



Haga lo que hicimos en aula

Si aún no lo ha abierto, abra el proyecto en *Empleado*.

Cree una clase llamada *SistemaInterno* que tenga una contraseña como atributo e implemente el método de *autenticar* de la siguiente manera:

```
public class SistemaInterno {  
  
    private int contraseña = 2222;  
  
    public void autenticar(Gerente g) {  
  
        boolean autentico = g.autenticar(this.contraseña);  
  
        if (autentico) {  
            System.out.println("Puede entrar al sistema");  
        } else {  
            System.out.println("No puede entrar al sistema");  
        }  
    }  
}
```

Cree una clase de prueba llamada *TestSistema* con el método *main* ya generado.

Cree una instancia de un *Gerente*, use el método *setContraseña* e ingrese la contraseña correcta, además, cree una instancia del *SistemaInterno* e intente autenticarse.

El siguiente código debe estar dentro del método *main*:

```
Gerente g = new Gerente ();  
g.setContraseña (2222);  
  
SistemaInterno si = new SistemaInterno();  
Si.autenticar(g);
```

Cree una clase de *Administrador* que debe heredar de la clase de *Empleado*. Date cuenta de que Eclipse ya nos da el método *getBonificacion*

Cree una clase abstracta *EmpleadoAutenticado* que herede de *Empleado*

Complete el código de *EmpleadoAutenticado* con el siguiente código:

```
public class EmpleadoAutenticado {  
  
    private int contraseña;  
  
    public void setContraseña(int contraseña) {  
        this.contraseña = contraseña;  
    }  
  
    public boolean autenticar(int contraseña) {  
        if (this.contraseña == contraseña) {  
            return true;  
        } else {  
            return false;  
        }  
    }  
}
```

```
}
```

Modifique el código de *Gerente* haciendo que herede de *EmpleadoAutenticado*.

```
public class Gerente extends EmpleadoAutenticado {  
    //resto de código  
}
```

En el *SistemaInterno*, cambie la referencia del método *autenticar* para que reciba un *EmpleadoAutenticado*. Vea continuación:

```
public class SistemaInterno {  
  
    private int contraseña = 2222;  
  
    public void autenticar(EmpleadoAutenticado ea) {  
  
        boolean autentico = ea.autenticar(this.contraseña);  
  
        if (autentico) {  
            System.out.println("Puede entrar al sistema");  
        } else {  
            System.out.println("No puede entrar al sistema");  
        }  
    }  
}
```

Tenga en cuenta que, si necesitamos que un cliente esté autenticado, tendremos un problema, ya

que la clase *Cliente* necesitaría heredar de *EmpleadoAutenticado* y esto daría como resultado que un cliente tuviera una bonificación ya que la clase *EmpleadoAutenticado* hereda de *Empleado*

Para solucionar esto, haremos uso de las Interfaces, transformando la antigua clase abstracta *EmpleadoAutenticado* en la interfaz *Autenticacion* como se muestra a continuación:

```
public abstract interface Autenticacion {  
  
    public abstract void setContraseña(int contraseña);  
    public abstract boolean autenticar(int contraseña);  
}
```

En la clase *Cliente*, use la palabra reservada `implements` para usar la interfaz creada, vea a continuación:

```
public class Cliente implements Autenticacion {  
  
}
```

Complete el código para la clase *Cliente*, implementando los métodos que faltan:

```
public class Cliente implements Autenticacion {  
  
    private int contraseña;  
  
    @Override  
    public void setContraseña(int contraseña) {  
        this.contraseña = contraseña;  
    }  
}
```

```

@Override

public boolean autenticar(int contraseña) {

    if (this.contraseña == contraseña) {

        return true;

    } else {

        return false;

    }

}

}

```

Haga que las clases *Gerente* *y **Administrador* implementen la interfaz * *Autenticacion*, *no olvide implementar los métodos y declarar la atributo *contraseña* privado!

```

public class Gerente extends Empleado implements Autenticacion {

    private int contraseña;

    public int getContraseña() {

        // implementación da bonificación de gerente omitida;

    }

    @Override

    public void setContraseña(int contraseña) {

        this.contraseña = contraseña;

    }

    @Override

    public boolean autenticar(int contraseña) {

        if (this.contraseña == contraseña) {

```

```

        return true;
    } else {
        return false;
    }
}

}

```

```

public class Administrador extends Empleado implements Autenticacion {

    private int contraseña;

    public int getContraseña() {
        // implementación da bonificación de Administrador omitida;
    }

    @Override
    public void setContraseña(int contraseña) {
        this.contraseña = contraseña;
    }

    @Override
    public boolean autenticar(int contraseña) {
        if (this.contraseña == contraseña) {
            return true;
        } else {
            return false;
        }
    }
}

```

```
}
```

En *SistemaInterno* modifique el método de autenticar para que ahora reciba una *Autenticacion*

```
public class SistemaInterno {  
  
    private int contraseña = 2222;  
  
    public void autenticar(Autenticacion au) {  
  
        boolean autentico = au.autenticar(this.contraseña);  
  
        if (autentico) {  
            System.out.println("Puede entrar al sistema");  
        } else {  
            System.out.println("No puede entrar al sistema");  
        }  
    }  
}
```

Tenga en cuenta que ahora podemos usar la *Autenticacion* como un tipo, vea el ejemplo a continuación:

```
Autenticacion referencia = new Gerente();  
  
Autenticacion referencia = new Cliente();
```

```
Autenticacion referencia = new Administrador();
```

¡Pruebe todo y vea si todo está bien!

Opinión del instructor

Caso tengas alguna dificultad, ¡no dudes en utilizar nuestro foro!

