Fundamentos del lenguaje Java

1. Creación de clases
   I. Atributos
   II. Constructores
   III. Getters y Setters
   IV. Métodos típicos heredados de Object: toString, equals, hashCode
2. Jerarquías de herencia
   I. Constructores en jerarquías de herencia
   II. Llamadas a través de super
3. Clases abstractas
4. Creación de interfaces
   I. Métodos con código en interfaces (deafult, static)
5. Polimorfismo (redefinición de métodos, @Override)
6. Manejo de colecciones: listas (List), conjuntos (Set), mapas (Map)
7. Gestión de excepciones

Prácticas

1. Creación de clases
   1. Definir la clase Persona con dos atributos, nombre y edad.
   2. Crear dos constructores, uno sin argumentos, otro con nombre y edad
   3. Crear getters y setters
   4. Redinir los métodos típicos
   5. Crear un par de objetos de tipo Persona y llamar a sus métodos
2. Jerarquías de herencia
   1. Crear la clase Empleado descenciente de Persona
   2. Añadir un atributo sueldo numérico
   3. Escribir constructores, getters y setters
   4. Redefinir el método toString para que llame al del ancestro (usar super)
   5. Crear un empleado y probar sus métodos
3. Clases abstractas
   1. Definir la clase abstracta Figura con un método abstracto, dibujar
   2. Definir varios descendientes
   3. Instanciar los descendientes y llamar a sus métodos
4. Interfaces
5. Definir la interfaz Figura con un método abstracto, dibujar, un método default y un método estático con trazas por consola
   1. Definir varias clases que implementen esa interfaz
   2. Instanciarlos y llamar a sus métodos
6. Polimorfismo: recogido en las prácticas sobre jerarquías de herencia e

interfaces

7. Manejo de colecciones
    1. Definir la clase Aficion
    2. Hacer que las personas puedan tener 0..N aficiones.
    3. Emplear primero una lista, después un conjunto
    4. Mostrar las aficiones por consola
    5. Definir una clase con un mapa con atributo que asocie una palabra con el número de veces que aparece en una frase
    6. Mostrar por consola las palabras y el número de veces que aparecen
8. Gestión de excepciones
    1. En cualquier práctica que use la clase Persona, modificarla para que si le damos un nombre null o vacío, se genere una excepción

Aplicación completa: Agenda de contactos