Se trata de realizar una aplicación para gestionar los empleados de una empresa. Tenemos la siguiente jerarquía de clases (se incluirá la Clase Cuenta del ejercicio 2):

1. Clase Empleados: de esta clase no se podrán crear objetos (*clase abstracta*).
2. * Atributos, que sólo pueden verse desde esta clase:
     + nombre, de tipo String.
     + dni, de tipo String.
     + salario, de tipo double.
     + cuentaEmpleado, de tipo Cuenta, la clase Cuenta del ejercicio 2.
   * Métodos:
     + Constructor con parámetro. Inicializará los atributos y creará la cuenta del empleado donde el número de cuenta será el dni; el saldo de la cuenta será el valor de su salario y el tipo de interés, 5.
     + getter y setter.
     + El método abstracto extras que devuelve un double y no recibe parámetros. Calculará los extras del salario del empleado.
     + El método datosEmpleados que mostrará los datos del empleado, incluyendo los extras.
3. Clase Programador: clase hija de Empleado.
4. * Atributos:
     + horasExtras, de tipo int.
   * Métodos:
     + Constructor con parámetro.
     + getter y setter.
     + El método extras devolverá el producto de las horasExtras por 20.
     + El método datosEmpleados mostrará "Programador" y el resto de datos.
5. Clase Gerente: clase hija de Empleado.
6. * Atributos:
     + comision, de tipo double.
     + proyectos, de tipo int.
   * Métodos:
     + Constructor con parámetro.
     + getter y setter.
     + El método extras devolverá el producto de los atributos comision y proyectos.
     + El método datosEmpleados mostrará "Gerente" y el resto de datos.
7. Clase Empresa:
8. * Atributos:
     + nombreEmpresa, de tipo String.
     + cif, de tipo String.
     + numeroEmpleados, de tipo int.
     + cuentaEmpresa, de tipo Cuenta, igual que cuentaEmpleado.
     + empleados, un array de tipo Empleados.
   * Métodos:
     + Constructor con parámetro. Recibirá el nombre de la empresa, el cif y el saldo de la cuenta. Inicializará sus atributos a esos valores, creará un vector del tamaño 20. El atributo numeroEmpleados se inicializará al valor 0. Al construir el objeto cuentaEmpresa, el nombre del cliente será el nombre de la empresa; el número de cuenta, el cif; el tipo de interés del 10% y el saldo, el recibido como parámetro
     + getter y setter.
     + contratar recibirá como parámetro un objeto del tipo Empleado y, si se puede contratar, lo insertará en la posición que indique el atributo numeroEmpleados, incrementándolo. El método será del tipo boolean, devolviendo el resultado de la operación si se ha contratado o no.
     + devolverEmpleado recibirá como parámetro una posición y devolverá un objeto del tipo Empleado que se encuentre en esa posición del *array* o *null* si esa posición es errónea.
     + pagarNomina, de tipo boolean, que recibirá una posición del vector empleados y le hará una transferencia a la cuenta del empleado desde la cuenta de la empresa. El importe de la transferencia será la suma de su salario más los extras que le correspondan. Devolverá *true* o *false* si se ha hecho la transferencia o no.
     + listarEmpleados, de tipo *void* sin parámetros, mostrará los datos de todos los empleados contratados.
     + datosEmpresa, de tipo *void* sin parámetros, mostrará el nombre, el CIF y el saldo de la cuenta empresa.
9. Clase GestionarEmpresa: será la clase de la aplicación.
10. * Métodos:
      + El método main tendrá un objeto del tipo Empresa que se construirá con unos datos concretos (no se pedirán por teclado). A través de un menú se podrán realizar las siguientes operaciones:  
        1. *Ver/modificar los datos de la empresa*: mostrará el nombre y el CIF de la empresa junto con el saldo de la cuenta. Se podrán modificar los dos primeros.
        2. *Contratar a un empleado*.
        3. *Ver los datos de un empleado*: pedirá una posición del vector y mostrará los datos de eese empleado.
        4. *Ver los datos de todos los empleado*.
        5. *Pagar todas las nóminas*.
        6. *Ingresar en la cuenta de la empresa*: pedirá la cantidad por teclado y la ingresará en la cuenta de la empresa.
      + El método pedirDatosEmpleados que devolverá un objeto del tipo Empleado. Pedirá por teclado el tipo de empleado, gerente o programador, y todos su datos.
      + Todos los métodos que se crean convenientes para el diseño del programa.