

Parte 4 - POO | Ignacio CAVALLO

Clase 22 | 28/05

Conceptos Aprendidos

En Clase:

- Herencia
- Polifrmosmo
- El uso de *extends*

Fuera de clase:

- Creación de métodos con argumentos de la misma Clase.
- Como interactuan las clases Padres con sus hijos.
- Buenas practicas en la implemetnación de Constructores.
- Algo más sobre Recursión.
- Una vez terminado de programar, iterar removiendo parámetros inecesarios.
- Representación String de una clase, *String toString()*.
- Hacer links dentro de un documento Markdown, por ejemplo [este enlace al primer ejercicio](#).

Primera Parte

En clase se planteó un ejercicio que consistía en una Cuenta bancaria que permitiera giros,transferencias y depósitos. Los atributos:

```
private String nombre;  
private String nCuenta;  
private double saldo;
```

Las restricciones:

1. Si quiere hacer giros, el monto del giro no puede ser mayor al saldo.
2. Si se desea hacer una transferencia, la cuenta de origen debe tener fondos.

Los giros y depositos eran algo sencillo de abordar, resolviendo el primero con un condicional:

```
public void giro(double montoGiro) {  
    if (this.saldo >= montoGiro) {  
        this.saldo -= montoGiro;  
    } else {....
```

El **DESAFÍO** radicaba en la segunda restricción. Ya que debía relacionar dos instancias del objeto. Hasta ahora, siempre interactuábamos con instancias, no con la Clase en sí. Por eso tuve que idear como referencia a una futura instancia sin que esta estuviese creada. Luego de varios intentos concluí que lo más inteligente, era *utilizar como argumento la misma clase*. Para claficiar aquí esta el código:

```
public void transferencia(double montoTrans, Cuenta emisor) {
    if (this.saldo >= montoTrans) {
        this.saldo -= montoTrans;
        emisor.deposito(montoTrans);
        System.out.println(this.nombre + " transfirió a " + emisor.getNombre()
+
        " un monto de: $" + montoTrans);
    } else {.....
```

Básicamente, el método pide como argumento un objeto de tipo **Cuenta**, y con ella podemos relacionar al emisor con el remitente a traves de:

```
receptor = new Cuenta();
receptor.deposito()
```

El ejercicio completo está en esta [sección](#)

Segunda Parte

El segundo ejercicio fue la creación de una clase Libro con los siguientes argumentos:

```
String nombre;
String autor;
int numEjemplares;
int numPrestados;
```

Las restricciones:

1. En función a la cantidad de ejemplares que posee la biblioteca, el numero de libros prestados no puede superior al total de libros en existencias.

Para sobrellevar esta condición decidí crear un método boolean se puedePrestar(), que devuelve verdadero si hay existencias. Este me permite implementar la función requerida de prestar, la cual si puedePrestar es verdadero, entonces prestar el libro y aumentar en uno la cantidad de libros.

La función devolver simplemente resta en uno la existencia de libros prestados. Quedaría así:

```

public boolean sePuedeprestar(){
    return this.numPrestados<this.numEjemplares;
}

public int ejemplaresRemanentes(){
    return getNumEjemplares() - getNumPrestados();
}

public void prestar(){
    if(sePuedeprestar()){
        System.out.println("Tenemos ejemplares de " + getNombre());
        System.out.println("    Stock Remanente: " + ejemplaresRemanentes());
        System.out.println();
        this.numPrestados += 1;
    }
}

```

El ejercicio completo está [aquí](#)

Cuenta Bancaria

```

package com.nacho;

public class Cuenta<getSaldo> {
    private String nombre;
    private String nCuenta;
    private double saldo;

    public Cuenta(String nombre, String nCuenta, double saldo) {
        this.nombre = nombre;
        this.nCuenta = nCuenta;
        this.saldo = saldo > 0 ? saldo : 0;
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getnCuenta() {
        return this.nCuenta;
    }

    public void setnCuenta(String nCuenta) {
        this.nCuenta = nCuenta;
    }

    public double getSaldo() {

```

```

        return this.saldo;
    }

    public void setSaldo(double saldo) {
        this.saldo = saldo;
    }

    public void deposito(double montoDeposito) {
        this.saldo += montoDeposito;
    }

    public void giro(double montoGiro) {
        if (this.saldo >= montoGiro) {
            this.saldo -= montoGiro;
        } else {
            System.out.println("No hay platica, estás peor que las pitucas sin
lucas!");
            System.out.println("Su saldo de $" + getSaldo() + " no es suficiente
para completar su solicitud");
        }
    }

    public void transferencia(double montoTrans, Cuenta receptor) {
        if (this.saldo >= montoTrans) {
            this.saldo -= montoTrans;
            receptor.deposito(montoTrans);
            System.out.println(this.nombre + " transfirió a " +
receptor.getNombre() +
                " un monto de: $" + montoTrans);
        } else {
            System.out.println();
            System.out.println("**** AVISO ****");
            System.out.println("No hay platica, estás peor que las pitucas sin
lucas");
            System.out.println("Su saldo de $" + getSaldo() + "no es suficiente
para transferir su solicitud");
            System.out.println();
        }
    }

    @Override
    public java.lang.String toString() {
        return "Cuenta nº " + this.nCuenta + " de " + getNombre() +
            " tiene un saldo de: $" + saldo;
    }
}
/* ----- RESULTADO ----- */

```

Cuenta de Raymundo

Cuenta nº 001 de Raymundo Rodriguez tiene un saldo de: \$500.0

Cuenta de Fulgencio

Cuenta nº 002 de Fulgencio Fernandez tiene un saldo de: \$1200.0

Transferencia

Fulgencio -200 ==> Raymundo +200

Fulgencio Fernandez transfirió a Raymundo Rodriguez un monto de: \$200.0

Cuenta nº 001 de Raymundo Rodriguez tiene un saldo de: \$700.0

Cuenta nº 002 de Fulgencio Fernandez tiene un saldo de: \$1000.0

Intento Transferencia NO VALIDA

*** AVISO ***

No hay plastica, estás peor que las pitucas sin lucas

Su saldo de \$700.0 no es suficiente para transferir su solicitud

Libro

```
package com.nacho;

public class Libro {
    String nombre;
    String autor;
    int numEjemplares;
    int numPrestados;

    public Libro(){ }

    public Libro(String nombre, String autor, int numEjemplares, int
numPrestados){
        this.nombre = nombre;
        this.autor = autor;
        this.numEjemplares = numEjemplares;
        this.numPrestados = numPrestados>this.numEjemplares? numEjemplares:
numPrestados ;
    }

    public String getAutor() {
        return autor;
    }

    public void setAutor(String autor) {
        this.autor = autor;
    }

    public String getNombre() {
        return nombre;
    }
}
```

```

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public int getNumEjemplares() {

        return numEjemplares;
    }

    public void setNumEjemplares(int numEjemplares) {
        this.numEjemplares = numEjemplares;
    }

    public int getNumPrestados() {
        return numPrestados;
    }

    public void setNumPrestados(int numPrestados) {
        this.numPrestados = numPrestados;
    }

    public boolean sePuedeprestar(){
        return this.numPrestados<this.numEjemplares;
    }

    public int ejemplaresRemanentes(){
        return getNumEjemplares() - getNumPrestados();
    }

    public void prestar(){
        if(sePuedeprestar()){
            System.out.println("Tenemos ejemplares de " + getNombre());
            System.out.println("    Stock Remanente: " + ejemplaresRemanentes());
            System.out.println();
            this.numPrestados += 1;
        }
        else{
            System.out.println("No tenemos más ejemplares de " + getNombre() );
            System.out.println();
        }
    }

    public void devolver(){
        this.numPrestados-=1;
        System.out.println("Se devolvió un ejemplar de " + getNombre());
        System.out.println();
    }

    @Override
    public String toString() {
        return " El libro " + getNombre() + " del autor " + getAutor() +
            " posee un total de " + getNumEjemplares() + " ejemplares con un
stock disponible a préstamo de "
            + ejemplaresRemanentes();
    }
}

```

```

/* ---- MAIN ---- */

    var aleph = new Libro("El Aleph","Jorge Luis Borges", 5, 1);
    System.out.println("Valor Inicial==>" + aleph);
    loopRecursivo(7, aleph);
    aleph.devolver();
    aleph.getNumPrestados();
    System.out.println();
    System.out.println(aleph);
    aleph.prestar();
    System.out.println(aleph);
    aleph.prestar();

}
public static void loopRecursivo(int cantPrestar, Libro libroAPrestar){
    if(libroAPrestar.ejemplaresRemanentes() == 0) {
        System.out.println("*****");
        System.out.println("No hay más stock");
        System.out.println("*****");
    }else{
        libroAPrestar.prestar();
        loopRecursivo(cantPrestar - 1, libroAPrestar);
    }
}

/* ----- RESULTADO ----- */
Valor Inicial==> El libro El Aleph del autor Jorge Luis Borges posee un total de 5
ejemplares con un stock disponible a préstamo de 4
Tenemos ejemplares de El Aleph
    Stock Remanente: 4

Tenemos ejemplares de El Aleph
    Stock Remanente: 3

Tenemos ejemplares de El Aleph
    Stock Remanente: 2

Tenemos ejemplares de El Aleph
    Stock Remanente: 1

*****
No hay más stock
*****
Se devolvió un ejemplar de El Aleph

El libro El Aleph del autor Jorge Luis Borges posee un total de 5 ejemplares con
un stock disponible a préstamo de 1
Tenemos ejemplares de El Aleph
    Stock Remanente: 1

El libro El Aleph del autor Jorge Luis Borges posee un total de 5 ejemplares con
un stock disponible a préstamo de 0

```

No tenemos más ejemplares de El Aleph