

From Processing 2 Java

Polymorphism



TU / **e**

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

Recap...

- **Data abstraction**
 - **classes, encapsulation**
- **Inheritance**
 - **extension**

Upcasting

- Treat an object as if it was of its base type?

```
public void setup() {  
    ...  
    // create the boids  
    for (int i = 1; i <= xsize; i++) {  
        for (int j = 1; j <= ysize; j++) {  
            if (squareboid) {  
                boids.add(new SquareBoid(this, i * step, j * step, 0));  
            } else {  
                boids.add(new TriangleBoid(this, i * step, j * step, 0));  
            }  
        }  
    }  
    ...  
    // draw the boids  
    private void drawBoid() {  
        for (int i = 0; i < boids.size(); i++) {  
            Boid b = boids.get(i);  
            b.render();  
        }  
    }  
}
```

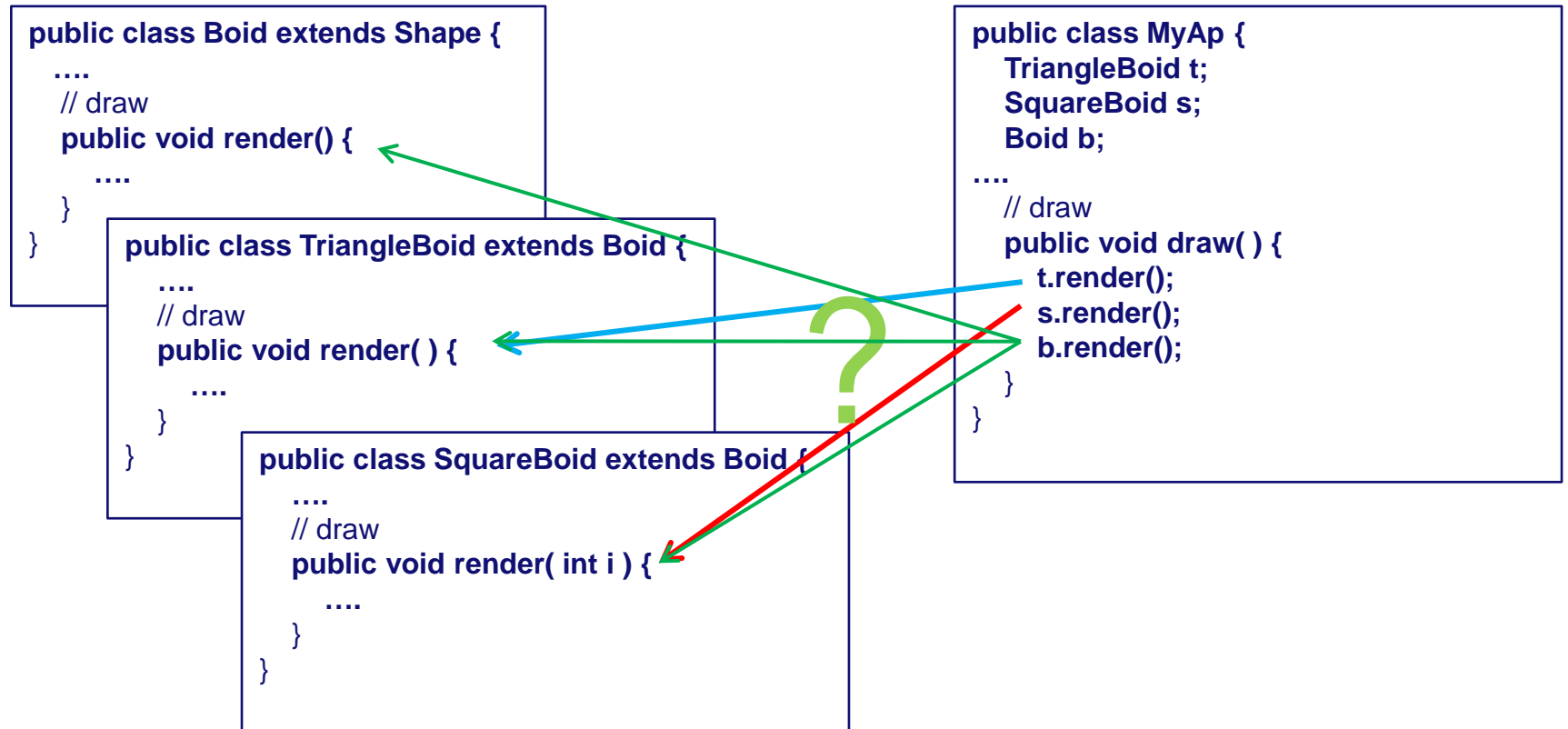
boids stores different
types of boids...

get as "Boid",
you don't
know what it
actually is...

Treat as...Boid?
SquareBoid?
TriangleBoid?

Overriding / overloading

- Beware...



Abstract classes and methods

- **classes that are not meant to be instantiated**
 - *can* contain non-abstract methods!
 - or even *only* non-abstract methods !
- **compile time check for solving abstractness**

```
abstract class Shape {  
    ....  
    public void setY(int y) {  
        ....  
    }  
  
    public void setX(int x) {  
        ....  
    }  
}
```

Constructors / Finalization

- **Construction order**
 1. **Object storage is zeroed**
 2. **Base class constructor is called (repeated hierarchy)**
 - **it's the default constructor(s) that is called if you don't explicitly change it!**
 3. **Properties initialization**
 4. **Derived class constructor**
- **Finalization ties up loose ends**
 - **protected finalize()** to do special cleanup
 - **don't forget to call super.finalize()**

Additional documentation

- Inheritance
- Overriding and hiding
- Polymorphism
- Java Quick Reference
 - (look at the bottom of the pages...)

- Using AdMoVeo in eclipse...
 - www.admoveo.nl → programming AdMoVeo → in eclipse
- Examples
- [AdMoVeo In Processing](#)

Homework

1. Read chapter 7 TIJ2
2. Exercise 8 of chapter 7 TIJ2
3. Show if there is a difference between
 - using `super()` in the constructor of a derived class and
 - using the default constructor of a derived class
4. Write a program that shows that your AdMoVeo robot reacts on one of the sensors by *movement* and reacts on another sensor by *colorchange*.