

# Constructores, setters & getters. Método equals



# ITESO

Universidad Jesuita  
de Guadalajara

Materia: Programación Orientada a objetos

Maestr@: Victor Hugo Zaldivar Carrillo

Alumn@: José Luis Almendarez Gonzalez(737807)

## Corridas de cada ejercicio(Narradas)

```
package mx.iteso.calendar;

public class TestDate {
    public static void main(String[] args) {

        //A)
        Date d1= new Date();
        //B)
        Date d2= new Date( day: 31, month: 12, year: 2016);
        //C)
        Date d3= new Date().clone(d1);
        //D)
        Date d4= new Date(d1.getDay(),d2.getMonth(), d3.getYear(), format: 1);
        //E)
        Date d5= new Date().clone(d2);
        d5.next();
        //F)
        d3.setMonth(4);
        //G)
        Date[] elementos={d1,d2,d3,d4,d5};
        for (int i = 0; i < elementos.length; i++) {
            System.out.println("La fecha de es: "+elementos[i]);
        }
    }
}
```

Esto es lo que podemos ver cuando intentamos probar el objeto Date(). Primero creamos 5 objetos diferentes y los fuimos modificando a como nos pedía el ejercicio, hasta tener el resultado final. Más adelante se explicara de manera más a detalle de qué trataba cada ejercicio, que método se usó para resolver y cómo funciona ese método, para darle mayor comprensión al lector. Por ahora podemos decir que el ejercicio fue un éxito.

```
La fecha de es: 01/01/17
La fecha de es: 31/12/16
La fecha de es: 01/04/17
La fecha de es: 1-Dic-2017
La fecha de es: 01/01/17
```

### A)

El ejercicio A) nos pide que imprimamos la fecha predeterminada, esto se hace únicamente creando un objeto y no rellenando con valores nuevos. Pues, gracias a los constructores nosotros ya declaramos anteriormente cual es la fecha default.

```
4 usages
public Date() { this( day: 1, month: 1, year: 2017, format: 0); }
1 usage
public Date(int day, int month, int year) { this(day,month,year, format: 0); }
3 usages
public Date(int day, int month, int year, int format){
    setDay(day);
    setMonth(month);
    setYear(year);
    setFormat(format);
}
```

B)

El siguiente ejercicio es que tenemos que rellenar valores como día mes y año, sin el format. El format se pone el predeterminado pues así lo declaramos en los constructores. Los otros valores los podemos modificar gracias a la ayuda de los setters que construimos.

```
public void setDay(int day){
    //El set de day esta hecho asi para dejar en claro que si alguno de los valores se pasan de rango en un mes en
    // especifico se ponga el valor predeterminado
    if ((day>31||day<1)&(this.month ==1||this.month ==3||this.month ==5||this.month ==7||this.month ==8||this.month ==10||this.month ==12)){
        this.day= 1;
    } else if ((day>30 || day < 1)& (this.month ==4||this.month==6||this.month==9||this.month==11)){
        this.day= 1;
    } else if ((day>28 || day < 1)& (this.month ==2)) {
        this.day= 1;
    }
    else {
        this.day=day;
    }
}
```

```
public void setYear(int year){
    //Este set year unicamente corrige la salida de rangos a los valores predeterminados
    //como los anteriores
    if(year>3000||year<1900){
        this.year=2017;
    } else {
        this.year=year;
    }
}

1 usage

public void setFormat(int format){
    //Como los anteriores, corrige la salida de rangos y apunta al propio valor del input
    if (format<0||format>2){
        this.format = 0;
    }else {
        this.format = format;
    }
}
```

```
public void setMonth(int month){
    //El set de month tambien corrige los rangos asi como que dependiendo del valor de month tambien cambia el valor
    //de Monthname
    if(month>12||month<1){
        this.month=1;
    } else {
        if (month==1){
            this.month=1;
            this.monthName="Enero";
        } else if (month==2){
            this.month=2;
            this.monthName="Febrero";
        } else if (month==3){
            this.month=3;
            this.monthName="Marzo";
        } else if (month==4){
            this.month=4;
            this.monthName="Abril";
        } else if (month==5){
            this.month=5;
            this.monthName="Mayo";
        } else if (month==6){
            this.month=6;
        }
    }
}
```

En los sets se declara un rango de valores válidos, si el valor queda fuera de ese rango automáticamente se asigna el valor predeterminado. Si no, se le asigna el valor que se introdujo en el input. Lo unico a destacar es que en setMonth tambien seteamos monthName, vinculando su numero de mes con el nombre de este mismo.

### C)

En el ejercicio tenemos que crear un objeto llamado d3, este es una copia de d1. Lo hacemos a través del método clone.

```
public Date clone(Date obj1) {  
    Date objeto = new Date();//Se crea un objeto provisional  
  
    //se le asignan los valores del objeto copiado al objeto provisional  
    objeto.day = obj1.day;  
    objeto.month = obj1.month;  
    objeto.year = obj1.year;  
    return objeto;//se regresa la informacion del objeto provisional para darsela al objeto que recibira la clonacion  
}
```

Lo que hace este método es crear un objeto provisional que va a tener los mismos valores que el objeto que asignamos en el input y después los valores se asignan a d3, pues los regresamos.

### D)

Lo que hacemos aquí es rellenar los valores que nos piden con los valores de otros objetos, para esto usamos los getters, lo que los getters hacen es apuntar a la información encapsulada del objeto, entonces lo que básicamente hacemos es invocar la información para poderla usar.

```
//Getters  
// Estos getters te consiguen los valores ya establecidos  
11 usages  
public int getDay(){return day;}  
27 usages  
public int getMonth(){return month;}  
2 usages  
public int getYear(){return year;}  
public String getMonthName(){ return monthName;}  
public int getFormat(){return format;}
```

E)

Para este ejercicio lo que había que hacer era copiar los valores de d2 en un nuevo objeto llamado d5 y pasar al día siguiente después de ese. Lo primero se hace con el método clone que ya expliqué. Lo segundo con el método next() que lo que hace es que dependiendo de ciertos factores como el día en el que estamos y el mes en el que nos encontramos decidiremos si sumar un día, un mes o pasar al próximo año. No es nada más que un par de decisiones if-else que apuntan a valores encapsulados.

```
public void next(){// Aquí dependiendo del rango de valores del día y el mes decidimos si sumar un día, ajustar un día y sumar un mes o
//ajustar día, mes y sumar anno
//Es exactamente igual para los meses que tienen 30,31 y 28 días

if (getDay()==31 && (getMonth() ==1|| getMonth()==3|| getMonth()==5||getMonth()== 7||getMonth()==8||getMonth()==10)){
    setDay(1);
    setMonth(getMonth()+1);
}else if (getDay()<31 && ((getMonth() ==1|| getMonth()==3|| getMonth()==5||getMonth()== 7||getMonth()==8||getMonth()==10))){
    setDay(getDay()+1);
}else if(getDay()==31&& getMonth()==12){
    setDay(1);
    setMonth(1);
    setYear(getYear()+1);
}

if (getDay()<30 && (getMonth() ==4||getMonth() ==6||getMonth() ==9||getMonth() ==11)){
    setDay(getDay()+1);
} else if(getDay()>=30 &&(getMonth() ==4||getMonth() ==6||getMonth() ==9||getMonth() ==11)){
    setDay(1);
    setMonth(getMonth()+1);
}

if (getDay()<28 && (getMonth() ==2)){
    setDay(getDay()+1);
} else if(getDay()>=28 &&(getMonth() ==2)){
    setDay(1);
    setMonth(getMonth()+1);
}
}
```

F)

El siguiente ejercicio nos pide que cambiemos el mes de d3 a abril, esto es muy sencillo pues tenemos que usar los setters explicados anteriormente para hacerlo. Solo volvemos a ajustar el mes del objeto.

G)

El último ejercicio, los print únicamente tenemos que hacer un arreglo de tipo date y mediante un ciclo for podemos imprimir todos los valores del arreglo iterando por el número de valores en el mismo.

## Conclusiones

La verdad me hubiera encantado entender cómo funciona todo esto antes de haber tenido el examen pues ahora sé cómo funciona la construcción de objetos, sus métodos, sus alcances, etc. Me ha quedado muy claro como para poder entender que esto es todo, todo lo que vayamos a aprender a partir de ahora solo son adiciones a esta estructura de datos. Pero teniendo las bases claras es obvio que todo lo que haremos tratará acerca de la encapsulación de los datos de un objeto y como jugamos y modificamos estos mismos. Realmente no se que mas agregar a la conclusión pues hemos estado viendo esto desde el principio del semestre y todas las dudas que me surgen siempre las intento tratar en clase. Únicamente puedo expresar mi emoción hacia el nuevo tema de para hacer `@override` a los objetos y añadir más, falta aprenderse bien como se llama todo y la sintaxis pero los conceptos los tengo claros, pero bueno, eso es tema futuro. Por ahora todo bien con este tema.

## Código indentado y Documentado

### TestDate.java

```
package mx.iteso.calendar;

public class TestDate {
    public static void main(String[] args) {

        //A)
        Date d1= new Date();
        //B)
        Date d2= new Date(31,12,2016);
        //C)
        Date d3= new Date().clone(d1);
        //D)
        Date d4= new Date(d1.getDay(),d2.getMonth(), d3.getYear(),1);
        //E)
        Date d5= new Date().clone(d2);
        d5.next();
        //F)
        d3.setMonth(4);
        //G)
        Date[] elementos={d1,d2,d3,d4,d5};
        for (int i = 0; i < elementos.length; i++) {

            System.out.println("La fecha de es: "+elementos[i]);
        }
    }
}
```

### Date.java

```
package mx.iteso.calendar;
public class Date {

    //Declaracion de caracteristicas
    private int day=1;
    private int month=1;
    private int year=2017;
    private String monthName="Enero";
    private int format = 0;

    //Constructores
    //Estos son los 3 cronstructores hechos para dejar en claro los valores con las
    que se van a trabajar asi como
```

```

//sus valores predeterminados
public Date(){
    this(1,1,2017,0);
}
public Date(int day, int month, int year){
    this(day,month,year,0);
}
public Date(int day, int month, int year, int format){
    setDay(day);
    setMonth(month);
    setYear(year);
    setFormat(format);
}

//Setters
public void setDay(int day){
    //El set de day esta hecho asi para dejar en claro que si alguno de los valores
    se pasan de rango en un mes en
    // especifico se ponga el valor predeterminado
    if ((day>31||day<1)&(this.month ==1||this.month ==3||this.month ==5||this.month
==7||this.month ==8||this.month ==10||this.month ==12)){
        this.day= 1;
    } else if ((day>30 || day < 1)& (this.month
==4||this.month==6||this.month==9||this.month==11)){
        this.day= 1;
    } else if ((day>28 || day < 1)& (this.month ==2)) {
        this.day= 1;
    }
    else {
        this.day=day;
    }
}

public void setMonth(int month){
    //El set de month tambien corrige los rangos asi como que dependiendo del
    valor de month tambien cambia el valor
    //de Monthname
    if(month>12||month<1){
        this.month=1;
    } else {
        if (month==1){
            this.month=1;
            this.monthName="Enero";
        } else if (month==2){
            this.month=2;

```



```

        this.monthName="Febrero";
    } else if (month==3){
        this.month=3;
        this.monthName="Marzo";
    } else if (month==4){
        this.month=4;
        this.monthName="Abril";
    } else if (month==5){
        this.month=5;
        this.monthName="Mayo";
    } else if (month==6){
        this.month=6;
        this.monthName="Junio";
    } else if (month==7){
        this.month=7;
        this.monthName="Julio";
    } else if (month==8){
        this.month=8;
        this.monthName="Agosto";
    } else if (month==9){
        this.month=9;
        this.monthName="Septiembre";
    } else if (month==10){
        this.month=10;
        this.monthName="Octubre";
    } else if (month==11){
        this.month=11;
        this.monthName="Noviembre";
    } else if (month==12){
        this.month=12;
        this.monthName="Diciembre";
    }
}
}

public void setYear(int year){
    //Este set year unicamente corrige la salida de rangos a los valores
predeterminados
    //como los anteriores
    if(year>3000||year<1900){
        this.year=2017;
    } else {
        this.year=year;
    }
}
}

```

```

public void setFormat(int format){
    //Como los anteriores, corrige la salida de rangos y apunta al propio valor del
input
    if (format<0||format>2){
        this.format = 0;
    }else {
        this.format = format;
    }
}

```

//Getters

// Estos getters te consiguen los valores ya establecidos

```

public int getDay(){return day;}
public int getMonth(){return month;}
public int getYear(){return year;}
public String getMonthName(){ return monthName;}
public int getFormat(){return format;}

```

//Metodos

// Dependiendo del resultado de format se regresara un String con los datos y el formato a como se desea que se imprima

```

public String toString(){
    String result;
    if (format== 0){
        String annoStr = String.valueOf(year);// transformamos el int del anno a string
        String ultimos2 = annoStr.substring(2,4);// La funcion substring sirve para
agarrar un rango de caracteres dentro del arreglo del string
        String diaM10;
        if (this.day<10){
            diaM10 = "0"+day; // Para que el formato quede como se pide para los que
no tienen 2 caracteres se suma un 0 string al principio
        } else {
            String diaStr = String.valueOf(day);
            diaM10 = diaStr;
        }
    }

    //se repite lo que hicimos con el formato de los valores monocaracteristicos
    String mesM10;
    if (this.month<10){
        mesM10 = "0"+month;
    } else {
        String mesStr = String.valueOf(month);
        mesM10 = mesStr;
    }
}

```

```

        result = diaM10+"/"+mesM10+"/"+ultimos2;//Se imprime como se pide
        return result;
    }else if(format== 1){
        String primerosTres =monthName.substring(0,3);//usamos substring de nuevo
        para seleccionar los primeros 3 caracteres de monthname
        result=day+"-"+primerosTres+"-"+year;
        return result;
    }else if(format== 2){
        result = day + " de " + monthName+ " de " +year; //esto unicamente creamos un
        string dedicado para imprimir
        return result;
    }
    return null;
}

```

```

    public boolean equals(Date obj2){//se comparan los datos encapsulados con los
    datos del objeto, hecho eso se llega a un veredicto
        boolean result;
        if (this.day==obj2.day && this.month==obj2.month && this.year==obj2.year){
            System.out.println("verdadero");
            return true;
        }else {
            System.out.println("falso");
            return false;
        }
    }
}

```

```

    public Date clone(Date obj1) {
        Date objeto = new Date();//Se crea un objeto provisional

        //se le asignan los valores del objeto copiado al objeto provisional
        objeto.day = obj1.day;
        objeto.month = obj1.month;
        objeto.year = obj1.year;
        return objeto;//se regresa la informacion del objeto provisional para darsela al
        objeto que recibira la clonacion
    }

    public void next(){// Aqui dependiendo del rango de valores del dia y el mes
    decidimos si sumar un dia, ajustar un dia y sumar un mes o
        //ajustar dia, mes y sumar anno

        //Es exactamente igual para los meses que tienen 30,31 y 28 dias

```

```

        if (getDay() == 31 && (getMonth() == 1 || getMonth() == 3 ||
getMonth() == 5 || getMonth() == 7 || getMonth() == 8 || getMonth() == 10)) {
            setDay(1);
            setMonth(getMonth() + 1);
        } else if (getDay() < 31 && ((getMonth() == 1 || getMonth() == 3 ||
getMonth() == 5 || getMonth() == 7 || getMonth() == 8 || getMonth() == 10))) {
            setDay(getDay() + 1);
        } else if (getDay() == 31 && getMonth() == 12) {
            setDay(1);
            setMonth(1);
            setYear(getYear() + 1);
        }

        if (getDay() < 30 && (getMonth() == 4 || getMonth() == 6 || getMonth()
== 9 || getMonth() == 11)) {
            setDay(getDay() + 1);
        } else if (getDay() >= 30 && (getMonth() == 4 || getMonth() == 6 || getMonth()
== 9 || getMonth() == 11)) {
            setDay(1);
            setMonth(getMonth() + 1);
        }

        if (getDay() < 28 && (getMonth() == 2)) {
            setDay(getDay() + 1);
        } else if (getDay() >= 28 && (getMonth() == 2)) {
            setDay(1);
            setMonth(getMonth() + 1);
        }

    }

}

```