

## Commenting the Public Interface & Instance Fields

## Commenting on the Public Interface

```
/**  
 * Withdraws money from the bank account.  
 * @param the amount to withdraw  
 */  
public void withdraw(double amount)  
{  
    // implementation filled in later  
}
```

```
/**  
 * Gets the current balance of the bank account.  
 * @return the current balance  
 */  
public double getBalance()  
{  
    // implementation filled in later  
}
```

## Class Comment

```
/**  
 * A bank account has a balance that can  
 * be changed by deposits and withdrawals.  
 */  
public class BankAccount  
{  
    . . .  
}
```

### • Provide documentation comments for

- every class
- every method
- every parameter
- every return value.

## Javadoc Method Summary

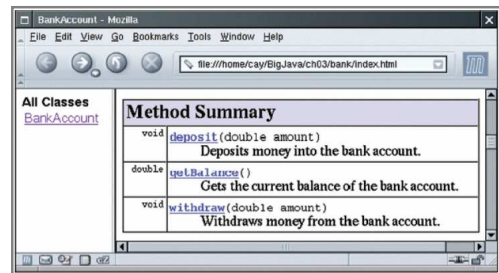


Figure 3:  
A Method Summary Generated by javadoc

## Javadoc Method Detail



Figure 4:  
Method Detail Generated by javadoc

## Instance Fields

- An object stores its data in instance fields
- Field: a technical term for a storage location inside a block of memory
- Instance of a class: an object of the class
- The class declaration specifies the instance fields:

```
public class BankAccount  
{  
    . . .  
    private double balance;  
}
```

## Instance Fields

- An instance field declaration consists of the following parts:
  - access specifier (such as `private`)
  - type of variable (such as `double`)
  - name of variable (such as `balance`)
- Each object of a class has its own set of instance fields
- You should declare all instance fields as `private`

## Instance Fields

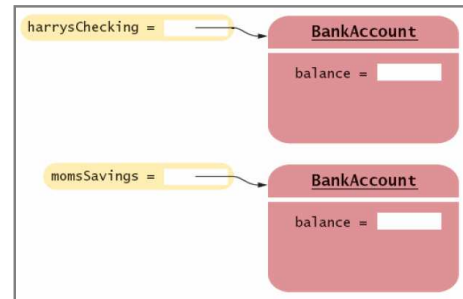


Figure 5:  
Instance Fields

## Syntax 3.4: Instance Field Declaration

```
accessSpecifier class ClassName
{
    . . .
    accessSpecifier fieldType fieldName;
    . . .
}
```

**Example:**

```
public class BankAccount
{
    . . .
    private double balance;
    . . .
}
```

**Purpose:**

To define a field that is present in every object of a class

## Accessing Instance Fields

- The `deposit` method of the `BankAccount` class can access the `private` instance field:

```
public void deposit(double amount)
{
    double newBalance = balance + amount;
    balance = newBalance;
}
```

*Continued..*

## Accessing Instance Fields

- Other methods cannot:

```
public class BankRobber
{
    public static void main(String[] args)
    {
        BankAccount momsSavings = new BankAccount(1000);
        . . .
        momsSavings.balance = -1000; // ERROR
    }
}
```

- Encapsulation = Hiding data and providing access through methods