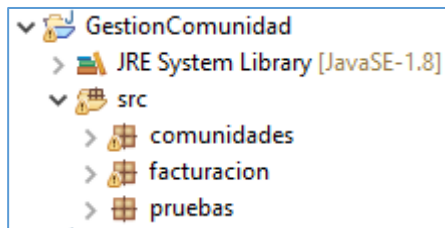


ALUMNO/A:

El examen consiste en realizar un proyecto que gestione una **comunidad de vecinos**. En concreto, habrá que controlar:

- **Los pisos** que componen la comunidad.
- **Los proveedores** que contrata la comunidad.

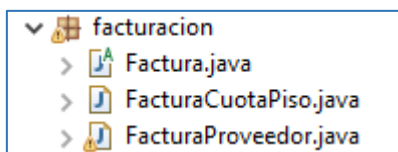
Crea un nuevo proyecto java llamado **GestionComunidad** que contenga los siguientes paquetes:



➤ **Paquete Facturación:**

En este paquete se encuentran las clases relacionadas con las facturas de la comunidad. Hay dos clases de facturas:

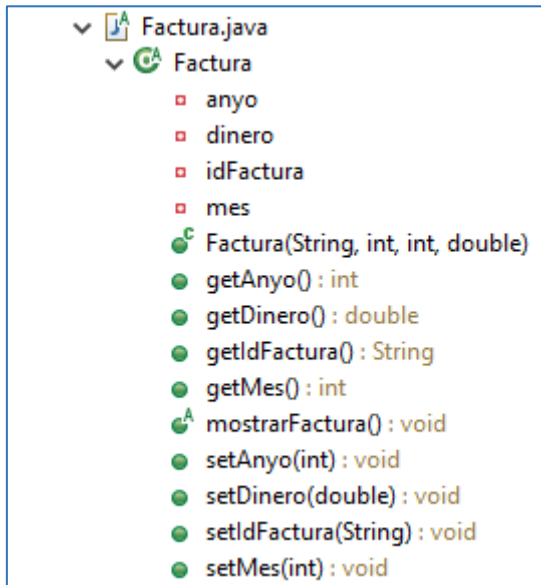
- Las **cuotas de los pisos**, que se cobran mensualmente a los vecinos y representan los ingresos de la comunidad.
- Las **facturas de los proveedores**, que se pagan por los trabajos realizados para la comunidad de vecinos (representan los gastos de la comunidad).



ALUMNO/A:

**Clase Factura:** (1 PUNTO)

Es una **clase abstracta** que almacena los datos de la factura. Se estructura de la siguiente manera:



- **Atributos** (todos son privados)

**idFactura:** Identificador de la factura (**String**)

**anyo:** Año de emisión de la factura (**int**)

**mes:** Mes de emisión de la factura (**int**)

**dinero:** Importe de la factura (**double**)

- **Constructor:**

```
public Factura(String idFactura, int anyo, int mes, double dinero)
```

- **Métodos:**

Métodos getter y setter para cada atributo

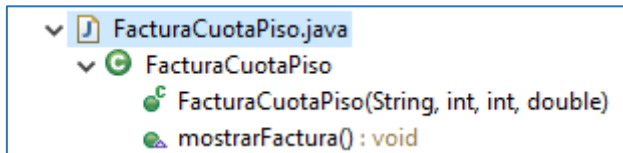
mostrarFactura() es un método abstracto no implementado

---

ALUMNO/A:

**Clase FacturaCuotaPiso** (hereda de la clase Factura): (1 PUNTO)

Representa la factura mensual (cuota piso) que tienen que pagar los propietarios de los pisos.



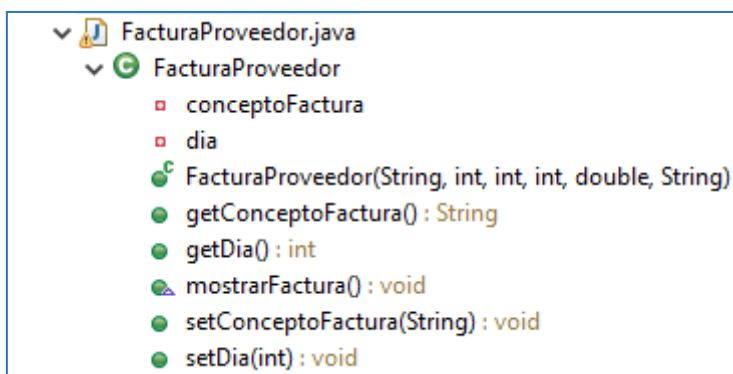
- **Atributos** (los heredados del padre)
- **Constructor:**  
`public FacturaCuotaPiso (String idFactura, int anyo, int mes, double dinero)`
- **Métodos:**  
`mostrarFactura() // muestra los datos de una cuota mensual del piso`

Id. factura: cuot1	año: 2019	mes: 3	importe: 20.3
--------------------	-----------	--------	---------------

**Nota:** Para imprimir así las líneas puedes utilizar el tabulador (símbolo \t)

**FacturaProveedor** (hereda de la clase Factura): (1 PUNTO)

Corresponde a una factura de un proveedor.



- **Atributos** (a parte de los atributos heredados de la clase Factura)

**concepto:** Trabajo realizado por el proveedor (**String**)

**dia:** Día de emisión de la factura (**int**)

ALUMNO/A:

- **Constructor:**

```
public FacturaProveedor(String idFactura, int anyo, int mes, int dia, double dinero, String conceptoFactura)
```

- **Métodos:**

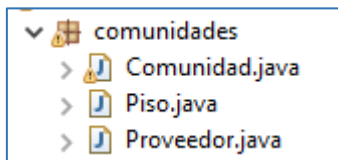
Métodos getter y setter para cada atributo

mostrarFactura() // muestra datos de la factura de un proveedor

Id. factura: fact1	Concepto: iluminacion escalera	dia: 20/4/2019	importe: 10.3
--------------------	--------------------------------	----------------	---------------

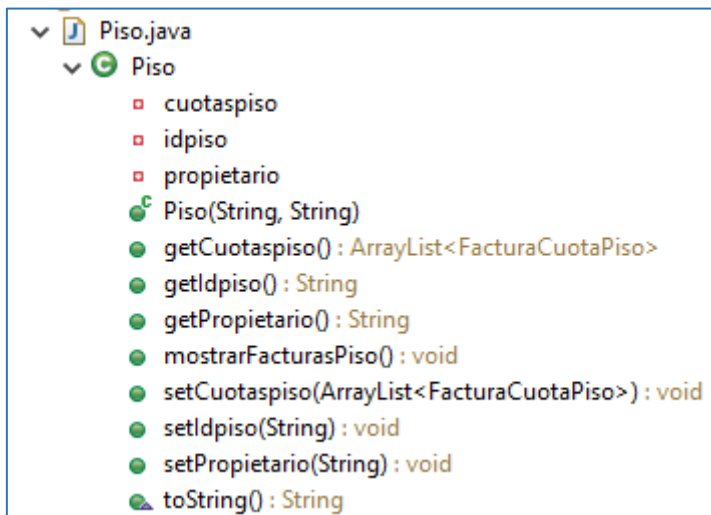
➤ **Paquete Comunidades:**

En este paquete se encuentran las clases relacionadas con la creación de una comunidad de vecinos (Piso, Proveedor y Comunidad).



**Clase Piso:** (1,5 PUNTOS)

Guarda los datos del piso (incluido las cuotas mensuales pagadas por el mismo)



**Atributos**

**idpiso:** Identificador del piso (**String**)

**propietario:** Vecino propietario del piso (**String**)

**cuotasPiso:** Array con las cuotas mensuales pagadas por el piso (**ArrayList**)

ALUMNO/A:

- **Constructor:**

```
public Piso(String idpiso, String propietario)
// Inicialmente el array con las cuotas del piso está vacío (no se han pagado
cuotas aún).
```

- **Métodos:**

Métodos getter y setter para cada atributo

toString() // muestra los datos del piso

```
Piso [idpiso=piso1, propietario=propietario1]
```

mostrarFacturasPiso() // muestra todas las cuotas pagadas por el piso

CUOTAS PAGADAS EN EL PISO CON ID piso1			
Id. factura: cuot1	año: 2019	mes: 3	importe: 20.3
Id. factura: cuot2	año: 2019	mes: 4	importe: 20.3

### **Clase Proveedor:** (1,5 PUNTOS)

Guarda la información de cada proveedor contratado por la comunidad de vecinos (incluidas todas las facturas emitidas por dicho proveedor)

```
Proveedor.java
└─ Proveedor
   ├── cif
   ├── direccion
   ├── facturasproveedor
   ├── nombre
   ├── Proveedor(String, String, String)
   ├── getCif() : String
   ├── getDireccion() : String
   ├── getFacturasproveedores() : ArrayList<FacturaProveedor>
   ├── getNombre() : String
   ├── mostrarFacturasProveedor() : void
   ├── setCif(String) : void
   ├── setDireccion(String) : void
   ├── setFacturasproveedores(ArrayList<FacturaProveedor>) : void
   ├── setNombre(String) : void
   └─ toString() : String
```

ALUMNO/A:

### Atributos

**cif:** Identificador de la empresa (**String**)

**nombre:** Nombre del proveedor (**String**)

**dirección:** Dirección del proveedor (**String**)

**facturasProveedor:** Array con las facturas del proveedor (**ArrayList**)

- **Constructor:**

```
public Proveedor(String cif, String nombre, String direccion)
// Inicialmente el array con las facturas del proveedor está vacío (no se han
emitido facturas aún).
```

- **Métodos:**

Métodos getter y setter para cada atributo

toString() // muestra datos del proveedor

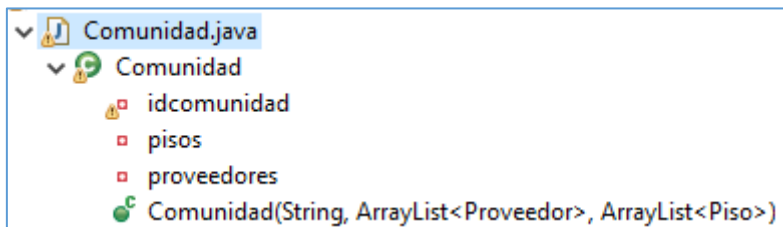
```
Proveedor [cif=cif1, nombre=nombre1, direccion=direccion1]
```

mostrarFacturasProveedor() // muestra todas las facturas emitidas por el  
proveedor a la comunidad de vecinos.

FACTURAS DEL PROVEEDOR: cif1			
Id. factura: fact1	Concepto: iluminacion escalera	dia: 20/4/2019	importe: 10.3
Id. factura: fact2	Concepto: canaletas bloque	dia: 20/4/2019	importe: 10.3

### **Clase Comunidad:** (1 PUNTO)

Almacena la información de los proveedores y pisos de la comunidad



### **Atributos**

**idcomunidad:** Identificador de la comunidad (**String**)

**pisos:** Array con los pisos de la comunidad de vecinos (**ArrayList**)

**proveedores:** Array con los proveedores de la comunidad de vecinos (**ArrayList**)

ALUMNO/A:

- **Constructor:**

```
public Comunidad(String idcomunidad, ArrayList<Proveedor>
proveedores, ArrayList<Piso> pisos)
```

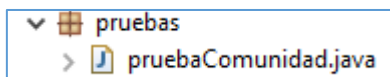
- **Métodos:**

Métodos getter y setter para cada atributo

---

➤ **Paquete pruebas:**

En este paquete se debe implementar el programa principal.



**Programa principal:** PruebaComunidad (3 PUNTOS)

Haz un programa principal que cree una comunidad de vecinos con tres pisos y tres proveedores. A continuación, implementa el siguiente menú:

```
HA SIDO CREADA UNA COMUNIDAD CON 3 PISOS Y 3 PROVEEDORES...
Elige una opción:
1- Añadir una factura de un proveedor de la comunidad:
2- Añadir cuota piso de la comunidad:
3- Mostrar Ingresos/gastos de un año-mes:
4- Mostrar Pisos que no han pagado la cuota de un año-mes:
5- Salir:
```

**Opciones:**

- 1- Pide el **cif del proveedor** y **los datos de la factura** y añádela al proveedor correspondiente.
- 2- Pide **el identificador del piso** y **los datos de la cuota** y añádela al piso correspondiente.
- 3- Pide **el mes, el año**, y muestra **los ingresos** (suma de todas las cuotas pagadas por los pisos en ese mes), **los gastos** (suma de todas las facturas de los proveedores en ese mes) contabilizados y el **total contable** (ingresos menos gastos)

```
CONTABILIDAD DEL MES 4 DEL AÑO 2019

INGRESOS: 30.3
GASTOS: 20.6
-----
TOTAL: 9.7
```

- 4- Pide **el mes, el año**, y muestra los pisos que **no han pagado aún la cuota de ese mes**

```
PISOS QUE NO HAN PAGADO LA CUOTA DEL MES 4 DEL AÑO 2019
Piso [idpiso=piso2, propietario=propietario2]
```

ALUMNO/A:

**Nota:** Debido a lo limitado del tiempo del examen, elige entre implementar la opción 3 o la opción 4, **no implementes las dos opciones!!**

**Los criterios de calificación serán:**

- **Clase Factura:** (1 PUNTO)
- **Clase FacturaCuotaPiso** (1 PUNTO)
- **Clase FacturaProveedor** (1 PUNTO)
- **Clase Piso:** (1,5 PUNTOS)
- **Clase Proveedor:** (1,5 PUNTOS)
- **Clase Comunidad:** (1 PUNTO)
- **Programa principal:** PruebaComunidad (3 PUNTOS)