**PROGRAMACIÓN ORIENTADA A OBJETOS**

**Guía de Laboratorio**

**Objetivo: Arreglos de Objetos**

**CASO DE NEGOCIO 01: SALARIO DE EMPLEADOS**

Una empresa desea un sistema capaz de calcular el salario semanal de sus empleados de acuerdo con la **cantidad de horas trabajadas**, **el sueldo por hora** y tomando en cuenta los siguientes criterios:

* Si las horas trabajadas son más de 40, entonces el excedente se considera hora extra.
* Si las horas trabajadas están entre el rango de 41 a 45, entonces cada hora extra se paga doble.
* Si las horas trabajadas son más de 45, entonces cada hora extra se paga triple.

Se pide:

Diseñar la **clase Empleado**, con sus atributos (código de empleado, constructor con parámetros, métodos accesadores y modificadores, así mismo el método para **calcular salario**.

Diseñar la **clase GestionEmpleado**, con:

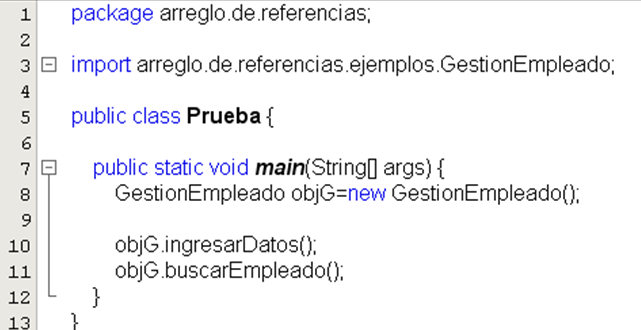
**Atributos:**

* Un arreglo capaz de almacenar 10 objetos de la clase Empleado y
* Una variable de tipo entero que contiene el total de objetos almacenados en el arreglo.

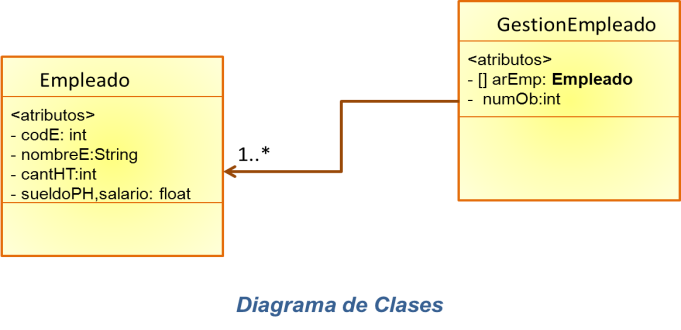
**Métodos:**

* Un constructor sin parámetros que permita crear el arreglo de objetos vacío.
* Un método para ingresar los datos del empleado al arreglo, hasta que el usuario decida y verifique que exista espacio disponible.
* Un método que permita buscar el empleado con el mayor salario y muestre todos sus datos. (considere que puede haber más de uno)

Diseñar la **clase Prueba**:







**CASO DE NEGOCIO 02: GESTION EN UNA GRANJA.**

Se desea gestionar la información de pollitos en una granja.

La clase **Pollo** tiene como variables de instancia: nombre, peso y tiempo en días.

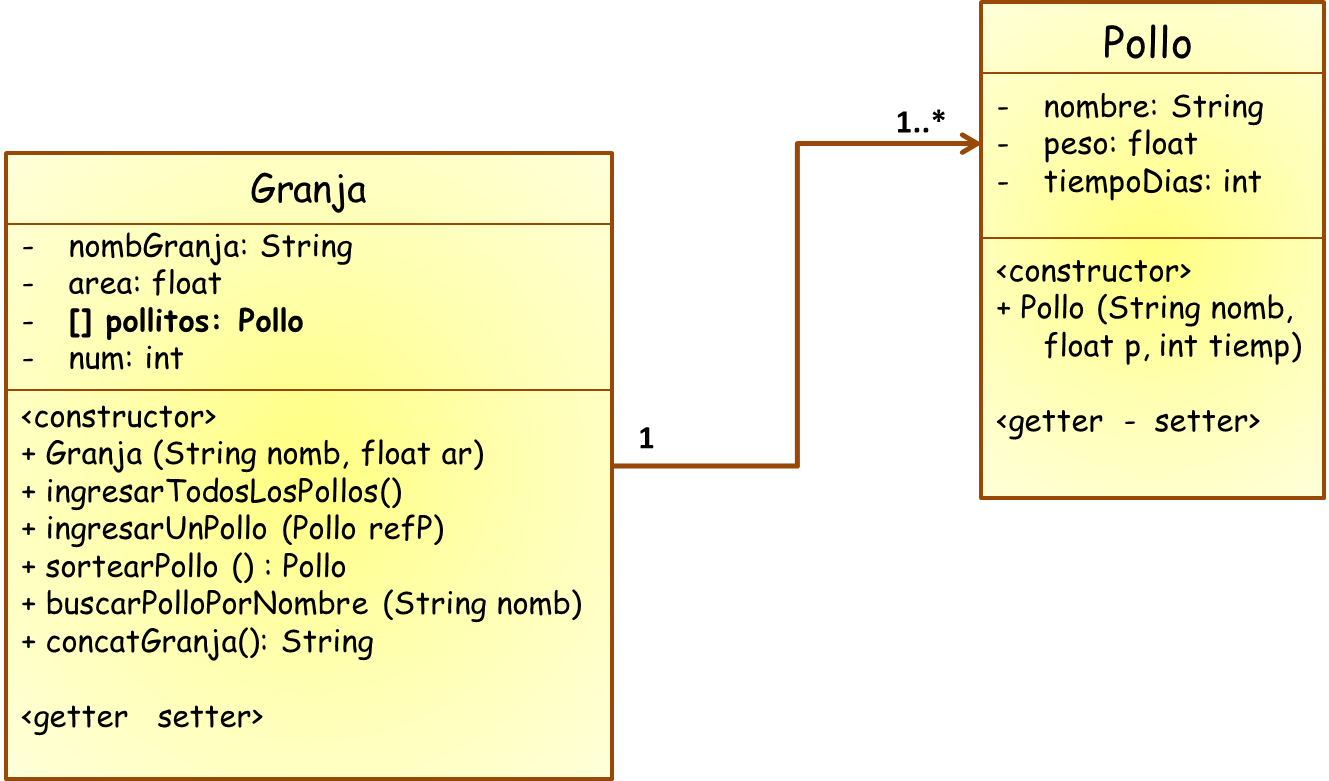
Un constructor con parámetros y sus métodos getter y setter.

La clase **Granja** tiene como variables de instancia: nombre de la granja, área, un arreglo de referencias a pollitos y el número de pollitos.

Un constructor con parámetros, nombre de la granja y área. Aquí se crea el arreglo de pollitos de tamaña 10.

Y los siguientes métodos:

* Ingresar al arreglo todos los pollitos, los datos se ingresan por pantalla.
* Ingresar un pollito, el cual es pasado como parámetro.
* Sortear un pollo aleatoriamente, el cual es retornado.
* Realizar la búsqueda de un pollo por su nombre.
* Concatenar la información de toda la granja.



Diseñe la clase **UnDiaEnLaGranja**, que permita realizar y probar todos los métodos solicitados.

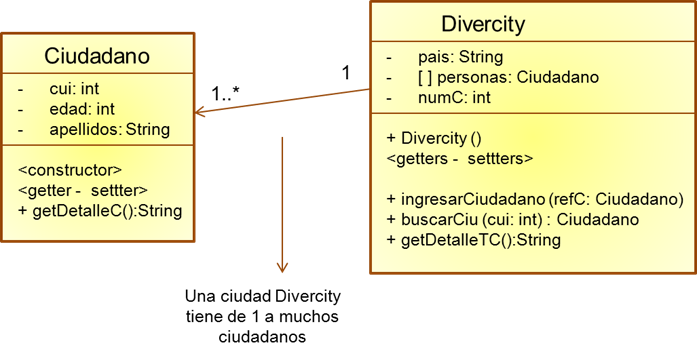
**CASOS DE NEGOCIO 03: Proyecto Ciudad Divertida (DiverCity)**

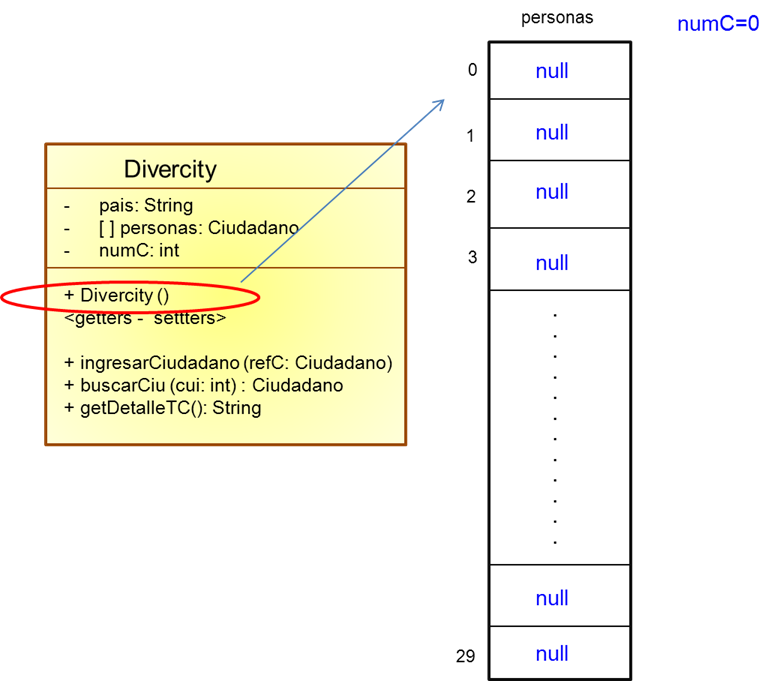
**Divercity**, es un parque temático para los niños y niñas entre tres y trece años, ubicado en cuatro ciudades de Colombia: Bogotá, Medellín, Barranquilla; y en Lima Perú.

* Desarrollar una aplicación Java de nombre “Proyecto DiverCity” que permita controlar el ingreso de ciudadanos a DiverCity, listar a los ciudadanos que han ingresado, realizar un sorteo entre todos los ciudadanos y mostrar al ganador.
* Para la solución del caso de negocio se deberá diseñar clases cuyos métodos permitan crear objetos ciudadanos que sean referenciados por un arreglo de referencias, realizar la búsqueda de un ciudadano por cui, el cual será ingresado por teclado, y listar a todos los ciudadanos de DiverCity.

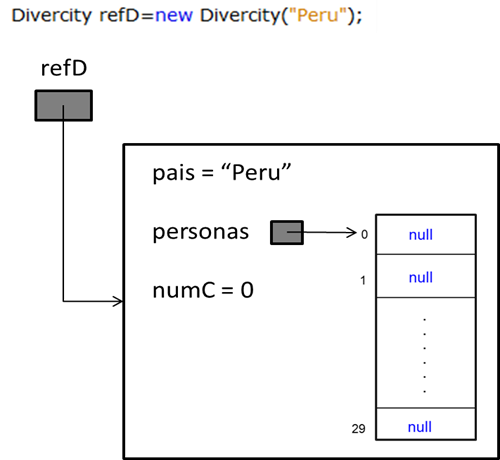
La clase **Ciudadano** tendrá como variables de instancia (atributos) el dni, edad, nombres y apellidos. El máximo número de ciudadanos, para efectos de simulación, es de 30.

* La clase **DiverCity** tendrá como variables de instancia al país de ubicación, y para gestionar a sus ciudadanos, un arreglo de referencias tipo Ciudadano y a un contador de ciudadanos.
* La clase Prueba, deberá instanciar a las clases Ciudadano y DiverCity para solucionar el caso de negocio. Para efectos de simulación, crear tres objetos ciudadanos del país Perú. Debe controlar que el dni no se repita, envíe un mensaje de error al respecto





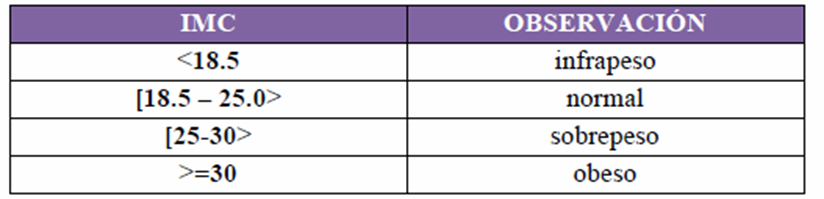
Un objeto referencia de tipo Divercity tendrá la siguiente representación:



**CASO DE NEGOCIO 04: PROYECTO PACIENTES DE UNA CLINICA.**

Diseñe clases y desarrolle una aplicación Java que permita registrar y guardar los datos de los pacientes de una clínica: nombre, peso en kilogramos, talla en metros, imc y su observación correspondiente.

* La aplicación deberá calcular el índice de masa corporal (imc= peso/(talla)^2 ) y la observación correspondiente, de los pacientes , de acuerdo a la siguiente tabla

****

* La solución deberá mostrar el listado de los pacientes consultados, los datos de los pacientes con observación Obeso.

Diseñe la clase **Paciente**. Identifique los atributos o variables de instancia de la clase Paciente, así como su constructor y sus métodos.

Diseñe la clase **Clínica**, la cual tendrá como atributos principales un arreglo de referencias tipo Paciente y un contador de pacientes de la clínica. Desarrolle su constructor y los métodos de la clase.

Diseñe una clase Principal (de prueba) y solucione el caso de negocio, utilizando las clases Paciente y Clínica.

**CASO DE NEGOCIO 05: Proyecto Aplicación Académica**

* Diseñar Clases (clase Alumno y clase Asignatura) y desarrollar una aplicación Java que permita registrar un máximo de 30 alumnos en una asignatura. Cada alumno deberá disponer de sus nombres y apellidos, de su código de alumno y finalmente de su nota final.
* Esta aplicación también deberá realizar consultas por código de alumno para mostrar sus nombres y apellidos y su nota final.
* Información adicional:

La nota final permitida está en el rango de 0 a 20.

**CASO DE NEGOCIO 06: Proyecto Colección de Partes**

Desarrollar una aplicación Java con la siguiente funcionalidad:

* Coleccionar partes de un auto, en un arreglo de referencias.
* Los códigos de identificación (id) de las partes del auto son únicos, no se permite ingresar a la colección una parte con un mismo id.
* Ubicar partes del auto en la colección (arreglo de referencias) por id. Utilizar búsqueda secuencial.

Para la solución del caso de negocio se deberá diseñar clases cuyos métodos permitan crear objetos (partes del auto) y agregarlos a la colección (arreglo de referencias), sólo si no existe en la colección.

La clase Parte tendrá como variables de instancia el código de identificación (id) y el nombre de la parte del auto. Considerar para efectos de simulación, que la colección tiene un máximo de 30 partes.

La clase Colección tendrá como variable de instancia a un arreglo de referencias tipo Parte y a un contador de miembros de la colección.

**CASO DE NEGOCIO 07: PROMEDIO NOTAS.**

Se desea gestionar el promedio de prácticas (en total 6 notas) de un alumno considerando para ello sus 5 mejores notas.

Se pide:

* Diseñar la clase **Alumno**, con variables de instancia: código y nombre.
* Un constructor sin parámetros que inicialice por defecto por atributos de la clase.
* Los métodos getter y setter.
* Diseñar la clase **GestionNotas** que tiene como variables de instancias: un objeto referencia a la clase Alumno, un arreglo para almacenar 6 notas y el promedio de notas.
* Un constructor con un parámetro, el objeto referencia a alumno, y tiene como código la creación del arreglo de las 6 notas y el promedio de notas inicializado en cero.
* Un método que genera aleatoriamente las 6 notas del alumno y los almacene en el arreglo.
* Un método que calcule el promedio de notas considerando las 5 mejores.
* Un método que retorna la concatenación de toda la información, como son los datos de alumno, su promedio; así como todas sus notas.

Y otros métodos que usted considere.

* Diseñar la clase de prueba llamada **ProcesarNotas** que permite realizar las siguientes tareas:
* Crear un objeto referencia **objAlumno** de tipo Alumno, en el cual el código, y nombre es ingresado por pantalla.
* Crear un objeto referencia de tipo **GestionNotas**.
* Se generan las 6 notas, se calcula su promedio y se muestra toda la información.

