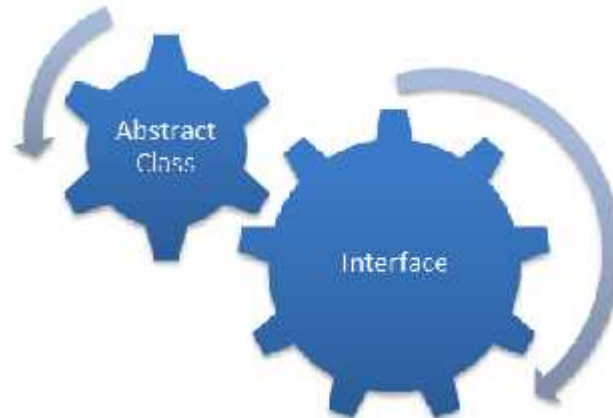


DIFERENCIA DE UNA CLASE ABSTRACTA Y UNA INTERFAZ



Las clases abstractas a diferencia de las interfaces, pueden tener constructores, implementaciones de método por defecto y solo se pueden heredar 1 sola vez de ellas 1 sola vez:

```
public abstract class Producto
{
    private int precio;
    public Producto(int precio)
    {
        this.precio = precio;
    }
    public int getPrecio()
    {
        return this.precio;
    }

    public abstract void getName();
}

public class Banana extends Producto
{
    public Banana()
    {
        super(200);
    }

    @Override
    public String getNamen()
    {
        return "banana";
    }
}
```

Aunque en java 8, se permiten las implementaciones por defecto, estas no pueden contener constructores.

En el caso de las interfaces, si se puede implementar de múltiples de ellas:

```
public class MiClase implements OnClickListener, OnTouchListener, Runnable
{
    @Override
    public void onClick(View view) {}

    @Override
    public void onTouch(View view) {}

    @Override
    public void run(){}
}
```

Esto permite emular la herencia múltiple:

```
MiClase miClase = new MiClase();

new Thread(miClase).run();
elementoVista.setOnClickListener(miClase);
elementoVista.setOnTouchListener(miClase);
```

Las interfaces se les conoce como un contrato. Esto porque te obligan a la implementación de sus métodos lo que le asegura a toda lógica de negocio que todo objeto que implemente de ella, tendrá el acceso a los métodos definidos en la misma.

Mientras que las clases abstractas son más utilizadas para objetos de tipo base. Es como la principal en la jerarquía en un conjunto de objetos que comparten el mismo padre.

-) En una clase abstracta pueden existir variables static, final o static final con cualquier modificador de acceso (public, private, protected o default). En una interfaz sólo puedes tener constantes (public static final).