Examen de Complementos de Programación

3 de noviembre de 2016

Tiempo máximo: 3 horas

Problema 1 (4 puntos)

Se quiere gestionar información de *startups* (empresas emergentes) en el ámbito tecnológico y de servicios.

Una *startup* dispone de un CIF (código único compuesto por letras y dígitos numéricos) y un nombre que lo identifican y que nunca se pueden cambiar. También se gestionarán otros datos como son el capital social, el importe dedicado a inversiones, el número de trabajadores y una pequeña descripción del negocio.

Para las *startups* de ámbito tecnológico interesa saber si han registrado alguna patente o desarrollado software propio y si es así cuantas patentes y/o aplicaciones y el valor medio de las mismas. En el caso concreto de las *startups* tecnológicas que se dedican a la biotecnología interesa saber en qué ámbito trabajan: biotecnología blanca (industrial), verde (agrícola), azul (marina) o gris (medio ambiente) y la valoración de sus laboratorios. Para las tecnológicas dedicadas a las TIC interesa el costo de cada puesto de trabajo, que actualmente es de 2300 euros pero puede variar en el tiempo.

Para las del ámbito de servicios interesa saber si disponen de un *call center*.

Se pide diseñar las clases necesarias para gestionar *startups*.

Se pide diseñar los métodos que realicen las siguientes funcionalidades:

- Alta de trabajadores. El método incrementará en uno el número de trabajadores de la *startup*.
- Baja de trabajadores. El método decrementará en uno el número de trabajadores de la *startup*. Este método devolverá un booleano indicando si se ha podido realizar la actualización. Se puede actualizar el número de trabajadores siempre que al menos quede un trabajador en la *startup*.
- Cálculo del inmovilizado inmaterial. Devolverá el valor total de las patentes y/o aplicaciones de la *startup*. Para el resto de las *startups* el valor será 0.
- Cálculo del inmovilizado material. En el caso de las *startups* del ámbito de la biotecnología el valor se corresponde con la valoración de sus laboratorios. Para las *startups* del ámbito de las TIC el valor se corresponde con el coste del total de los puestos de trabajo. Para el resto de las *startups* el valor será 0.
- Cálculo del inmovilizado. Se corresponde con la suma del inmovilizado material, el inmaterial y las inversiones.

Para todas las clases se deberá diseñar los métodos *getter* y *setter* necesarios. Se deberá verificar el ámbito para el caso de las *startup* biotecnológicas, en caso de valor incorrecto la información del ámbito no se modificará.

Diseñar también los constructores necesarios, los métodos que se han pedido anteriormente y los métodos *toString* y *equals*. Dos *startups* son iguales si tienen el mismo CIF.

Examen de Complementos de Programación

3 de noviembre de 2016

Problema 2 (3 puntos)

El ayuntamiento de Terrassa quiere promover la creación de nuevas empresas (*startups*) para lo que ha creado un vivero de *startups*. El vivero de *startups* consiste en una serie de locales que el ayuntamiento pone a disposición de las *startups* de forma gratuita. Los locales se identifican por un número que va desde 0 hasta el número de locales -1.

Diseñar una clase vivero que permita la gestión de los locales proporcionados por el ayuntamiento a las *startups*. Concretamente se quiere mantener por cada local todos los datos de la *startup* que lo ocupa. Cuando se crea el vivero se indica el número de locales que se pone a disposición. Inicialmente todos estarán vacíos. La clase debe tener:

- Un constructor que crea el vivero según el número de locales que se ponen a disposición de las *startups*.
- Un método para asignar un local a una *startup*. El método ha de retornar el número de local asignado y ha de guardar los datos de la *startup*. El local asignado debe ser el primero libre, en caso de que todos estén ocupados el método debe retornar un valor negativo.
- Un método que indique cuantos locales hay libres.
- Un método que dado el CIF de una *startup* retorne el número de local que ocupa.
- Un método que retorne la *startup* que ocupa un local. En caso que el local este libre debe retornar un *null*.
- El método *toString* que retorne un *String* con los datos de las *startups* que ocupan cada uno de los locales.

NOTA: Si no se ha realizado el problema 1 se puede utilizar la clase StartupPorDefecto que se encuentra en Atenea.

Problema 3 (3 puntos)

Hacienda quiere poder controlar el patrimonio de los contribuyentes que están bajo investigación y nos ha pedido que diseñemos una clase Patrimonio para mantener el patrimonio de un contribuyente. El patrimonio consiste en todos los objetos de valor del contribuyente. De cada uno de ellos se debe poder consultar el precio de adquisición y el valor actual. La clase Patrimonio debe tener:

- Un constructor que dada una tabla de objetos de valor inicializa el patrimonio. Esta tabla no será modificada nunca desde fuera de la clase Patrimonio.
- Un método que retorne el valor actual de todo el patrimonio.
- Un método que retorne la plusvalía de todo el patrimonio. La plusvalía de cada objeto de valor se calcula como la diferencia entre el valor actual y el precio de adquisición.