UNIVERSIDAD DE GUADALAJARA
SISTEMA DE UNIVERSIDAD VIRTUAL

UDGVIRTUAL

# Polimorfismo

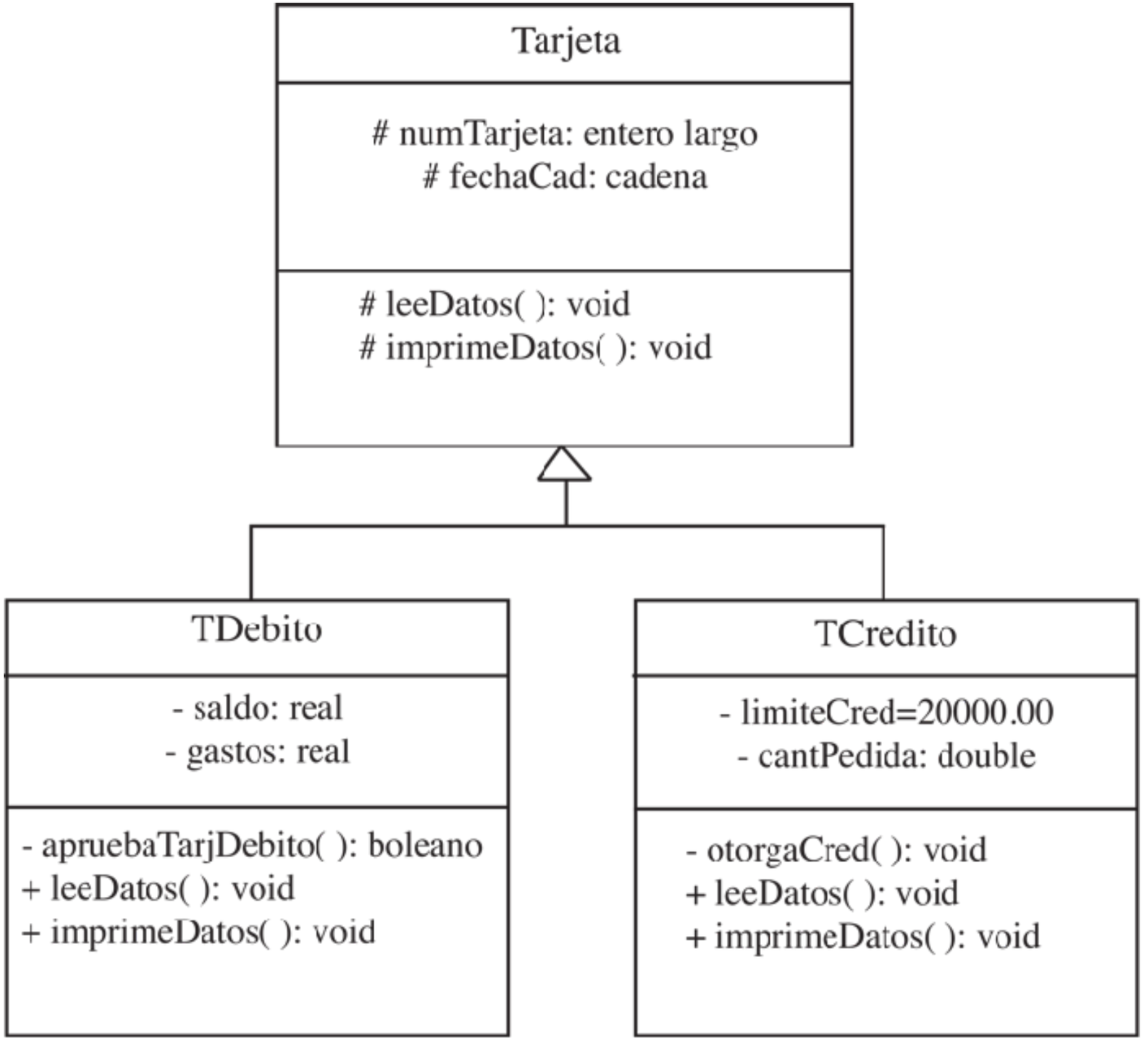
Polimorfismo

Agosto, 2018
Guadalajara, Jalisco. México

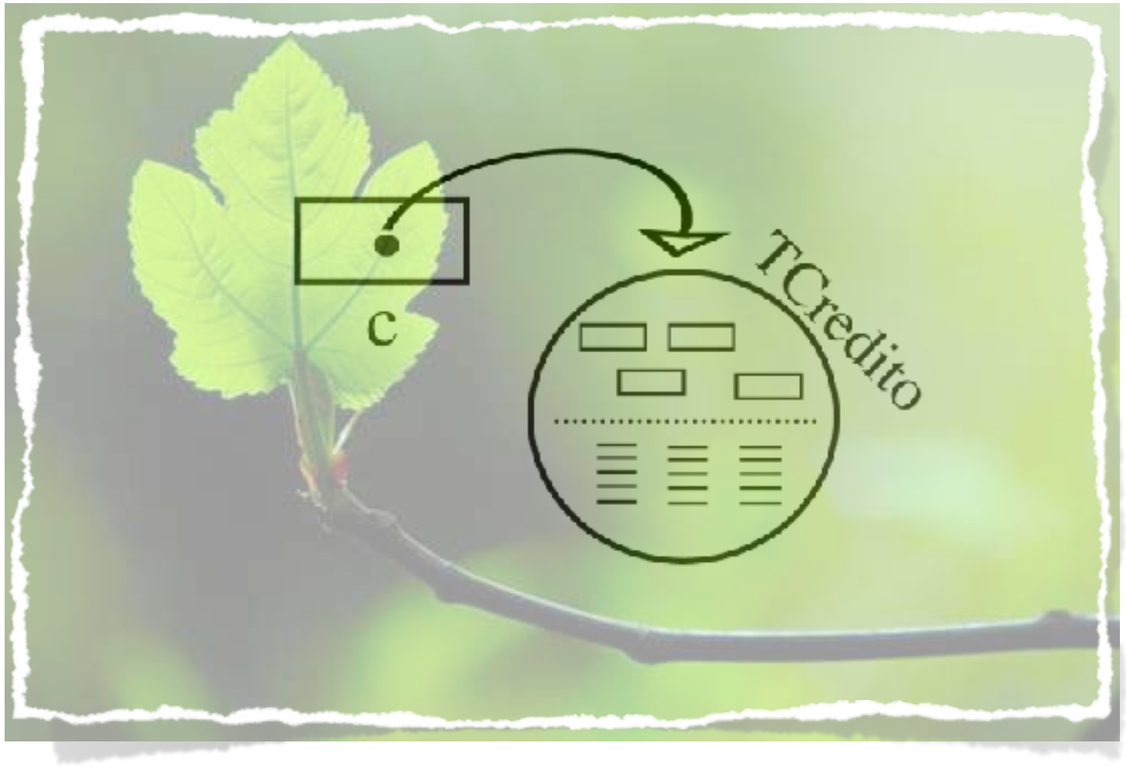
Realizado por
Prof. Sabrina Lizbeth Vega Maldonado

# Definición

es la forma de responder
más congruente al momento
de la solicitud y a la
naturaleza del objeto
implicado


Tarjeta
# numTarjeta: entero largo
# fechaCad: cadena
# leeDatos( ): void
# imprimeDatos( ): void
TDebito
- saldo: real
- gastos: real
- apruebaTarjDebito( ): boleano
+ leeDatos( ): void
+ imprimeDatos( ): void
TCredito
- limiteCred=20000.00
- cantPedida: double
- otorgaCred( ): void
+ leeDatos( ): void
+ imprimeDatos( ): void


d
TDebito


c
TCredito

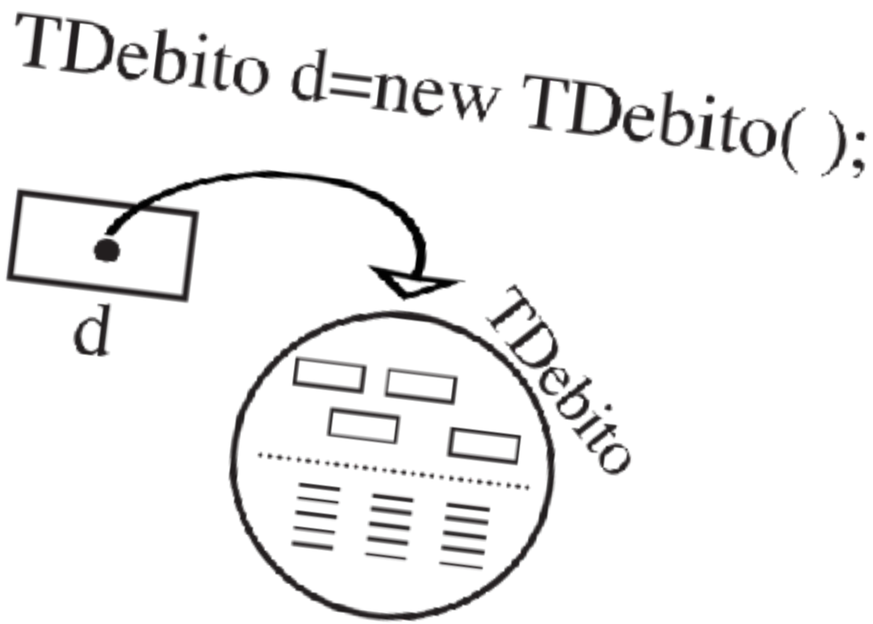
# Código en el lenguaje de Java - Herencia

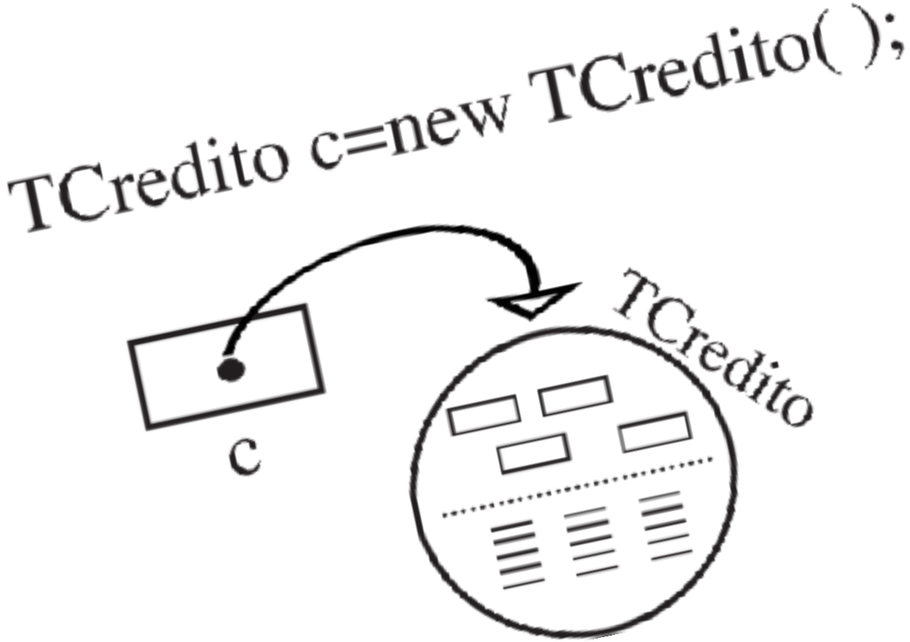
```
class Tarjeta {
    protected long numTarjeta;
    protected String fechaCad;

    protected void leeDatos(){
        System.out.println("\nALTA TARJETA BANCARIA");
        System.out.print("Dame tu numero de Tarjeta: ");
        numTarjeta=Leer.ent(); //archivo Leer.java (U2-a2 Herencia.pdf, diapositiva 20)
        System.out.print("Dame la fecha de caducidad de la Tarjeta: ");
        fechaCad=Leer.cad(); //archivo Leer.java (U2-a2 Herencia.pdf, diapositiva 20)
    }
    protected void imprimeDatos(){
        System.out.println("\nCONSULTA TARJETA BANCARIA");
        System.out.println("El numero de la Tarjeta es: "+numTarjeta);
        System.out.println("La fecha de caducidad de la Tarjeta es: "+fechaCad);
    }
}
```

```
class TDebito extends Tarjeta{
    private double saldo;
    private double gastos;
    private boolean apruebaTarjDebito(){
        if (gastos<=saldo)
            return true;
        else
            return false;
    }
    public void leeDatos(){
        super.leeDatos();
        System.out.print("Dame tu saldo actual: $");
        saldo=Leer.real(); //archivo Leer.java (U2-a2 Herencia.pdf, diapositiva 20)
        System.out.print("Escribe la cantidad a pagar: $");
        gastos=Leer.real(); //archivo Leer.java (U2-a2 Herencia.pdf, diapositiva 20)
    }
    public void imprimeDatos(){
        super.imprimeDatos();
        System.out.println("El saldo actual es: "+saldo);
        System.out.print("La cantidad a pagar es: $"+gastos);
        if(apruebaTarjDebito())        System.out.println("\nTarjeta Aprobada \n$"+gastos+" pagados\n");
        else       System.out.println("\nTarjeta No Aprobada\n");
    }
}
```

```
class TCredito extends Tarjeta{
    private final double LIMITE_CRED=20000.00; //constante
    private double cantPedida;
    private void otorgaCred(){
        if(cantPedida<LIMITE_CRED)
            System.out.println("\nCredito Aprobado: $"+cantPedida+"\n");
        else
            System.out.println("\nCredito No Aprobado\n");
    }
    public void leeDatos(){
        super.leeDatos();
        System.out.print("Cantidad de credito a pedir: $");
        cantPedida=Leer.real(); //archivo Leer.java (U2-a2 Herencia.pdf, diapositiva 20)

    }
    public void imprimeDatos(){
        super.imprimeDatos();
        System.out.println("Cantidad pedida de credito: $"+cantPedida);
        otorgaCred();
    }
}
```

```
public class HerenciaApp {
   public static void main(String[] args) {
      TDebito d=new TDebito(); //creación del objeto
      TCredito c=new TCredito();
      d.leeDatos(); //llamada método sobreescrito por cada objeto
      d.imprimeDatos();
      c.leeDatos();
      c.imprimeDatos();
   }
}
```



# Ligadura estática

El polimorfismo se puede aplicar pasando el mismo mensaje (método) a diferentes objetos, pero de esta manera se obliga desde tiempo de compilación a hacer las llamadas desde los diferentes objetos


t

# Implementando Polimorfismo con ligadura estática

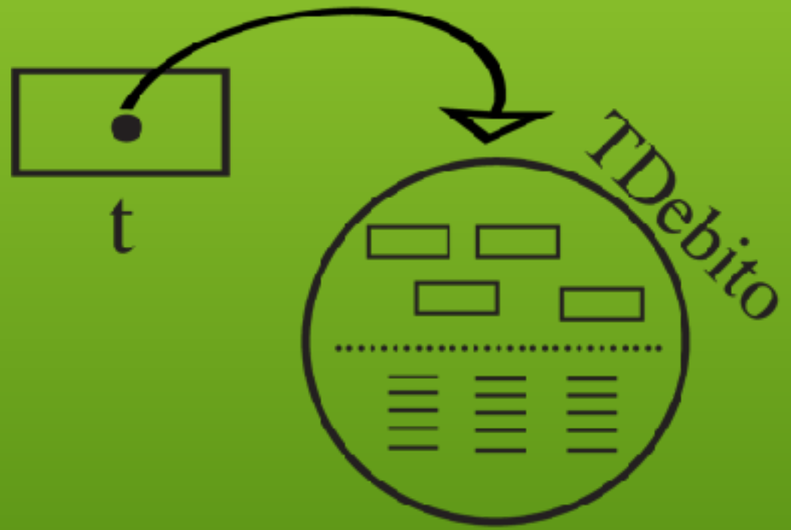
```
public class PolimorfismoApp {

    public static int menu(){
        int opc;
      System.out.println("MENU TARJETAS
      BANCARIAS");
      System.out.println("1. Tarjeta Debito");
      System.out.println("2. Tarjeta Credito");
      System.out.println("3. Salir");
      System.out.print("Elige una opcion: ");
      opc=Leer.ent(); /*archivo Leer.java (U2-a2
      Herencia.pdf, diapositiva 20) */
      return opc;
    }
```

```
    public static boolean ejecutaMenu(int opc){
        Tarjeta t; //declaración del objeto t
        switch(opc){
            case 1: t=new TDebito(); //creación con el mismo objeto para debito
                    t.leeDatos(); //llamada método polimórfico
                    t.imprimeDatos(); //llamada método polimórfico
            break;
            case 2: t=new TCredito(); //creación con el mismo objeto para credito
                    t.leeDatos(); //llamada método polimórfico
                    t.imprimeDatos(); //llamada método polimórfico
            break;
            case 3: System.out.println("Saliendo del programa...");
                    return false;
        }
        return true;
    }

    public static void main(String[] args) {
            while(ejecutaMenu(menu())); //se repite la llamada mientras sea true
    }
}
```

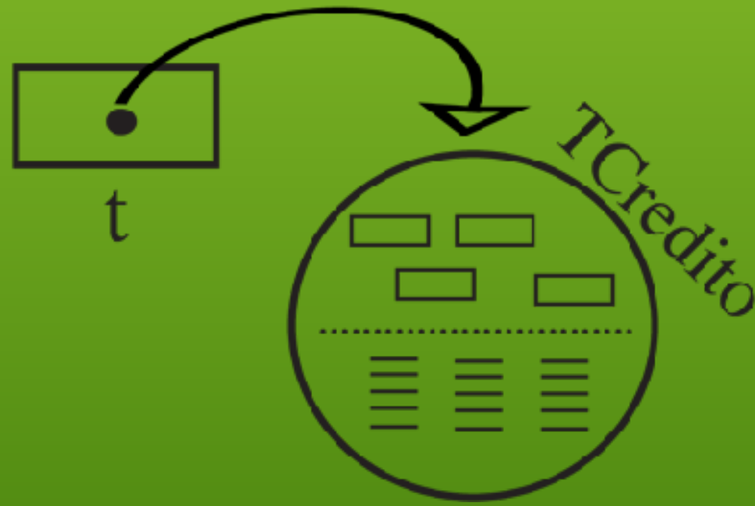
# En ejecución….

Tarjeta t;


t=new TDebito( );




t=new TCredito( );



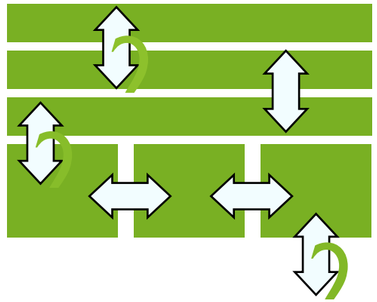
# Ligadura dinámica

ligadura dinámica, es que el tipo de objeto con el cual se hará la llamada del método no es necesario decidirlo en compilación sino hasta ejecución.

# Implementando Polimorfismo con ligadura dinámica

```
public class PolimorfismoArreglos {

   public static void main(String[] args) {
      Tarjeta t[ ]; //declaración local de un arreglo de objetos
      int tipo, cant=0; //variables locales
      System.out.print("Cuantas tarjetas deseas introducir: ");
      cant=Leer.ent( ); //archivo Leer.java (U2-a2 Herencia.pdf, diapositiva 20)
      t=new Tarjeta [cant]; //creación del arreglo de objetos
            for (int i=0; i<t.length; i++){
                  tipo=1+(int)(Math.random()*2);
                  switch(tipo){
                     case 1: t[i]=new TDebito(); //creación aleatoria del objeto TDebito
                              break;
                     case 2: t[i]=new TCredito(); //creación aleatoria del objeto TCredito
                              break;
                  }
            }
            System.out.println("ALTA TARJETAS");
            for (int i=0; i<cant; i++)
                  t[i].leeDatos( ); //llamada Polimórfica
            System.out.println("REPORTE TARJETAS");
            for (int i=0; i<cant; i++)
                  t[i].imprimeDatos( ); //llamada Polimórfica
   }
}
```

# Ejercicio para la actividad integradora 5



- Imprimir en pantalla el pago de impuestos para una persona Física. El programa leerá los datos de nombre, dirección, teléfono, rfc. Otro proceso será el cálculo de pago de impuestos que se calculará en base a un porcentaje de su salario percibido en un año si el salario es mayor a $50,000.00. (el porcentaje lo defines tú dentro del programa). También la impresión en pantalla de los datos y por último la impresión en pantalla del pago de impuestos.

- Imprimir en pantalla el pago de impuestos para una persona Moral. El programa leerá los datos de nombre, dirección, teléfono, rfc, sociedad mercantil, ingresos y gastos. Otro proceso será las deducciones de impuestos que la persona moral puede realizar las cuales se calcularan en base a los ingresos menos los gastos obtenidas en un periodo (lo defines tú dentro del programa). También la impresión en pantalla de los datos y por último la impresión en pantalla del pago de impuestos menos las deducciones desglosado (ingresos, deducciones, pago total).

- El programa debe preguntar a través de un menú si la persona es física o moral. También se debe considerar que al final del programa se pregunte si quiere salir del programa en caso de que la respuesta sea no el programa debe regresar al menú.