Java Programming 2 Visibility modifiers, static members

Mary Ellen Foster
MaryEllen.Foster@glasgow.ac.uk

Semester 1 2020/2021

Question

For the `BankAccount` class …

How do we stop other code from directly modifying the fields of individual `BankAccount` objects?

```
class BankAccount {
  int balance;
  String name;
  void deposit(int value) { this.balance += value; }
  void withdraw(int value) { this.balance -= value; }
  BankAccount(String name, int initialAmount) {
    this.name = name;
    this.balance = initialAmount;
  }
}
```

```
BankAccount b = new BankAccount("Mary", 1000);
b.value += 1000;
b.name = "Eve";
```

Visibility modifiers

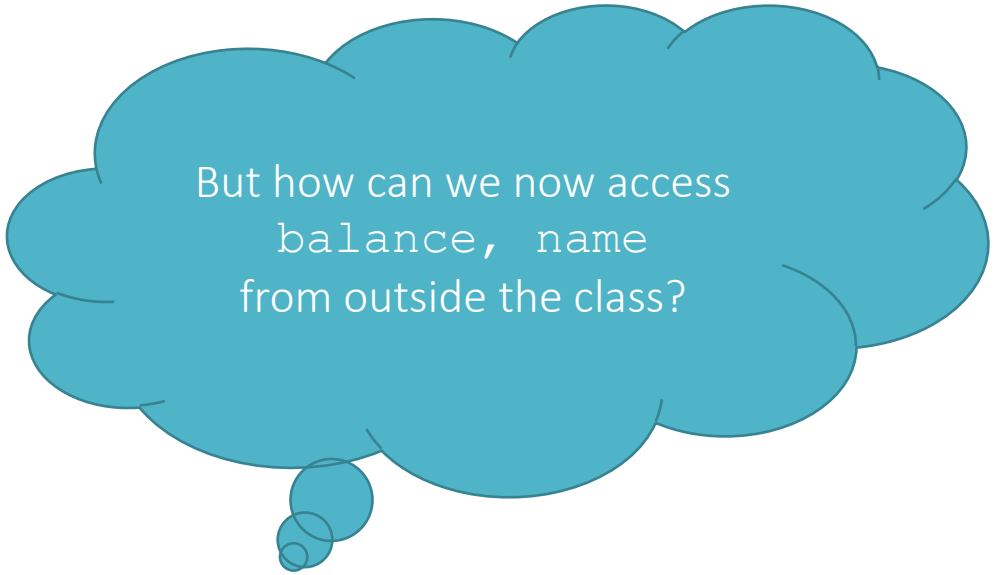
Modifier	Same class	Same package	Any subclass	Any class
`public`	●	●	●	●
`protected`	●	●	●	
(default)	●	●		
`private`	●			

Used to limit the visibility of class members (fields and methods)

Specify as part of member declaration – `private int balance;`

Bank account revisited (again)

```
public class BankAccount {

  private int balance;
  private String name;

  public void deposit(int value) { this.balance += value; }

  public void withdraw(int value) { this.balance -= value; }

  public BankAccount(String name, int initialAmount) {

    this.name = name;
    this.balance = initialAmount;
  }
}
```



Getters and setters

Give controlled access to private properties

```
public String getName() {
      return name;
}

public void setName (String name) {
      this.name = name;
}

// … and so on
```

Bank account class again – added ID field

```
public class BankAccount {

   private int balance;
  private String name;
  private int id;

   public BankAccount(String name, int initialAmount) {

      this.name = name;
     this.balance = initialAmount;
     this.id = // ??? What to do here ???

   }
}
```

Bank account class again – added ID field

```
public class BankAccount {

   private int balance;
   private String name;
   private int id;

   private static int NEXT_ID = 0;

   public BankAccount(String name, int initialAmount) {

      this.name = name;
      this.balance = initialAmount;
      this.id = BankAccount.NEXT_ID++;

   }
}
```

Static members

Associated directly with the class itself, not with any object of that class

Static field: only one variable, no matter how many objects of the class have been created (even zero)

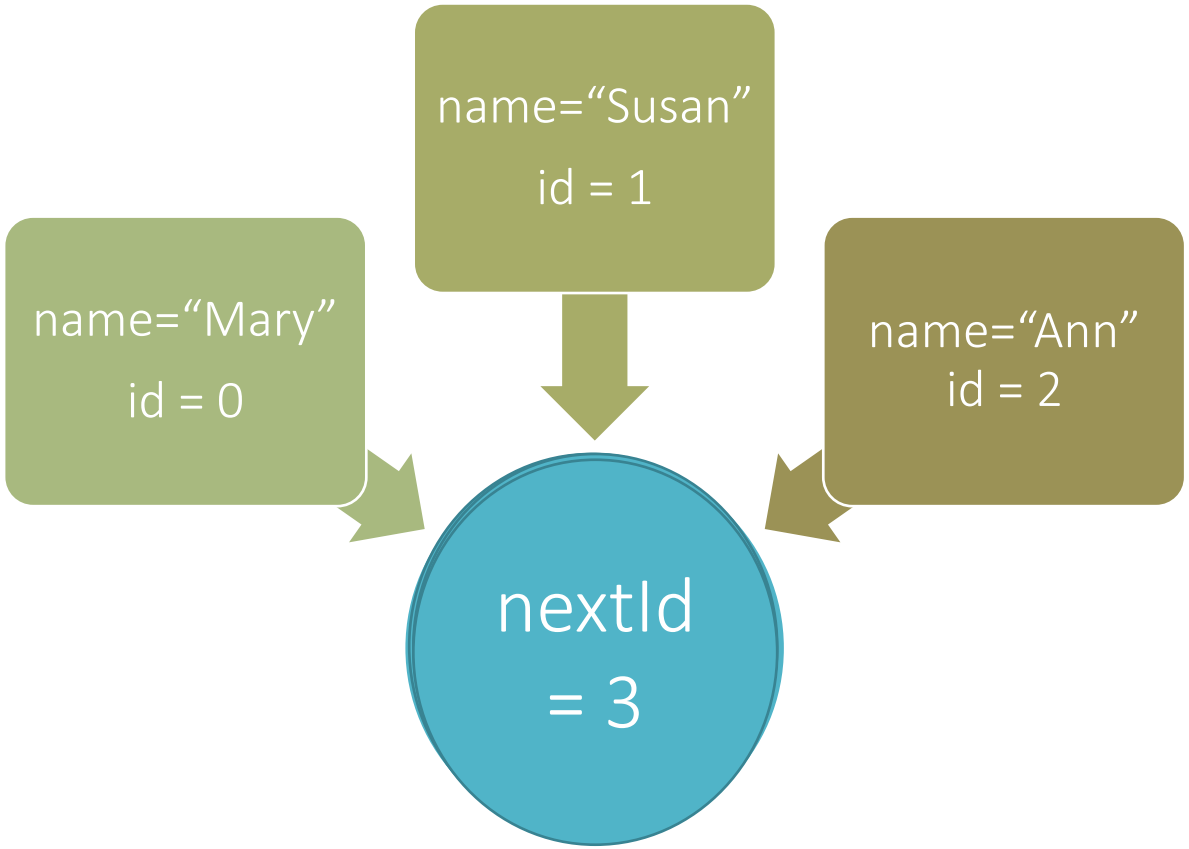
Examples you have seen: `Double.MAX_VALUE`, `System.out`

Static method: performs a general task for the class; can only access other static members

Examples you may have seen: `Math.random()`, `Integer.parseInt()`

In updated `BankAccount` class, `NEXT_ID` field is static (why?)

Static and instance fields in BankAccount



Understanding `System.out.println("hello")`