

Recap of OO concepts

Objects, classes, methods and more.

Produced Ms. Mairead Meagher
by: Dr. Siobhán Drohan



Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics
<http://www.wit.ie/>

Classes and Objects

- A **class**
 - defines a group of related **methods** (functions) and **fields** (variables / properties).

The screenshot shows the Oracle Java Platform SE 7 API documentation for the `String` class. The browser address bar shows the URL `https://docs.oracle.com/javase/7/docs/api`. The navigation bar includes links for Overview, Package, Class (selected), Use, Tree, Deprecated, Index, and Help. Below the navigation bar, there are links for Prev Class, Next Class, Frames, No Frames, and All Classes. The main content area is divided into two sections: a left sidebar and a right main area.

Left Sidebar:

- java.lang
- Class String**
- java.lang.Object
- java.lang.String

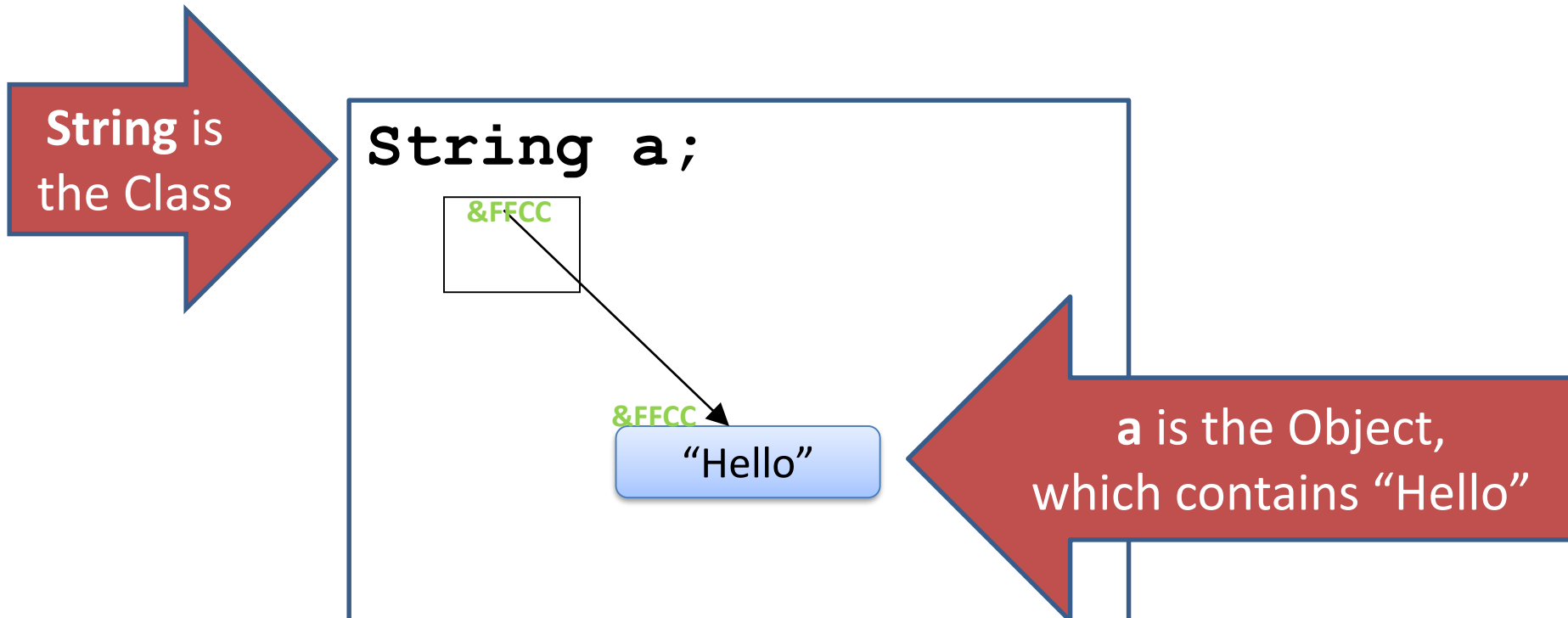
Right Main Area:

Method Summary

Modifier and Type	Method and Description
char	<code>charAt(int index)</code> Returns the char value at the specified index.
int	<code>codePointAt(int index)</code> Returns the character (Unicode code point) at the specified index.
int	<code>codePointBefore(int index)</code> Returns the character (Unicode code point) before the specified index.
int	<code>codePointCount(int beginIndex, int endIndex)</code> Returns the number of Unicode code points in the specified text range of this String.
int	<code>compareTo(String anotherString)</code> Compares two strings lexicographically.
int	<code>compareToIgnoreCase(String str)</code> Compares two strings lexicographically, ignoring case differences.
String	<code>concat(String str)</code> Concatenates the specified string to the end of this string.
boolean	<code>contains(CharSequence s)</code> Returns true if and only if this string contains the specified sequence of char values.
boolean	<code>contentEquals(CharSequence cs)</code> Compares this string to the specified CharSequence.
boolean	<code>contentEquals(StringBuffer sb)</code> Compares this string to the specified StringBuffer.
static String	<code>copyValueOf(char[] data)</code> Returns a String that represents the character sequence in the array specified.
static String	<code>copyValueOf(char[] data, int offset, int count)</code> Returns a String that represents the character sequence in the array specified.
boolean	<code>endsWith(String suffix)</code> Tests if this string ends with the specified suffix.
boolean	<code>equals(Object anObject)</code> Compares this string to the specified object.

Classes and Objects

- An **object**
 - is a single instance of a class
 - i.e. an object is created (instantiated) from a class.



Classes and Objects – Many Objects

- Many **objects** can be constructed from a single **class** definition.
- Each **object** must have a unique name within the program.

Ver 1.0

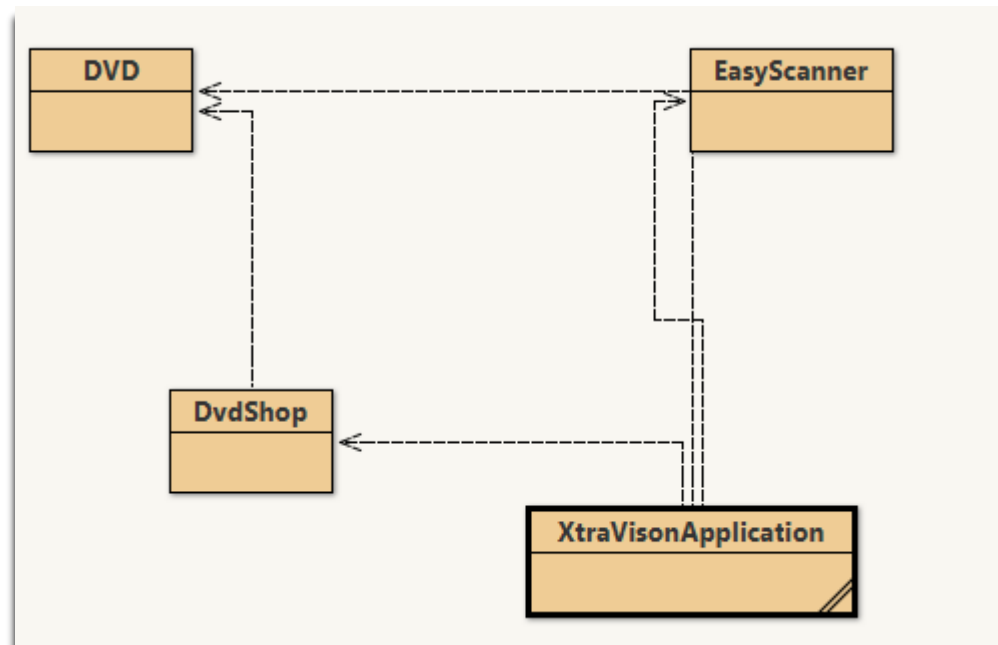
XTRAVISION APPLICATION



XtraVisionApplication



- We will recap object oriented concepts through the study of your first year assignment XtraVisionApplication.



DVD



DVD

- The **DVD** class stores **details** about a product
 - id
 - name
 - ageClassification
 - category
 - numMinutes
 - lenofTime
 - rating

DvdShop



DvdShop

- The **DvdShop** class stores a **collection of dvds and other details about the collection**
 - dvdList
 - total (of dvds)

EasyScanner

- The **EasyScanner** class has the following methods to make reading from keyboard simple:
 - **int** nextInt()
 - **double** nextDouble()
 - **String** nextString()
 - **char** nextChar()

A blue rounded rectangle containing the text "EasyScanner" in white.

EasyScanner

A DVD Class... fields



```
private String dvdId;  
private String dvdName;  
private int ageClassification;  
private String category;  
private int numMinutes;  
private static int lenOfTime;  
private int rating;
```

A DVD Class... fields and constructor

```
public class Class {
```

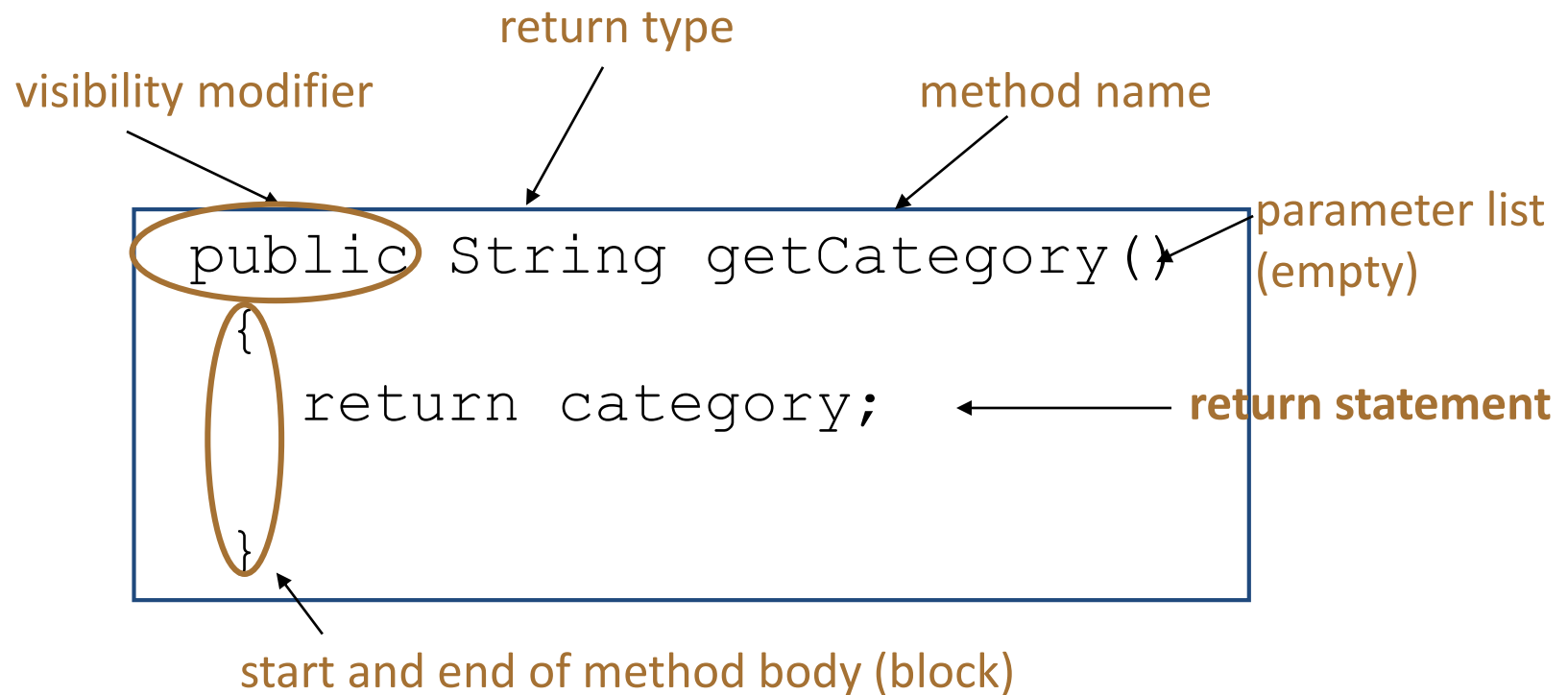
```
    private String dvdId;  
    private String dvdName;  
    private int ageClassification;  
    private String category;  
    private int numMinutes;  
    private static int lenOfTime;  
    private int rating;
```

```
    public DVD(String dvdIdIn, String nIn, String categoryIn,  
               int ageClassificationIn, int numMinutesIn)  
    {  
        dvdId = dvdIdIn;  
        dvdName = nIn;  
        category = categoryIn;  
        ageClassification = ageClassificationIn;  
        numMinutes = numMinutesIn;  
        rating = 0;           //rating will be decided later  
        lenOfTime = 0;       //lenOfTime will be changed later  
    }  
}
```

Getters (Accessor Methods)

- **Accessor** methods
 - return information about the **state** of an object
 - i.e. the values stored in the fields.
- A **'getter'** method
 - is a specific type of **accessor** method and typically:
 - contains a **return statement**
(as the last executable statement in the method).
 - defines a **return type**.
 - **does NOT change the object state**.

Getters



A Product Class...getters

```
public String getDvdId()
{
    return dvdId;
}

public String getDvdName()
{
    return dvdName ;
}

public int getAgeClassification()
{
    return ageClassification;
}

public String getCategory()
{
    return category;
}
```

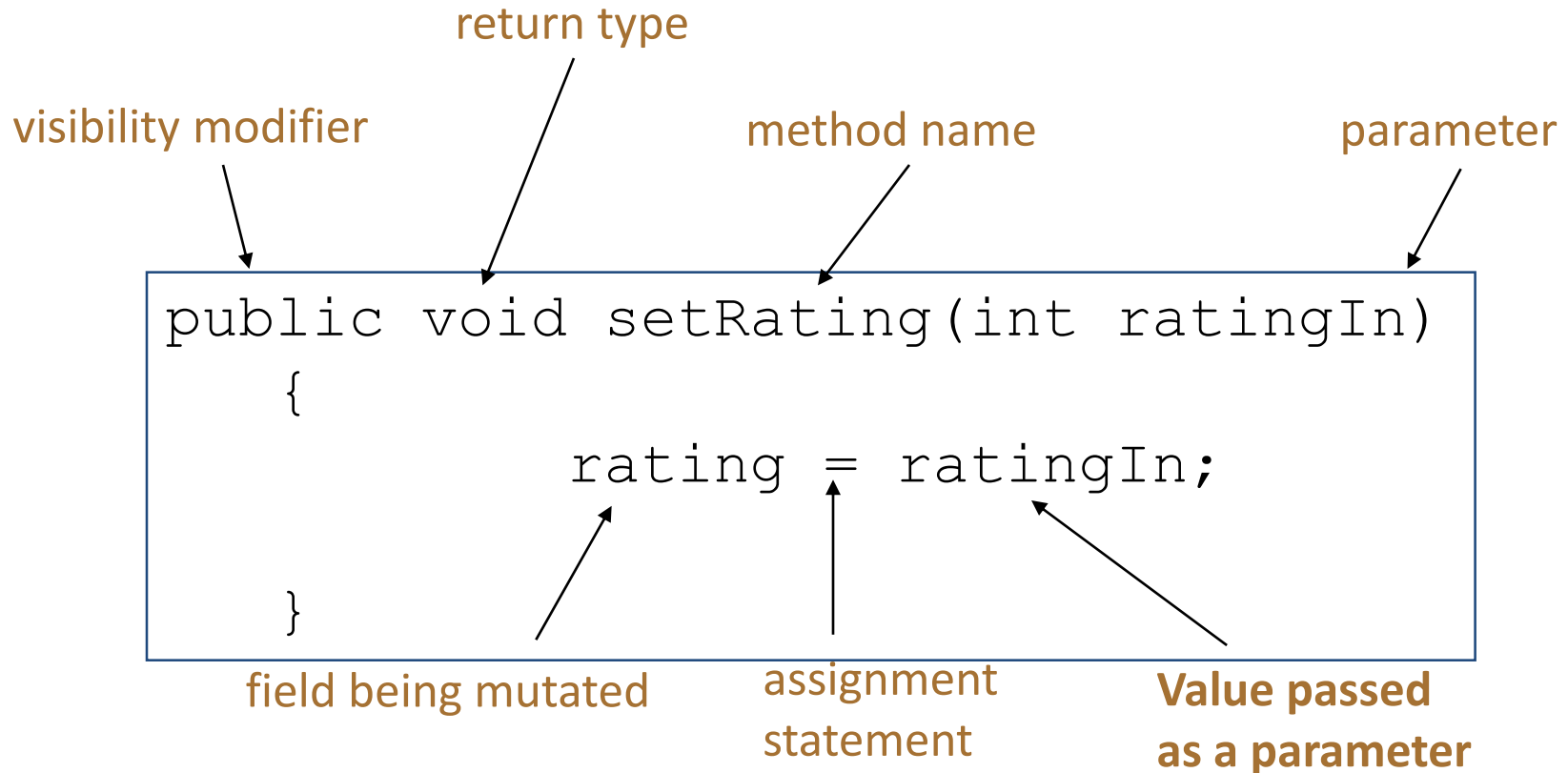
```
public int getNumMinutes()
{
    return numMinutes;
}

public static int getLenOfTime()
{
    return lenOfTime;
}
```

Setters (Mutator methods)

- **Mutator** methods
 - change (i.e. mutate!) an object's state.
- A **'setter'** method
 - is a specific type of **mutator** method and typically:
 - contains an **assignment statement**
 - takes in a **parameter**
 - **changes the object state.**

Setters



A DVD Class...setters

```
public void setDvdId(String dvdIdIn )
{
    dvdId = dvdIdIn;
}
public void setDvdName(String dvdNameIn)
{
    dvdName = dvdNameIn;
}
public void setAgeClassification(int ageClassificationIn)
{
    ageClassification = ageClassificationIn;
}

public void setCategory(String categoryIn)
{
    category = categoryIn;
}
```

A DVD Class...setters contd

```
public void setRating(int ratingIn)
{
    rating = ratingIn;
}
public void setNumMinutes(int numMinutesIn)
{
    numMinutes = numMinutesIn;
}
public static void setLenOfTime()
{
    lenOfTime = lenOfTime + 1;
}
```

Getters/Setters

- For **each instance field** in a class, you are normally asked to write:
 - A **getter**
 - Return statement
 - A **setter**
 - Assignment statement

Encapsulation in Java – steps 1-3

Encapsulation Step	Approach in Java
1. Wrap the data (fields) and code acting on the data (methods) together as single unit.	<pre>public class <i>ClassName</i> { <i>Fields</i> <i>Constructors</i> <i>Methods</i> }</pre>
2. Hide the fields from other classes.	Declare the fields of a class as <u>private</u>.
3. Access the fields only through the methods of their current class.	Provide <u>public</u> setter and getter methods to modify and view the fields values.

A DVD Class... An Encapsulated Class

1. DVD class **wraps** the data (fields) and code acting on the data (methods) together as **single unit**.

Fields are **hidden** from other classes.

```
public class DVD
{
    private String dvdId;
        : fields

    public DVD ( ...) { .. }
    public other methods
        :
}
```

Methods are **available** from other classes.

Using the DVD Class

1

```
private DVD product;
```

Declaring an object **dvd**,
of type **DVD**.

dvd

2

```
dvd = new DVD("01", "First Man", 12, "Some Category", 105, 3);
```

Calls the **DVD**
constructor to build
the **dvd** object in
memory.

dvd : DVD

private String dvdId

"01"

private String dvdName

"First Man"

private int ageClassification

12

private String category

"Some Category"

private int numMinutes

3

private int rating

0

Inspect

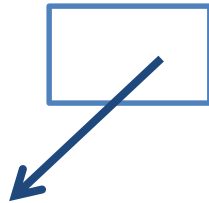
Get

Multiple Product Objects

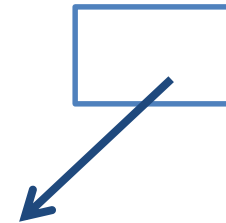
```
dvd = new DVD("01","First Man", 12, "Some Category", 12, 3);
```

```
dvd2 = new DVD("02","Avengers Endgame", 15, "SuperHeroes", 15, 4);
```

dvd



dvd2



dvd : DVD

private String dvdId	<input type="text" value="01"/>	<input type="button" value="Inspect"/> <input type="button" value="Get"/> <input type="button" value="Close"/>
private String dvdName	<input type="text" value="First Man"/>	
private int ageClassification	<input type="text" value="12"/>	
private String category	<input type="text" value="Some Category"/>	
private int numMinutes	<input type="text" value="3"/>	
private int rating	<input type="text" value="0"/>	

dvd2 : DVD

private String dvdId	<input type="text" value="02"/>	<input type="button" value="Inspect"/> <input type="button" value="Get"/> <input type="button" value="Close"/>
private String dvdName	<input type="text" value="Avengers Endgame"/>	
private int ageClassification	<input type="text" value="15"/>	
private String category	<input type="text" value="SuperHeroes"/>	
private int numMinutes	<input type="text" value="4"/>	
private int rating	<input type="text" value="0"/>	

Questions?

