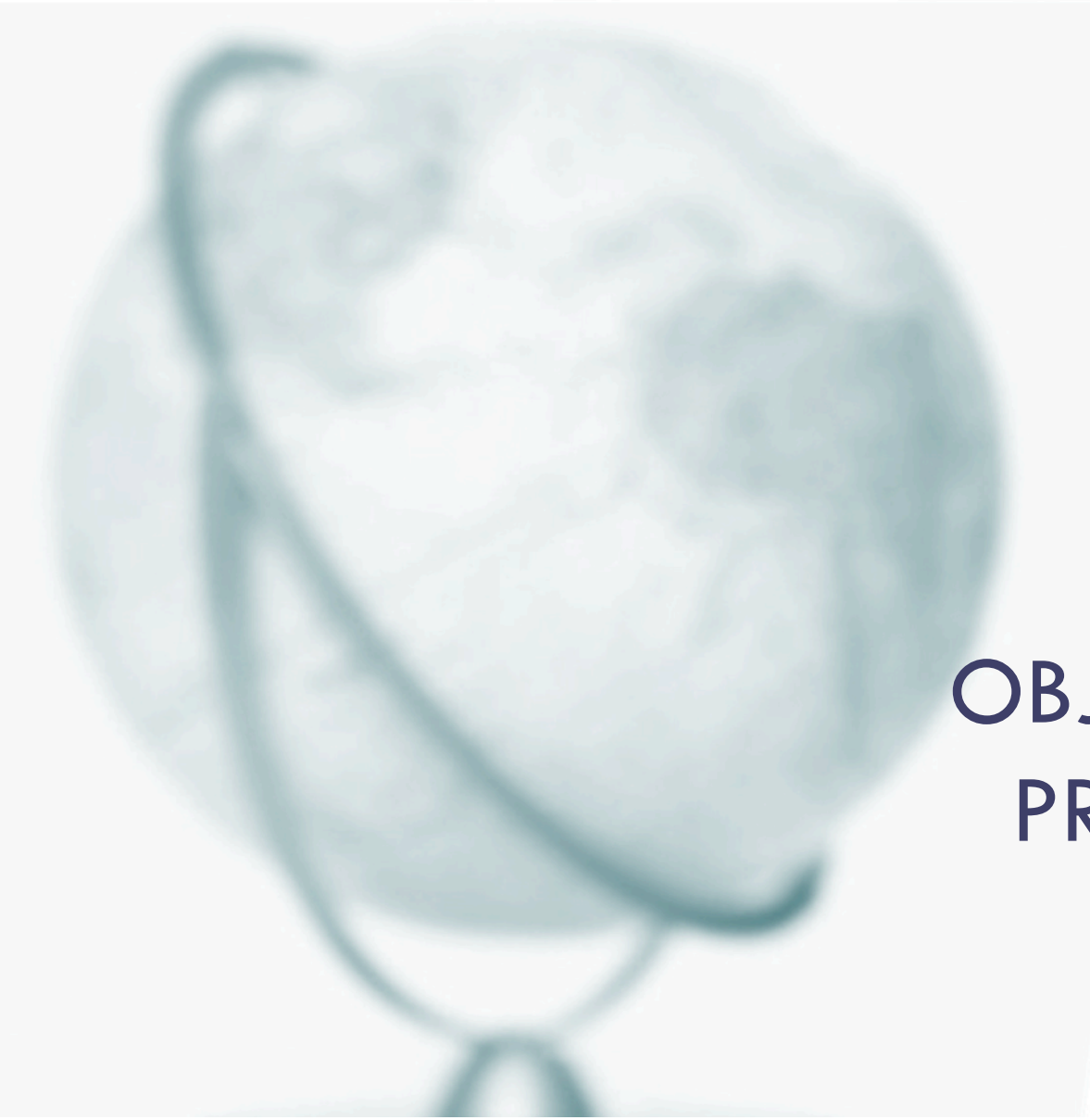


OBJECT-ORIENTED PROGRAMMING FOR ABMs

LECTURE 2

Dr. Simone Giansante
EC910 – L2

Lecture 2 outline

- Java classes
  - Inheritance
  - Interface
- OOP in an ABM
  - example
  - Objects in ABMs

Classes in Java

- A **class** is the model from which individual objects are created

- In object-oriented terms, an object is an ***instance*** of the class of objects.

Example of class

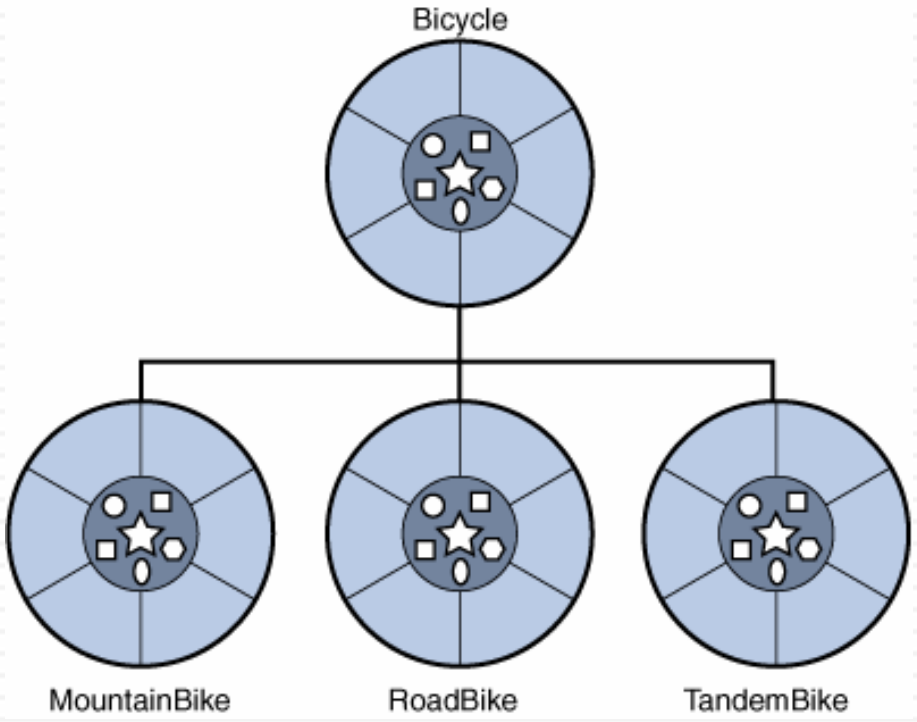

```
class Bicycle {

  int cadence = 0;
  int speed = 0;
  int gear = 1;

  void changeCadence(int newValue) {
    cadence = newValue;
  }

  void changeGear(int newValue) {
    gear = newValue;
  }

  void speedUp (int increment) {
    speed = speed + increment;
  }

  void applyBrakes(int decrement) {
    speed = speed - decrement;
  }

  void printStates() {
    System.out.println("cadence:"+cadence+"
    speed:"+speed+" gear:"+gear);
  }

}
```

Inheritance in Java (1)

- Different kinds of objects often have a certain amount in common with each other.
- Object-oriented programming allows classes to ***inherit*** commonly used state and behavior from other classes.
- In the Java programming language, each class is allowed to have one direct ***superclass***, and each superclass has the potential for an unlimited number of ***subclasses***

Inheritance in Java (2)

Example:




```
class MountainBike extends Bicycle {
  // new fields and methods defining a mountain bike would go here
}
```

Interfaces in Java

- An **interface** is a group of related methods with empty bodies. Methods form the object's *interface* with the outside world.
- Example:

```
interface Bicycle {
  void changeCadence (int newValue);
  void changeGear (int newValue);
  void speedUp (int increment);
  void applyBrakes (int decrement);
}

class ACMEBicycle implements Bicycle {
  // remainder of this class implemented as before
}
```

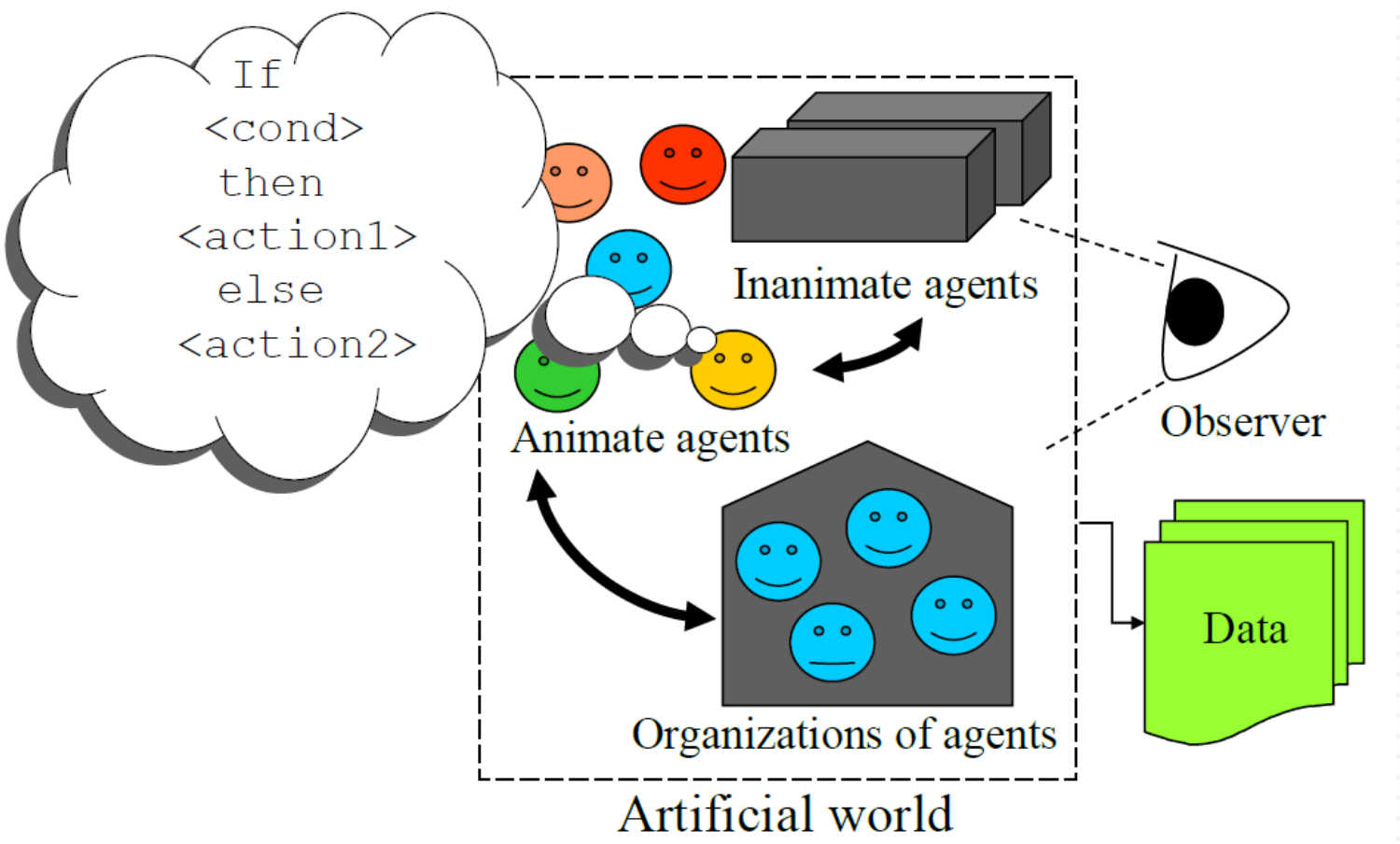
Simple OOP program (1)


```
public class Bicycle {
  // the Bicycle class has three fields
  public int cadence;
  public int gear;
  public int speed;

  // the Bicycle class has one constructor
  public Bicycle(int startCadence, int startSpeed, int
startGear) {
  gear = startGear;
  cadence = startCadence;
  speed = startSpeed;
  }
  // the Bicycle class has four methods
  public void setCadence(int newValue) {
  cadence = newValue;
  }
  public void setGear(int newValue) {
  gear = newValue;
  }
  public void applyBrake(int decrement) {
  speed -= decrement;
  }
  public void speedUp(int increment) {
  speed += increment;
  }
}
```

Simple OOP program (2)


```
public class MountainBike extends Bicycle {
  // the MountainBike subclass has one field
  public int seatHeight;

  // the MountainBike subclass has one constructor
  public MountainBike(int startHeight, int startCadence, int startSpeed, int startGear) {
  super (startCadence, startSpeed, startGear);
  seatHeight= startHeight;
  }

  // the MountainBikesubclass has one method
  public void setHeight(int newValue) {
  seatHeight= newValue;
  }
}
```

Objects in an Agent-Based Model (1)



Objects in an Agent-Based Model (2)