

Sesión 05: Terminología

Programación 2

Ángel Herranz

Febrero 2019

Universidad Politécnica de Madrid

En capítulos anteriores

- Sobre los IDEs
- Clases y objetos (intro)
- **Objetos**, **referencias** y variables (y **primitivos**)
- **Clases**: plantilla para crear objetos

En capítulos anteriores

- Sobre los IDEs
- Clases y objetos (intro)
- **Objetos**, **referencias** y variables (y **primitivos**)
- **Clases**: plantilla para crear objetos

Encapsular

datos y comportamiento

- Racionales

Objetos, referencias y variables

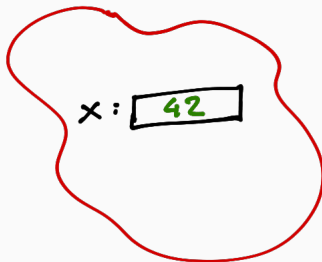
```
public class Sesion02 {  
    public static void main(String args[]) {  
        A a;  
        a = new A();  
    }  
}
```

a: null

Objetos, referencias y variables

```
public class Sesión02 {  
    public static void main(String args[]) {  
        A a;  
        a = new A();  
    }  
}
```

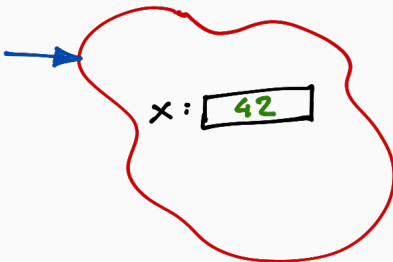
a: null



Objetos, referencias y variables

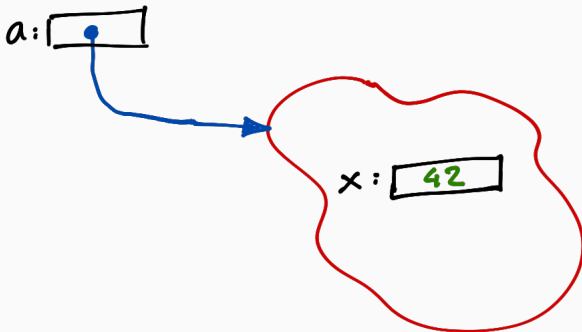
```
public class Sesion02 {  
    public static void main(String args[]) {  
        A a;  
        a = new A();  
    }  
}
```

a: null



Objetos, referencias y variables

```
public class Sesion02 {  
    public static void main(String args[]) {  
        A a;  
        a = new A();  
    }  
}
```



Clases: datos + comportamiento

```
class Cancion {  
    String titulo;  
    String interprete;  
    int duracion;  
    ...  
    Cancion (String titulo,  
             String interprete,  
             int duracion) {  
        this.titulo = titulo;  
        this.interprete = interprete;  
        this.duracion = duracion;  
    }  
    ...  
}
```

```
String duracion() {  
    String minutos =  
        duracion / 60;  
    String segundos =  
        duracion % 60;  
    if (segundos.length() == 1)  
        segundos  
        = "0" + segundos;  
    }  
    return  
        minutos + ":" + segundos;  
    }  
}
```


Clases: datos + comportamiento

```
class Cancion {
    String titulo;
    String interprete;
    int duracion;
    ...
    Cancion (String titulo,
             String interprete,
             int duracion) {
        this.titulo = titulo;
        this.interprete = interprete;
        this.duracion = duracion;
    }
    ...

    String duracion() {
        String minutos =
            duracion / 60;
        String segundos =
            duracion % 60;
        if (segundos.length() == 1)
            segundos
                = "0" + segundos;
        }
        return
            minutos + ":" + segundos;
    }
}
```

Clases: datos + comportamiento

```
class Cancion {  
    String titulo;  
    String interprete;  
    int duracion;  
    ...  
    Cancion (String titulo,  
             String interprete,  
             int duracion) {  
        this.titulo = titulo;  
        this.interprete = interprete;  
        this.duracion = duracion;  
    }  
    ...  
}
```

```
String duracion() {  
    String minutos =  
        duracion / 60;  
    String segundos =  
        duracion % 60;  
    if (segundos.length() == 1)  
        segundos  
        = "0" + segundos;  
    }  
    return  
        minutos + ":" + segundos;  
    }  
}
```

Clases: datos + comportamiento

```
class Cancion {  
    String titulo;  
    String interprete;  
    int duracion;  
    ...  
    Cancion (String titulo,  
             String interprete,  
             int duracion) {  
        this.titulo = titulo;  
        this.interprete = interprete;  
        this.duracion = duracion;  
    }  
    ...  
}
```

```
String duracion() {  
    String minutos =  
        duracion / 60;  
    String segundos =  
        duracion % 60;  
    if (segundos.length() == 1)  
        segundos  
        = "0" + segundos;  
    }  
    return  
        minutos + ":" + segundos;  
}
```

Clases: datos + comportamiento

```
class Cancion {  
    String titulo;  
    String interprete;  
    int duracion;  
    ...  
    Cancion (String titulo,  
             String interprete,  
             int duracion) {  
        this.titulo = titulo;  
        this.interprete = interprete;  
        this.duracion = duracion;  
    }  
    ...  
}
```

```
String duracion() {  
    String minutos =  
        duracion / 60;  
    String segundos =  
        duracion % 60;  
    if (segundos.length() == 1)  
        segundos  
        = "0" + segundos;  
    }  
    return  
        minutos + ":" + segundos;  
    }  
}
```

Terminología OO + Java i

atributo, clase, constructor, dato
básico, instancia, invocación, mensaje,
miembro, método, modificador,
objeto, observador, puntero,
referencia, tipo, variable

Terminología OO + Java i

*attribute, class, constructor, basic
data, instance, invocation, message,
member, method, modifier,
object, observer, pointer,
reference, type, variable*

Datos y comportamiento

- Nadie usa los términos *datos* y *comportamiento*
- Datos: **atributos** (*attribute*)
- Comportamiento: **métodos** (*method*)
 - **Constructor**: mismo nombre que la clase
 - **Observador**: observa, devuelve algo, no modifica
 - **Modificador** modifica, no devuelve nada (**void**)
- Ámbos (datos y métodos): **miembros** (*member*)²

²Terminología Java

Datos y comportamiento

- Nadie usa los términos *datos* y *comportamiento*
- Datos: **atributos** (*attribute*)
- Comportamiento: **métodos** (*method*)
 - **Constructor**: mismo nombre que la clase
 - **Observador**: observa, devuelve algo, no modifica
 - **Modificador** modifica, no devuelve nada (**void**)
- ↻ ¿Puede haber métodos *observamodificadores*?¹
- Ámbos (datos y métodos): **miembros** (*member*)²

¹Sí, pero hagamos caso a Bertrand Meyer y a su principio CQS
(*Command Query Separation Principle*)

²Terminología Java

Algunos sinónimos

Objeto = Instancia

Puntero = Referencia

Método = Mensaje

Invocar = Enviar mensaje

Clase \subseteq Tipo

recordar Cuando usas un código

- No quieres saber cómo está hecho
- Sólo quieres saber qué hace
- No quieres repetir trabajo

Ocultación i

- Estás usando `Racional`

```
System.out.println(  
    "El valor decimal es " + r.num / r.den  
);
```

- El desarrollador de `Racional` decide renombrar `num` y `den` por `numerador` y `denominador`

¿Qué va a pasar?

- Estás usando `Racional`

```
System.out.println(  
    "El valor decimal es " + r.num / r.den  
);
```
- El desarrollador de `Racional` decide renombrar `num` y `den` por `numerador` y `denominador`

Ocultación ii

- Has desarrollado una clase para representar puntos en 2D

```
class Punto2D {  
    double x;  
    double y;  
    ...  
}
```

- Tienes un modificador **void** rotar(**double** rad)

Ocultación ii

- Has desarrollado una clase para representar puntos en 2D

```
class Punto2D {  
    double x;  
    double y;  
    ...  
}
```

- Tienes un modificador **void** rotar(**double** rad)
- Y ahora sabes que sabes que va a ser usado masivamente

Ocultación ii

¿Qué vas a hacer?

- Has desarrollado una clase para representar puntos en 2D

```
class Punto2D {  
    double x;  
    double y;  
    ...  
}
```

- Tienes un modificador **void** rotar(**double** rad)
- Y ahora sabes que sabes que va a ser usado masivamente

Ocultación ii

¿Qué vas a hacer?

¿Qué va a pasar?

- Has desarrollado una clase para representar puntos en 2D

```
class Punto2D {  
    double x;  
    double y;  
    ...  
}
```

- Tienes un modificador **void** rotar(**double** rad)
- Y ahora sabes que sabes que va a ser usado masivamente

¿Ideas?

