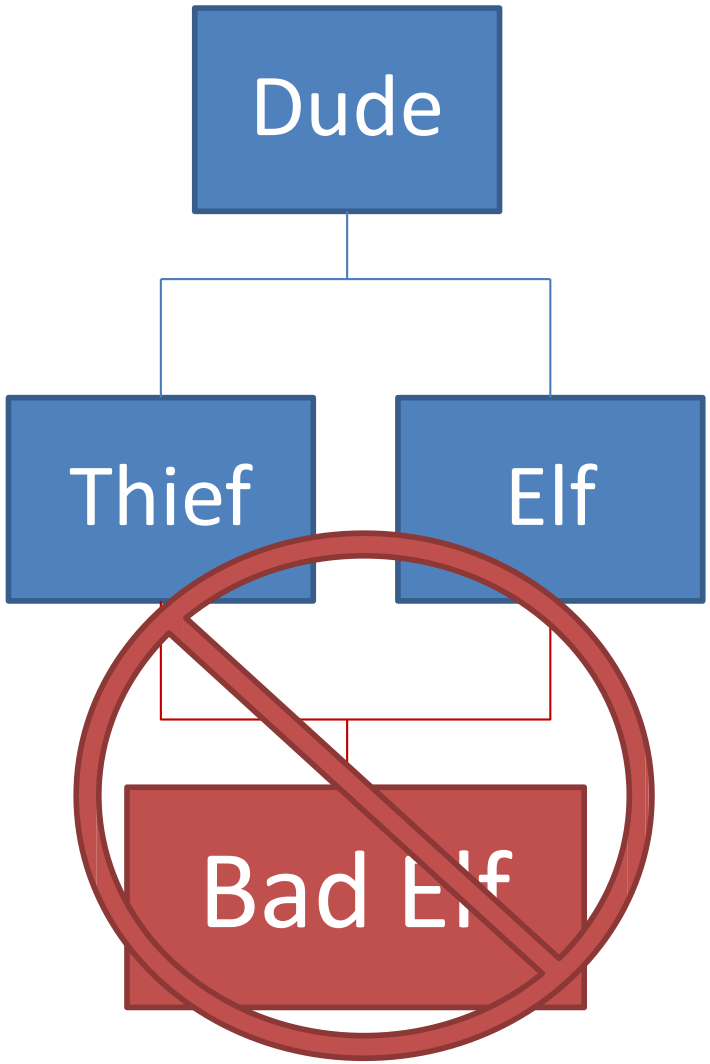
Subclass of Wizard



You can only inherit from one class



You can only inherit from one class



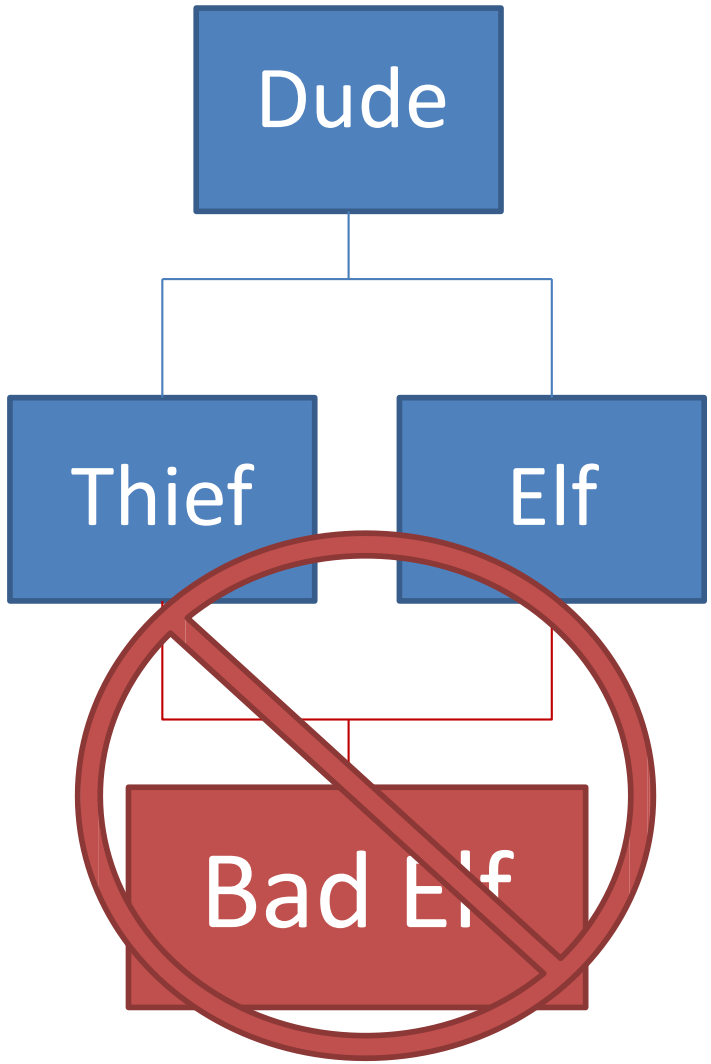
You can only inherit from one class

What if Thief and Elf both implement

```
public void sneakUp()
```

If they implemented differently, which `sneakUp()` does BadElf call?

Java Doesn't Know!!



Inheritance Summary

- class A extends B {} == A is a subclass of B
- A has all the fields and methods that B has
- A can add it's own fields and methods
- A can only have 1 parent
- A can replace a parent's method by re-implementing it
- If A doesn't implement something Java searches ancestors

LAB#5-1 Family Simulation

- Simulate your family's earning and spending for this month.
  - Discuss your family earning and spending structure
    - Who earn money?
    - How many do family members spending it?
    - Is there any regulations or characteristics?
  - Select one's family
  - Create more classes and their inheritance
  - Simulate the earning and spending of your family, using the main method.

LAB#5-2 Animal Sound

- Exercise method overriding using animals' hierarchy:
  - Parent class: Animal
  - Child classes: Dog, Cat, Cow, Lion
  - Their methods: bark()

- Create each class for each animal
- Create the bark() method in the Animal class
- Make each class inherit the method from Animal
- Implement the bark() method in each class

These slides are from:

- 6.092 Introduction to Programming in Java, January (IAP) 2010, MIT OpenCourseWare http://ocw.mit.edu

So much more to learn!

- http://java.sun.com/docs/books/tutorial/java/IandI/subclasses.html
- http://home.cogeco.ca/~ve3ll/jatutor5.htm
- http://en.wikipedia.org/wiki/Inheritance_(computer_science)
- http://www.google.com